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

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

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



32-Bit Microcontroller

FM0+ Family Peripheral Manual

Doc. No. 002-04969 Rev. *C

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

© Cypress Semiconductor Corporation, 2013-2018. 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.

Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Preface

Thank you for your continued use of Cypress 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.
For the descriptions on Analog macro, Timer, and Communication Macro, see the respective separate peripheral manual.

Note:

- *This manual explains the configuration and operation of the peripheral functions, but does not cover the specifics of each device in the series. Users should refer to the respective data sheets of devices for device-specific details.*
- *Whether a peripheral function is on board or not is dependent on product type. See data sheets for details.*

Sample programs and development environment

Cypress offers sample programs free of charge for using the peripheral functions of the FM0+ family. 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:

https://community.cypress.com/community/MCU

Note:

- *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.

Overall Organization of This Manual

Peripheral Manual Timer Part has 14 chapters and APPENDIXES as shown below.

CHAPTER 1: System Overview
CHAPTER 2-1:Clock
CHAPTER 2-2:Peripheral Clock Gating

CHAPTER 2-3:High-Speed CR Trimming
CHAPTER 2-4:Low-Speed CR Prescaler
CHAPTER 3: Clock Supervisor
CHAPTER 4: Resets
CHAPTER 5-1:Low-voltage Detection Overview
CHAPTER 5-2:Low-voltage Detection (TYPE1)
CHAPTER 5-3:Low-voltage Detection (TYPE2)
CHAPTER 5-3:Low-voltage Detection (TYPE3)
CHAPTER 6-1:Low Power Consumption Mode
CHAPTER 6-2:VBAT Domain
CHAPTER 7-1:Interrupts Overview
CHAPTER 7-2:Interrupts Top (TYPE1)
CHAPTER 7-3:Interrupts(TYPE1-A)
CHAPTER 7-4:Interrupts(TYPE1-B)
CHAPTER 7-5:Interrupts Top (TYPE2)
CHAPTER 7-6:Interrupts(TYPE2-A)
CHAPTER 7-7:Interrupts(TYPE2-B)
CHAPTER 7-8:Interrupts Top (TYPE3)
CHAPTER 8: External Interrupt and NMI Control Sections
CHAPTER 9: DMAC
CHAPTER 10-1:I/O Port
CHAPTER 10-2:Fast GPIO
CHAPTER 11: CRC (Cyclic Redundancy Check)
CHAPTER 12: Debug Interface
CHAPTER 13: Micro Trace Buffer Data Watchpoint and Trace
CHAPTER 14: Flash Memory
CHAPTER 15: Unique ID Register
CHAPTER 16: DSTC
APPENDIXES

Related Manuals

The manuals related to this family are listed below. See the manual appropriate to the applicable conditions.

The contents of these manuals are subject to change without notice. Contact us to check the latest versions available.

Peripheral Manual

- FM0+ Family PERIPHERAL MANUAL (this manual)
(Called "PERIPHERAL MANUAL" hereafter)
- FM0+ Family PERIPHERAL MANUAL Timer Part
(Called "Timer Part" hereafter)
- FM0+ Family PERIPHERAL MANUAL Analog Macro Part
(Called "Analog Macro Part" hereafter)
- FM0+ Family PERIPHERAL MANUAL Communication Macro Part
(Called "Communication Macro Part" hereafter)

Data sheet

For details about device-specific, electrical characteristics, package dimensions, ordering information etc., see the following document.

- 32-bit Microcontroller FM0+ Family DATA SHEET

Note:

- *The data sheets for each series are provided.*
See the appropriate data sheet for the series that you are using.

CPU Programming manual

For details about Arm Cortex-M0+ core, see the following documents that can be obtained from http://www.arm.com/.

- Cortex-M0+ Technical Reference Manual
- Armv6-M Architecture Application Level Reference Manual

Flash Programming manual

For details about the functions and operations of the built-in flash memory, see the following document.

- FM0+ Family FLASH PROGRAMMING MANUAL

Note:

- *The Flash Programming manuals for each series are provided.*
  See the appropriate Flash Programming manual for the series that you are using.

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. Register Map" in "APPENDIXES".

About the chapters

Basically, this manual explains 1 peripheral function per chapter.

Terminology

This manual uses the following terminology.

Term	Explanation
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
  - Attribute : Attributes for read and write of each bit
    - R : Read only
    - W : Write only
    - R/W : Readable/Writable
    - - : Undefined
  - Initial value : Initial value of the register after reset
    - 0 : Initial value is "0"
    - 1 : Initial value is "1"
    - X : Initial value is undefined

- 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)

The target products in this manual

- In this manual, the products are classified into the following groups and are described as follows. For the descriptions such as "TYPE1-M0+", “TYPE2-M0+” and “TYPE3-M0+”, see the relevant items of the target FM0+ product in the lists below.

Table 1 FM0+ family TYPE1 Product list

TYPE	Flash memory size	
	56 Kbytes	88 Kbytes
TYPE1-M0+	S6E1A11B S6E1A11C	S6E1A12B S6E1A12C

Table 2 FM0+ family TYPE2 Product list

TYPE	Flash memory size	
	304K bytes	560K bytes
TYPE2-M0+	S6E1B84E S6E1B84F S6E1B84G S6E1B34E S6E1B34F S6E1B34G	S6E1B86E S6E1B86F S6E1B86G S6E1B36E S6E1B36F S6E1B36G

Table 3 FM0+ family TYPE3 Product list

TYPE	Flash memory size	
	64K bytes	128K bytes
TYPE3-M0+	S6E1C31B S6E1C31C S6E1C31D S6E1C11B S6E1C11C S6E1C11D	S6E1C32B S6E1C32C S6E1C32D S6E1C12B S6E1C12C S6E1C12D

Table of Contents

Preface **3**
Related Manuals **5**
How to Use This Manual **7**
Table of Contents **9**
CHAPTER 1: System Overview **23**
1. Bus Architecture 24
1.1 Bus Block Diagram 26
1.2 Memory Architecture 27
1.3 Memory Map 28
1.4 Peripheral Address Map 30
2. Cortex-M0+ Architecture 33
2.1 Option Configuration 35
3. Mode 36
CHAPTER 2-1: Clock **39**
1. Overview 40
2. Configuration 42
3. Operations 45
3.1 Selecting Clock Mode 45
3.2 Internal Bus Clock Frequency Division Control 46
3.3 PLL Clock Control 47
3.4 Oscillation Stabilization Wait Time 49
3.5 Interrupt Factors 52
4. Clock Setup Procedure Examples 53
4.1 Setup Procedure Examples (TYPE1-M0+, TYPE3-M0+) 53
4.2 Setup Procedure Examples (TYPE2-M0+) 57
5. Registers 62
5.1 System Clock Mode Control Register (SCM_CTL) 63
5.2 System Clock Mode Status Register (SCM_STR) 65
5.3 Base Clock Prescaler Register (BSC_PSR) 67
5.4 APB0 Prescaler Register (APBC0_PSR) 68
5.5 APB1 Prescaler Register (APBC1_PSR) 69
5.6 Software Watchdog Clock Prescaler Register (SWC_PSR) 70
5.7 Clock Stabilization Wait Time Register (CSW_TMR) 71
5.8 PLL Clock Stabilization Wait Time Setup Register (PSW_TMR) 73
5.9 PLL Control Register 1 (PLL_CTL1) 75
5.10 PLL Control Register 2 (PLL_CTL2) 76
5.11 Debug Break Watchdog Timer Control Register (DBWDT_CTL) 77
5.12 Interrupt Enable Register (INT_ENR) 78
5.13 Interrupt Status Register (INT_STR) 79

5.14 Interrupt Clear Register (INT_CLR) 80
6. Usage Precautions 82
CHAPTER 2-2: Peripheral Clock Gating 85
1. Peripheral Clock Gating Overview 86
2. Peripheral Clock Gating Configuration 89
3. Peripheral Clock Gating Control 92
3.1 Peripheral Clock Control Procedures 93
4. Peripheral Clock Gating Function Registers 97
4.1 Peripheral Function Clock Control Register 0 (CKEN0) 98
4.2 Peripheral Reset Control Register 0 (MRST0) 101
4.3 Peripheral Clock Control Register 1 (CKEN1) 104
4.4 Peripheral Function Reset Control Register 1 (MRST1) 107
4.5 Peripheral Clock Control Register 2 (CKEN2) 109
4.6 Peripheral Function Reset Control Register 2 (MRST2) 112
5. Peripheral Clock Gating Function Usage Precautions 115
CHAPTER 2-3: High-Speed CR Trimming 119
1. High-Speed CR Trimming Function Overview 120
2. High-Speed CR Trimming Function Configuration and Block Diagram 121
3. High-Speed CR Trimming Function Operation 122
4. High-Speed CR Trimming Function Setup Procedure Example 124
5. High-Speed CR Trimming Function Register List 134
5.1 High-Speed CR Oscillation Frequency Division Setup Register (MCR_PSR) 135
5.2 High-Speed CR Oscillation Frequency Trimming Register (MCR_FTRM) 136
5.3 High-Speed CR Oscillation Temperature Trimming Setup Register (MCR_TTRM) 138
5.3.1 MCR_TTRM (TYPE1-M0+) 138
5.3.2 MCR_TTRM (TYPE2-M0+) 139
5.3.3 MCR_TTRM (TYPE3-M0+) 140
5.4 High-Speed CR Oscillation Register Write-Protect Register (MCR_RLR) 141
6. High-Speed CR Trimming Function Usage Precautions 142
CHAPTER 2-4: Low-Speed CR Prescaler 143
1. Low-speed CR Prescaler Overview 144
2. Low-speed CR Prescaler Configuration 145
3. Low-speed CR Prescaler Operation and Setup Procedure Example 146
3.1 Setup Procedures of Low-speed CR Prescaler 146
3.2 Operation of Low-speed CR Prescaler 147
3.3 Low-speed CR Correction Example 148
4. Low-speed CR Prescaler Register 149
4.1 Low-speed CR Prescaler Control Register (LCR_PRSLD) 150
CHAPTER 3: Clock Supervisor 151
1. Overview 152
2. Configurations and Block Diagrams 153
2.1 Clock Failure Detection 153
2.2 Anomalous Frequency Detection 154
3. Explanation of Operations 155
4. Setup Procedure Examples 156
5. Operation Examples 158
5.1 Clock Failure Detection 158
5.2 Anomalous Frequency Detection 160

5.3 Example of Anomalous Frequency Detection Function Window Setting 161
6. Registers 163
6.1 CSV Control Register (CSV_CTL) 164
6.2 CSV Status Register (CSV_STR) 166
6.3 Frequency Detection Window Setting Register (Upper) (FCSWH_CTL) 167
6.4 Frequency Detection Window Setting Register (Lower) (FCSWL_CTL) 168
6.5 Frequency Detection Counter Register (FCSWD_CTL) 169
7. Usage Precautions 170
CHAPTER 4: Resets 171
1. Overview 172
2. Configuration 173
3. Explanation of Operations 175
3.1 Reset Factors 176
3.2 Resetting Inside Device 180
3.2.1 Resets to Cortex-M0+ 181
3.2.2 Resets to Peripheral Circuit 182
3.3 Reset Sequence 183
3.4 Operations after Resets are Cleared 185
4. Register 186
4.1 Reset Factor Register (RST_STR) 187
CHAPTER 5-1: Low-voltage Detection Overview 191
1. Overview 192
CHAPTER 5-2: Low-voltage Detection (TYPE1) 195
1. Configuration 196
2. Operations 198
3. Setup Procedure Examples 201
4. Registers 203
4.1 Low-Voltage Detection Voltage Control Register (LVD_CTL) 204
4.2 Low-Voltage Detection Interrupt Factor Register (LVD_STR) 207
4.3 Low-Voltage Detection Interrupt Factor Clear Register (LVD_CLR) 208
4.4 Low-Voltage Detection Voltage Protection Register (LVD_RLR) 209
4.5 Low-Voltage Detection Circuit Status Register (LVD_STR2) 210
5. Usage Precautions 211
CHAPTER 5-3: Low-voltage Detection (TYPE2) 213
1. Configuration 214
2. Operations 216
3. Setup Procedure Examples 219
3.1 Example for Low-voltage detection reset 219
3.2 Example for Low-voltage detection interrupt (LVDRLIE = 0) 220
3.3 Example for Low-voltage detection interrupt (LVDRLIE = 1) 221
3.4 Example for Vref calibration 223
4. Registers 224
4.1 Low-voltage Detection Voltage Control Register (LVD_CTL) 225
4.2 Low-voltage detection release voltage Control Register (LVD_CTL2) 227
4.3 Low-voltage Detection (ch.2) Voltage Control Register (LVD2_CTL) 229
4.4 Low-voltage Detection (ch.2) release voltage Control Register (LVD2_CTL2) 231
4.5 Low-voltage Detection Interrupt Factor Register (LVD_STR) 233
4.6 Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) 235

4.7 Low-voltage Detection Voltage Protection Register (LVD_RLR) 236
4.8 Low-voltage Detection Circuit Status Register (LVD_STR2) 237
4.9 Vref Calibration Control Register (CAL_CTL) 239
4.10 Vref Calibration Value Register (CAL_SET) 240
4.11 Vref Calibration Security Key Register (CAL_KEY) 241
5. Usage Precautions 242
CHAPTER 5-4: Low-voltage Detection (TYPE3) 243
1. Configuration 244
2. Operations 246
3. Setup Procedure Examples 249
3.1 Example for Low-voltage detection reset 249
3.2 Example for Low-voltage detection interrupt 250
3.3 Example for Vref calibration 251
4. Registers 252
4.1 Low-voltage Detection Voltage Control Register (LVD_CTL) 253
4.2 Low-voltage Detection Interrupt Factor Register (LVD_STR) 255
4.3 Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) 256
4.4 Low-voltage Detection Voltage Protection Register (LVD_RLR) 257
4.5 Low-voltage Detection Circuit Status Register (LVD_STR2) 258
4.6 Vref Calibration Control Register (CAL_CTL) 259
4.7 Vref Calibration Security Key Register (CAL_KEY) 261
5. Usage Precautions 263
CHAPTER 6-1: Low Power Consumption Mode 265
1. Overview 266
2. Configuration of CPU Operation Modes 271
3. Operations of Standby Modes 281
3.1 Operations of SLEEP Modes (High-Speed CR Sleep, Main Sleep, PLL Sleep, Low-Speed CR Sleep, and Sub Sleep Modes) 284
3.2 Operations of TIMER Modes (High-Speed CR Timer, Main Timer, PLL Timer, Low-Speed CR Timer, and Sub Timer Modes) 286
3.3 Operation of RTC Mode 288
3.4 Operations of STOP Mode 291
4. Standby Mode Setting Procedure Examples 294
5. Description of Deep Standby Mode Operation 299
5.1 Operation of Deep Standby Mode 299
5.1.1 Clock Operation Status in Deep Standby Mode 299
5.1.2 Return Factors from Deep Standby Mode 299
5.1.3 Internal Power Supply Status and Reset Status in Deep Standby Mode 300
5.2 Operation of Deep Standby RTC Mode 301
5.3 Operation of Deep Standby Stop Mode 303
6. Deep Standby Mode Setting Procedure Examples 305
7. Deep Standby Return Factor Determination Procedure 307
8. Registers 308
8.1 Standby Mode Control Register (STB_CTL) 310
8.2 Sub Oscillation Circuit Power Supply Control Register (REG_CTL) 312
8.3 Sub Clock Control Register (RCK_CTL) 313
8.4 RTC Mode Control Register (PMD_CTL) 314
8.5 Deep Standby Return Factor Register 1 (WRFSR) 315

8.6 Deep Standby Return Factor Register 2 (WIFSR) 316
8.7 Deep Standby Return Enable Register (WIER) 318
8.8 WKUP Pin Input Level Register (WILVR) 320
8.9 Deep Standby RAM Retention Register (DSRAMR) 321
8.10 Backup Registers 01 to 16 (BUR01 to 16) 322
8.11 Main OSCillation Crystal Type Selection ConTroL Register (MOSC_CTL) 323
8.12 IO State hold control Register (WIOLC_CTL) 324
8.13 Sub Oscillator IO Control Register (SUBOSC_CTL) 326
8.14 CEC Input/Output Control Register (CEC_CTL) 327
8.15 Serial Wire Debug Control Register (DEBUG_SW_CTL) 328
9. Usage Precautions 329
CHAPTER 6-2: VBAT Domain 331
1. Overview of VBAT Domain 332
2. Configuration of VBAT Domain 334
2.1 Interfacing with Always-on Domain 335
2.2 RTC 345
2.3 32 kHz Oscillation Circuit 346
2.4 Power-on Circuit 348
2.5 Backup Registers 349
2.6 VBAT I/O Ports 351
3. Chip Power Supply Control 354
4. Hibernation Control 358
5. Procedure for Setting 32 kHz Clock 362
6. Procedure for Setting VBAT I/O Port 367
7. Registers 370
7.1 VB_CLKDIV Register 371
7.2 WTOSCCNT Register 372
7.3 CCS/CCB Register 373
7.4 BOOST Register 374
7.5 EWKUP Register 375
7.6 HIBRST Register 376
7.7 VDET Register 377
7.8 Port Function Set Register (VBPFR) 378
7.9 Pull-Up Set Register (VBPCR) 380
7.10 Port I/O Direction Set Register (VBDDR) 381
7.11 Port Input Data Register (VBDIR) 382
7.12 Port Output Data Register (VBDOR) 383
7.13 Port Pseudo-Open Drain Set Register (VBPZR) 384
8. Usage Precautions 385
CHAPTER 7-1: Interrupts Overview 387
1. Explanation of Reference Chapter of Interrupts 388
CHAPTER 7-2: Interrupts Top (TYPE1) 389
1. Overview 390
2. Configuration 391
CHAPTER 7-3: Interrupts (TYPE1-A) 395
1. Exception and Interrupt Factor Vectors 396
2. Registers 398
2.1 DMA Request Selection Register (DRQSEL) 399

2.2 EXC02 Batch Read Register (EXC02MON) 400
2.3 IRQ00 Batch Read Register (IRQ00MON) 401
2.4 IRQ01 Batch Read Register (IRQ01MON) 402
2.5 IRQ02 Batch Read Register (IRQ02MON) 403
2.6 IRQ03 Batch Read Register (IRQ03MON) 404
2.7 IRQ04 Batch Read Register (IRQ04MON) 406
2.8 IRQ05 Batch Read Register (IRQ05MON) 407
2.9 IRQ06 Batch Read Register (IRQ06MON) 409
2.10 IRQ07/09/11/13/15/17 Batch Read Register (IRQxxMON) 411
2.11 IRQ19/21 Batch Read Register (IRQxxMON) 412
2.12 IRQ08/10/12/14/16/18 Batch Read Register (IRQxxMON) 413
2.13 IRQ20/22 Batch Read Register (IRQxxMON) 415
2.14 IRQ23 Batch Read Register (IRQ23MON) 416
2.15 IRQ24 Batch Read Register (IRQ24MON) 417
2.16 IRQ25/26 Batch Read Register (IRQxxMON) 419
2.17 IRQ27 Batch Read Register (IRQ27MON) 420
2.18 IRQ28 Batch Read Register (IRQ28MON) 421
2.19 IRQ29 Batch Read Register (IRQ29MON) 423
2.20 IRQ30 Batch Read Register (IRQ30MON) 425
2.21 IRQ31 Batch Read Register (IRQ31MON) 427
3. Usage Precautions 429

CHAPTER 7-4: Interrupts (TYPE1-B) 431

1. Exception and Interrupt Factor Vectors 432
2. Registers 434
2.1 DMA Request Selection Register (DRQSEL) 435
2.2 EXC02 Batch Read Register (EXC02MON) 436
2.3 IRQ00 Batch Read Register (IRQ00MON) 437
2.4 IRQ01 Batch Read Register (IRQ01MON) 438
2.5 IRQ02 Batch Read Register (IRQ02MON) 439
2.6 IRQ03 to IRQ10 Batch Read Register (IRQ03MON to IRQ10MON) 440
2.7 IRQ11 Batch Read Register (IRQxxMON) 442
2.8 IRQ12 Batch Read Register (IRQ12MON) 443
2.9 IRQ13 Batch Read Register (IRQ13MON) 444
2.10 IRQ14 Batch Read Register (IRQ14MON) 446
2.11 IRQ15 to IRQ18 Batch Read Register (IRQ15MON to IRQ18MON) 447
2.12 IRQ19/21 Batch Read Register (IRQxxMON) 448
2.13 IRQ20/22 Batch Read Register (IRQxxMON) 449
2.14 IRQ23 Batch Read Register (IRQ23MON) 450
2.15 IRQ24 Batch Read Register (IRQ24MON) 451
2.16 IRQ25/26 Batch Read Register (IRQxxMON) 453
2.17 IRQ27 Batch Read Register (IRQ27MON) 455
2.18 IRQ28/29 Batch Read Register (IRQxxMON) 457
2.19 IRQ30 Batch Read Register (IRQ30MON) 459
2.20 IRQ31 Batch Read Register (IRQ31MON) 461
2.21 Interrupt Factor Vector Relocate Setting Register (IRQCMODE) 463
2.22 Interrupt Factor Selection Register 0 (RCINTSEL0) 464
2.23 Interrupt Factor Selection Register 1 (RCINTSEL1) 465
3. Usage Precautions 468

CHAPTER 7-5: Interrupts Top (TYPE2) 469
1. Overview 470
2. Configuration 470
CHAPTER 7-6: Interrupts (TYPE2-A) 475
1. Lists of Interrupts 476
1.1 List of Exceptions and Interrupts 476
1.2 Interrupt Signals Input to DSTC 481
2. Registers 483
2.1 EXC02 Batch Read Register (EXC02MON) 484
2.2 IRQxx Batch Read Register (IRQxxMON) 485
3. Usage Precautions 486
CHAPTER 7-7: Interrupts (TYPE2-B) 487
1. Lists of Interrupts 488
1.1 List of Exceptions and Interrupts 488
1.2 Interrupt Signals Input to DSTC 493
2. Registers 495
2.1 EXC02 Batch Read Register (EXC02MON) 496
2.2 IRQxx Batch Read Register (IRQxxMON) 497
2.3 Interrupt Factor Vector Relocate Setting Register (IRQCMODE) 498
2.4 Interrupt Factor Selection Register 0 (RCINTSEL0) 499
2.5 Interrupt Factor Selection Register 1 (RCINTSEL1) 500
3. Usage Precautions 502
CHAPTER 7-8: Interrupts (TYPE3) 503
1. Overview 504
2. Configuration and Function 504
2.1 Connection diagram 504
2.2 NVIC (Nested Vectored Interrupt Controller) 505
2.3 Interrupt Aggregation and Batch Read Registers 505
2.4 Vector Indicate Register 506
2.5 Connection of DSTC transfer request and transfer completion 507
2.6 Connection of DMA Transfer Acknowledge Signal 508
2.7 Connection of DMA Transfer Stop Signal 508
3. Lists of Interrupts 509
3.1 Lists of exceptions and interrupts 509
3.2 Lists of transfer request signals Input to DSTC 514
4. Registers 515
4.1 Lists of registers 515
4.2 EXC02 Batch Read Register (EXC02MON) 516
4.3 IRQxx Batch Read Register (IRQxxMON) 517
4.4 VIR Offset Register (VIR_OFFSET) 518
4.5 Vector Indicate Register xx (VIRxx) 519
4.6 USB ch.0 Odd Packet Size DMA Enable Register (ODDPKS) 520
5. Usage Precautions 522
CHAPTER 8: External Interrupt and NMI Control Sections 523
1. Overview 524
2. Block Diagram 525
3. Operations and Setting Procedure Examples 527
3.1 Operations of External Interrupt Control Section 528

3.2 Operations of NMI Control Section 531
3.3 Returning from the Timer, Stop, RTC Mode 532
4. Registers 534
4.1 External Interrupt Enable Register (ENIR) 535
4.2 External Interrupt Factor Register (EIRR) 536
4.3 External Interrupt Factor Clear Register (EICL) 537
4.4 External Interrupt Factor Level Register (ELVR) 538
4.5 External Interrupt Factor Level Register 1 (ELVR1) 539
4.6 Non Maskable Interrupt Factor Register (NMIRR) 540
4.7 Non Maskable Interrupt Factor Clear Register (NMICL) 541
4.8 External Interrupt Factor Level Register 2 (ELVR2) 542
4.9 Non Maskable Interrupt Enable Register (NMIENR) 543
CHAPTER 9: DMAC 545
1. Overview of DMAC 546
2. Configuration of DMAC 548
2.1 DMAC and System Configuration 549
2.2 I/O Signals of DMAC 551
3. Functions and Operations of DMAC 553
3.1 Software-Block Transfer 554
3.2 Software-Burst Transfer 557
3.3 Hardware-Demand Transfer 559
3.4 Hardware-Block Transfer & Burst Transfer 560
3.5 Channel Priority Control 562
4. DMAC Control 563
4.1 Overview of DMAC Control 564
4.2 DMAC Operation and Control Procedure for Software Transfer 565
4.3 DMAC Operation and Control Procedure for Hardware (EM=0) Transfer 574
4.4 DMAC Operation and Control Procedure for Hardware (EM=1) Transfer 585
5. Registers of DMAC 590
5.1 List of Registers 591
5.2 Entire DMAC Configuration Register (DMACR) 592
5.3 Configuration A Register (DMACA) 594
5.4 Configuration B Register (DMACB) 598
5.5 Transfer Source Address Register (DMACSA) 602
5.6 Transfer Destination Address Register (DMACDA) 603
6. Usage Precautions 604
CHAPTER 10-1: I/O Port 605
1. Overview 606
2. Configuration, Block Diagram, and Operation 607
2.1 Configuration of the I/O Port 607
2.2 Initially Selected Functions for the I/O Port 612
2.3 Relocation Function 613
2.4 Fixed Priority of EPFR Outputs 615
2.5 Operation in Deep Standby Mode (TYPE2-M0+) 616
2.6 Operation in Deep Standby Mode (TYPE3-M0+) 617
2.7 Deep Standby Transition Reset 620
3. Setup Procedure Example 621
3.1 Setup of the I/O Port 621

3.2 How to use I/O State Hold function (TYPE3-M0+) 622
4. Registers 624
4.1 Port Function Setting Register (PFRx) 628
4.2 Pull-Up Setting Register (PCRx) 630
4.3 Port Input/Output Direction Setting Register (DDRx) 632
4.4 Port Input Data Register (PDIRx) 634
4.5 Port Output Data Register x (PDORx) 636
4.6 Analog Input Setting Register (ADE) 638
4.7 Extended Pin Function Setting Register (EPFRx) 639
4.8 Extended Pin Function Setting Register 00 (EPFR00) 641
4.9 Extended Pin Function Setting Register 01 (EPFR01) 644
4.10 Extended Pin Function Setting Register 02 (EPFR02) 649
4.11 Extension Function Pin Setting Register 03 (EPFR03) 654
4.12 Extended Pin Function Setting Register 04 (EPFR04) 659
4.13 Extended Pin Function Setting Register 05 (EPFR05) 664
4.14 Extended Pin Function Setting Register 06 (EPFR06) 669
4.15 Extended Pin Function Setting Register 07 (EPFR07) 674
4.16 Extended Pin Function Setting Register 08 (EPFR08) 679
4.17 Extended Pin Function Setting Register 09 (EPFR09) 684
4.18 Extended Pin Function Setting Register 12 (EPFR12) 689
4.19 Extended Pin Function Setting Register 13 (EPFR13) 694
4.20 Extended Pin Function Setting Register 14 (EPFR14) 699
4.21 Extended Pin Function Setting Register 15 (EPFR15) 701
4.22 Extended Pin Function Setting Register 16 (EPFR16) 706
4.23 Extended Pin Function Setting Register 17 (EPFR17) 711
4.24 Extended Pin Function Setting Register 18 (EPFR18) 716
4.25 Extended Pin Function Setting Register 21 (EPFR21) 718
4.26 Extended Pin Function Setting Register 22 (EPFR22) 720
4.27 Extended Pin Function Setting Register 23 (EPFR23) 722
4.28 Extended Pin Function Setting Register 31 (EPFR31) 725
4.29 Extended Pin Function Setting Register 33 (EPFR33) 727
4.30 Extended Pin Function Setting Register 34 (EPFR34) 731
4.31 Extended Pin Function Setting Register 37 (EPFR37) 733
4.32 Extended Pin Function Setting Register 38 (EPFR38) 737
4.33 Special Port Setting Register (SPSR) 739
4.34 Port Pseudo Open Drain Setting Register (PZRx) 742
4.35 LVDI Input Setting Register (LVDIE) 744
5. Usage Precautions 745
CHAPTER 10-2: Fast GPIO 747
1. Overview 748
2. Configuration 749
3. Setup Procedure Example 755
4. Registers 757
4.1 Fast GPIO Input Data Register (FPDIRx) 760
4.2 Fast GPIO Output Data Register x (FPDORx) 762
4.3 Mirror Fast GPIO Input Data Register (M_FPDIRx) 764
4.4 Mirror of Fast GPIO Output Data Register (M_FPDORx) 766
4.5 Fast GPIO Output Enable Register (FPOERx) 768

5. Bit Manipulation Base Address 770
CHAPTER 11: CRC (Cyclic Redundancy Check) 771
1. Overview of CRC 772
2. CRC Operations 774
2.1 CRC Calculation Sequence 775
2.2 CRC Use Examples 776
3. CRC Registers 781
3.1 CRC Control Register (CRCCR) 782
3.2 Initial Value Register (CRCINIT) 784
3.3 Input Data Register (CRCIN) 785
3.4 CRC Register (CRCR) 786
CHAPTER 12: Debug Interface 787
1. Overview 788
2. Pin Description 789
2.1 Pins for Debug Purposes 790
2.2 Functions Initially Assigned to Pins 791
2.3 Internal Pull-Up of SW-DP Pins 792
CHAPTER 13: Micro Trace Buffer Data Watchpoint and Trace 793
1. Overview 794
2. Block Diagram 795
3. Configuration and Setting Procedure Examples 796
3.1 Configurations of MTB_DWT 797
3.2 Setting Procedure 798
4. Registers 800
4.1 MTB_DWT Address Compare Start Trace Register (CMP_ADDR_START) 801
4.2 MTB_DWT Data Compare Start Trace Register (CMP_DATA_START) 802
4.3 MTB_DWT Mask Data Compare Start Trace Register (CMP_MASK_START) 803
4.4 MTB_DWT Address Compare Stop Trace Register (CMP_ADDR_STOP) 804
4.5 MTB_DWT Data Compare Stop Trace Register (CMP_DATA_STOP) 805
4.6 MTB_DWT Mask Data Compare Stop Trace Register (CMP_MASK_STOP) 806
4.7 MTB_DWT Function Register (FCT) 807
4.8 Peripheral ID0-7 Register (PID0-7) 809
4.9 Component ID0-3 Register (CID0-3) 810
CHAPTER 14: Flash Memory 811
CHAPTER 15: Unique ID Register 813
1. Overview 814
2. Registers 814
2.1 Unique ID Register 0 (UIDR0: Unique ID Register 0) 815
2.2 Unique ID Register 1 (UIDR1: Unique ID Register 1) 816
CHAPTER 16: DSTC 817
1. Overview of DSTC 818
2. DSTC Operations Overview and DSTC System Configuration 819
2.1 Operations Overview of DSTC 819
2.2 DSTC and System Configuration 821
3. Functions and Operations of DSTC 824
3.1 Settings of DES 825
3.1.1 Specifying Transfer Data Size 825
3.1.2 Setting Transfer Addresses 826

3.1.3 Specifying OuterReload 828
3.1.4 Setting Chain Start and Transfer End Interrupt Notification 831
3.1.5 Other DES Settings 833
3.2 Control Functions of DSTC 836
3.2.1 DSTC internal Block Diagram 836
3.2.2 DESTP Register 836
3.2.3 Control of SW Transfer 836
3.2.4 Control of HW Transfer 837
3.2.5 Arbitration of Transfer Requests 838
3.2.6 Read Skip Buffer Function 840
3.2.7 Operation of the Transfer End 840
3.2.8 MONERS Register 841
3.2.9 Standby Function 843
3.3 Operation Flows of DSTC 845
3.3.1 SW Transfer Flow 845
3.3.2 HW Transfer Flow 848
3.3.3 Operation Flow after Specifying of DESP 850
4. Examples of DSTC Operations and Control 853
4.1 Transfer Operation Example 1 854
4.2 Transfer Operation Example 2 857
4.3 Transfer Operation Example 3 859
4.4 Transfer Operation Example 4 861
4.5 Transfer Operation Example 5 863
4.6 Examples of Controlling DSTC 866
5. Registers and Descriptors of DSTC 870
5.1 Lists of Control Registers and DES 871
5.2 DESTP Register 872
5.3 HWDESP[n] Register 873
5.4 CMD Register 874
5.5 CFG Register 875
5.6 SWTR Register 877
5.7 MONERS Register 878
5.8 DREQENB[n] Register 881
5.9 HWINT[n] Register 882
5.10 HWINTCLR[n] Register 883
5.11 DQMSK[n] Register 884
5.12 DQMSKCLR[n] Register 885
5.13 Descriptor 0 (DES0) 886
5.14 Descriptor 1 (DES1) 890
5.15 Descriptor 2 (DES2) 892
5.16 Descriptor 3 (DES3) 892
5.17 Descriptor 4 (DES4) 893
5.18 Descriptor 5 (DES5) 893
5.19 Descriptor 6 (DES6) 893
APPENDIXES 895
A. Product Type 897
1. Product TYPE List 898
B. Register Map (TYPE1-M0+) 899

1. Register Map 900
1.1 Flash I/F 901
1.2 Unique ID 902
1.3 Clock/Reset 903
1.4 HW WDT 905
1.5 SW WDT 906
1.6 Dual Timer 907
1.7 MFT 908
1.8 PPG 912
1.9 Base Timer 916
1.10 IO Selector for Base Timer 917
1.11 QPRC 918
1.12 QPRC NF 919
1.13 A/DC 920
1.14 D/AC 922
1.15 CR Trim 923
1.16 EXTI 924
1.17 INT-Req. READ 925
1.18 GPIO 927
1.19 HDMI-CEC 934
1.20 LVD 935
1.21 DS Mode 936
1.22 MFS 937
1.23 CRC 939
1.24 Watch Counter 940
1.25 RTC 941
1.26 Low-speed CR Prescaler 942
1.27 Peripheral Clock Gating 943
1.28 DMAC 944
1.29 MTB_DWT 945
1.30 Fast GPIO 947
C. Register Map (TYPE2-M0+) 951
1. Register Map 952
1.1 Flash I/F 953
1.2 Unique ID 954
1.3 Clock/Reset 955
1.4 HW WDT 957
1.5 SW_WDT 958
1.6 Dual Timer 959
1.7 MFT 960
1.8 PPG 964
1.9 Base Timer 968
1.10 IO Selector for Base Timer 969
1.11 QPRC 970
1.12 QPRC NF 971
1.13 A/DC 972
1.14 D/AC 974
1.15 CR Trim 975

1.16 EXTI 976
1.17 INT-Req. READ 977
1.18 LCDC 979
1.19 GPIO 980
1.20 HDMI-CEC 987
1.21 LVD 988
1.22 DS Mode 989
1.23 USB Clock 991
1.24 MFS 992
1.25 CRC 994
1.26 Watch Counter 995
1.27 RTC 996
1.28 Low-speed CR Prescaler 1000
1.29 Peripheral Clock Gating 1001
1.30 Smart Card I/F 1002
1.31 MFSI2S 1003
1.32 High Resilience 1004
1.33 USB 1005
1.34 DSTC 1007
1.35 MTB_DWT 1008
1.36 Fast GPIO 1010
D. Register Map (TYPE3-M0+) 1015
1. Register Map 1016
1.1 Flash I/F 1017
1.2 Unique ID 1018
1.3 Clock/Reset 1019
1.4 HW WDT 1021
1.5 SW WDT 1022
1.6 Dual Timer 1023
1.7 Base Timer 1024
1.8 IO Selector for Base Timer 1025
1.9 A/DC 1026
1.10 CR Trim 1028
1.11 EXTI 1029
1.12 INT-Req. READ 1030
1.13 GPIO 1032
1.14 HDMI-CEC 1037
1.15 LVD 1038
1.16 DS Mode 1039
1.17 USB Clock 1041
1.18 I2CSLAVE 1042
1.19 MFS 1043
1.20 CRC 1045
1.21 Watch Counter 1046
1.22 RTC 1047
1.23 Low-speed CR Prescaler 1048
1.24 Peripheral Clock Gating 1049
1.25 Smart Card I/F 1050

1.26 MFSI2S 1051
1.27 USB 1052
1.28 DSTC 1054
1.29 MTB_DWT 1055
1.30 Fast GPIO 1057
1.31 VIR 1059
E. List of Notes 1061
1. Notes when High-speed CR Is Used for Master Clock 1062
F. Major Changes 1063
Revision History 1065

CHAPTER 1: System Overview

This chapter explains the system overview.

1. Bus Architecture
2. Cortex-M0+ Architecture
3. Mode

CODE: 9AFSYSTEM-E03.0

1. Bus Architecture

This section explains the bus architecture.

For this series bus, AHB Bus Matrix circuit actualizes a multi-layer bus. Master and slave architectures are shown below:

- Master
  - Cortex-M0+ CPU (AHB-Lite)
  - DMAC
  - DSTC
- Slave
  - On-chip Flash Memory
  - On-chip SRAM (MTB sharable)
  - AHB-AHB Bus Bridge
  - AHB-APB Bus Bridge (APB0, APB1)
  - USB ch.0/ch.1

See Figure 1-1 for the bus block diagram.

Features

- RAM Architecture

The user SRAM area can be shared with the MTB SRAM area. The two areas are divided according to user configuration.

- APB Extension Bus

APB1 Peripheral Bus is an APB extension bus to which the following functions are originally added based on AMBA3.0. (APB0 is not included.)

  - Supporting Halfword (16 bits) and Byte(8 bits) Accesses

    For supported registers, halfword access and byte access are enabled.
    See "A. Register Map" in "APPENDIXES" for the supported registers.

  - Adding Read-Modify-Write (RMW) Signal

    HMASTLOCK signal in bit-band operations is used to generate.
    RMW signal is a signal added to prevent that an unrelated flag is cleared mistakenly in read-modify-write process of bit-band operations.
    The corresponding flag reads "1" in read during the read-modify-write process and is designed to ignore "1" write.
    This prevents any unrelated flag from being mistakenly cleared in the next write when the flag is set immediately after the read in the sequence from read to modify to write.
    For the corresponding flags and registers, it is described that "regardless of bit values, "1" can be read in "Read-Modify-Write".

Notes:

- *Bit-band operation must not be performed to a register which RMW is prohibited*
- *When Read-Modify-Write process is performed over the software without bit-band operation, RMW signal is not output.*
  Therefore, in this case, the flag value can be read in read operation although a register supports RMW process, and it is necessary not to be cleared an unrelated flag mistakenly in write operation.

- *For the details of bit-band operations, see the Cortex-M3 Technical Reference Manual, because this bit-band operations has the compatibility with Cortex-M3.*

■ Priority Level

A priority of the bus right is determined in round-robin scheme.

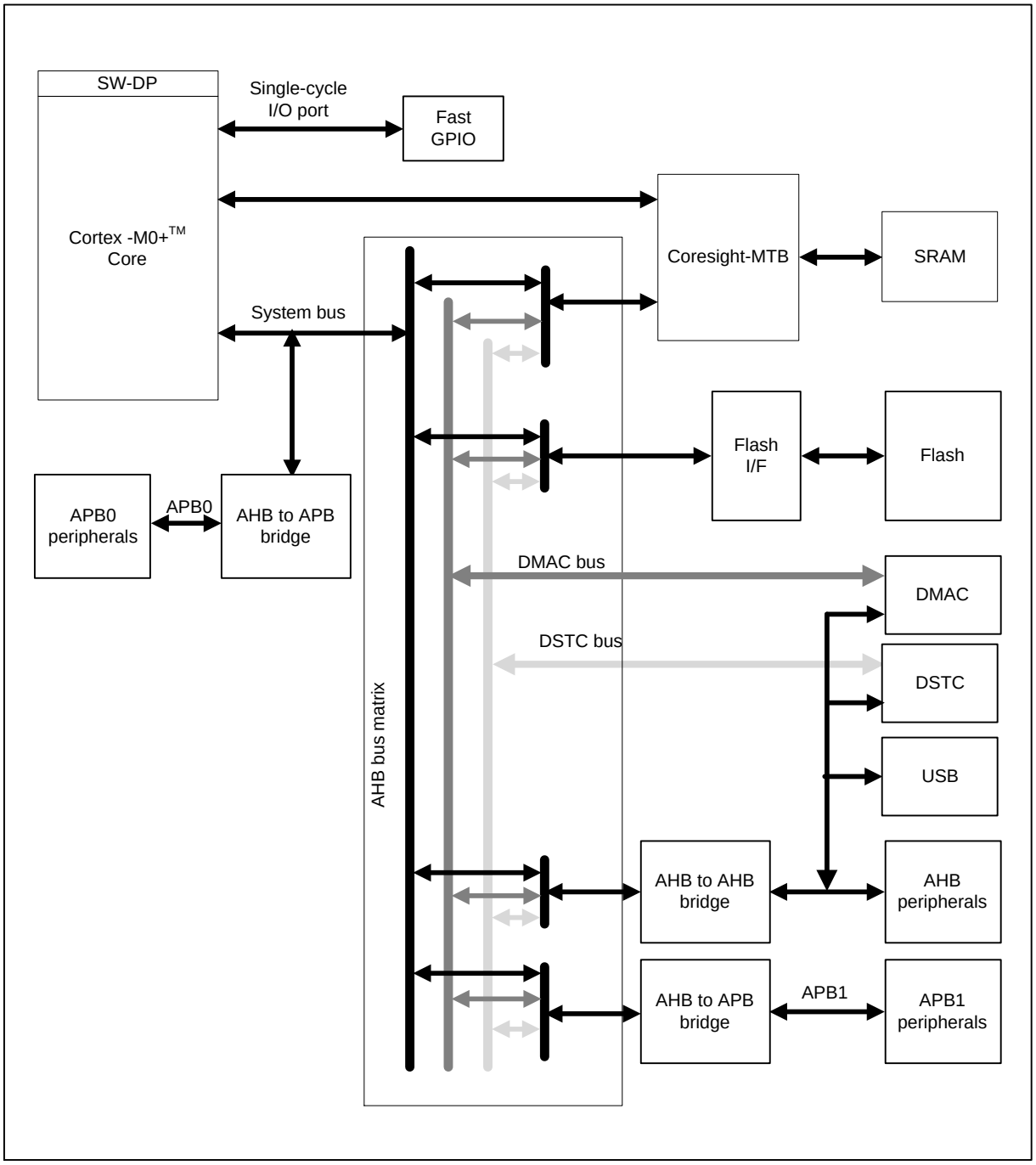
■ Endian

This family uses little endian byte order.

1.1 Bus Block Diagram

Figure 1-1 illustrates the bus block diagram.

Figure 1-1 Bus Block Diagram



Note:

- *There are some areas which no DMA transfer can be performed. For details, see the notes in "1.3 Memory Map" and DMA Transfer column in Table 1-1.*

1.2 Memory Architecture

This section shows the memory architecture.

For this family, 4 GB address space is available.

A Flash memory area up to 1 MB in size and an SRAM area up to 512 KB in size are defined.

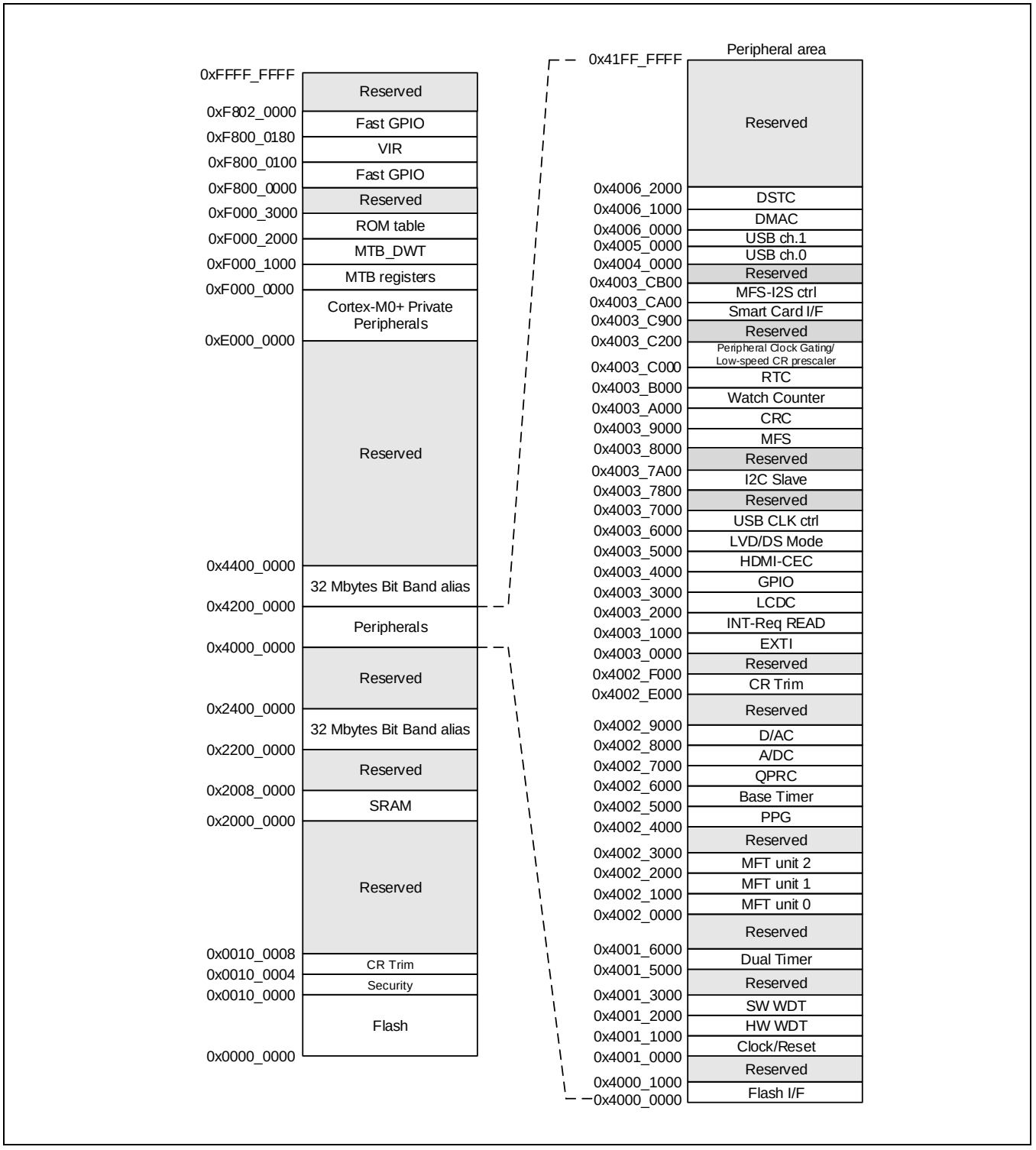
Section "1.3 Memory Map" illustrates the memory map, and Section "1.4 Peripheral Address Map" illustrates the peripheral address map.

For the details of Cortex-M0+ private peripheral shown in Figure **1**-**2**, see "Cortex-M0+ Technical Reference Manual".

1.3 Memory Map

Figure **1-2** illustrates the memory map.

Figure 1-2 **Memory Map**



Notes:

- *Do not access to a reserved area.*
- *For details of the flash memory, see "Flash Programing Manual" of the product used.*
- *Do not perform DMA transfer to following area.*
  - *Bit Band Alias area*
  - *Fast GPIO*
  - *VIR*
  - *ROM table*
  - *MTB_DWT*
  - *MTB registers(SFR)*
  - *Coretex-M0+ Private Peripherals*

1.4 Peripheral Address Map

Table 1-1 shows the peripheral address map.

Table 1-1 Peripheral Address Map

Start Address	End Address	Bus	DMA Transfer	Peripheral	Register Map	CHAPTER
0x4000_0000	0x4000_0FFF	AHB	Disabled	FLASH IF Register / Unique ID Register	FLASH_IF/ Unique ID	* Chapter15
0x4000_1000	0x4000_FFFF			Reserved	-	-
0x4001_0000	0x4001_0FFF	APB0	Disabled	Clock and Reset Control	Clock / Reset	Chapter 2-1 Chapter 3 Chapter 4
0x4001_1000	0x4001_1FFF			Hardware Watchdog Timer	HWWDT	Chapter 1 in Timer Part
0x4001_2000	0x4001_2FFF			Software Watchdog Timer	SWWDT	
0x4001_3000	0x4001_4FFF			Reserved	-	-
0x4001_5000	0x4001_5FFF			Dual Timer	Dual_ Timer	Chapter 2 in Timer Part
0x4001_6000	0x4001_FFFF			Reserved	-	-
0x4002_0000	0x4002_0FFF	APB1	Enabled	Multi-function Timer unit 0	MFT	Chapter 6 in Timer Part
0x4002_1000	0x4002_1FFF			Multi-function Timer unit 1	MFT	
0x4002_2000	0x4002_2FFF			Multi-function Timer unit 2	MFT	
0x4002_3000	0x4002_3FFF			Reserved	-	-
0x4002_4000	0x4002_4FFF			PPG	PPG	Chapter 7-1 Chapter 7-2 Chapter 7-3 in Timer Part
0x4002_5000	0x4002_5FFF			Base Timer	Base Timer/ Base Timer Selector	Chapter 5-1 Chapter 5-2 in Timer Part
0x4002_6000	0x4002_6FFF			QPRC	QPRC	Chapter 8-1 Chapter 8-2 in Timer Part
0x4002_7000	0x4002_7FFF			A/D Converter	A/DC	Chapter 1 in Analog Macro Part
0x4002_8000	0x4002_8FFF			D/A Converter	D/AC	Chapter 2 in Analog Macro Part
0x4002_9000	0x4002_DFFF			Reserved	-	-
0x4002_E000	0x4002_EFFF			High speed CR trimming	CR Trim	Chapter 2-3
0x4002_F000	0x4002_FFFF			Reserved	-	-

Start Address	End Address	Bus	DMA Transfer	Peripheral	Register Map	CHAPTER
0x4003_0000	0x4003_0FFF	APB1	Enabled	External Interrupt	EXTI	Chapter 8
0x4003_1000	0x4003_1FFF			Interrupt Source Check Register	INT-Req READ	Chapter 7
0x4003_2000	0x4003_2FFF			LCDC	LCDC	Chapter 3 in Analog Macro Part
0x4003_3000	0x4003_3FFF			GPIO	GPIO	Chapter 10
0x4003_4000	0x4003_4FFF			HDMI-CEC/Remote Control Reception	HDMI-CEC	Chapter 3 in Communication Macro Part
0x4003_5000	0x4003_50FF			Low Voltage Detection	LVD	Chapter 5
0x4003_5100	0x4003_5FFF			Deep standby control block	DS_Mode	Chapter 6
0x4003_6000	0x4003_6FFF			USB clock generation block/USB	USB Clock	Chapter 4 in Communication Macro Part
0x4003_7800	0x4003_79FF			I2C Slave Wakeup	I2CSLAVE	Chapter 7 in Communication Macro Part
0x4003_7A00	0x4003_7FFF			Reserved	-	-
0x4003_8000	0x4003_8FFF			Multi-function serial	MFS	Chapter 1-2 Chapter 1-3 Chapter 1-4 Chapter 1-5 in Communication Macro Part
0x4003_9000	0x4003_9FFF			CRC	CRC	Chapter 11
0x4003_A000	0x4003_AFFF			Watch counter	Watch Counter	Chapter 3 in Timer Part
0x4003_B000	0x4003_BFFF			Real time clock	RTC	Chapter 4 in Timer Part
0x4003_C000	0x4003_C1FF			Peripheral Clock Gating /Low Speed CR Prescaler	Peripheral Clock Gating	Chapter 2-2 Chapter 2-4
0x4003_C200	0x4003_C8FF			Reserved	-	-
0x4003_C900	0x4003_C9FF			IC Card (Smart Card) Interface	Smart Card I/F	Chapter 6 in Communication Macro Part
0x4003_CA00	0x4003_CAFF			I^2S Clock Generator	MFS-I2S ctrl	Chapter 1-6 in Communication Macro Part
0x4003_CB00	0x4003_FFFF			Reserved	-	-
0x4004_0000	0x4005_FFFF	AHB	Enabled	USB ch.0/ch.1	USB	Chapter 5 in Communication Macro Part
0x4006_0000	0x4006_0FFF			DMAC	DMAC	Chapter 9

Start Address	End Address	Bus	DMA Transfer	Peripheral	Register Map	CHAPTER
0x4006_1000	0x4006_1FFF			DSTC	DSTC	Chapter 16
0x4006_4000	0x41FF_FFFF			Reserved	-	-

*: For the details of "Flash IF Register", see "Flash Programing Manual" of the product used.

2. Cortex-M0+ Architecture

This section explains the core architecture used in this family.

The Cortex-M0+ core block architecture used in this family is as follows:

- Cortex-M0+ Core
- NVIC
- Data watchpoint unit
- BPU
- MTB
- SW-DP
- ROM Table
- Single-cycle I/O port

*: The architecture varies depending on the products. For details, see “2.1Option Configuration”.

Cortex-M0+ Core

This family is equipped with a highly energy-efficient 32-bit processor core (Arm Cortex-M0+ core).
This peripheral manual does not describe the details of Cortex-M0+ core.
For the details, see "Cortex-M0+ Technical Reference Manual".

- Cortex-M0+ core version

For the version of Cortex-M0+ core, see "data sheet" of the product used.

NVIC (Nested Vectored Interrupt Controller)

For this family, one NMI (non-maskable interrupt) and maximum 32 peripheral interrupts (IRQ0 to IRQ31)*1 can be used.
In addition, the interrupt priority register (from 0xE000E400) has 2 bits, and 4 interrupt priority levels can be configured.

For the details of peripheral interrupts, see the chapter of the target "Interrupts" after check the product currently used with "Configuration of interrupts", and for NMI operations, see also another chapter "External Interrupt and NMI Control Block".

The NMIX pin is shared with a general-purpose port. After a reset is released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the port setting. For details, see Chapter “I/O port” and “External Interrupt and NMI Controller”.

*1: Cortex-M0+ Technical Reference Manual defines an exception type: IRQ as an external interrupt. In this peripheral manual, to distinguish from an interrupt by an external pin "External Interrupt and NMI Control Block", the exception type: IRQ is indicated as a peripheral interrupt.

When other interruption with a high priority is processing, the interrupt request from peripheral functions notified to NVIC is pended inside NVIC. When canceling the interrupt request which was pended inside NVIC, clear the interrupt request from peripheral functions, and clear the interrupt request which was pended inside NVIC by the Interrupt Clear-Pending Registers (addresses: 0xE000E280) installed in the NVIC.

SysTick Timer

SysTick Timer is a system timer for OS task management integrated into NVIC.
This family generates STCLK through dividing HCLK by eight and sets the values of SysTick Calibration Value Register (Address: 0xE000E01C) as shown below:

bit31	:	NOREF = 0
bit30	:	SKEW = 1
bit23:0	:	TENMS = 0x00C350 (50000)*1

*1: TENMS value is set to a value which becomes 10 ms when 1/8 clock of HCLK is input to STCLK and that HCLK is in 40 MHz (5 MHz in 1/8 case).
The value of TENMS is not always 10ms because HCLK can be changed to another frequency in the clock control block. Therefore, it is required to calculate an appropriate interrupt timing according to HCLK frequency.

Data Watchpoint Unit

This family is equipped with watchpoint to use as the debug function.
Data watchpoint unit contains two comparators can be use as data address matching and instruction address matching.

BPU (Breakpoint Unit)

BPU provides support for breakpoint functionality on instruction fetches.

MTB (Micro Trace Buffer)

This family is equipped with a Cortex-M0+ optional component MTB to support instruction trace.
MTB_DWT controls the start/stop of trace.

SW-DP

This family is equipped with SW-DP to support the serial wire protocol.

ROM Table

ROM table provides the address information of a debug component to an external debug tool.

Single-Cycle I/O Port

This family is equipped with Single-cycle I/O port to use as the high speed access to tightly-coupled peripherals.

2.1 Option Configuration

Table2-1 shows the option configuration of this series for Cortex-M0+ core.

Table 2-1 Option Configuration

Feature	TYPE1-M0+ TYPE2-M0+ TYPE3-M0+
Interrupts	32
Data endianness	Little-endian
SysTick timer	Present
Number of watchpoint comparators	2
Number of breakpoint comparators	4
Halting debug support	Present
Multiplier	Fast
Single-cycle I/O port	Present
Wake-up interrupt controller	Not present
Vector Table Offset Register	Present
Unprivileged/Privileged support	Not present
Memory Protection Unit	Not present
Reset all registers	Present
Instruction fetch width	32-bit
MTB	Present
Debug port	Serial wire (SW-DP) only
Serial wire multi drop support	Not present

3. Mode

This section explains the function of operating modes.

In this family, the following operating modes can be used:

- User Mode

Internal ROM (Flash) Startup: CPU obtains a reset vector from Flash memory and starts operations.

- Serial Writer Mode

Flash serial write is enabled.

*: For the details of this mode, see "Flash Programing Manual" of the product used.

Operating modes are determined after a release of respective power-on reset, low voltage detection reset, and INITX pin input reset.

*: For the details of power consumption control and clock selection modes, see other chapters "Low Power Consumption Mode" and "Clock".

How to Set Operating Mode

Operating modes are configured by MD pin (MD0) input.

TYPE1-M0+ products

MD Pins	Operating Mode
MD0	
0	User Mode Internal ROM(Flash) Startup
1	Serial Writer Mode

Products other than TYPE1-M0+

MD Pins		Operating Mode
MD1	MD0	
-	0	User Mode Internal ROM(Flash) Startup
0	1	Serial Writer Mode
1	1	Setting is prohibited

Startup Sequence

Processes to determine operating modes in the startup sequence are shown below:

1. MD Pin Sampling
2. Determining Operating Mode and Retaining Mode Data

The descriptions of these processes are as follows:

1. MD Pin Sampling

Operating mode is configured by MD pin input (MD0/MD1). This input is sampled by power-on reset, low-voltage detection reset, and INITX pin input reset.
Determine the MD0/MD1 pin input before a reset that is the sampling factor is released.

2. Determining Operating Mode and Retaining Mode Data

MD0/MD1, which is sampled by a reset, is retained until another reset is input again.
Operating modes are determined by the retained MD0/MD1. Therefore, even MD0/MD1 is changed after a reset is released, it does not affect an operating mode.

CHAPTER 2-1: Clock

This chapter explains the operating clock.

1. Overview
2. Configuration
3. Operations
4. Clock Setup Procedure Examples
5. Registers
6. Usage Precautions

CODE: 9AFCLOCK-E03.0

1. Overview

This section provides an overview of the clock generation unit.

The clock generation unit generates various types of clocks used to operate the MCU.

Source clock is the generic name for external and internal oscillation clocks of this MCU.

The following five types of clocks are source clocks:

- Main clock (CLKMO)
- Sub clock (CLKSO)
- High-speed CR clock (CLKHC)
- Low-speed CR clock (CLKLC)
- Main PLL clock (CLKPLL)

Select one from the source clocks. In this chapter, the selected clock is referred to as the master clock. The master clock is a source of internal bus clocks used to operate this MCU.

Dividing the master clock frequency can generate a base clock. In addition, dividing the base clock can generate each bus clock.

In this chapter, the base clock and bus clocks are referred to as internal bus clocks. The following three types of clocks are internal bus clocks:

- Base clock (HCLK/FCLK)
- APB0 bus clock (PCLK0)
- APB1 bus clock (PCLK1)

In addition to source clocks, the master clock, and internal bus clocks, the following clocks are provided:

- PLLOUT clock for USB (TYPE3-M0+)
- USB clock (TYPE2-M0+, TYPE3-M0+)
- Software watchdog timer count clock

The following shows the features of the clock generation unit.

- It can set the oscillation stabilization wait time of the main clock (CLKMO).
- It can set the interrupt which generates at completing the oscillation stabilization wait time of the main clock (CLKMO).
- It can set the oscillation stabilization wait time of the sub clock (CLKSO).
- It can set the interrupt which generates at completing the oscillation stabilization wait time of the sub clock (CLKSO).
- It can set the oscillation stabilization wait time of the main PLL clock (CLKPLL).
- It can set the interrupt which generates at completing the oscillation stabilization wait time of the main PLL clock (CLKPLL).
- It can set the PLL multiplication ratio.
- It can select the master clock.
- It can set the frequency division ratio of each internal bus clock frequency.
- It can select run or stop of the APB1 bus clock.
- It can set the frequency division ratio of the software watchdog timer count clock frequency.
- It can set run/stop of the software watchdog timer count clock.
- It can set the watchdog timer count operation in debug mode.

- It includes registers for enabling clock-related interrupts, checking interrupt status, and clearing interrupt factors.

2. Configuration

This section explains configuration of the clock generation unit.

Source Clock

Source clock is the generic name for external and internal oscillation clocks of this MCU. The following five types of clocks are source clocks:

- Main clock (CLKMO)

CLKMO is generated by connecting a crystal oscillator etc. to the main clock oscillation pins (X0, X1), or input using an external clock.

- Sub clock (CLKSO)

CLKSO is generated by connecting a crystal oscillator etc. to the sub clock oscillator pins (X0A, X1A), or input using an external clock.

- High-speed CR clock (CLKHC)

CLKHC is an output clock for the high-speed CR oscillator.

- Low-speed CR clock (CLKLC)

CLKLC is an output clock for the low-speed CR oscillator.

(Notes) In TYPE1,2 product, the low-speed CR clock is a clock after a prerscaler.
For details on the low-speed CR clock prescaler, see Chapter "Low-speed CR Clock Prescaler".

In TYPE3 product, the low-speed CR clock is connected the low-speed CR oscillator directly. The low-speed CR clock prescaler is not used.

- Main PLL clock (CLKPLL)

CLKPLL is generated by multiplying the main oscillation clock or high-speed CR clock using the PLL Clock Multiplication Circuit (PLL Oscillation Circuit).

Master Clock

The signal selected from source clocks are referred to as the master clock.The master clock is a source for all bus clocks. The master clock value should not be larger than the maximum value in "Internal operating clock frequency: Fcc (Base clock HCLK/FCLK)"of data sheet.

Note: See "1. Notes when high-speed CR is used for the master clock" in "E. List of Notes" when you use the following clock for the master clock.

- High-speed CR clock
- Main PLL clock (When selecting high-speed CR clock for the input clock of PLL)

Internal Bus Clocks

The following signals are bus clocks generated internally.

- Base clock (HCLK/FCLK)

HCLK and FCLK are collectively called the base clock. Both HCLK and FCLK are supplied to the CPU.
HCLK is a clock for macro connected to the AHB bus.
The clock frequency can be set to between 1/1 and 1/16 frequency of the master clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop

mode.
In sleep mode, the CPU stops the supply of HCLK while continuing the supply of FCLK.

- APB0 bus clock (PCLK0)

PCLK0 is a clock for peripheral macro connected to the APB0 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.

- APB1 bus clock (PCLK1)

PCLK1 is a clock for peripheral macro connected to the APB1 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.
The supply of the clock can be also stopped by setting a register.

Clocks Other than Source Clocks and Internal Bus Clocks

- PLLOUT clock for USB (TYPE3-M0+)

In addition to using this clock internally in main PLL block, TYPE3-M0+ product with USB can use this clock as a source of USB clock. See Chapter “USB Clock Generation” in Communication Macro part.

- USB clock (TYPE2-M0+,TYPE3-M0+)

This clock is a 48MHz clock used for USB Communication. For details about how to create the clock and use it, see Chapter “USB Clock Generation” in Communication Macro part.

- Software watchdog timer count clock (SWDOGCLK)

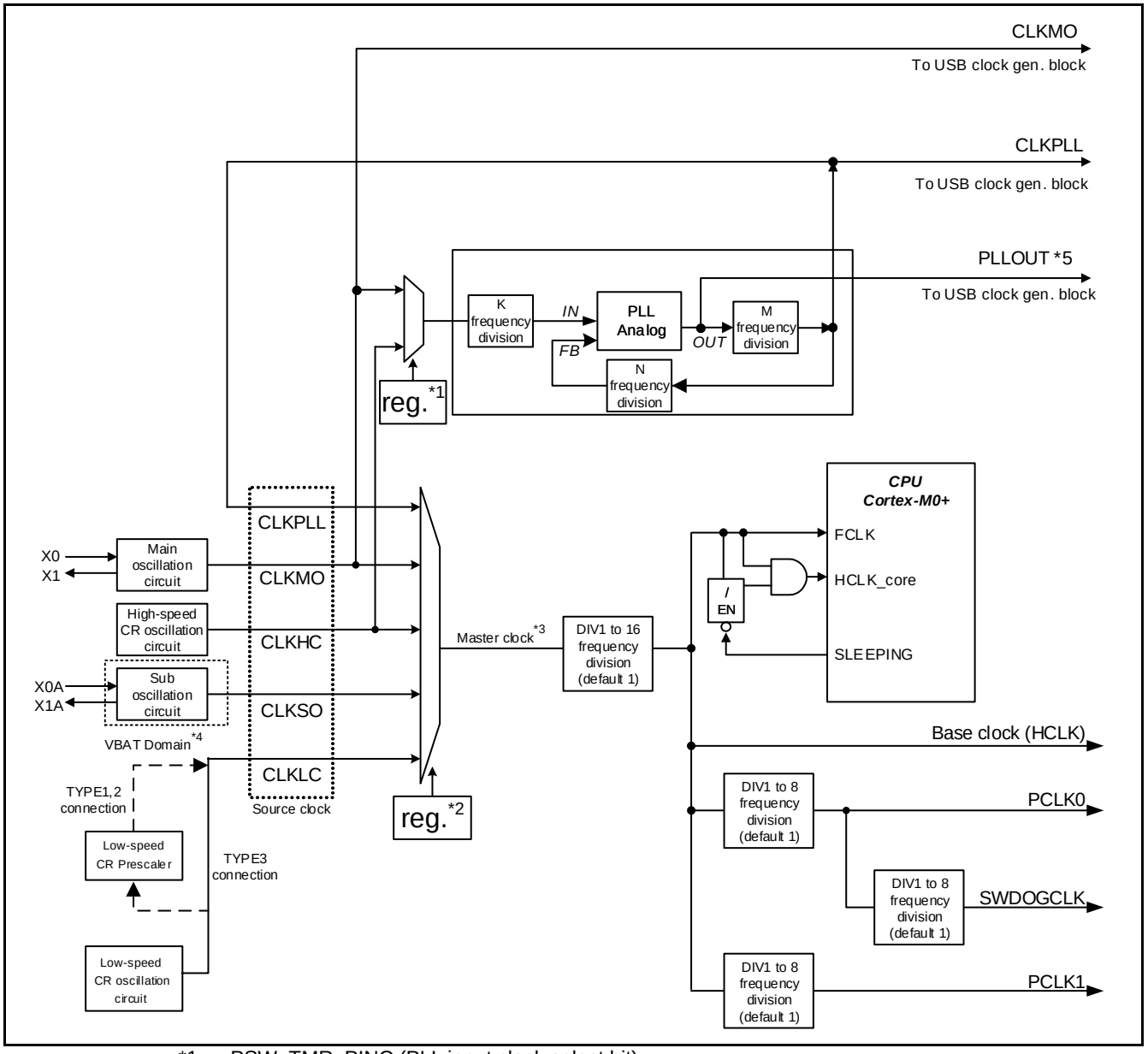
SWDOGCLK is a clock for the software watchdog timer connected to the APB0 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the APB0 bus clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.

For operation settings of the software watchdog timer, see Chapter "Watchdog Timer" in "Timer part".

Block Diagram

Figure 2-1 shows the block diagram of the clock generation unit.

Figure 2-1 Block Diagram of Clock Generation Unit



*1: PSW_TMR: PINC (PLL input clock select bit)

*2: SCM_CTL: RCS[2:0] (Master clock switch control bits)

*3: The master clock frequency should not be larger than the maximum frequency of base clock (HCLK/FCLK). For the maximum frequency of base clock (HCLK/FCLK), see data sheet of the product used.

*4: In TYPE2-M0+ product, Sub Oscillation Circuit is in different independent power domain. For details, please refer to Chapter “VBAT Domain”

*5: In TYPE3-M0+ product, PLLOUT clock is used as a source clock of USB clock.

3. Operations

This section explains the clock generation unit.

3.1 Selecting Clock Mode

Definition of Clock Mode (Selecting the Master Clock)

The MCU clock mode is defined by the source clock selected by the system clock mode control register. Five types of clock modes are provided: Main clock mode, sub clock mode, high-speed CR clock mode, low-speed CR clock mode, and main PLL clock mode.

- Main clock mode

  In main clock mode, the main clock (CLKMO) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

  The status of the PLL clock (CLKPLL) depends on the setting of the PLLE bit in the System Clock Mode Control Register (SCM_CTL), the status of the sub clock (CLKSO) on the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL), and the status of the high-speed CR clock (CLKHC) on the setting of the HCRE bit in the System Clock Mode Control Register (SCM_CTL). Besides, the MCSVE/FCSDE in CSV_CTL also activate the high-speed CR clock (CLKHC).The low-speed CR clock (CLKLC) cannot be stopped by user program.

- Sub-clock mode

  In sub clock mode, the sub clock (CLKSO) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

  The main clock (CLKMO), high-speed CR clock (CLKHC), and main PLL clock (CLKPLL) are stopped by hardware. The low-speed CR clock (CLKLC) cannot be stopped by user program.

- High-speed CR clock mode

  In high-speed CR clock mode, the high-speed CR clock (CLKHC) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

  Statuses of the main clock (CLKMO), main PLL clock (CLKPLL), and sub clock (CLKSO) differ depending on the settings of MOSCE, PLLE, and SOSCE bits in the System Clock Mode Control Register (SCM_CTL). The high-speed CR clock (CLKHC) and low-speed CR clock (CLKLC) cannot be stopped by user program.

- Low-speed CR clock mode

  In low-speed CR clock mode, the low-speed CR clock (CLKLC) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

  In low-speed CR clock mode, the main clock (CLKMO), high-speed CR clock (CLKHC), and main PLL clock (CLKPLL) are stopped by hardware. Status of the sub clock (CLKSO) differs depending on the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL).

- Main PLL clock mode

  In main PLL clock mode, the main PLL clock (CLKPLL) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

  Status of the sub clock (CLKSO) differs depending on the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL). The status of the high-speed CR clock (CLKHC) depends on the setting of some registers. For details, see Chapter "Low Power Consumption Mode". The status of main clock (CLKMO) depends on the setting of PINC in PSW_TMR or MOSCE in SCM_CTL. The low-speed CR clock (CLKLC) cannot be stopped by user program.

3.2 Internal Bus Clock Frequency Division Control

This section explains the internal bus clock frequency division.

Frequency division ratio from the base clock can be set independently for each internal bus clock. This function can set the operating frequency optimized for each circuit.

The maximum frequency of the internal bus clock differs by product. For details, see "Data Sheet" of the product used.

To set the frequency division ratio of internal bus clocks, use the Base Clock Prescaler Register (BSC_PSR), APB0 Prescaler Register (APBC0_PSR) and APB1 Prescaler Register (APBC1_PSR). For details of each register, see 5. Registers.

Setting the Bus Clock Frequency Division

- The set frequency division ratio is not cleared by a software reset. The latest value is retained before the software reset.
- The value is initialized by a reset other than software resets.
  Before changing the initially set master clock to a faster source clock, be sure to set the frequency division ratio.
- If a combined value of master clock, PLL multiplication, and frequency division ratio settings exceeds the maximum operating frequency of each internal bus, the operation corresponding to the setting is not guaranteed.

3.3 PLL Clock Control

This section explains the PLL clock control.

The PLL Clock Control Circuit is used to generate the main PLL clock from the main clock or high-speed CR clock. The PLL Oscillation Circuit can enable/disable operation (oscillation), select the input clock, set the stabilization wait time, and set the multiplication.

PLL Operation

The following explains operation of the main PLL clock.

- Configure the following settings using the PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR).
  - Selecting the PLL input clock
  - Setting the main PLL clock stabilization wait time
- The PLL oscillation enable bit (PLLE) of the System Clock Mode Control Register (SCM_CTL) must be enabled to let the PLL Circuit start oscillating.
- When the PLL clock stabilization wait time has elapsed, and the PLL oscillation stable bit of the System Clock Mode Status Register (SCM_STR) indicates a stable state, the preparation for transition to main PLL clock mode completes.
- Master clock switch control bit (RCS[2:0]) of the System Clock Mode Control Register (SCM_CTL) must be set to main PLL clock mode (RCS[2:0]=010) to change to main PLL clock mode.

Setting the Main PLL Clock Oscillation Stabilization Wait Time

The details are given in 5.8 PLL Clock Stabilization Wait Time Setup Register (PSW_TMR).

Notes:

- *For block diagram of the PLL Clock Control Circuit, see 2.Configuration.*
- *For the order of frequency division settings for each internal bus clock, see 4 Clock Setup Procedure Examples.*
- *For the oscillation stabilization wait time, see 3.4 Oscillation Stabilization Wait .*
- *When selecting high-speed CR in the input clock of PLL, see 1. Notes when high-speed CR is used for the master clock in Appendixes E. List of Notes.*

Setting the Multiplication Ratio to Generate the Main PLL Clock

Each frequency division clock in the PLL Multiplication Circuit must be set using PLL Control Register 1 (PLL_CTL1) and PLL Control Register 2 (PLL_CTL2). Table 3-1 and Table 3-2 provide an example of frequency division settings.

Table 3-1 Example of PLL Multiplication Ratio Settings (TYPE1-M0+, TYPE2-M0+)

Input Clock	K	PLLin	N	PLLout	M	CLKPLL
4 MHz	1	4 MHz	2	80 MHz	10	8 MHz
4 MHz	1	4 MHz	4	80 MHz	5	16 MHz
4 MHz	1	4 MHz	5	80 MHz	4	20 MHz
4 MHz	1	4 MHz	6	120 MHz	5	24 MHz
4 MHz	1	4 MHz	9	108 MHz	3	36 MHz
4 MHz	1	4 MHz	10	80 MHz	2	40 MHz
8 MHz	1	8 MHz	5	80 MHz	2	40 MHz
8 MHz	2	4 MHz	10	80 MHz	2	40 MHz
12 MHz	3	4 MHz	10	80 MHz	2	40 MHz
16 MHz	2	8 MHz	5	80 MHz	2	40 MHz
16 MHz	4	4 MHz	10	80 MHz	2	40 MHz
24 MHz	3	8 MHz	5	80 MHz	2	40 MHz

Table 3-2 Example of PLL multiplication ratio settings (TYPE3-M0+)

Input clock	K	PLLin	N	PLLout	M	CLKPLL
8 MHz	1	8 MHz	2	80 MHz	5	16 MHz
8 MHz	1	8 MHz	3	96 MHz	4	24 MHz
8 MHz	1	8 MHz	4	96 MHz	3	32 MHz
8 MHz	1	8 MHz	5	80 MHz	2	40 MHz
12 MHz	1	12 MHz	3	144 MHz	4	36 MHz
16 MHz	2	8 MHz	5	80 MHz	2	40 MHz
24 MHz	3	8 MHz	5	80 MHz	2	40 MHz
48MHz	6	8MHz	5	80MHz	2	40 MHz

Notes:

- *For PLL characteristics, see data sheet of the product used.*
- *Set the PLLin within the value PLL input clock frequency: f_{PLL} shown in the data sheet.*
- *The value M×N is a multiplication ratio for the PLLin. Set this value within the range shown in the "PLL multiple rate" of the data sheet.*
- *The frequency of the PLLin multiplied by M×N becomes PLLout. Set this value within the range shown in the PLL macro oscillation clock frequency: f_{PLLO} of the data sheet.*
- *The value of the PLLout divided by M becomes CLKPLL.*
- *See Figure 2-1 for the configurations of PLL and divider.*
- *The master clock / CLKPLL value should not be larger than the maximum value in "Internal operating clock frequency: Fcc (Base clock HCLK/FCLK) of data sheet.*
- *In TYPE3-M0+ product, PLLout clock can be used as a source clock for USB clock generation. For details, see USB clock generation section.*

3.4 Oscillation Stabilization Wait Time

This section explains the oscillation stabilization wait time.

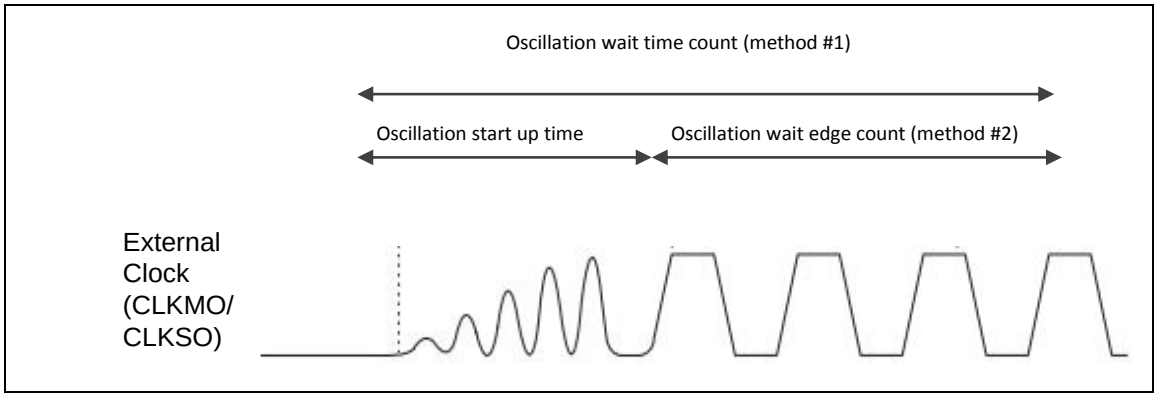
An oscillation stabilization wait time is required if the source clock is not in a stable operating state. During the oscillation stabilization wait time, internal and external clocks stop the supply, There are two method to wait until the stabilization wait time passes, a time value set in the Clock Stabilization Wait Time Register (CSW_TMR) or PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR). When the wait time has been passed, the corresponding oscillator is ready to operate, and the clock can be used as a master clock.

Method for Clock Stabilization Wait Count.

- There are two kind methods to count external clock (CLKMO/CLKSO) stabilization time.
  1. The one of method is that a time of external clock (CLKMO/CLKSO) stabilization is counted by internal CR clock (CLKHC/CLKLC). In that way, after a particular time exceeded, external clock (CLKMO/CLKSO) becomes stable.
  2. The other method is that a number of the external clock (CLKMO/CLKSO) positive edges are counted for themselves wait count.
- In each those methods there are selection how long time / how many clock edge must be counted to it assumes the external clock becomes stable.

- Below is the example of clock stabilization time / number of edges selection.

Figure 3-1 Oscillation Wait Count Method.



- The clock stabilization method is selected by whether clock supervisor is enabled or disabled.

  CSV_CTL.FCSDE=0 and CSV_CTL.MCSVE=0: Number of CLKMO clock edge count method 2 is selected.

  CSV_CTL.FCSDE=1 or CSV_CTL.MCSVE=1: Time counted by CLKHC method 1 is selected.

  CSV_CTL.SCSVE=0: Number of CLKSO clock edge count method 2 is selected.

  CSV_CTL.SCSVE=1: Time counted by CLKLC method 1 is selected.

Order of Priority for Oscillation Stabilization Wait Times

If some clocks start oscillation by mode transition, the clock controller counts the respective oscillation stabilization wait times of clocks according to a designated order of priority.

- Transition to Main clock mode
  - If MCSV or FCS are enabled

    Low-speed CR -> Sub OSC -> High-speed CR -> PLL(High-speed CR input) -> Main OSC -> PLL(Main OSC input)
  - If MCSV and FCS are disabled.

    Low-speed CR -> Sub OSC -> Main OSC -> PLL(Main OSC input) -> High-speed CR -> PLL(High-speed CR input)
- Transition to Sub clock mode

  Low-speed CR -> Sub OSC
- Transition to High-speed CR clock mode

  Low-speed CR -> Sub OSC -> High-speed CR -> PLL (High-speed CR input) -> Main OSC -> PLL(Main OSC input)
- Transition to Low-speed CR clock mode

  Low-speed CR -> Sub OSC
- Transition to Main clock PLL mode
  - If MCSV or FCS are enabled

    Low-speed CR -> Sub OSC -> High-speed CR -> Main OSC-> PLL(Main OSC input)
  - If MCSV and FCS are disabled.

    Low-speed CR -> Sub OSC -> Main OSC -> PLL(Main OSC input) -> High-speed CR
- Transition to High-speed clock PLL mode

  Low-speed CR -> Sub OSC -> High-speed CR -> PLL(High-speed CR input) -> Main OSC

Setting the Oscillation Stabilization Wait Time

- Main clock (CLKMO)

  Set the stabilization wait time of the main clock using the Clock Stabilization Wait Time Register (CSW_TMR). The set time value is counted by CLKHC (if Clock supervisor for CLKMO is enabled)/CLKMO (if Clock supervisor for CLKMO is disabled).
- Sub clock (CLKSO)

  Set the stabilization wait time of the sub clock using the Clock Stabilization Wait Time Register (CSW_TMR). The set time value is counted by CLKLC (if CSV for CLKSO is enabled)/CLKSO (if CSV for CLKSO is disabled).
- Main PLL clock

  Configure the following settings using the PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR). The set time value is counted by CLKPLL.
- Selecting the PLL input clock
- Setting the main PLL clock stabilization wait time

Cause of Waiting for Oscillation Stability

- After the oscillation is enabled via software

  If the PLLE, SOSCE, MOSCE and HCRE bits in the System Clock Mode Control Register (SCM_CTL) are set to "1", each relevant oscillator waits during the oscillation stabilization wait time.

- When returning to watch counter interrupt, RTC interrupt, and external interrupt from RTC mode

  It returns to the clock mode before RTC mode by watch counter interrupt, RTC interrupt, and external interrupt. Hardware of a source clock waits for the oscillation stabilization wait time automatically.

- When returning from stop mode using an external interrupt

  The status returns to clock mode, a state before stop mode, using an external clock. During stop mode, all source clocks stop and, therefore, the hardware automatically waits during the oscillation stabilization wait time.

- After PLL operation is enabled

  After PLL operation is enabled, the PLL oscillation stabilization wait time is waited.

Notes:

- *Each set value of the oscillation stabilization wait time must be changed before the clock is enabled.*
- *After software reset, the oscillation stabilization wait time is not applied.*
- *The oscillation stabilization wait completion flag will be activated when the counting is complete. The oscillation stabilization wait time may be completed before oscillator stabilization if the setting of the oscillation stabilization wait time is too short.*
- *As the stabilization wait times for main clock and sub clock oscillators depend on the type of the oscillator (crystal, ceramics, etc.), proper oscillation stabilization wait time must be chosen for the oscillator to be used.*
- *Set the PLL oscillation stabilization wait time by referring to PLL Clock LOCKUP Time of the electric characteristics described in data sheet" of the product used.*

3.5 Interrupt Factors

This section explains interrupt factors relevant to clocks.

The clock generation unit has the following interrupt factors.

Interrupt Factors

The clock generation unit has the following four types of interrupt factors:

- FCS (anomalous frequency detection) interrupt

When the FCS (anomalous frequency detection) is enabled, and an anomalous frequency of the main clock is detected, an interrupt occurs.

- Main PLL clock oscillation stabilization wait completion interrupt

When the main PLL clock oscillation stabilization wait time ends, an interrupt occurs.

- Sub clock oscillation stabilization wait completion interrupt

When the sub clock oscillation stabilization wait time ends, an interrupt occurs.

- Main clock oscillation stabilization wait completion interrupt

When the main clock oscillation stabilization wait time ends, an interrupt occurs.

Registers

The following three types of registers are provided for each interrupt factor:

- Interrupt Enable Register (INT_ENR)

This register enables each interrupt.

- Interrupt Status Register (INT_STR)

This register indicates each interrupt status. This register is read-only.

- Interrupt Clear Register (INT_CLR)

This register clears each interrupt factor. This register is write-only.

4. Clock Setup Procedure Examples

4.1 Setup Procedure Examples (TYPE1-M0+, TYPE3-M0+)

This section explains procedure examples of setting up clocks in TYPE1-M0+, TYPE3-M0+ product.

Figure 4-1 Example of Clock Setup Procedure (Power-on -> High-Speed CR Run Mode -> Desired Clock Mode)

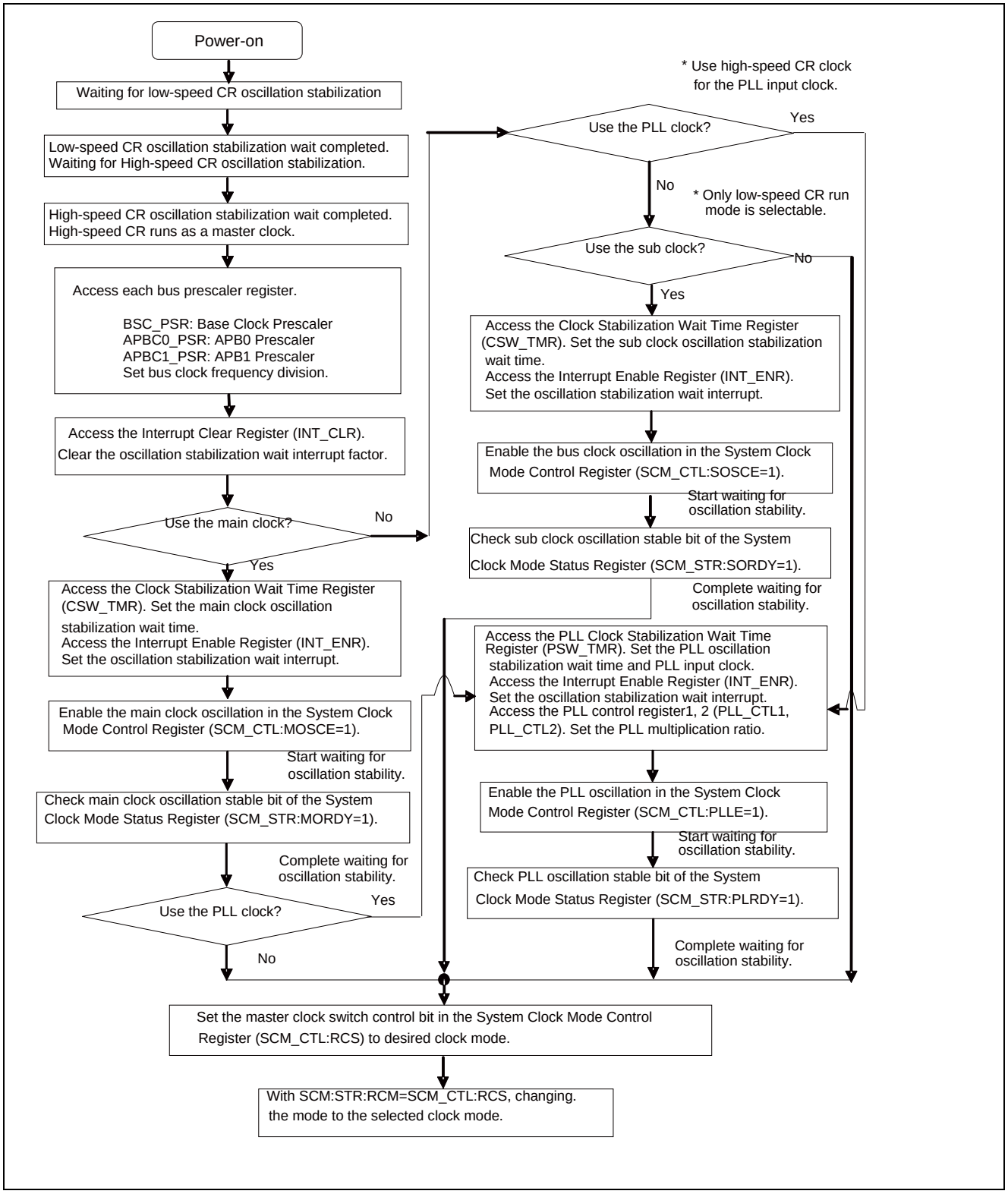
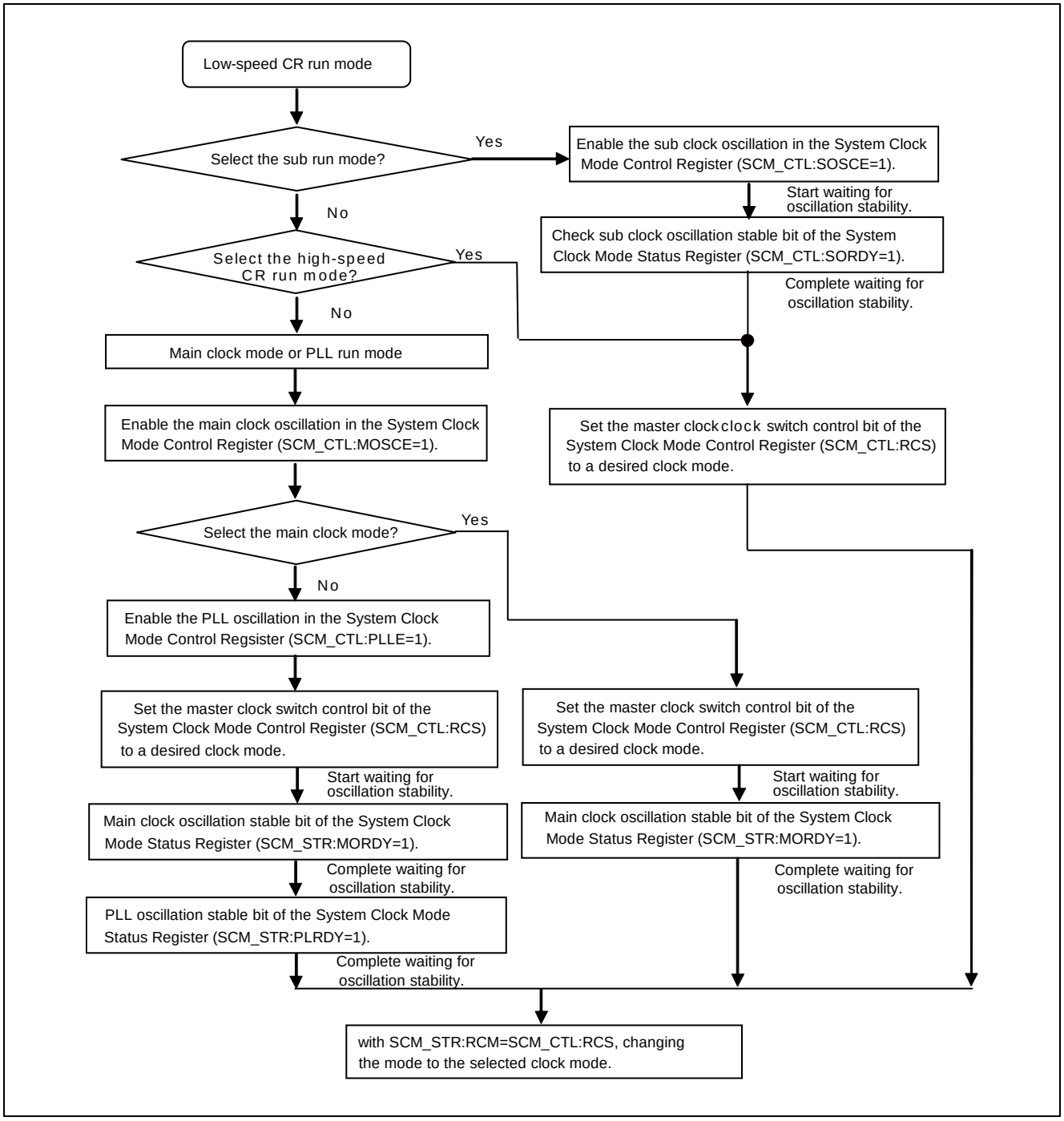


Figure 4-2 Example of Clock Setup Procedure (Low-Speed CR Run Mode -> Desired Clock Run Mode)



Notes:

- *Figure 4-2 assumes that settings of the oscillation stabilization wait time, interrupts, PLL multiplication ratio and bus clock frequency division for each clock have been configured previously, and they are omitted from the flowchart.*

- *In the sub clock mode/low-speed CR clock mode, the main clock (CLKMO), high-speed CR (CLKHC), main PLL clock (CLKPLL) is stopped by hardware. So CLKMO/CLKHC/CLKPLL does not start oscillation only setting oscillation enable bit=1. These oscillations will start by changing the SCM_CTL:RCS bit with setting oscillation enable bit=1.*
- *If the main clock/sub clock oscillation stabilization wait times are short and the oscillation stabilization wait times run out before oscillators stabilize, reset may be applied by the clock supervisor function.*

4.2 Setup Procedure Examples (TYPE2-M0+)

This section explains procedure examples of setting up clocks in TYPE2-M0+ product.

Figure 4-3 Example of Clock Setup Procedure (Power-on -> High-Speed CR Run Mode -> Desired Clock Mode)

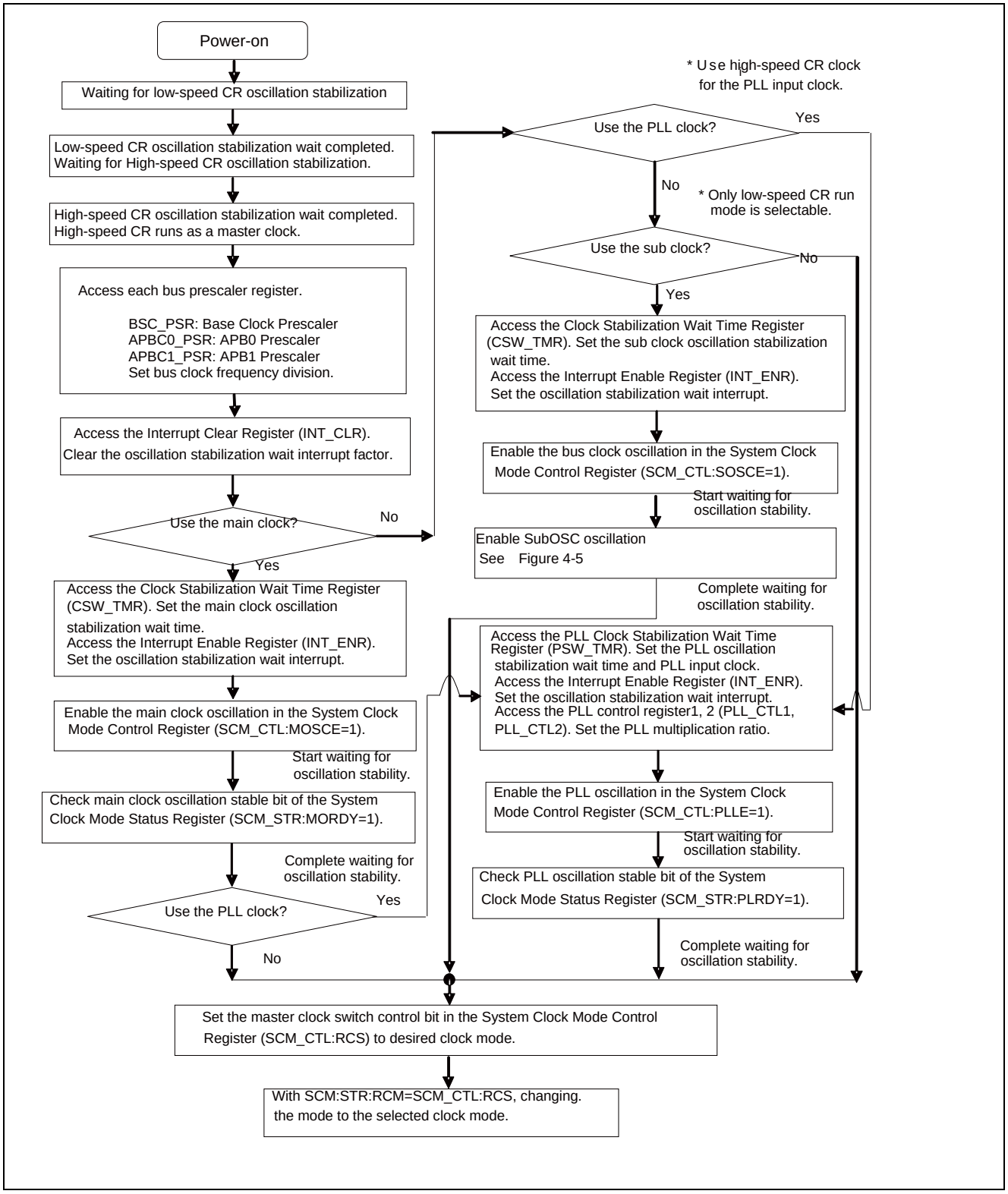
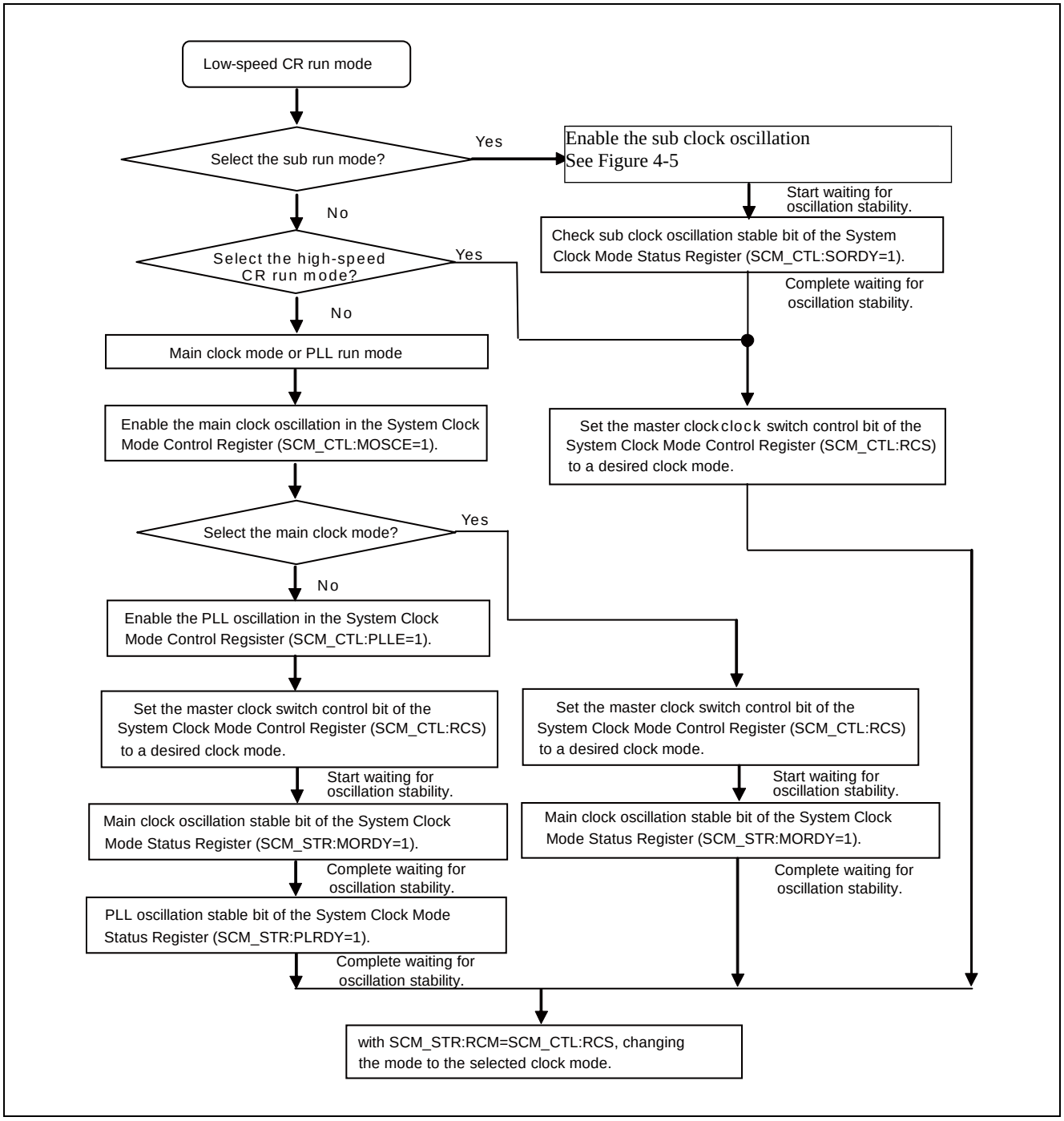


Figure 4-4 Example of Clock Setup Procedure (Low-Speed CR Run Mode -> Desired Clock Run Mode)

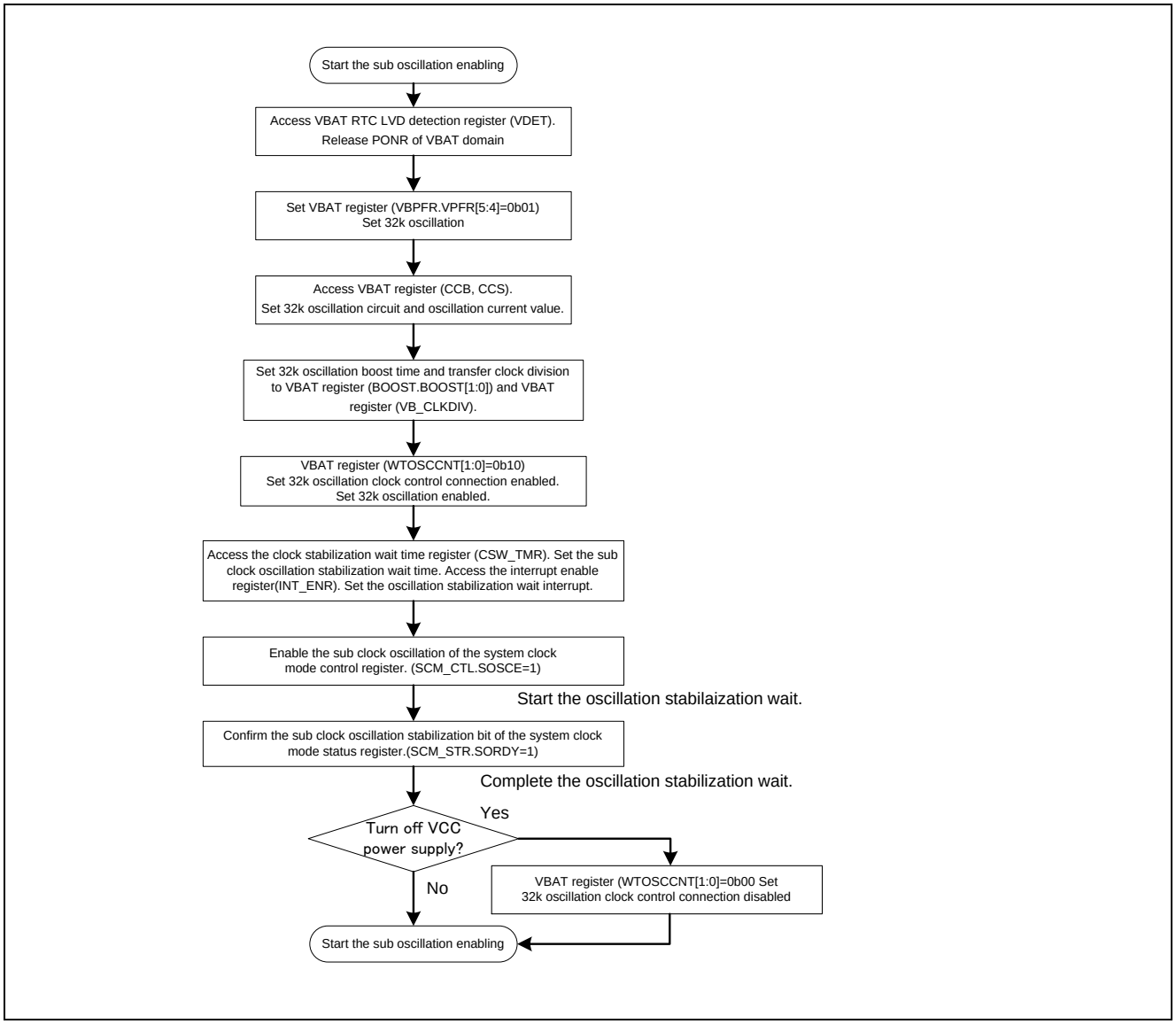


Notes:

- *Figure 4-4 assumes that settings of the oscillation stabilization wait time, interrupts, PLL multiplication ratio and bus clock frequency division for each clock have been configured previously, and they are omitted from the flowchart.*

- *In the sub clock mode/low-speed CR clock mode, the main clock (CLKMO), high-speed CR (CLKHC), main PLL clock (CLKPLL) is stopped by hardware. So CLKMO/CLKHC/CLKPLL does not start oscillation only setting oscillation enable bit=1. These oscillations will start by changing the SCM_CTL:RCS bit with setting oscillation enable bit=1.*
- *If the main clock/sub clock oscillation stabilization wait times are short and the oscillation stabilization wait times run out before oscillators stabilize, reset may be applied by the clock supervisor function.*

Figure 4-5 Example of Sub Oscillation Setup Procedure



Notes:

- *Set the sub clock stabilization wait time (SCM_CTL.SOWT) longer than the VBAT RTC 32 k oscillation boost setting time [BOOST:BOOST[1:0]].*
- *The following combination of settings is prohibited:*

- *When 32k oscillation clock control linkage of VBAT register is disabled (WTOSCCNT.SOSCNTL=0) and 32k oscillation is disabled (WTOSCCNT.SOSCEX=1), the setting combination of sub clock mode oscillation of system clock mode control register enabled (SCM_CTL.SOSCE=1) and sub CSV function of CSV control register enabled (CSV_CTL.SVSVE=1) is prohibited.*

- *For details of VBAT RTC, see Chapter VBAT Domain.*

5. Registers

This section describes the registers of the clock generation unit.

List of Clock Generation Unit Registers

Table 5-1 List of Clock Generation Unit Registers

Abbreviation	Register Name	Reference
SCM_CTL	System Clock Mode Control Register	5.1
SCM_STR	System Clock Mode Status Register	5.2
BSC_PSR	Base Clock Prescaler Register	5.3
APBC0_PSR	APB0 Prescaler Register	5.4
APBC1_PSR	APB1 Prescaler Register	5.5
SWC_PSR	Software Watchdog Clock Prescaler Register	5.6
CSW_TMR	Clock Stabilization Wait Time Register	5.7
PSW_TMR	PLL Clock Stabilization Wait Time Setup Register	5.8
PLL_CTL1	PLL Control Register 1	5.9
PLL_CTL2	PLL Control Register 2	5.10
DBWDT_CTL	Debug Break Watchdog Timer Control Register	5.11
INT_ENR	Interrupt Enable Register	5.12
INT_STR	Interrupt Status Register	5.13
INT_CLR	Interrupt Clear Register	5.14

5.1 System Clock Mode Control Register (SCM_CTL)

The System Clock Mode Control Register (SCM_CTL) selects the master clock and enables/disables the clock oscillation.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	RCS[2:0]			PLLE	SOSCE	Reserved	MOSCE	HCRE
Attribute	R/W			R/W	R/W	-	R/W	R/W
Initial value	000			0	0	-	0	1

Register Functions

[bit7:5] RCS[2:0]: Master clock switch control bits

bit7	bit6	bit5	Description
0	0	0	High-speed CR clock [Initial value]
0	0	1	Main clock
0	1	0	Main PLL clock
0	1	1	Setting is prohibited
1	0	0	Low-speed CR clock
1	0	1	Sub clock
1	1	0	Setting is prohibited
1	1	1	Setting is prohibited

[bit4] PLLE: PLL oscillation enable bit

Bit	Description
0	Disables PLL oscillation [Initial value]
1	Enables PLL oscillation

[bit3] SOSCE: Sub clock oscillation enable bit

Bit	Description
0	Disables sub clock oscillation [Initial value]
1	Enables sub clock oscillation

[bit2] Reserved: Reserved bit

The read value is undefined. These bits have no effect when written.

[bit1] MOSCE: Main clock oscillation enable bit

Bit	Description
0	Disables main clock oscillation [Initial value]
1	Enables main clock oscillation

[bit0] HCRE: High-speed CR clock oscillation enable bit

Bit	Description
0	Disables high-speed CR oscillation.
1	Enables the high-speed CR oscillation. [Initial value]

Notes:

- *This register is not initialized by software reset.*
- *When you change the clock mode, you should set the enable bit to transition for desired clock oscillation. Then, you can change the clock switch control bits (SCM_CTL:RCS[2:0]).*
- *When RTCE bit (PMD_CTL:RTCE) is 1, it becomes a sub clock oscillation enable state regardless of the SOSCE bit and SORDY bit values.*
- *Writing 1 to RTCE bit (PMD_CTL:RTCE) is enabled only when SORDY bit is 1.*
- *RTCE bit (PMD_CTL:RTCE) does not exist in the products that do not have RTC mode and deep standby RTC mode. See Table 1-1 in the Chapter Low Power Consumption Mode.*
- *When FCSDE bit (CSV_CTL:FCSDE) or MCSVE bit(CSV_CTL:MCSVE) is "1", it becomes a High-speed CR clock oscillation enable state regardless of the HCRE bit value.*
- *In TYPE2-M0+ products, first of all, after the power supply is turned on, it is required to set the register of VBAT RTC for enabling the sub-clock oscillation. For the sub-clock oscillation enable, see Figure 4-5.*
- *When any clock is in the stabilization waiting state, don't enable other clocks oscillation.*

5.2 System Clock Mode Status Register (SCM_STR)

The System Clock Mode Status Register (SCM_STR) indicates a clock selected for the master clock and a waiting state for clock oscillation stability.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	RCM[2:0]			PLRDY	SORDY	Reserved	MORDY	HCRDY
Attribute	R			R	R	-	R	R
Initial value	000			0	0	-	0	1

Register Functions

[bit7:5] RCM[2:0]: Master clock selection bits

bit7	bit6	bit5	Description
0	0	0	High-speed CR clock [Initial value]
0	0	1	Main clock
0	1	0	Main PLL clock
0	1	1	Setting is prohibited
1	0	0	Low-speed CR clock
1	0	1	Sub clock
1	1	0	Setting is prohibited
1	1	1	Setting is prohibited

[bit4] PLRDY: PLL oscillation stable bit

Bit	Description
0	In the stabilization wait or the oscillation stop state [Initial value]
1	In the stable state

[bit3] SORDY: Sub clock oscillation stable bit

Bit	Description
0	In the stabilization wait or the oscillation stop state [Initial value]
1	In the stable state

[bit2] Reserved: Reserved bit

The read value is undefined. These bits have no effect when written.

[bit1] MORDY: Main clock oscillation stable bit

Bit	Description
0	In the stabilization wait or the oscillation stop state [Initial value]
1	In the stable state

[bit0] HCRDY: High-speed CR clock oscillation stable bit

Bit	Description
0	In the stabilization wait or the oscillation stop state
1	In the stable state [Initial value]

Notes:

- *This register is not initialized by software reset.*
- *When RTCE bit (PMD_CTL:RTCE) is 1, it becomes a sub clock oscillation enable state regardless of the SOSCE bit and SORDY bit values.*
- *Writing "1" to RTCE bit (PMD_CTL:RTCE) is enabled only when SORDY bit is 1.*

5.3 Base Clock Prescaler Register (BSC_PSR)

The Base Clock Prescaler Register (BSC_PSR) sets the frequency division ratio of the base clock.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						BSR	
Attribute	-						R/W	
Initial value	-						000	

Register Functions

[bit7:3] Reserved: Reserved bits

0b00000 is read from these bits. Set these bits to 0b00000 when writing.

[bit2:0] BSR: Base clock frequency division ratio setting bits

bit2	bit1	bit0	Description
0	0	0	1/1 [Initial value]
0	0	1	1/2
0	1	0	1/3
0	1	1	1/4
1	0	0	1/6
1	0	1	1/8
1	1	0	1/16
1	1	1	Setting is prohibited

Note:

- *This register is not initialized by software reset.*

5.4 APB0 Prescaler Register (APBC0_PSR)

The APB0 Prescaler Register (APBC0_PSR) sets the APB0 bus clock frequency division.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						APBC0	
Attribute	-						R/W	
Initial value	-						00	

Register Functions

[bit7:2] Reserved: Reserved bits

0b000000 is read from these bits. Set these bits to 0b000000 when writing.

[bit1:0] APBC0: APB0 bus clock frequency division setting bits

bit1	bit0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

Note:

- *This register is not initialized by software reset.*

5.5 APB1 Prescaler Register (APBC1_PSR)

The APB1 Prescaler Register (APBC1_PSR) sets the APB1 bus clock frequency division.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	APBC1EN	Reserved		APBC1RST	Reserved		APBC1	
Attribute	R/W	-		R/W	-		R/W	
Initial value	1	-		0	-		00	

Register Functions

[bit7] APBC1EN: APB1 clock enable bit

Bit	Description
0	Disables PCLK1 output
1	Enables PCLK1 output [Initial value]

[bit6:5] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit4] APBC1RST: APB1 bus reset control bit

Bit	Description
0	APB1 bus reset, inactive [Initial value]
1	APB1 bus reset, active

[bit3:2] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit1:0] APBC1: APB1 bus clock frequency division setting bits

bit1	bit0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

Note:

- *This register is not initialized by software reset.*

5.6 Software Watchdog Clock Prescaler Register (SWC_PSR)

The Software Watchdog Clock Prescaler Register (SWC_PSR) sets the frequency division and enables the output of the software watchdog clock.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	TESTB	Reserved					SWDS	
Attribute	R/W	-					R/W	
Initial value	X	-					00	

Register Functions

[bit7] TESTB: TEST bit

Bit	Description
0	Setting is prohibited
1	Always written by "1"

Note: The read value of this bit is undefined.

[bit6:2] Reserved: Reserved bits

0b00000 is read from these bits. Set these bits to 0b00000 when writing.

[bit1:0] SWDS: Software watchdog clock frequency division ratio setting bits

bit1	bit0	Description
0	0	Sets 1/1 frequency of PCLK0. [Initial value]
0	1	Sets 1/2 frequency of PCLK0.
1	0	Sets 1/4 frequency of PCLK0.
1	1	Sets 1/8 frequency of PCLK0.

Notes:

- *This register is not initialized by software reset.*
- *Be sure to set the TESTB bit to "1" when writing a value to this register.*

5.7 Clock Stabilization Wait Time Register (CSW_TMR)

The Clock Stabilization Wait Time Register (CSW_TMR) sets the stabilization wait time of the main/sub clock.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	SOWT				MOWT			
Attribute	R/W				R/W			
Initial value	0000				0000			

Register Functions

[bit7:4] SOWT: Sub clock stabilization wait time setup bits

The stabilization wait time is counted by the CLKLC or CLKSO.

bit7	bit6	bit5	bit4	Description	Calculation Example1 CLKLC=100kHz	Calculation Example2 CLKSO=32.768kHz
0	0	0	0	2^{10} cycles [Initial value]	Approx. 10.3 ms	Approx. 31.3 ms
0	0	0	1	2^{11} cycles	Approx. 20.5 ms	Approx. 62.5 ms
0	0	1	0	2^{12} cycles	Approx. 41 ms	Approx. 125 ms
0	0	1	1	2^{13} cycles	Approx. 82 ms	Approx. 250 ms
0	1	0	0	2^{14} cycles	Approx. 164 ms	Approx. 500 ms
0	1	0	1	2^{15} cycles	Approx. 327 ms	Approx. 1.00 s
0	1	1	0	2^{16} cycles	Approx. 655 ms	Approx. 2.00 s
0	1	1	1	2^{17} cycles	Approx. 1.31 s	Approx. 4.00 s
1	0	0	0	2^{18} cycles	Approx. 2.62 s	Approx. 8.00 s
1	0	0	1	2^{19} cycles	Approx. 5.24 s	Approx. 16.0 s
1	0	1	0	2^{20} cycles	Approx. 10.49 s	Approx. 32.0 s
1	0	1	1	2^{21} cycles	Approx. 20.97 s	Approx. 64.0 s
1	1	0	0	2^{1} cycles	Approx. 0.02 ms	Approx. 0.06 ms
1	1	0	1	2^{2} cycles	Approx. 0.04 ms	Approx. 0.12 ms
1	1	1	0	2^{3} cycles	Approx. 0.08 ms	Approx. 0.24 ms
1	1	1	1	2^{4} cycles	Approx. 0.16 ms	Approx. 0.49 ms

[bit3:0] MOWT: Main clock stabilization wait time setup bits

The stabilization wait time is counted by the CLKHC or CLKMO.

bit3	bit2	bit1	bit0	Description	Calculation Example		
					CLKHC=4MHz or CLKMO=4MHz	**CLKHC=8MHz or CLKMO=8MHz**	CLKMO=40MHz
0	0	0	0	2^1 cycles [Initial value]	Approx. 500 ns	Approx. 250 ns	Approx. 50 ns
0	0	0	1	2^5 cycles	Approx. 8 µs	Approx. 4 µs	Approx. 0.8 µs
0	0	1	0	2^6 cycles	Approx. 16 µs	Approx. 8 µs	Approx. 1.6 µs
0	0	1	1	2^7 cycles	Approx. 32 µs	Approx. 16 µs	Approx. 3.2 µs
0	1	0	0	2^8 cycles	Approx. 64 µs	Approx. 32 µs	Approx. 6.4 µs
0	1	0	1	2^9 cycles	Approx. 128 µs	Approx. 64 µs	Approx. 12.8 µs
0	1	1	0	2^{10} cycles	Approx. 256 µs	Approx. 128 µs	Approx. 25.6 µs
0	1	1	1	2^{11} cycles	Approx. 512 µs	Approx. 256 µs	Approx. 51.2 µs
1	0	0	0	2^{12} cycles	Approx. 1.0 ms	Approx. 512 µs	Approx. 0.1 ms
1	0	0	1	2^{13} cycles	Approx. 2.0 ms	Approx. 1.0 ms	Approx. 0.2 ms
1	0	1	0	2^{14} cycles	Approx. 4.1 ms	Approx. 2.0 ms	Approx. 0.4 ms
1	0	1	1	2^{15} cycles	Approx. 8.2 ms	Approx. 4.1 ms	Approx. 0.8 ms
1	1	0	0	2^{17} cycles	Approx. 33.0 ms	Approx. 16.4 ms	Approx. 3.3 ms
1	1	0	1	2^{19} cycles	Approx. 131 ms	Approx. 65.5 ms	Approx. 13.1 ms
1	1	1	0	2^{21} cycles	Approx. 524 ms	Approx. 262 ms	Approx. 52.4 ms
1	1	1	1	2^{23} cycles	Approx. 2.0 s	Approx. 1.0 s	Approx. 0.2 s

Notes:

- *Set each oscillation stabilization wait time before enabling each oscillation enable bit (SOSCE, MOSCE) of the SCM_CTL register.*
  If you change MOWT or SOWT bit while waiting for oscillation stability of each oscillator, each oscillation stabilization wait time is not guaranteed.
- *This register is not initialized by software reset.*

5.8 PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)

The PLL Clock Stabilization Wait Time Setup Register (PSW_TMR) sets the main PLL clock stabilization wait time.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved			PINC	Reserved	POWT		
Attribute	-			R/W	-	R/W		
Initial value	-			0	-	000		

Register Functions

[bit7:5] Reserved: Reserved bits

0b000 is read from these bits. Set these bits to 0b000 when writing.

[bit4] PINC: PLL input clock select bit

Bit	Description
0	Selects CLKMO (main clock oscillation) [Initial value]
1	Selects CLKHC (high-speed CR clock)

Note: Setting this bit to "1" has some restrictions.

See 1. Notes when high-speed CR is used for the master clock in E. List of Notes.

[bit3] Reserved: Reserved bit

0 is read from this bit. Set this bit to 0 when writing.

[bit2:0] POWT: Main PLL clock stabilization wait time setup bits

bit2	bit1	bit0	Description	Calculation Example1 CLKPLL=20MHz	Calculation Example2 CLKPLL=40MHz
0	0	0	2^9 cycles [Initial value]	Approx. 25.6 µs	Approx. 12.8 µs
0	0	1	2^{10} cycles	Approx. 51.2 µs	Approx. 25.6 µs
0	1	0	2^{11} cycles	Approx. 102.4 µs	Approx. 51.2 µs
0	1	1	2^{12} cycles	Approx. 204.8 µs	Approx. 102.4 µs
1	0	0	2^{13} cycles	Approx. 409.6 µs	Approx. 204.8 µs
1	0	1	2^{14} cycles	Approx. 819.2 µs	Approx. 409.6 µs
1	1	0	2^{15} cycles	Approx. 1638.4 µs	Approx. 819.2 µs
1	1	1	2^{16} cycles	Approx. 3276.8 µs	Approx. 1638.4 µs

Notes:

- *Set each oscillation stabilization wait time before enabling the PLL oscillation enable bit (PLLE) of the SCM_CTL.*
  If you change POWT bit while waiting for oscillation stability of the PLL oscillator, the oscillation stabilization wait time is not guaranteed.
- *This register is not initialized by software reset.*
- *In Main PLL clock mode, When PINC bit is 1, it becomes a High-speed CR clock oscillation enable state regardless of the HCRE bit(SCM_CTL:HCRE) value. When PINC bit is 0, The status of the high-speed CR clock (CLKHC) depends on the setting of HCRE bit, FCSDE bit(CSV_CTL:FCSDE) and MCSVE bit(CSV_CTL:MCSVE).*

- *In Main PLL clock mode, When PINC bit is 1, The status of the Main clock oscillation depends on the setting of MOSCE(SCM_CTL).When PINC bit is 0, it becomes a Main clock oscillation enable state regardless of the MOSCE bit value.*

5.9 PLL Control Register 1 (PLL_CTL1)

The PLL Control Register 1 (PLL_CTL1) sets the PLL frequency division ratio.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	PLLK				PLLM			
Attribute	R/W				R/W			
Initial value	0000				0000			

Register Functions

[bit7:4] PLLK: PLL input clock frequency division ratio setting bits

bit 7:4	Description
0000	The frequency division is (PLLK value +1). (Frequency division: 1 to 16) Example: PLLK value (0000) +1 => 1/1 frequency [Initial value]
0001	
•	
•	
1111	

[bit3:0] PLLM: PLL VCO clock frequency division ratio setting bits

bit3:0	Description
0000	The frequency division is (PLLM value +1). (Frequency division: 1 to 16) Example: PLLM value (0000) +1 => 1/1 frequency [Initial value]
0001	
•	
•	
1111	

Notes:

- *Set each frequency division ratio before enabling the PLL oscillation enable bit (PLLE) of the SCM_CTL register.*
- *This register is not initialized by software reset.*

5.10 PLL Control Register 2 (PLL_CTL2)

The PLL Control Register 2 (PLL_CTL2) sets the PLL frequency division ratio.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		PLLN					
Attribute	-		R/W					
Initial value	-		000000					

Register Functions

[bit7:6] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit5:0] PLLN: PLL feedback frequency division ratio setting bits

bit5:0	Description
000000	The frequency division is (PLLN value +1). (Frequency division: 1 to 50) Example: PLLN value (000000) +1 => 1/1 division [Initial value]
000001	
.	
.	
110001	
110010	Setting is prohibited
.	
111111	

Notes:

- *Set the frequency division ratio before enabling the PLL oscillation enable bit (PLLE) of the SCM_CTL register.*
- *This register is not initialized by software reset.*

5.11 Debug Break Watchdog Timer Control Register (DBWDT_CTL)

The Debug Break Watchdog Timer Control Register (DBWDT_CTL) sets the watchdog timer count operation for debug mode tool break.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	DPHWBE	Reserved	DPSWBE	Reserved				
Attribute	R/W	-	R/W	-				
Initial value	0	-	0	-				

Register Functions

[bit7] DPHWBE: HW-WDG debug mode break bit

Bit	Description
0	HW-WDG stops counting at the tool break [Initial value]
1	HW-WDG continues counting at the tool break

[bit6] Reserved: Reserved bit

0 is read from this bit. Set this bit to 0 when writing.

[bit5] DPSWBE: SW-WDG debug mode break bit

Bit	Description
0	SW-WDG stops counting at the tool break [Initial value]
1	SW-WDG continues counting at the tool break

[bit4:0] Reserved: Reserved bits

0b00000 is read from these bits. Set these bits to 0b00000 when writing.

Note:

- *This register is not initialized by software reset.*

5.12 Interrupt Enable Register (INT_ENR)

The Interrupt Enable Register (INT_ENR) enables/disables interrupts.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		FCSE	Reserved		PCSE	SCSE	MCSE
Attribute	-		R/W	-		R/W	R/W	R/W
Initial value	-		0	-		0	0	0

Register Functions

[bit7:6] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit5] FCSE: Anomalous frequency detection interrupt enable bit

Bit	Description
0	Disables FCS interrupts
1	Enables FCS interrupts

[bit4:3] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit2] PCSE: PLL oscillation stabilization wait completion interrupt enable bit

Bit	Description
0	Disables PLL oscillation stabilization wait completion interrupts
1	Enables PLL oscillation stabilization wait completion interrupts

[bit1] SCSE: Sub clock oscillation stabilization wait completion interrupt enable bit

Bit	Description
0	Disables sub clock oscillation stabilization wait completion interrupts
1	Enables sub clock oscillation stabilization wait completion interrupts

[bit0] MCSE: Main clock oscillation stabilization wait completion interrupt enable bit

Bit	Description
0	Disables main clock oscillation stabilization wait completion interrupts
1	Enables main clock oscillation stabilization wait completion interrupts

Note:

- *For Anomalous frequency detection, see Chapter Clock supervisor.*

5.13 Interrupt Status Register (INT_STR)

The Interrupt Status Register (INT_STR) indicates the status of interrupts.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		FCSI	Reserved		PCSI	SCSI	MCSI
Attribute	-		R	-		R	R	R
Initial value	-		0	-		0	0	0

Register Functions

[bit7:6] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit5] FCSI: Anomalous frequency detection interrupt status bit

Bit	Description
0	No FCS interrupt has been asserted.
1	An FCS interrupt has been asserted.

[bit4:3] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit2] PCSI: PLL oscillation stabilization wait completion interrupt status bit

Bit	Description
0	No PLL oscillation stabilization wait completion interrupt has been asserted.
1	A PLL oscillation stabilization wait completion interrupt has been asserted.

[bit1] SCSI: Sub clock oscillation stabilization wait completion interrupt status bit

Bit	Description
0	No sub clock oscillation stabilization wait completion interrupt has been asserted.
1	A sub clock oscillation stabilization wait completion interrupt has been asserted.

[bit0] MCSI: Main clock oscillation stabilization wait completion interrupt status bit

Bit	Description
0	No main clock oscillation stabilization wait completion interrupt has been asserted.
1	A main clock oscillation stabilization wait completion interrupt has been asserted.

5.14 Interrupt Clear Register (INT_CLR)

The Interrupt Clear Register (INT_CLR) clears interrupt factors.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		FCSC	Reserved		PCSC	SCSC	MCSC
Attribute	-		W	-		W	W	W
Initial value	-		0	-		0	0	0

Register Functions

[bit7:6] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit5] FCSC: Anomalous frequency detection interrupt factor clear bit

Bit	Description
When 0 is written	The FCS interrupt factor is not affected by the written value.
When 1 is written	Clears the FCS interrupt factor.
When read	The fixed value "0" is read.

[bit4:3] Reserved: Reserved bits

0b00 is read from these bits. Set these bits to 0b00 when writing.

[bit2] PCSC: PLL oscillation stabilization wait completion interrupt factor clear bit

Bit	Description
When 0 is written	The PLL oscillation stabilization wait completion interrupt factor is not affected by the written value.
When 1 is written	Clears the PLL oscillation stabilization wait completion interrupt factor.
When read	The fixed value "0" is read.

[bit1] SCSC: Sub clock oscillation stabilization wait completion interrupt factor clear bit

Bit	Description
When 0 is written	The sub clock oscillation stabilization wait completion interrupt factor is not affected by the written value.
When 1 is written	Clears the sub clock oscillation stabilization wait completion interrupt factor.
When read	The fixed value "0" is read.

[bit0] MCSC: Main clock oscillation stabilization wait completion interrupt factor clear bit

Bit	Description
When 0 is written	The main clock oscillation stabilization wait completion interrupt factor is not affected by the written value.
When 1 is written	Clears the main clock oscillation stabilization wait completion interrupt factor.
When read	The fixed value "0" is read.

Note:

- *When this register is cleared, each interrupt status bit (FCSI, PCSI, SCSI, MCSI) in the INT_STR register is also cleared.*

6. Usage Precautions

This section explains the precautions for using the clock generation unit.

■ The oscillation stabilization wait time of main clock and sub clock oscillators

Because the stabilization wait time of main clock/sub clock oscillator depends on the oscillator type (crystal, ceramic, etc.), the oscillation stabilization wait time suitable for the oscillator type must be selected.

■ Changing the frequency division under stabilized PLL oscillation

When the PLL frequency division ratio is changed after stabilization of PLL oscillation, stop the PLL oscillation once, change the frequency division ratio, and then re-enable the PLL oscillation.

■ Peripherals independent of clock control by the clock generation unit

The following peripherals run independently of clock control by the clock generation unit.
For information about how to handle each operating clock, see the following chapter.

- USB clock gen. unit : See Chapter “USB Clock Generation” in Communication Macro Part”.
- Clock supervisor : See Chapter Clock supervisor.
- Watchdog timer : See Chapter Watchdog Timer in Timer part.
- Watch counter : See Chapter Watch Counter in Timer part.
- Real-time clock : See Chapter REAL-TIME CLOCK in Timer part.

■ Setting the oscillation stabilization wait time

Set the oscillation stabilization wait time of the main clock, sub clock, and PLL oscillators with relevant oscillation stabilization wait time setup registers, and then enable each oscillator.
Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

■ Checking main clock oscillation while using the main PLL clock

It is prohibited to stop main clock oscillation while using PLL oscillation.

■ Switching clock modes

Clock modes can be switched by changing the RCS[2:0] bits of the SCM_CTL register.
To switch clock modes, take the following steps:

1. Set the oscillation stabilization wait time of each oscillator.
2. Set the oscillation enable bit of the desired clock (SCM_CTL:xxxE) to "1".
3. Check the oscillation stable bit of the desired clock (SCM_CTL:xxxRDY) to "1".
4. Switch SCM_CTL:RCS[2:0].
5. Wait until SCM_STR:RCM[2:0] = SCM_CTL:RCS[2:0].

■ Correlation between the clock mode switching and the oscillation stable bit

The timings when the oscillation stable bit (SCM_STR:xxxRDY) turns to 1 vary for the following clock mode switching.

- When switching from the high-speed CR run, main run, or PLL run to another clock mode:

  Setting SCM_CTL:xxxE to 1 can start the oscillation stabilization wait time. You can check that SCM_STR:xxxRDY is 1 after the oscillation stabilization wait time has elapsed.

- When switching from the low-speed CR run or sub run to the high-speed CR run, main run, or PLL run:

  Even if SCM_CTL:MOSCE (or PLLE) set to 1, oscillation of main clock does not start. To start the main clock (or high-speed CR or PLL) oscillation stabilization wait time, SCM_CTL:RCS [2:0] must be switched after setting SCM_CTL:MOSCE (or PLLE) to 1. After the oscillation stabilization wait time has elapsed, you can check that SCM_STR:xxxRDY is 1.

- If the standby mode is released by an interrupt, the device restarts in the clock mode that indicated by the RCS[2:0] bits in the SCM_CTL register.
- If any reset occurs other than software resets, the high-speed CR clock (CLKHC) is set as a master clock. High-speed CR clock mode is set as clock mode.
- If any reset other than software resets is executed, the main clock and sub clock oscillators, and PLL oscillation stop. If you want to use those oscillators again after the reset, enable them using the SCM_CTL register.
- For the correlation between each clock mode and start/stop of the oscillator, see Chapter Low Power Consumption Mode.
- To turn off the power supply on the chip side and operate only VBAT domain, be sure to set WTOSCCNT.SOSCNTL=0 and then turn off the power supply on the chip side.

CHAPTER 2-2: Peripheral Clock Gating

This chapter explains the functions of Peripheral Clock Gating.

1. Peripheral Clock Gating Overview
2. Peripheral Clock Gating Configuration
3. Peripheral Clock Gating Control
4. Peripheral Clock Gating Function Registers
5. Peripheral Clock Gating Function Usage Precautions

CODE: 9BFPCG-FM0-E03.0

1. Peripheral Clock Gating Overview

This section shows an overview of the Peripheral Clock Gating which stops the operation clocks of peripheral functions individually. By using these functions, the system can reduce the current consumption of the total system with gating the operation clocks of peripheral functions not used.

Overview of Peripheral Clock Gating

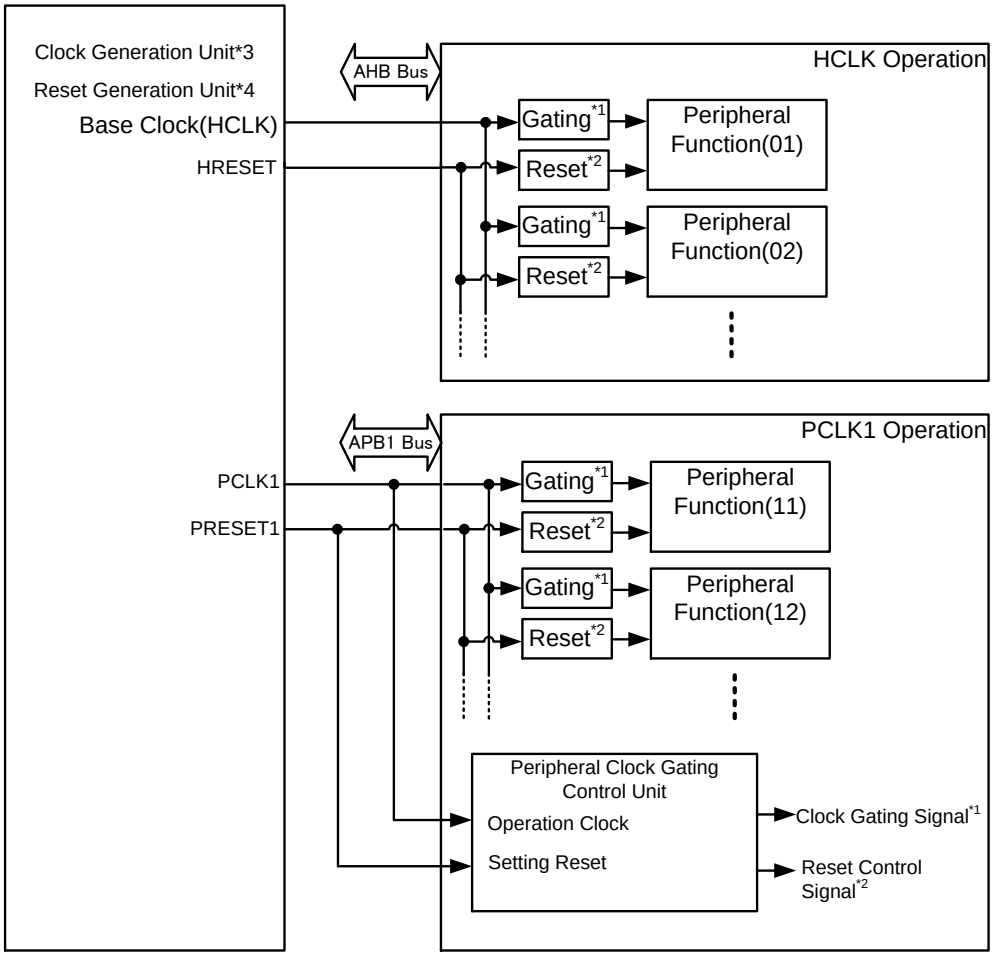
- The operation clocks of peripheral functions not used in the system operation are gated individually.
- For target clocks and units of the Peripheral Clock Gating, see "Gating units and their initial states of Peripheral Clock"
- When a clock is gated or before a clock is supplied, the internal states of peripheral functions can be reset.

The above peripheral clock gating and reset control are implemented by the setting of a register connecting to APB1 bus.

Overview of Connection with Clock and Reset Generation Units

Figure 1-1 shows the connection between peripheral clock gating and clock generation unit or reset generation unit. The peripheral clock gating exists between peripheral function and clock generation unit or reset generation unit and gates clocks and controls resets in the unit of peripheral function. When the internal bus clock supply from the reset control units are stopped, the priority is given to the settings of the clock control unit and the operation clock supplies to peripheral functions are gated To use the peripheral clock gating, be sure to make the settings which enable the output of APB1 bus clock (PCLK1) in the clock generation unit to control the rest.

Figure 1-1 Clock/Reset Connection Related to Peripheral Clock Gating



*1: Clock gating (internal bus clock gating) in the unit of peripheral function
*2: Forcible reset or bus reset (see Chapter "Reset") by peripheral function unit
*3: For details, see Chapter "Clock".
*4: For details, see Chapter "Reset".

Gating Units and Their Initial States of Peripheral Clock

For gating units and their initial states of Peripheral Clock Gating, see Table 1-1.

Table 1-1Control Units and Their Initial States of Peripheral Clock Gating

Peripheral Functions	Unit of structure to be stopped clock	Initial States	Remarks
Multi-function Serial Interface	One channel	Clock supply	
Base timer	Four channels	Clock supply	The clock gating can be controlled with every four channels Ch.0 to Ch.3, Ch.4 to Ch.7, Ch.8 to Ch.11, and Ch.12 to Ch.15.
Multi-function timer	One unit	Clock supply	-
PPG	Eight channels	Clock supply	The clock gating can be controlled with every eight channels Ch.0 to Ch.7, Ch.8 to Ch.15, Ch.16 to Ch.23 and Ch.24 to Ch.31.
Quad counter	One unit	Clock supply	-
DMAC	One unit	Clock supply	-
A/D converter	One unit	Clock supply	-
IO Port (GPIO/Fast GPIO)	All ports	Clock supply	For constrains at clock gating, "Usage Precautions" in 5. Peripheral Clock Gating.
DSTC	One unit	Clock supply	-
Smart card Interface (IC-Card)	One channel	Clock supply	-
MFSI2S	One channel	Clock supply	-
LCD controller	One unit	Clock supply	-
HDMI-CEC/ Remote Control Reception	One unit	Clock supply	-
Programmable-CRC	One unit	Clock supply	-
USB (Device/Host)	One channel	Clock stop	-

Notes:

- *For types and the number of mounted peripheral functions, see 'Data sheet' of the product used*
- *The clock control of PPG shares the setting bits with the multi-function timer. For details, see 4.3Peripheral Clock Control Register 1 (CKEN1).*
- *Execute the clock control of the DSTC unit alone with 5. DSTC Register in DSTC.*

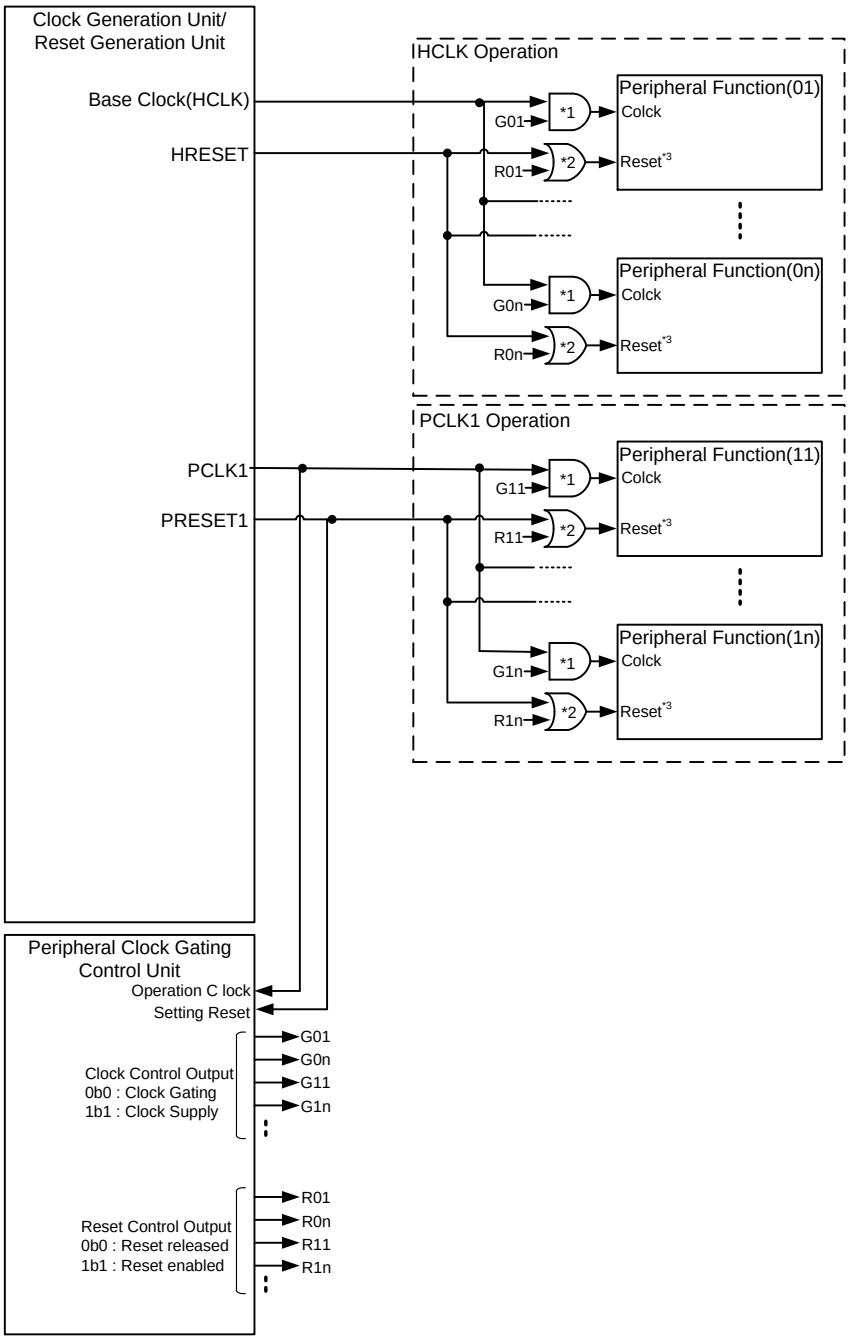
2. Peripheral Clock Gating Configuration

This section explains the configuration of the Peripheral Clock Gating.

Block Diagram

Figure 2-1 shows the system configuration of Peripheral Clock Gating.

Figure 2-1 Block Diagram of Peripheral Clock Gating



*1: Gates the operation clocks of peripheral functions

*2: Resets peripheral functions

*3: At High-level signal input: Reset enabled, At Low-level signal input: Reset released

Explanation on Block Diagram

- Peripheral Clock Gating Control Unit

The clock control or the reset control of each peripheral function is executed by changing the register setting value via the APB1 bus. Be sure to rewrite this register with setting APB1 clock enable bit (APBC1EN) in APB1 prescaler register (APBC1_PSR) of the clock control unit to the output enable and permitting PCLK1 output.

The clock of each peripheral function stops when the bit field of the target function is set to 0. When the bit field is set to 1, the clock is supplied. The initial value of a register is different by peripheral function. For details, see Table 1-1.

The reset of each peripheral function is issued when the bit field of the target function is set to1. When the bit field is set to "0", the reset is released. The initial value of each register is always 0 to release the reset.

- Peripheral Clock Gating Logic

Internal bus clock (HCLK, PCLK1) is supplied or gated by each specific peripheral function according to clock gating signal from the peripheral clock gating control unit.

- Peripheral Reset Control Logic

The reset is individually controlled by each peripheral function according to the reset control signal from the peripheral clock gating control unit. The reset control unit is the same with the peripheral clock control unit. However, it does not exceptionally have the bit field of this reset control for I/O Port alone

3. Peripheral Clock Gating Control

This section explains the control of the peripheral clock gating.

The register of the peripheral clock gating becomes an initial state by bus reset (PRESET1)*. Be sure to execute the clock control for necessary peripheral functions immediately after reset of the bus because the bus reset (PRESET1) is generated by all reset factors.

*: For the generating condition of bus reset (PRESET1), see Chapter Reset.

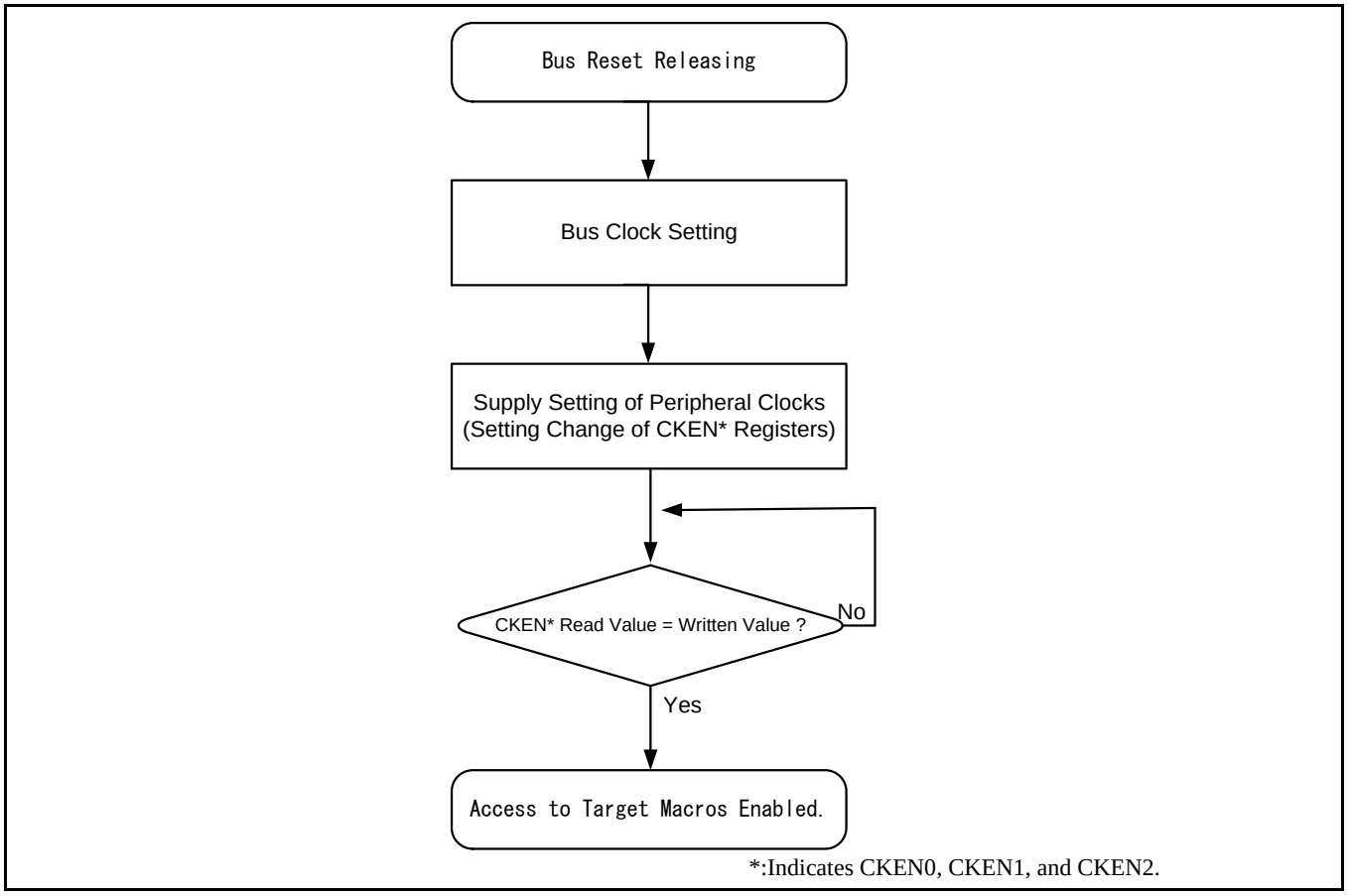
3.1 Peripheral Clock Control Procedures

This section explains the control procedures of supplying and stopping peripheral clocks.

Clock Supply Procedures

The settings of the bus clocks and the peripheral clocks are reset to the initial values immediately after the bus reset release. So, for the clocks of peripheral functions which have been stopped in the initial state, set the clock supplies conforming to the procedures in Figure 3-1.

Figure 3-1 Clock Supply Procedures



1. Bus clock setting

   Execute the setting of each bus clock by using the register of the clock generation part.

   For the setting details, see Chapter Clock.

2. Supply setting of peripheral clocks

   Change the setting of the bit corresponding to the peripheral function to which the clock is to be supplied for peripheral clock control registers (CKEN0, CKEN1, and CKEN2) of the clock control in the clock gating state of the initial state.

3. Set value confirmation of peripheral clock control register

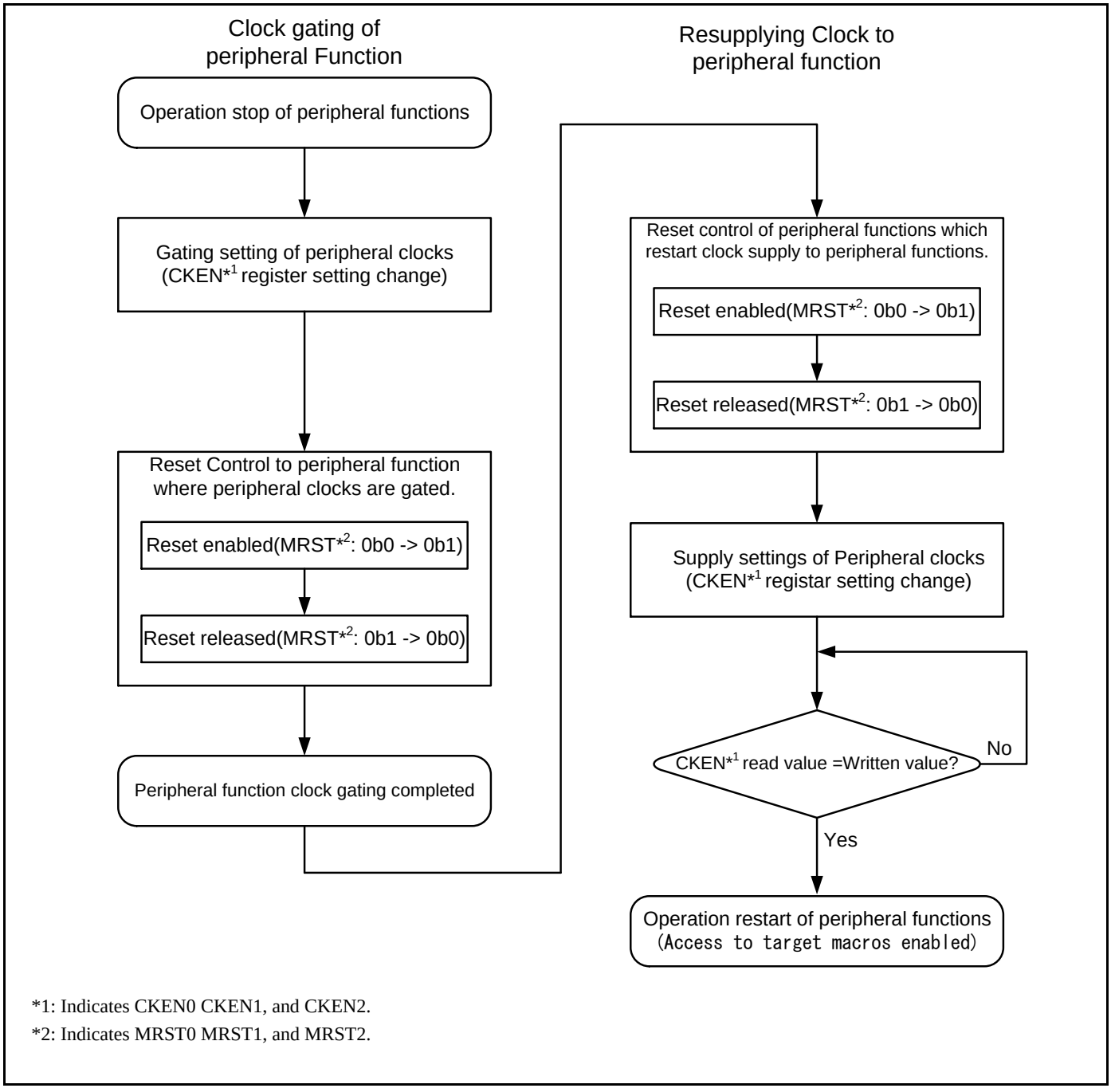
The peripheral clock registers (CKEN0, CKEN1, and CKEN2) updates the register value to the written value at the step of starting the clock supply to the peripheral function to which the setting is changed.

Be sure to start the access to the peripheral function after setting a change in the above-mentioned Item 2, reading this register, and then confirming the agreement with the written value because an access to peripheral function is invalid at clock gating.

Procedures of Gating and Resupplying Clocks

Figure 3-2 explains the procedures of gating the clocks of peripheral functions and resupplying clocks to peripheral functions.

Figure 3-2 Procedures of Gating Clocks of Peripheral Functions and Resupplying Clocks to Peripheral Functions



*1: Indicates CKEN0 CKEN1, and CKEN2.
*2: Indicates MRST0 MRST1, and MRST2.

■ Clock gating of peripheral functions

1. Gating setting of peripheral clocks

For the peripheral clock control registers (CKEN0 CKEN1, and CKEN2), change the bit corresponding to the peripheral function for which the clock supply is to be stopped to 0.

After gating the clock to the peripheral function to which the clock gating is instructed, the peripheral clock control registers (CKEN0 CKEN1, and CKEN2) updates the register value to the written value.

2. Reset control to peripheral functions whose peripheral clocks are gated

For the peripheral functions whose clocks are gated, to reset their internal state, execute the reset control of each peripheral function according to the following procedures.

Reset enabled:

Write 1 to the target bits of peripheral function reset control registers (MRST0 MRST1, and MRST2).

Reset released:

Write 0 to the target bits of peripheral function reset control registers (MRST0MRST1, and MRST2).

■ Resupplying clocks to peripheral functions

1. Reset control of peripheral functions which restart clock supply to peripheral functions

For the peripheral functions which gate the peripheral clocks, execute the reset control to each peripheral function by using peripheral function reset control registers (MRST0, MRST1, and MRST2) before restarting their operation. The procedures are the same as the above-mentioned procedures of reset control immediately after peripheral clocks gated.

2. Supply settings of peripheral clocks

For the peripheral clock control registers (CKEN0 CKEN1, and CKEN2), change the settings of bit corresponding to the peripheral function for which the clock is to be resupplied.

At this time, do not set the bit where the peripheral function is not provided and the bit whose bus clock has been gated to the values other than the initial value The reason is that the read value cannot coincide with the written value not to get out of the processing loop at the register set value confirmation in the following Item 3.

3. Confirmation of set values of peripheral clock control registers

At the step where the clock setting change is reflected to the peripheral function whose settings are changed, the peripheral clock control registers (CKEN0 CKEN1, and CKEN2) updates the register value to the written values.

Be sure to start the access to the peripheral function after executing the setting change of the above-mentioned Item 2 , reading the register, and then confirming the agreement with the written value because the access to the peripheral functions is invalid at clock gating.

4. Peripheral Clock Gating Function Registers

This section explains each register function of the peripheral clock gating functions.

Table 4-1 shows the list of registers of peripheral clock gating functions.

Table 4-1 Registers of the Peripheral Clock Gating Functions

Abbreviated Register Name	Register Name	Reference
CKEN0	Peripheral Function Clock Control Register 0	4.1
MRST0	Peripheral Function Reset Control Register 0	4.2
CKEN1	Peripheral Function Clock Control Register 1	0
MRST1	Peripheral Function Reset Control Register 1	4.4
CKEN2	Peripheral Function Clock Control Register 2	4.5
MRST2	Peripheral Function Reset Control Register 2	4.6

4.1 Peripheral Function Clock Control Register 0 (CKEN0)

This section explains Peripheral Function Reset Clock Register 0 (CKEN0).

bit	31	30	29	28	27	26	25	24
Field	Reserved			GIOCK	Reserved		DSTCCK	DMACK
Attribute		-		R/W		-	R/W	R/W
Initial value		-		1		-	1	1

bit	23	22	21	20	19	18	17	16
Field	Reserved				ADCCK[3:0]			
Attribute		-				R/W		
Initial value		-				1111		

bit	15	14	13	12	11	10	9	8
Field	MFSCK[15:8]							
Attribute				R/W				
Initial value				0xFF				

bit	7	6	5	4	3	2	1	0
Field	MFSCK[7:0]							
Attribute				R/W				
Initial value				0xFF				

[bit31:29] Reserved: Reserved bits

Write 0.to these bits.

[bit28] GIOCK: Software clock control of GPIO/Fast GPIO function

This bit controls the operation clock supplying and the gating to the I/O Port function. This bit controls all the operation clocks to the I/O Port functions collectively.

When this bit is set to 1, the bus clock is supplied to the I/O Port function block and the I/O Port function can be used.

When this bit is set to 0, the bus clock input to the I/O Port function block is gated. Note that the reading of the input level and the setting change of the output power level are disabled while the bus clock is gated. For details, see 5. Peripheral Clock Gating Functions Usage Precautions.

Bit	Description
0	The bus clock input to the I/O Port function block is gated.
1	The bus clock is supplied to the I/O Port function block. (Initial value) Be sure to set "1" in order to use I/O Port function.

[bit27:26] Reserved: Reserved bit

Write0 to this bit.

[bit25] DSTCCK: Software clock control of DSTC function

This bit controls the operation clock supplying and the gating to the DSTC function. When this bit is set to1, the bus clock is supplied to the DSTC block and the DSTC function can be used.
When this bit is set to 0, the bus clock input to the DSTC block is gated. While the bus clock input is gated, the DSTC function cannot be used..

Bit	Description
0	The bus clock input to the DSTC function block is gated.
1	The bus clock is supplied to the DSTC function block. (Initial value) Be sure to set "1" in order to use DSTC function.

NOTE: In TYPE3-M0+ product, this bit value is ignored. The standby control register in the DSTC can be used for the clock stop of the DSTC.

[bit24] DMACK: Supplying and gating settings of DMAC operation clock

This bit controls the operation clock supplying and the gating to the DMAC function. When this bit is set to1, the bus clock is supplied to the DMAC block and the DMAC function can be used.
When this bit is set to 0, the bus clock input to the DMAC block is gated. While the bus clock input is gated, the DMAC function cannot be used.

Bit	Description
0	The bus clock input to DMAC is gated.
1	The bus clock is supplied to DMAC. (Initial value)

[bit23:20] Reserved: Reserved bits

Write0 to these bits.

[bit19:16] ADCCK[3:0]: Settings for operation clock supplying and gating to AD converter

These bits control the operation clock supplying and gating to the AD converter. The following show the correspondence between each bit and the AD converter unit:

bit16 - ADCCK0: AD converter unit 0
bit17 - ADCCK1: AD converter unit 1
bit18 - ADCCK2: AD converter unit 2
bit19 - ADCCK3: AD converter unit 3

When the relevant bit is set to 1, the bus clock is supplied to the unit of the corresponding AD converter to enable the AD converter function. For products to which the corresponding AD converter is not mounted, do not change the relevant bit from the initial value.
When the relevant bit is set to 0, the bus clock input to the corresponding AD converter is gated. While the bus clock input is gated, the relevant AD converter cannot be used.

Bit	Description
0	The bus clock input to the A/D converter unit corresponding to the relevant bit is gated.
1	The bus clock is supplied to the A/D converter unit corresponding to the relevant bit. (Initial value)

[bit15:0] MFSCK[15:0]: Settings for operation clock supply and gating to multi-function serial interface

These bits control the operation clock supply and gating to the multi-function serial interface. The correspondence between each bit and the channel is shown below:

bit0 - MFSCK0: Multi-function serial interface channel 0
bit1 - MFSCK1: Multi-function serial interface channel 1
bit2 - MFSCK2: Multi-function serial interface channel 2
bit3 - MFSCK3: Multi-function serial interface channel 3
bit4 - MFSCK4: Multi-function serial interface channel 4
bit5 - MFSCK5: Multi-function serial interface channel 5
bit6 - MFSCK6: Multi-function serial interface channel 6
bit7 - MFSCK7: Multi-function serial interface channel 7
bit8 - MFSCK8: Multi-function serial interface channel 8
bit9 - MFSCK9: Multi-function serial interface channel 9
bit10 - MFSCK10: Multi-function serial interface channel 10
bit11 - MFSCK11: Multi-function serial interface channel 11
bit12 - MFSCK12: Multi-function serial interface channel 12
bit13 - MFSCK13: Multi-function serial interface channel 13
bit14 - MFSCK14: Multi-function serial interface channel 14
bit15 - MFSCK15: Multi-function serial interface channel 15

When the relevant bit is set to 1, the bus clock is supplied to the channel of the corresponding multi-function serial interface to enable the function of the multi-function serial interface. For products to which the relevant multi-function serial interface channel is not mounted, do not change the relevant bit from the initial value.
When the relevant bi is set to 0, the bus clock input to the channel of the corresponding multi-function serial interface is gated. While the bus clock input is gated, the multi-function serial interface function of the corresponding channel cannot be used.

Bit	Description
0	The bus clock input to the multi-function serial interface channel corresponding to the relevant bit is gated.
1	The bus clock is supplied to the multi-function serial interface channel corresponding to the relevant bit. (Initial value)

4.2 Peripheral Reset Control Register 0 (MRST0)

This section explains the peripheral reset control register 0 (MRST0).

bit	31	30	29	28	27	26	25	24
Field	Reserved						DSTCRST	DMARST
Attribute	-						R/W	R/W
Initial value	-						0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved				ADCRST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

bit	15	14	13	12	11	10	9	8
Field	MFSRST [15:8]							
Attribute	R/W							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	MFSRST [7:0]							
Attribute	R/W							
Initial value	0x00							

[bit31:26] Reserved: Reserved bits

Write0 to these bits.

[bit25] DSTCRST: Reset control of DSTC

This bit controls reset of the DSTC unit. If this bit is set to 1, DSTC becomes a reset state, the DMA transfer operation of DSTC stops, and all the register settings are initialized. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the DSTC reset. (Initial value)
1	Issues reset signal to DSTC.

NOTE: In TYPE3-M0+ product, this bit value is ignored.

[bit24] DMARST: Reset control of DMAC

This bit controls reset of the DMAC unit. If this bit is set to 1, DMAC becomes a reset state, the DMA transfer operation stops, and all the register settings are initialized. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the DMAC reset. (Initial value)
1	Issues reset signal to DMAC.

[bit23:20] Reserved: Reserved bits
Write0 to these bits.

[bit19:16] ADCRST[3:0]: Reset control of AD converter
These bits control the reset of each unit of the AD converter. The correspondence between each bit and AD converter unit is shown below:

bit16 - ADCRST 0: AD converter unit 0
bit17 - ADCRST 1: AD converter unit 1
bit18 - ADCRST 2: AD converter unit 2
bit19 - ADCRST 3: AD converter unit 3

If the relevant bit is set to 1, the corresponding AD converter unit becomes a reset state, the analog to digital conversion operation stops, and the register settings are initialized. For products to which the relevant AD converter unit is not mounted, do not change the relevant bit from the initial state. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of the AD converter unit corresponding to the relevant bit. (Initial value)
1	Issues the reset to the AD converter unit corresponding to the relevant bit.

[bit15:0] MFSRST[15:0]: Reset control of multi-function serial interface
These bits control the reset of each channel of the multi-function serial interface. The correspondence between each bit and the channel is shown below.

bit0 - MFSRST0: Multi-function serial interface channel 0
bit1 - MFSRST1: Multi-function serial interface channel 1
bit2 - MFSRST2: Multi-function serial interface channel 2
bit3 - MFSRST3: Multi-function serial interface channel 3
bit4 - MFSRST4: Multi-function serial interface channel 4
bit5 - MFSRST5: Multi-function serial interface channel 5
bit6 - MFSRST6: Multi-function serial interface channel 6
bit7 - MFSRST7: Multi-function serial interface channel 7
bit8 - MFSRST8: Multi-function serial interface channel 8
bit9 - MFSRST9: Multi-function serial interface channel 9
bit10 - MFSRST10: Multi-function serial interface channel 10
bit11 - MFSRST11: Multi-function serial interface channel 11
bit12 - MFSRST12: Multi-function serial interface channel 12
bit13 - MFSRST13: Multi-function serial interface channel 13
bit14 - MFSRST14: Multi-function serial interface channel 14
bit15 - MFSRST15: Multi-function serial interface channel 15

If the relevant bit is set to 1, the channel of the corresponding multi-function serial interface becomes a reset state, its serial communications stop, and the register setting is initialized. For products to which the relevant multi-function serial interface channel is not mounted, it is prohibited to set the relevant bit to 1. To release the above-mentioned reset, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of the multi-function serial interface channel corresponding to the relevant bit. (Initial value)
1	Issues the reset the multi-function serial interface channel corresponding to the relevant bit.

4.3 Peripheral Clock Control Register 1 (CKEN1)

This section explains the peripheral clock control register 1 (CKEN1).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved				QDUCK[3:0]			
Attribute	-				R/W			
Initial value	-				1111			

bit	15	14	13	12	11	10	9	8
Field	Reserved				MFTCK[3:0]			
Attribute	-				R/W			
Initial value	-				1111			

bit	7	6	5	4	3	2	1	0
Field	Reserved				BTMCK[3:0]			
Attribute	-				R/W			
Initial value	-				1111			

[bit31:20] Reserved: Reserved bits
Write0 to these bits.

[bit19:16] QDUCK[3:0]: Settings for operation clock supply and gating of quad counter
These bits control the operation clock supply and gating of quad counter. The correspondence between each bit and quad counter is shown below.

bit16 - QDUCK0: Quad counter unit 0
bit17 - QDUCK1: Quad counter unit 1
bit18 - QDUCK2: Quad counter unit 2
bit19 - QDUCK3: Quad counter unit 3

When the relevant bit is set to 1, the bus clock is supplied to the unit of the corresponding quad counter to use the quad counter function. For products to which the relevant quad counter unit is not mounted, do not change the relevant bit from the initial value.
When the relevant bit is set to 0, the bus clock input to corresponding quad counter is stopped. While the bus clock input is gated, the quad counter of the relevant unit cannot be used.

Bit	Description
0	Gates the bus clock input to the corresponding quad counter.
1	Supplies the bus clock to the quad counter corresponding to the relevant bit. (Initial value)

[bit15:12]Reserved: Reserved bits
Write0 to these bits.

[bit11:8] MFTCK[3:0]: Settings for operation clock supply and gating of multi-function timer and PPG
These bits control the operation clock supply and gating to the multi-function timer and PPG. The correspondence among each bit, the multi-function timer unit, and the PPG channel is shown below.

bit8 - MFTCK0: Multi-function timer unit 0 - PPG channels 0, 2, 4, 6
bit9 - MFTCK1: Multi-function timer unit 1 - PPG channels 8, 10, 12, 14
bit10 - MFTCK2: Multi-function timer unit 2 - PPG channels 16, 18, 20, 22
bit11 - MFTCK3: Multi-function timer unit 3 - PPG channels 24, 26, 28, 30

When the relevant bit is set to 1, the bus clock is supplied to corresponding multi-function timer unit and PPG channels to use the multi-function timer and PPG function. For products to which the relevant multi-function timer unit and PPG channels are not mounted, it is prohibited to change the relevant bit from the initial value.
When the relevant bit is set to 0, the bus clock input to the corresponding multi-function timer unit and PPG channels is gated. While the bus clock is gated, the relevant multi-function timer and PPG function cannot be used.

Bit	Description
0	The bus clock input to the multi-function timer unit and the PPG channel corresponding to the relevant bit is gated.
1	The bus clock is supplied to the multi-function timer unit and the PPG channel corresponding to the relevant bit. (Initial value)

[bit7:4] Reserved: Reserved bits
Write0 to these bits.

[bit3:0] BTMCK[3:0]: Settings operation clock supply and gating to base timer
These bits control the operation clock supply and gating to the base timer. The correspondence between each bit and the base timer channels is shown below.

bit0 - BTMCK0: Base timer channel 0, 1, 2, 3
bit1 - BTMCK1: Base timer channel 4, 5, 6, 7
bit2 - BTMCK2: Base timer channel 8, 9, 10, 11
bit3 - BTMCK3: Base timer channel 12, 13, 14, 15

When the relevant bit is set to 1, the bus clock is supplied to the corresponding base timer channel to use the base timer, do not change the relevant bit from the initial value.
When 0 is set to the relevant bit, the bus clock input to the corresponding base timer channel is gated. While the bus clock input is gated, the base timer function of the corresponding channel cannot be used.

Bit	Description
0	The bus clock input to the base timer channel corresponding to the relevant bit is gated.
1	The bus clock is supplied to the base timer channel corresponding to the relevant bit. (Initial value)

4.4 Peripheral Function Reset Control Register 1 (MRST1)

This section explains the peripheral function reset control register 1(MRST1).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved				QDURST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				MFTRST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved				BTMRST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

[bit31:20] Reserved: Reserved bits

Write0 to these bits.

[bit19:16] QDURST[3:0]: Reset control of quad counter

These bits control the reset of each unit of the quad counter. The correspondence between each bit and the quad counter unit is shown below.

bit16 - QDURST 0: Quad counter unit 0
bit17 - QDURST 1: Quad counter unit 1
bit18 - QDURST 2: Quad counter unit 2
bit19 - QDURST 3: Quad counter unit 3

If the relevant bit is set to 1, the unit of the corresponding quad counter becomes a reset state, the quad counter operation stops, and the register settings are initialized. For products to which the relevant quad counter is not mounted, do not change the relevant bit from the initial state. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of the quad counter corresponding to the relevant bit. (Initial value)
1	Issues the reset to the quad counter unit corresponding to the relevant bit.

[bit15:12] Reserved: Reserved bits

Write0 to these bits.

[bit11:8] MFTRST[3:0]: Control of multi-function timer and PPG reset control

These bits control multi-function timer reset of each unit and PPG reset of every four channels. The correspondence among each bit, quad counter unit, and the PPG channel is shown below.

bit8 - MFTRST0: Multi-function timer unit 0 - PPG channel 0, 2, 4, 6
bit9 - MFTRST1: Multi-function timer unit 1 - PPG channel 8, 10, 12, 14
bit10 - MFTRST2: Multi-function timer unit 2 - PPG channel 16, 18, 20, 22
bit11 - MFTRST3 Multi-function timer unit 3 - PPG channel 24, 26, 28, 30

If the relevant bit is set to 1, the corresponding multi-function timer unit and PPG channel become the reset states, the multi-function timer operation stops, and the register setting is initialized. For products to which the relevant multi-function timer unit and PPG channels are not mounted, do not change the relevant bit from the initial value. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Release the resets of the multi-function timer unit and the PPG channel corresponding to the relevant bit. (Initial value)
1	Issue the resets to the multi-function timer unit and the PPG channels corresponding to the relevant bit.

[bit7:4]Reserved: Reserved bits

Write0 to these bits.

[bit3:0] BTMRST[3:0]: Reset control of base timer

These bits control the reset for four units of the base timer. The correspondence among each bit and the base timer channels is shown below.

bit0 - BTMRST0: Base timer channels 0, 1, 2, 3
bit1 - BTMRST1: Base timer channels 4, 5, 6, 7
bit2 - BTMRST2: Base timer channels 8, 9, 10, 11
bit3 - BTMRST3: Base timer channels 12, 13, 14, 15

If the relevant bit is set to 1, the unit of the corresponding base timer channels becomes a reset state, the base timer operation stops, and the register setting is initialized. For products to which the relevant base timer channels are not mounted, do not change the relevant bit from the initial value. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Release the reset for the base timer channel corresponding to the relevant bit. (Initial value)
1	Issue the reset to the base timer channel corresponding to the relevant bit.

4.5 Peripheral Clock Control Register 2 (CKEN2)

This section explains the peripheral clock control register 2(CKEN2).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved			PCRCCK	Reserved	CECCK	Reserved	LCDCCK
Attribute	-			R/W	-	R/W	-	R/W
Initial value	-			1	-	1	-	1

bit	15	14	13	12	11	10	9	8
Field	I2SCCK [1:0]		ICCCK [1:0]		Reserved		Reserved	
Attribute	R/W		R/W		-		-	
Initial value	11		11		-		-	

bit	7	6	5	4	3	2	1	0
Field	Reserved						USBCK [1:0]	
Attribute	-						R/W	
Initial value	-						00	

[bit31:21] Reserved: Reserved bits

Write0 to these bits.

[bit20] PCRCCK: Settings for operation clock supply and gating to Programmable-CRC

This bit controls the operation clock supply and gating to the Programmable-CRC function. When this bit is set to 1, the bus clock is supplied to the Programmable-CRC unit to use the Programmable-CRC function. For products to which the relevant Programmable-CRC unit is not mounted, do not change the relevant bit from the initial value.

When this bit is set to 0, the bus clock input to the Programmable-CRC unit is gated. While the bus clock input is gated, the functions of the Programmable-CRC cannot be used.

Bit	Description
0	Gates the bus clock input to Programmable-CRC.
1	Supplies the bus clock to Programmable-CRC. (Initial value)

[bit19] Reserved: Reserved bit

Write0 to this bit.

[bit18] CECCK: Settings for operation clock supply and gating of HDMI-CEC/Remote Control Reception

This bit controls the operation clock supply and gating to HDMI-CEC/Remote Control Reception.

When the relevant bit is set to "1", the bus clock is supplied to the HDMI-CEC/Remote Control Reception function. For products to which the relevant HDMI-CEC/Remote Control Reception channel is not mounted, it is prohibited to change the relevant bit from the initial value.
When the relevant bit is set to "0", the bus clock input to the HDMI-CEC/Remote Control Reception function is gated. While the bus clock input is gated, the HDMI-CEC/Remote Control Reception function cannot be used

Bit	Description
0	Gates the bus clock input to the HDMI-CEC/Remote Control Reception function
1	Supplies the bus clock to the HDMI-CEC/Remote Control Reception function. (Initial value)

[bit17] Reserved: Reserved bit
Write0 to this bit.

[bit16] LCDCCK: Settings for operation clock supply and gating to LCD Controller
This bit controls the operation clock supply and gating to the LCDC function.

When this bit is set to 1, the bus clock is supplied to the LCDC unit to use the LCD function. For products to which the LCDC unit is not mounted, it is prohibited to change the relevant bit from the initial value.

When this bit is set to 0, the bus clock input to the LCDC unit is gated. While the bus clock input is gated, the functions of the LCD cannot be used.

Bit	Description
0	Gates the bus clock input to LCDC.
1	Supplies the bus clock to LCDC. (Initial value)

[bit15:14] I2SCCK[1:0]: Settings for operation clock supply and gating of MFSI2S
These bits control the operation clock supply and gating to MFSI2S. The correspondence between each bit and the MFSI2S channel is shown below.

[TYPE2-M0+ product]
bit14 – I2SCCK0: MFS I2S interface channel 5
bit15 – I2SCCK1: MFS I2S interface channel 6

[TYPE3-M0+ product]
bit14 – I2SCCK0: MFS I2S interface channel 4
bit15 – I2SCCK1: MFS I2S interface channel 6

When the relevant bit is set to 1, the bus clock is supplied to the corresponding MFSI2S channel to use the MFSI2S function. For products to which the relevant MFSI2S channel is not mounted, it is prohibited to change the relevant bit from the initial value.
When the relevant bit is set to 0, the bus clock input to the corresponding MFSI2S channel is gated. While the bus clock input is gated, the MFSI2S function of the corresponding channel cannot be used

Bit	Description
0	Gates the bus clock input to the MFSI2S channel corresponding to the relevant bit.
1	Supplies the bus clock to the MFSI2S channel corresponding to the relevant bit. (Initial value)

[bit13:12] ICCCK[1:0]: Settings for operation clock supply and gating of Smart-Card interface

These bits control the operation clock supply and gating to Smart-Card interface. The correspondence between each bit and the Smart-Card interface channel is shown below.

bit12 – ICCCK0: Smart-Card interface channel 0
bit13 – ICCCK1: Smart-Card interface channel 1

When the relevant bit is set to 1, the bus clock is supplied to the corresponding Smart-Card interface channel to use the Smart-Card interface function. For products to which the relevant Smart-Card interface channel is not mounted, it is prohibited to change the relevant bit from the initial value.
When the relevant bit is set to 0, the bus clock input to the corresponding Smart-Card interface channel is gated. While the bus clock input is gated, the Smart-Card interface function of the corresponding channel cannot be used

Bit	Description
0	Gates the bus clock input to the Smart-Card interface channel corresponding to the relevant bit.
1	Supplies the bus clock to the Smart-Card interface channel corresponding to the relevant bit. (Initial value)

[bit11:2] Reserved: Reserved bit

Write0 to these bits.

[bit1:0] USBCK[1:0]: Settings for operation clock supply and gating of USB(device/host)

These bits control the operation clock supply and gating to USB (device/host). The correspondence between each bit and the USB channel is shown below.

bit0 - USBCK0: USB channel 0
bit1 - USBCK1: USB channel 1

When the relevant bit is set to 1, the bus clock is supplied to the corresponding USB channel to use the USB function. For products to which the relevant USB channel is not mounted, it is prohibited to change the relevant bit from the initial value.
When the relevant bit is set to 0, the bus clock input to the corresponding USB channel is gated. While the bus clock input is gated, the USB function of the corresponding channel cannot be used

Bit	Description
0	Gates the bus clock input to the USB channel corresponding to the relevant bit. (Initial value)
1	Supplies the bus clock to the USB channel corresponding to the relevant bit.

4.6 Peripheral Function Reset Control Register 2 (MRST2)

This section explains the peripheral function reset control register 2 (MRST2).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved			PCRCRST	Reserved	CECRST	Reserved	LCDCRST
Attribute	-			R/W	-	R/W	-	R/W
Initial value	-			0	-	0	-	0

bit	15	14	13	12	11	10	9	8
Field	I2SCRST [1:0]		ICCRST [1:0]		Reserved		Reserved	
Attribute	R/W		R/W		-		-	
Initial value	00		00		-		-	

bit	7	6	5	4	3	2	1	0
Field	Reserved						USBRST [1:0]	
Attribute	-						R/W	
Initial value	-						00	

[bit31:21] Reserved: Reserved bits

Write0 to these bits.

[bit20] PCRCRST: Reset control of Programmable-CRC

This bit controls the reset of the Programmable-CRC unit. If this bit is set to 1, the Programmable-CRC becomes a reset state, the operation of the Programmable-CRC stops, and the register settings are initialized. For products to which the Programmable-CRC is not mounted, do not set this bit to 1. To release the above-mentioned reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of Programmable-CRC. (Initial value)
1	Issue the reset signal to Programmable-CRC.

[bit19] Reserved: Reserved bit

Write0 to this bit.

[bit18] CECRST: Reset control of HDMI-CEC/Remote Control Reception

This bit controls the reset of HDMI-CEC/Remote Control Reception.

If the relevant bit is set to 1, the HDMI-CEC/Remote Control Reception becomes a reset state, the HDMI-CEC/Remote Control Reception operation stops, and the register settings are initialized.

To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of HDMI-CEC/Remote Control Reception.(Initial value)
1	Issues the reset to HDMI-CEC/Remote Control Reception.

[bit17] Reserved: Reserved bit

Write0 to this bit.

[bit16] LCDCRST: Reset control of LCD Controller

This bit controls the reset of the LCDC function.

When this bit is set to 1, the LCDC becomes a reset state, the operation of the LCD stops, and the register settings are initialized. For products to which the LCDC is not mounted, do not set this bit to 1. To release the above-mentioned reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of LCDC. (initial value)
1	Issues the reset to LCDC.

[bit15:14] I2SCRST[1:0]: Reset control of MFSI2S

These bits control the reset of each channel of MFSI2S. The correspondence between each bit and the MFSI2S channel is shown below.

[TYPE2-M0+ product]
bit14 – I2SCRST0: MFS I2S interface channel 5
bit15 – I2SCRST1: MFS I2S interface channel 6

[TYPE3-M0+ product]
bit14 – I2SCRST0: MFS I2S interface channel 4
bit15 – I2SCRST1: MFS I2S interface channel 6

If the relevant bit is set to 1, the channel of corresponding MFSI2S becomes a reset state, the MFSI2S operation stops, and the register settings are initialized. For products to which the relevant MFSI2S channel is not mounted, do not change this bit from the initial value. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of MFSI2S channel corresponding to the relevant bit.(Initial value)
1	Issues the reset to MFSI2S channel corresponding to the relevant bit.

[bit13:12] ICCRST[1:0]: Reset control of Smart-Card interface

These bits control the reset of each channel of Smart-Card interface. The correspondence between each bit and the Smart-Card interface channel is shown below.

bit12 – ICCRST0: Smart-Card interface channel 0
bit13 – ICCRST1: Smart-Card interface channel 1

If the relevant bit is set to 1, the channel of corresponding Smart-Card interface becomes a reset state, the Smart-Card interface operation stops, and the register settings are initialized. For products to which the relevant Smart-Card interface channel is not mounted, do not change this bit from the initial value. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of Smart-Card interface channel corresponding to the relevant bit.(Initial value)
1	Issues the reset to Smart-Card interface channel corresponding to the relevant bit.

[bit11:2] Reserved: Reserved bit

Write0 to these bits.

[bit1:0] USBRST[1:0]: Reset control of USB (device/host)

These bits control the reset of each channel of USB (device/host). The correspondence between each bit and the USB channel is shown below.

bit0 - USBRST0: USB channel 0
bit1 - USBRST1: USB channel 1

If the relevant bit is set to 1, the channel of corresponding USB becomes a reset state, the USB operation stops, and the register settings are initialized. For products to which the relevant USB channel is not mounted, do not change this bit from the initial value. To release the reset state, be sure to set this bit to 0 again.

Bit	Description
0	Releases the reset of USB channel corresponding to the relevant bit.(Initial value)
1	Issues the reset to USB channel corresponding to the relevant bit.

5. Peripheral Clock Gating Function Usage Precautions

This section explains the precautions for using peripheral clock gating functions by peripheral function.

Overview

- Control of a peripheral function to which a clock supply is stopped

The register access to a peripheral function to which a clock supply is stopped, both read and write, is not guaranteed. The read value is undefined, and the write operation is prohibited.
The internal state can be reset by controlling peripheral function reset control register (MRST0 and MRST1 and MRST2) while the peripheral clock is gated.

- Combination of peripheral clock settings

Be sure to set all the target peripheral functions to the clock supply side by the peripheral clock registers (CKEN0, CKEN1 and CKEN2) for the functions operated by combining two or more peripheral functions. For example, set a relevant unit of the A/D converter used and a relevant channel of the base timer to the clock supply side respectively by the peripheral clock control registers (CKEN0 and CKEN1) when the base timer is selected for use by the timer trigger of the A/D converter.

- Initialization conditions of peripheral clock settings

The peripheral clock gating function is initialized by the following reset. After issuing the following reset, be sure to reconfigure the peripheral clock gating function.
For details of the following resets, see Chapter "Reset".

  - Power-on reset (PONR)
  - Low voltage detection reset (LVDH)
  - INITX pin input(INITX)
  - Software watchdog reset (SWDGR)
  - Hardware watchdog reset (HWDGR)
  - Clock failure detection reset (CSVR)
  - Anomalous frequency detection reset (FCSR)
  - Software reset (SRST)
  - APB1 bus reset (APBC1_PSR)
  - Deep standby transition reset (DSTR)

Multi-Function Serial Interface

- LIN Sync field detection: LSYN

Execute the setting of the operation clock supply to the corresponding multi-function timer (input capture) separately with the setting of the peripheral clock of multi-function serial interface when the input capture (ICU) is used in the LIN bus interface mode. For the connection between the multi-function serial interface and the input capture, see Extended Pin Function Setting Register (EPFR) in Chapter of I/O port of FM0+ Peripheral Manual.

- MFS I^2S interface

Execute the setting of the operation clock supply to the corresponding MFS I^2S separately with the setting of the peripheral clock of multi-function serial interface when the I^2S is used in the clock synchronous serial interface mode (CSIO).

Base Timer

■ Clock setting unit of base timer

The peripheral clock control of the base timer is executed in the unit of four channels described in Table 5-1.

Table 5-1 Correspondence between Peripheral Clock Gating Setting and Base Timer Channels

Setting Bit of Peripheral Clock Control Register (CKEN1)	Target Channels
bit 0	Base Timer ch3, ch2, ch1, ch0
bit 1	Base Timer ch7, ch6, ch5, ch4
bit 2	Base Timer ch11, ch10, ch9, ch8
bit 3	Base Timer ch15, ch14, ch13, ch12

Multi-Function Timer

FRT Selection Register

For using the following FRT selection function, set the operation clock of the multi-function timer unit on which source-side FRT is mounted to the supply side。

- OCU Connection FRT selection register (OCFS)
- ICU Connection FRT selection register (ICFS)
- ADC Start-up compare connection FRT selection register (ADCMP)

PPG

■ Clock Control of PPG

The control of input clock to PPG synchronizes with the settings of input clock to the multi-function timer. For PPG channel numbers and unit numbers of multi-function timer, see Table 5-2.

Table 5-2 Multi-Function Timer and PPG Input Clock Control

Unit Number of Multi-Function Timer	PPG Channel Number
Unit 0	Channel 0,1,2,3,4,5,6,7
Unit 1	Channel 8,9,10,11,12,13,14,15
Unit 2	Channel 16,17,18,19,20,21,22,23
Unit 3	Channel 24,25,26,27,28,29,30,31

USB (Device/Host)

■ Clock control target

The gating and supplying of the clock for the USB communication cannot be controlled with USBCK bit of the peripheral clock control register 2 (CKEN2). Execute the control of the clock for the USB communication with UCEN bit of USB clock control register (UCCR). For details, see USB Clock Generation in FM0+ Family Peripheral Manual Communication Macro part.

A/D Converter

■ A/D Timer Trigger Selection

When the base timer is used as a startup trigger of the A/D converter, set the operation clock of the selected base timer channel to the supply side.

GPIO/Fast GPIO

■ Restrictions when bus clock is gated

While the bus clock of GPIO/Fast GPIO is gated, some functions of I/O port cannot be used as shown in Table 5-3.

Be sure to confirm the using conditions and execute the bus clock control of GPIO/Fast GPIO.

For details on I/O port functions, see Chapter I/O Port and Fast GPIO

Table 5-3 Restrictions when GPIO/Fast GPIO Clock is Gated

Restrictions	Bus Clock Status	
	Supplied*	Gated*
I/O port Function-Input level reading (PDIR/FPDIR/M_FPDIR register reading)	Available	Prohibited
I/O port Function-Output Level Switching and Status Confirmation (PDOR/FPDOR/M_FPDOR register reading/writing)	Available	Prohibited
I/O port Mode Switching (Setting change of PFR, PCR, DDR, ADE, SPSR, EPFR, and PZR, FPOER registers)	Available	Prohibited
Peripheral Function Operation (Signal Input and Output)	Available	Available
External Interrupt/NMI Control	Available	Available
Reset Input (INITX)	Available	Available
Return from Deep Standby Mode (WKUP pin input)	Available	Available

*: Available: can be used, 、Prohibited: cannot be used.

HDMI-CEC/Remote Control Reception

■ Clock control target

The gating and supplying of the sub clock for the HDMI-CEC/Remote Control Reception cannot be controlled with CECCK bit of the peripheral clock control register 2 (CKEN2). Execute the control of the sub clock for the HDMI-CEC/Remote Control Reception with CECCKE bit of sub clock control register (RCK_CTL). For details, see Chapter Low Power Consumption Mode.

CHAPTER 2-3: High-Speed CR Trimming

This chapter explains the High-Speed CR Trimming Function.

1. High-Speed CR Trimming Function Overview
2. High-Speed CR Trimming Function Configuration and Block Diagram
3. High-Speed CR Trimming Function Operation
4. High-Speed CR Trimming Function Setup Procedure Example
5. High-Speed CR Trimming Function Register List
6. High-Speed CR Trimming Function Usage Precautions

CODE: 9BFCRTRIM_FM0-E03.0

1. High-Speed CR Trimming Function Overview

This section explains frequency trimming function of the high-speed CR oscillator.

The high-speed CR oscillators used for this device have fluctuation range in frequency accuracy due to process variation. The fluctuation range of frequency accuracy due to process variation and temperature change can be reduced by configuring the trimming function.

The high-speed CR trimming function consists of the frequency trimming setup unit and temperature trimming setup unit.

The frequency trimming setup unit has the following functions:

- It can be configured the high-speed CR frequency trimming by writing a trimming value to the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM).
- By using ch.0 of Base Timer, the setting value to the frequency trimming register can be calculated from count value of the specified period.

The temperature trimming setup unit has the following function:

It can be configured the high-speed CR temperature compensation by writing a trimming value to the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM).

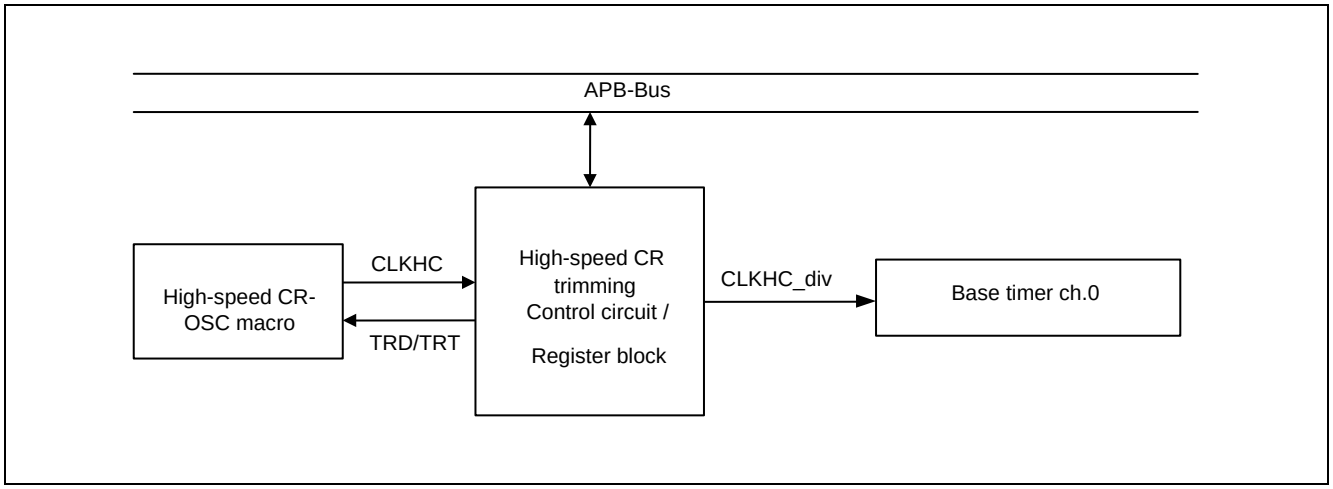
For the high-speed CR frequency accuracy, see electrical characteristics described in "Data Sheet" of the product used.

2. High-Speed CR Trimming Function Configuration and Block Diagram

This section explains the configuration and block diagram of high-speed CR oscillator frequency trimming function.

Figure 2-1 shows the block diagram of high-speed CR frequency trimming function.

Figure 2-1 Block Diagram of the High-Speed CR Oscillator Timing Circuit



Configuration

- High-speed CR OSC macro

A macro of the high-speed CR clock outputs CLKHC (high-speed CR clock).
In addition, the frequency trimming can be performed with TRD bit of high-speed CR oscillation frequency trimming register (MCR_FTRM) and TRT bit of high-speed CR oscillation temperature trimming register (MCR_TTRM).

- High-speed CR Trimming Control Circuit and register block

A control circuit and registers for trimming high-speed CR.
In addition, the high-speed CR clock (CLKHC_div) divided by the ratio set with CSR bit of high-speed CR oscillation frequency division setup register (MCR_PSR) is output to the base timer ch.0.

- Base timer

This block counts frequency before setting to calculate the frequency trimming data for high-speed CR.

Note:

- *For the clock definition, see Chapter "Clock".*

3. High-Speed CR Trimming Function Operation

This section explains operation conducted by frequency trimming function of the high-speed CR oscillator.

Operation of High-Speed CR Oscillation Frequency Trimming Function

- Frequency trimming setup

The setup process writes a trimming data value to the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM) to correct the misalignment of high-speed CR clock accuracy caused by process variation.

- Temperature trimming setup

The setup process writes a trimming data value to the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM) to correct the misalignment of high-speed CR clock accuracy caused by temperature change.

- Register lock function

Write protect function is provided for the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM) and the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM), a function that protects the register from being rewritten without authorization when the system runs out of control.

- Trimming data acquisition

Data written to the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM) can be acquired by one of the following three methods:

  - Use the factory preset value stored in the "CR trimming" area inside the flash memory. After reset is released, the value in the CR trimming area inside the flash memory is stored in the CR Trimming Data Mirror Register (CRTRMM). For data written to the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM) , use the TRMM bits of CR Trimming Data Mirror Register (CRTRMM).
  - Calculate by yourself the value set to the High-speed CR oscillation Frequency Trimming Register from the count value within a certain period by using base timer.
  - Output high-speed CR clock to an external pin, monitor the waveform to trim the frequency and calculate by yourself the value set to the High-speed CR oscillation Frequency Trimming Register.

Data written to the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM) can be acquired by the following method:

  - Use the factory preset value stored in the CR trimming area inside the flash memory.

After reset is released, the value in the CR trimming area inside the flash memory is stored in the CR Trimming Data Mirror Register (CRTRMM). For data written to the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM), use the TTRM bits of CR Trimming Data Mirror Register (CRTRMM).

Notes:

- *Erasing the flash memory also erases the "CR trimming" area inside the memory at the same time. If you use a value in the "CR trimming" area, therefore, save the data to other area (such as RAM) before erasing the flash memory, or only erase sectors other than in the "CR trimming" area.*

- *For the address of the "CR trimming" area, see "FLASH PROGRAMMING MANUAL" of the product used.*

4. High-Speed CR Trimming Function Setup Procedure Example

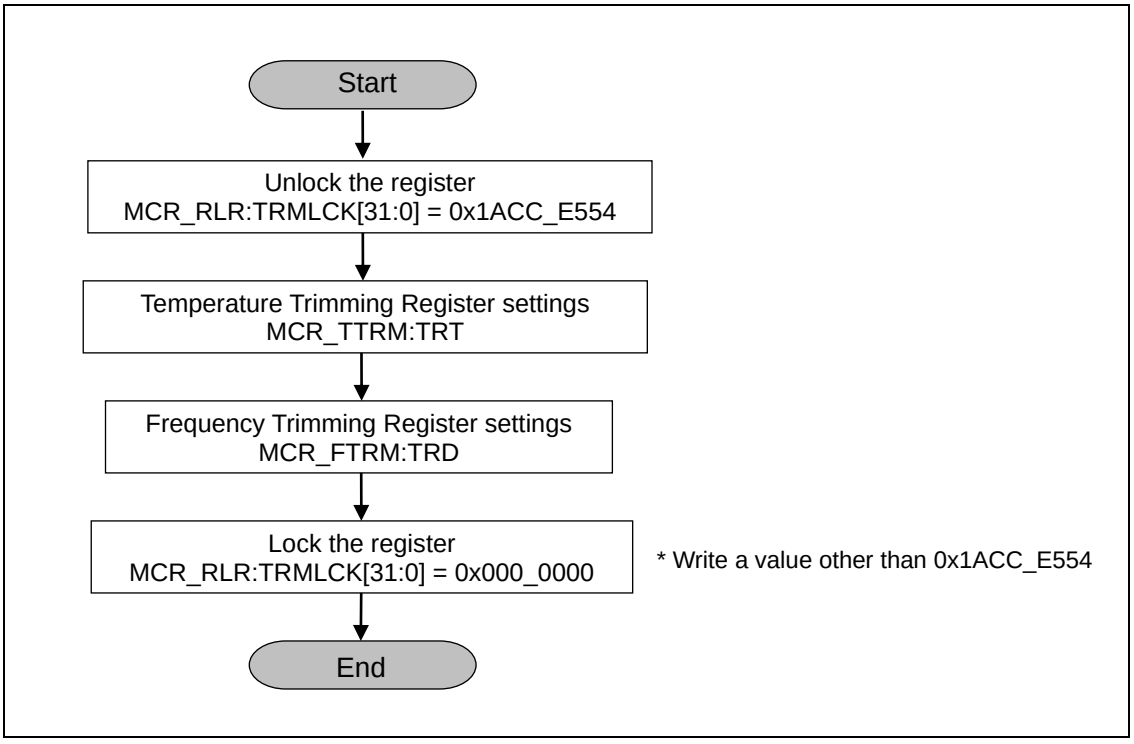
This section provides an example of setting up frequency trimming function of the high-speed CR oscillator.

Frequency Trimming Setup

Take the steps shown in Figure 4-1 to set up frequency trimming.

1. Write "0x1ACCE554" to TRMLCK[31:0] bits of High-speed CR frequency Register Write Protection register (MCR_RLR) to release the lock of Frequency Trimming Setup Register (MCR_FTRM)/ Temperature Trimming Setup Register (MCR_TTRM).
2. Set the trimming data to TRT bit of Temperature Trimming Setup Register (MCR_TTRM).
3. Set TRD bit of Frequency Trimming Setup Register (MCR_FTRM).
4. Write a value other than "0x1ACCE554" to TRMCLK[31:0] bits of High-speed CR Oscillation Register Write Protection Register (MCR_RLR) to lock the Frequency Trimming Setup Register (MCR_FTRM)/ Temperature Trimming Setup Register (MCR_TTRM).

Figure 4-1 Frequency/Temperature Trimming Setup



Frequency Trimming Data Acquisition Example

When acquiring the data from the "CR trimming" area in the flash memory;

Read the "CR trimming" area in the flash memory and get the data.
Write the acquired value to TRD bit of the High-speed CR oscillation Frequency Trimming Setup Register (MCR_FTRM).

How to Calculate the Frequency Trimming Data

The following explains how to calculate the trimming data of high-speed CR oscillation.

1. Let a target oscillation frequency be Ftgt[MHz] and its cycle time be Ttgt[ns]. Let Xtrm_coarse and Xtrm_fine be the TRD[9:5] bit values and TRD[4:0] bit values of the High-speed CR Oscilation Frequency Trimming Setup register at the time respectively.
2. Set “0b00000” to TRD[4:0] bits.
3. Let Xtrm_coarse be Xtrmmin_coarse when “0b00000” is set to TRD[4:0] bits. Let Tmax_coarse[sec] be the cycle at this time.
4. Let Xtrm_coarse be Xtrmmax_coarse when “0b11111” is set to TRD[9:5] bits. Let Tmin_coarse[sec] be the cycle at this time.
5. By calculating the following expression, obtain TRD[9:5] setting value, Xtrm_coarse giving the value more than target oscilation cycle, Ttgt.

$$Xtrm_coarse = \frac{Ttgt - \frac{T\max_coarse - T\min_corse}{31} - T\max_coarse}{\frac{T\min_coarse - T\max_coarse}{31}}$$

*: Round down decimals.

6. Set the obtained Xtrm_coarse to TRD[9:5] bits.
7. Confirm that the High-speed CR clock, F_{CRH}, after setting TRD bits is Ftgt or less. If the F_{CRH} exceed Ftgt, subtract “1” from Xtrm_coarse and then return to Step 6.When the F_{CRH} is Ftgt or less, go to Step 8.
8. Let the value when “0b00000” is set to TRD[4:0] be Xtrimmin_fine. Let Tmax_fine[sec] be the cycle at this time.
9. Let the value when “0b11111” is set to TRD[4:0] be Xtrimmax_fine. Let Tmin_fine[sec] be the cycle at this time.
10. By calculating the following expression, obtain TRD[4:0] setting value, Xtrm_fine giving the target oscilation cycle, Ttgt.

$$Xtrm_fine = \frac{Ttgt - \frac{T\max_fine - T\min_fine}{31} - T\max_fine}{\frac{T\min_fine - T\max_fine}{31}}$$

*: Round down decimals.

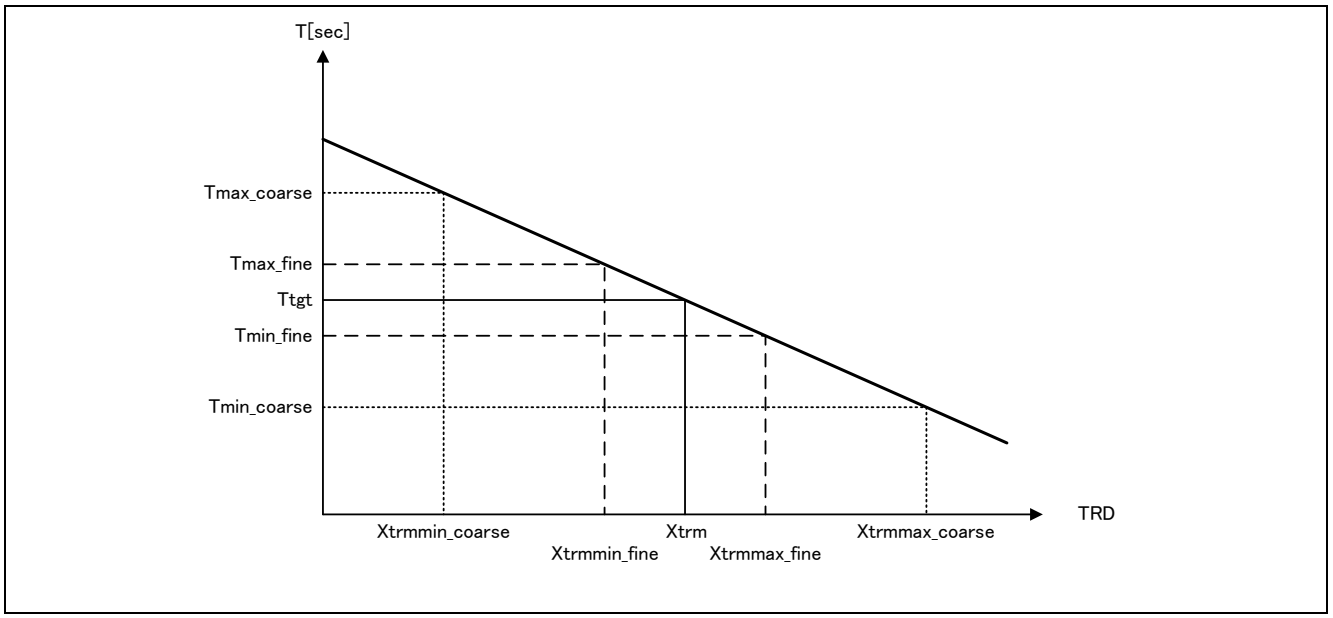
11. Set the obtained Xtrm_fine to TRD[4:0] bits.
12. Confirm whether the High-speed CR clock, F_{CRH}, after setting TRD bits is Ftgt or more and within the specification value of the High-speed CR clock oscillation frequency. If F_{CRH} exceeds the specification

value, subtract “1” from Xtrm_fine and return to Step 11. Moreover, if F_{CRH} is less than Ftgt, add “1” to Xtrm_fine and return to Step 11. When the value is within the specification values, the calculation of trimming data is finished.

Note:

- *For specifications of High-speed CR Clock Oscillation frequency, see “Data Sheet” of the product used.*

Figure 4-2 Method to Trim High-Speed CR Clock



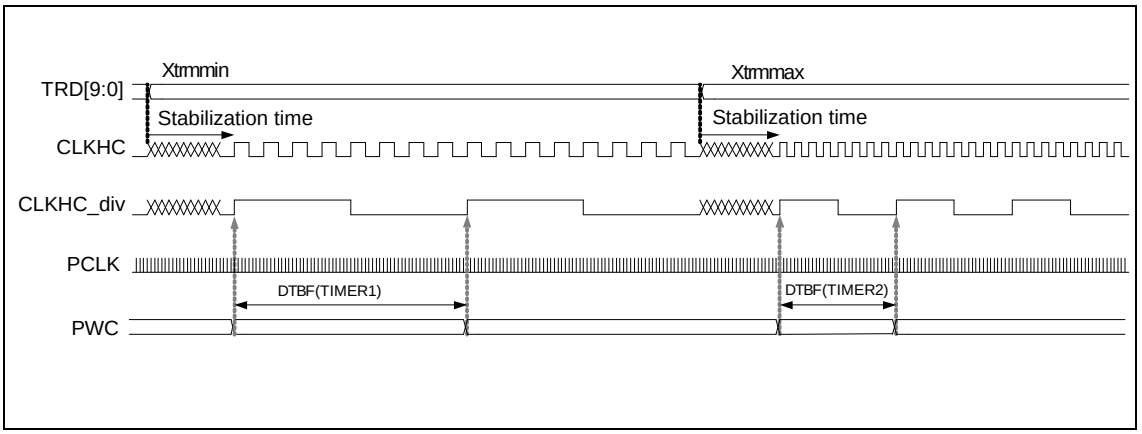
Note:

- *For information about how to measure Tmin_coarse/fine and Tmax_coarse/fine, see " Example of Trimming Data Acquisition Using Base ".*

Example of Trimming Data Acquisition Using Base Timer

Figure 4-3 shows the time chart of high-speed CR oscillation and the trimming process.

Figure 4-3 Time Chart of High-Speed CR Oscillation and the Trimming Process with Base Timer



Run the base timer by setting the main oscillation clock (CLKMO) as the master clock (measurement reference clock).

Activate a trigger on the rising of the high-speed CR frequency division clock (CLKHC_div) when setting Xtrmmin or Xtrmmax, read the base timer value at that time, and perform the following calculations.

Tmax = (TIMER1 × PCLK) / DIV

Tmin = (TIMER2 × PCLK) / DIV

- TIMER1, TIMER2: Count value of base timer (PWC)
- PCLK: APB1 bus clock
- DIV: Frequency division ratio set by CSR bit of Division Setting Register(MCR_PSR)

Example: When PCLK = 40 MHz (25 ns), frequency division ratio = 1/8, and TIMER1 = 100,

Tmax = (100 × 25 ns) / 8 = 312.5 ns

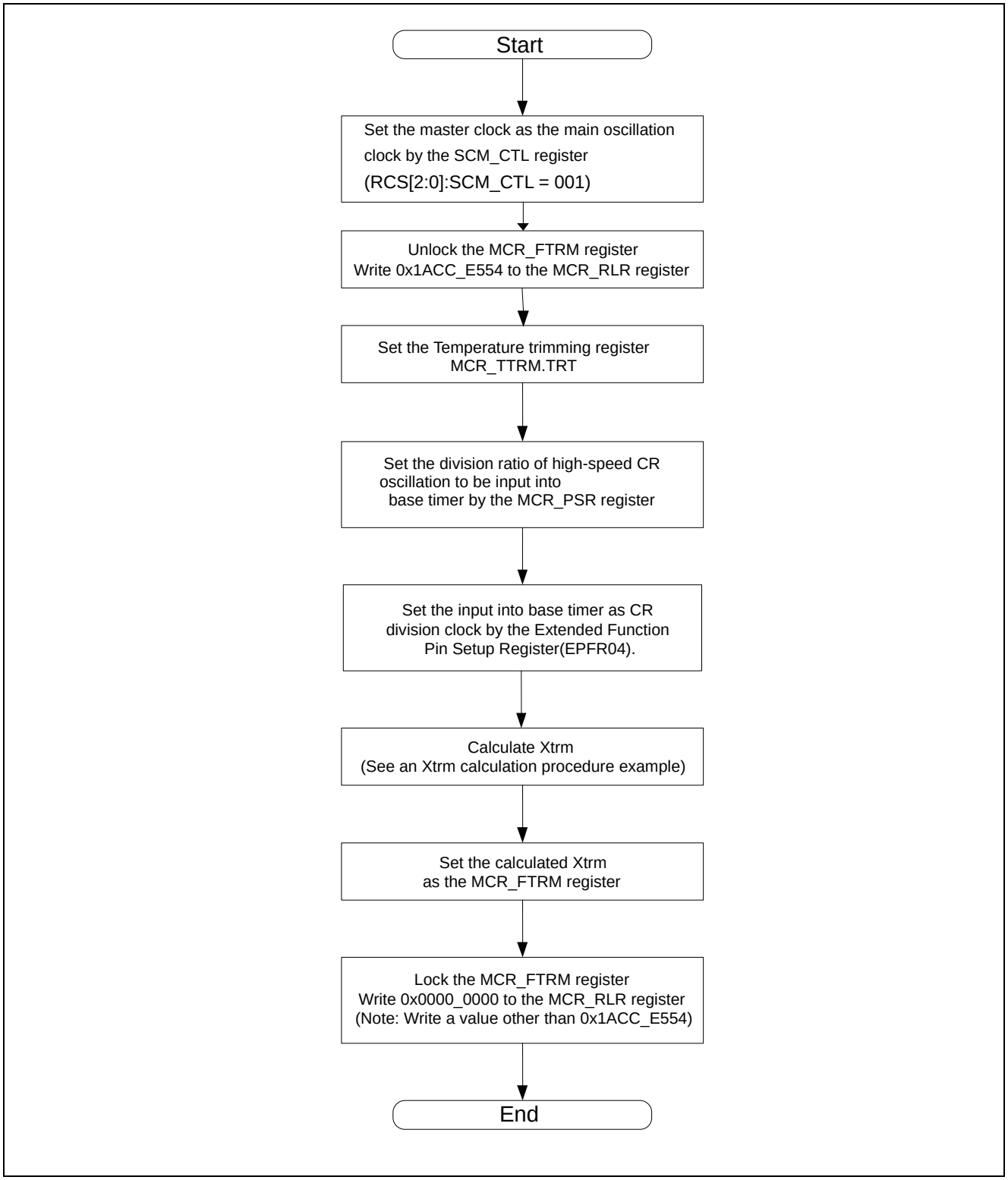
Note:

- *The base timer used for trimming is ch.0.*
  PCLK in Figure 4-3 is an APB1 bus clock.
  At this time, select the master clock as the main oscillation for PCLK.

Frequency Trimming Procedure Example

Figure 4-4 shows a trimming procedure example of high-speed CR oscillation.

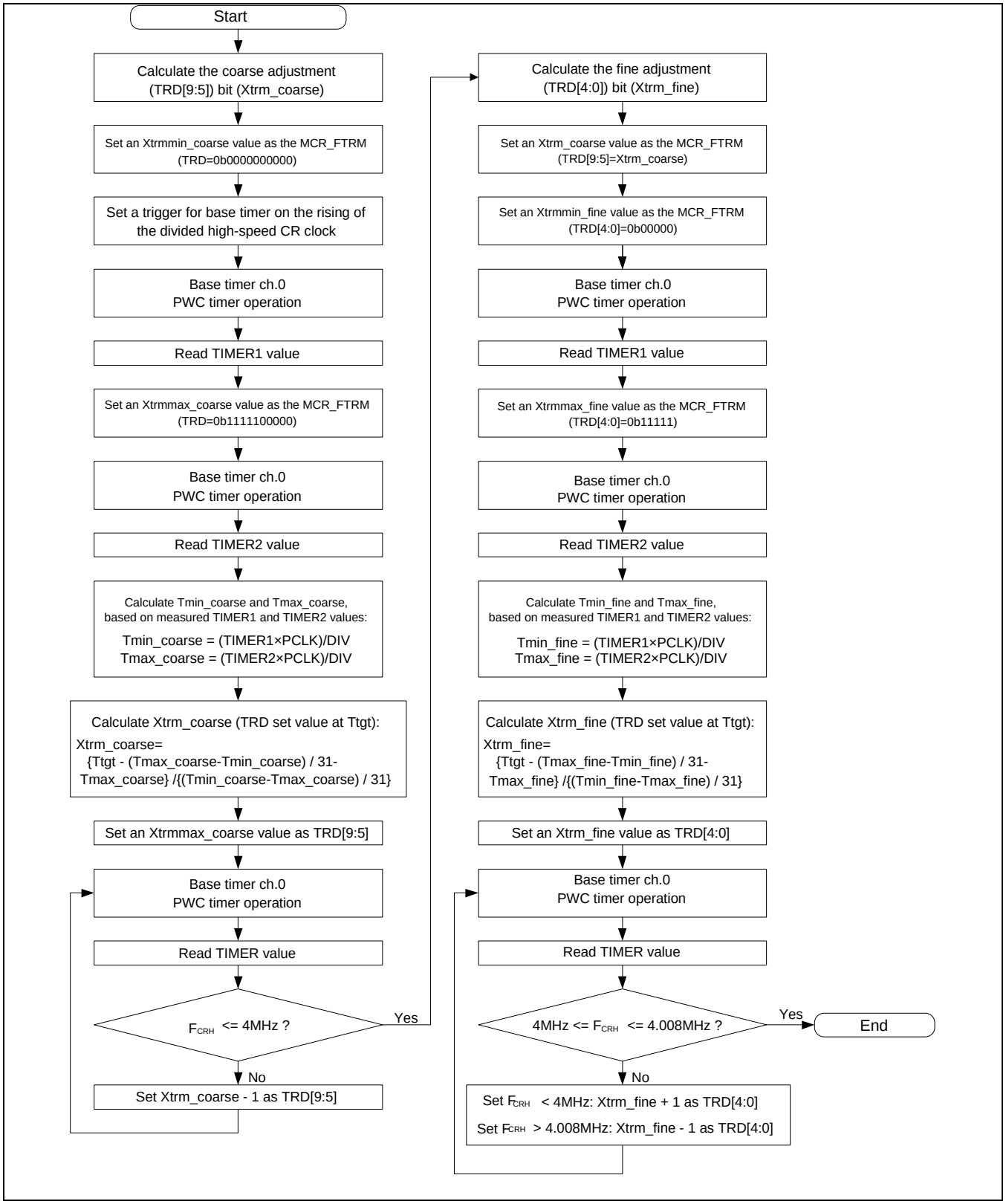
Figure 4-4 Trimming Procedure Example of High-Speed CR Oscillation



Xtrm Calculation Procedure Example (4MHz Oscillator)

Figure 4-5 shows an Xtrm calculation procedure example for 4MHz High-Speed CR Oscillator. Perform frequency trimming in the two stages of coarse adjustment and fine adjustment.

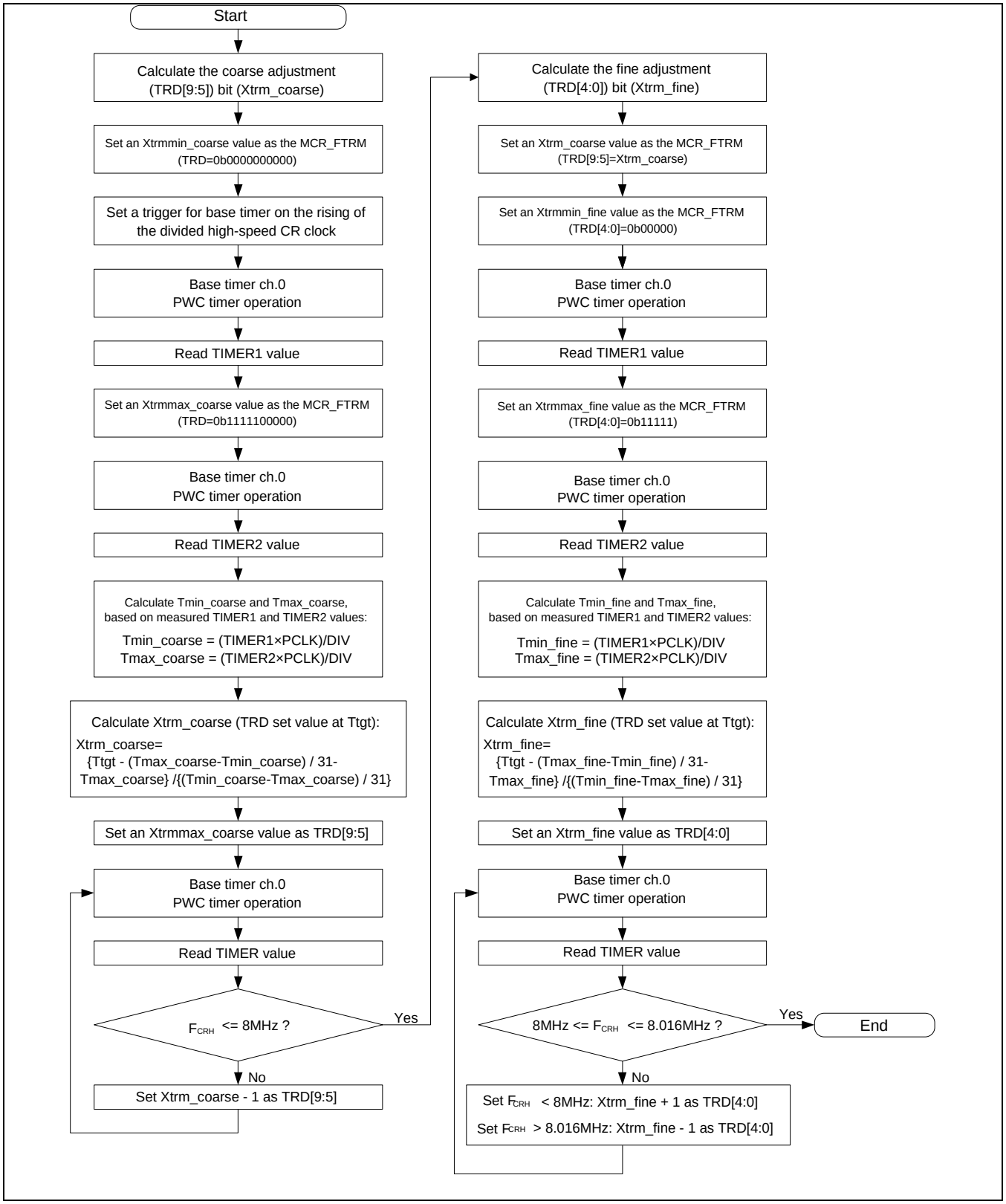
Figure 4-5 Xtrm Calculation Procedure Example (4MHz)



Xtrm Calculation Procedure Example (8MHz Oscillator)

Figure 4-6 shows an Xtrm calculation procedure example for 8MHz High-Speed CR Oscillator. Perform frequency trimming in the two stages of coarse adjustment and fine adjustment.

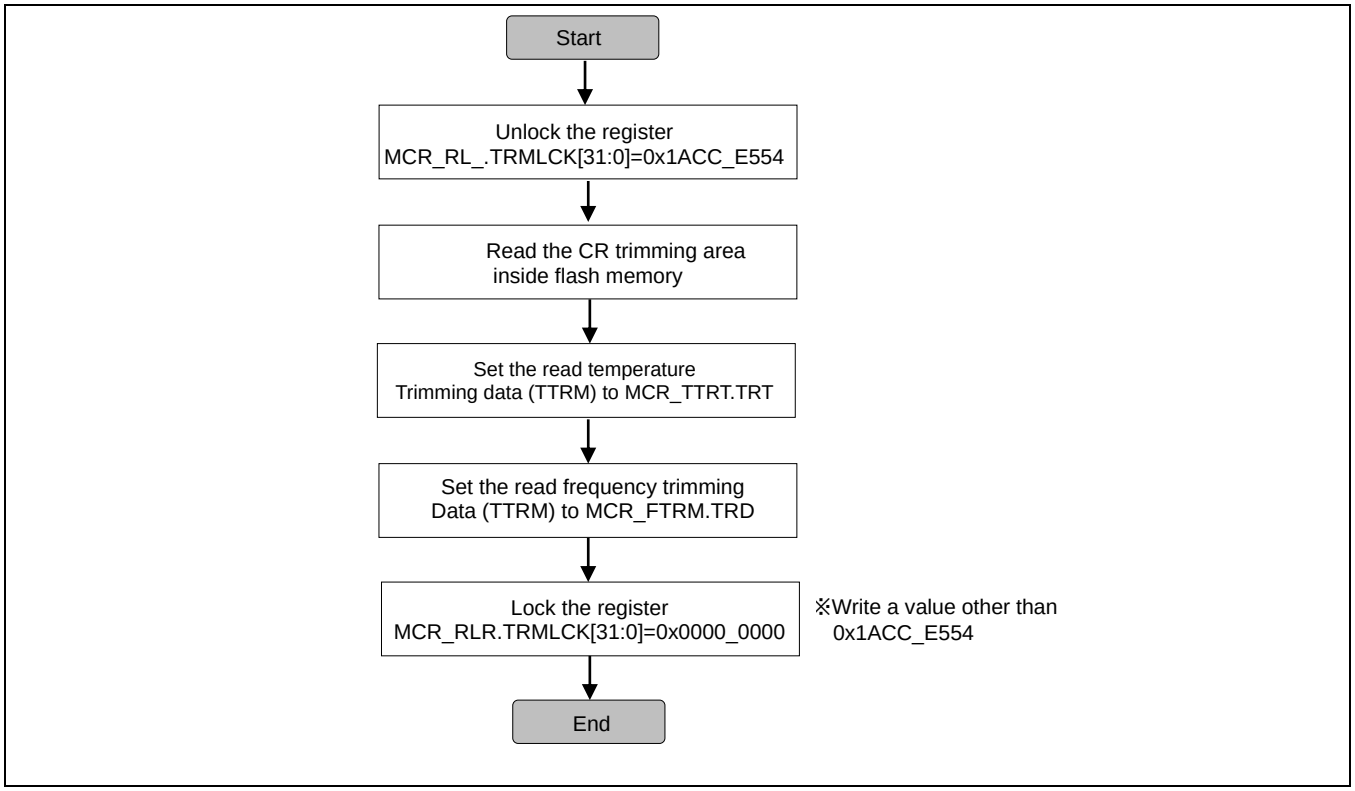
Figure 4-6 Xtrm calculation Procedure Example (8MHz)



Procedure Example of Using "CR Trimming" Area Storage Data Inside Flash Memory

Figure 4-7 shows a procedure example of reading trimming data stored in the "CR trimming" area inside the flash memory and setting it in the High-speed CR oscillation Frequency Trimming Register.

Figure 4-7 Procedure Example of Using "CR Trimming" Area Storage Data



Note:

- *For the address of the CR trimming area, see "FLASH PROGRAMMING MANUAL" for the product used.*

5. High-Speed CR Trimming Function Register List

The following lists and explains registers used for frequency trimming function of the high-speed CR oscillator.

Table 5-1 lists the registers.

Table 5-1 Register List

Abbreviation	Register Name	Reference
MCR_PSR	High-speed CR oscillation Frequency Division Setup Register	5.1
MCR_FTRM	High-speed CR oscillation Frequency Trimming Register	5.2
MCR_TTRM	High-speed CR oscillation Temperature Trimming Register	5.3
MCR_RLR	High-speed CR oscillation Register Write-Protect Register	5.4

5.1 High-Speed CR Oscillation Frequency Division Setup Register (MCR_PSR)

The MCR_PSR register sets the frequency division ratio of high-speed CR oscillation.
A divided clock can be input in base timer.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					CSR		
Attribute	-					R/W		
Initial value	-					001		

Register Functions

[bit7:3] Reserved: Reserved bits

"0b00000" is read from these bits.
Set these bits to "0b00000" when writing.

[bit2:0] CSR: High-speed CR oscillation frequency division ratio setting bits

bit2	bit1	bit0	Description
0	0	0	1/4
0	0	1	1/8 [Initial value]
0	1	0	1/16
0	1	1	1/32
1	0	0	1/64
1	0	1	1/128
1	1	0	1/256
1	1	1	1/512

5.2 High-Speed CR Oscillation Frequency Trimming Register (MCR_FTRM)

The MCR_FTRM register sets the frequency trimming value.

Register Configuration

bit	31															16
Field	Reserved															
Attribute	-															
Initial value	-															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved						TRD[9:0]									
Attribute	-						R/W									
Initial value	-						01111 01111 (TYPE1-M0+, TYPE2-M0+) 10000 00110 (TYPE3-M0+)									

Register Functions

[bit31:10] Reserved: Reserved bits

"0" is always read from these bits.

These bits have no effect when written.

[bit9:0] TRD[9:0]: Frequency trimming setup bits

bit9:5	Description
When write	These bits make coarse adjustment to the high-speed CR oscillator frequency. For values to be set, see trimming data acquisition in the operation explanation of the frequency trimming function. These bits fluctuate in frequency steps as below each time ±1 setting is made. approximately 1.0% (TYPE1-M0+) approximately 3.3% (TYPE2-M0+) approximately 1.8% (TYPE3-M0+)
When read	A specified value is read. As an initial value, "0b01111" is read. (TYPE1-M0+,TYP2-M0+) As an initial value, "0b10000" is read. (TYPE3-M0+)

bit4:0	Description
When write	These bits make fine adjustment to the high-speed CR oscillator frequency. For values to be set, see trimming data acquisition in the operation explanation of the frequency trimming function. These bits fluctuate in frequency steps as below each time ±1 setting is made. approximately 0.2% (TYPE1-M0+) approximately 0.3% (TYPE2-M0+) approximately 0.1% (TYPE3-M0+)
When read	A specified value is read. As an initial value, "0b01111" is read. (TYPE1-M0+,TYP2-M0+) As an initial value, "0b00110" is read. (TYPE3-M0+)

Notes:

- *This register is not initialized by software reset.*
- *For values to be set to the TRD bits, see trimming data acquisition in the operation explanation of the frequency trimming function.*

5.3 High-Speed CR Oscillation Temperature Trimming Setup Register (MCR_TTRM)

The MCR_TTRM register sets the temperature trimming value.

TYPE1-M0+ and TYPE2-M0+/TYPE3-M0+ have different register definition.

5.3.1 MCR_TTRM (TYPE1-M0+)

Register Configuration

bit	31															16
Field	Reserved															
Attribute	-															
Initial value	-															

bit	15											5	4	3	2	1	0
Field	Reserved												TRT[4:0]				
Attribute	-												R/W				
Initial value	-												10000				

Register Functions

[bit31:5] Reserved: Reserved bits

"0" is always read from these bits.

These bits have no effect when written.

[bit4:0] TRT[4:0]: Temperature trimming setup bits

bit4:0	Description
When write	These bits make adjustment to the high-speed CR oscillator frequency. Write the value read from Temperature Trimming bit storage area in Flash Memory. For Temperature Trimming bit storage area, see “FLASH PROGRAMING MANUAL” of the product used.
When read	A specified value is read. As an initial value, 0b10000 is read.

Notes:

- *This register is not initialized by software reset.*

Before obtaining the frequency trimming data, be sure to set this register.

5.3.2 MCR_TTRM (TYPE2-M0+)

Register Configuration

bit	31															16
Field	Reserved															
Attribute	-															
Initial value	-															

bit	15							6	5	4	3	2	1	0
Field	Reserved							TRT[6:0]						
Attribute	-							R/W						
Initial value	-							0111111						

Register Functions

[bit31:7] Reserved: Reserved bits

"0" is always read from these bits.
These bits have no effect when written.

[bit6:0] TRT[6:0]: Temperature trimming setup bits

bit4:0	Description
When write	These bits make adjustment to the high-speed CR oscillator frequency. Write the value read from Temperature Trimming bit storage area in Flash Memory. For Temperature Trimming bit storage area, see “FLASH PROGRAMING MANUAL” of the product used.
When read	A specified value is read. As an initial value, 0b0111111 is read.

Notes:

- *This register is not initialized by software reset.*

Before obtaining the frequency trimming data, be sure to set this register.

5.3.3 MCR_TTRM (TYPE3-M0+)

Register Configuration

bit	31															16
Field	Reserved															
Attribute	-															
Initial value	-															

bit	15														6	5	4	3	2	1	0
Field	Reserved														TRT[6:0]						
Attribute	-														R/W						
Initial value	-														1111111						

Register Functions

[bit31:7] Reserved: Reserved bits

"0" is always read from these bits.

These bits have no effect when written.

[bit6:0] TRT[6:0]: Temperature trimming setup bits

bit4:0	Description
When write	These bits make adjustment to the high-speed CR oscillator frequency. Write the value read from Temperature Trimming bit storage area in Flash Memory. For Temperature Trimming bit storage area, see “FLASH PROGRAMING MANUAL” of the product used.
When read	A specified value is read. As an initial value, 0b1111111 is read.

Notes:

- *This register is not initialized by software reset.*
- *Before obtaining the frequency trimming data, be sure to set this register.*

5.4 High-Speed CR Oscillation Register Write-Protect Register (MCR_RLR)

The MCR_RLR register controls the write-protect state of the frequency trimming register (MCR_FTRM)/high-speed CR oscillation temperature trimming register (MCR_TTRM).

Register Configuration

bit	31															16
Field	TRMLCK[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15															0
Field	TRMLCK[15:0]															
Attribute	R/W															
Initial value	0x0001															

Register Functions

[bit31:0] TRMLCK[31:0]: Register write-protect bits

bit31:0	Description
When read	When 0x00000000 is read, the MCR_FTRM/MCR_TTRM register is currently unlocked. When 0x00000001 is read, the MCR_FTRM/MCR_TTRM register is currently locked.
Writing a value other than 0x1ACCE554	Locks the MCR_FTRM/MCR_TTRM register
Writing 0x1ACCE554	Unlocks the MCR_FTRM/MCR_TTRM register

Note:

- *This register is not initialized by software reset.*

6. High-Speed CR Trimming Function Usage Precautions

This section explains the precautions for using the high-speed CR trimming function.

- Low-speed CR oscillator

  This trimming function is only enabled for the high-speed CR oscillator.
  It cannot apply to the low-speed CR oscillator.

- Data stored in the "CR trimming" area

  The CR trimming" area stores the factory preset frequency trimming data. For the address of the "CR trimming" area, see "FLASH PROGRAMMING MANUAL" for the product used.
  When Data in flash memory is erased, the data in “CR trimming” area is also erase at the same time. To use the data in the “CR trimming” area, save the data in the “CR trimming” area to other area such as RAM before erasing the data in flash memory.
  Otherwise, erase the sectors other than those in “CR trimming” area.

- For accuracy of oscillator oscillation frequency of High-seed CR oscillator

  Without setting High-speed CR oscillation temperature trimming register (MCR_TTRM) and High-speed CR oscillation frequency trimming register (MCR_FTRM), the accuracy of the High-speed CR oscillator described in “Data Sheet” cannot be guaranteed. So, be sure to set the above registers before use.

- How to use base timer

  For information about how to use base timer, see Chapters "Base Timer" in "Timer Part" and "I/O Port".

- FCS (Anomalous Frequency Detection)

  For FCS function (anomalous frequency detection), see Chapter "Clock supervisor". Do not perform CR trimming after the FCS function is enabled.

CHAPTER 2-4: Low-Speed CR Prescaler

This chapter shows the functions and operation of low-speed CR Prescaler.

1. Low-speed CR Prescaler Overview
2. Low-speed CR Prescaler Configuration
3. Low-speed CR Prescaler Operation and Setup Procedure Example
4. Low-speed CR Prescaler Register

CODE:9BFLCPC-FM0-E03.0

1. Low-speed CR Prescaler Overview

This section shows the overview of low-speed CR prescaler.

Low-speed CR Prescaler

By setting the low-speed CR prescaler load register(LCR_PRSLD), the low-speed CR prescaler divides low-speed CR and generates low-speed CR clock(CLKLC).
This macro can correct the accuracy of low-speed CR. For the correcting method, see the example of correcting low-speed CR.

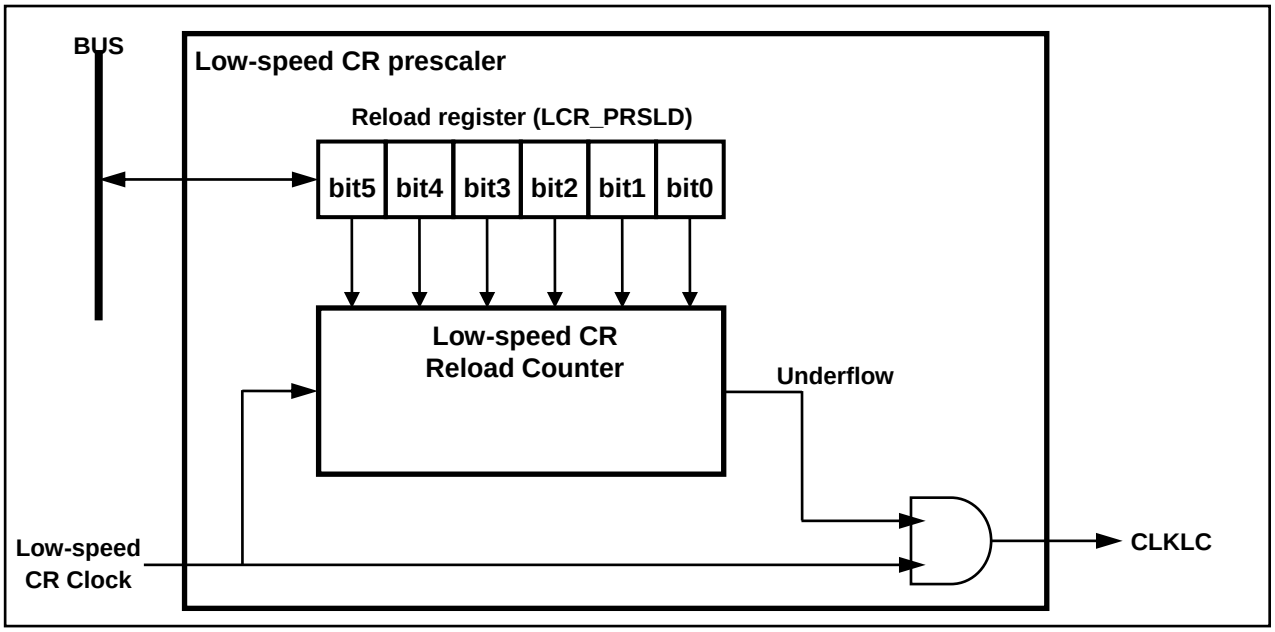
2. Low-speed CR Prescaler Configuration

This section shows the block diagram of low-speed CR prescaler.

Block Diagram of Low-speed CR Prescaler

For the block diagram of low-speed CR prescaler, see Figure 2-1.

Figure 2-1 Block Diagram of Low-speed CR Prescaler



- Low-speed CR Prescaler Load Register (LCR_PRSLD)

  Sets the division ratio (reload value) of Low-speed CR Prescaler.

- Low-speed CR Reload Counter

  This is the down counter which generates the Low-speed CR Division Clock (CLKLC).

3. Low-speed CR Prescaler Operation and Setup Procedure Example

This section explains the operation of Low-speed CR Prescaler. This section also shows the example of setup procedures.

3.1 Setup Procedures of Low-speed CR Prescaler

The Low-speed CR is asynchronous with the peripheral clock (PCLK).

For writing to the Low-speed CR Prescaler Reload Register, the peripheral clock is used. Therefore, if the setting change of the Low-speed CR Prescaler Load Register and the reload of the reload counter occur simultaneously, a value reloaded to the reload counter is not guaranteed.

So, execute the rewriting of the Low-speed CR Prescaler Reload Register conforming to the following procedures.

■ For Switching the division clock

The initial value of the Low-speed CR Prescaler Reload Register(LCR-PRSLD) is “0”.

Thus, for changing the value from the initial value, these procedures are unnecessary.

1. Set "0" to the Low-speed CR Prescaler Reload Register (LCR_PRSLD).
2. Wait until the value of the Low-speed CR Prescaler Reload Register (LCR_PRSLD) is reloaded to the reload counter.
3. The wait time is obtained by calculating the following formula:
   Low-speed CR cycle (50 kHz: 20 µs) × "the set value before changed to "0" in Item 1."
4. Write new setup value to the Low-speed CR Prescaler Reload Register (LCR_PRSLD).

For wait time at setup change, see Table 3-1.

Table 3-1 Setup Wait Time

Reload Value before Setup	Setup Value	Wait Time
0	0	Not exists.
1	0	20 µs (20 µs × 1)
2	0	40 µs (20 µs × 2)
3	0	60 µs (20 µs × 3)
:	:	:
60	0	1200 µs (20 µs × 60)
61	0	1220 µs (20 µs × 61)
62	0	1240 µs (20 µs × 62)
63	0	1260 µs (20 µs × 63)

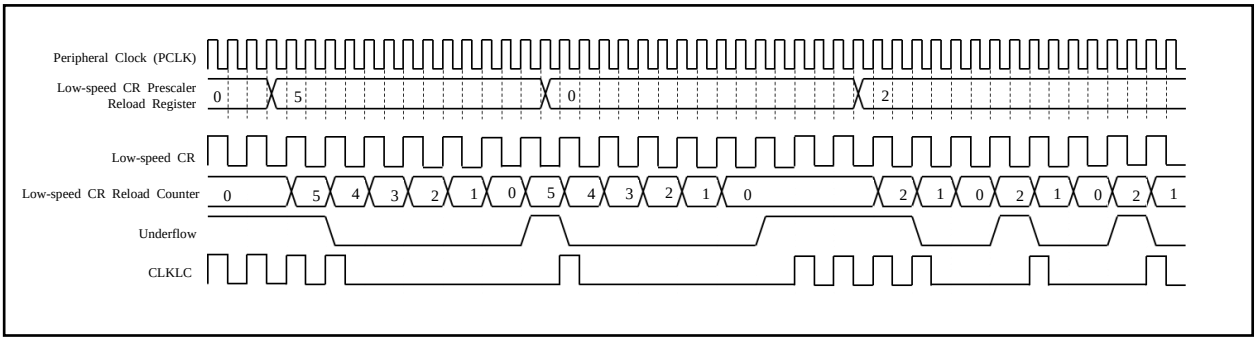
Notes:

- *The division clock cannot be stopped.*
- *The setting of the Low-speed CR Prescaler Reload Register (LCR_PRSLD) is executed at the underflow of the Low-speed CR Reload Counter.*

3.2 Operation of Low-speed CR Prescaler

For the operation of the Low-speed CR Prescaler, see Figure 3-1.

Figure 3-1 Low-speed CR Prescaler Operation



(1) Sets the Low-speed CR Prescaler Load Register (LCR_PRSLD) in synchronization with the peripheral clock (PCLK)

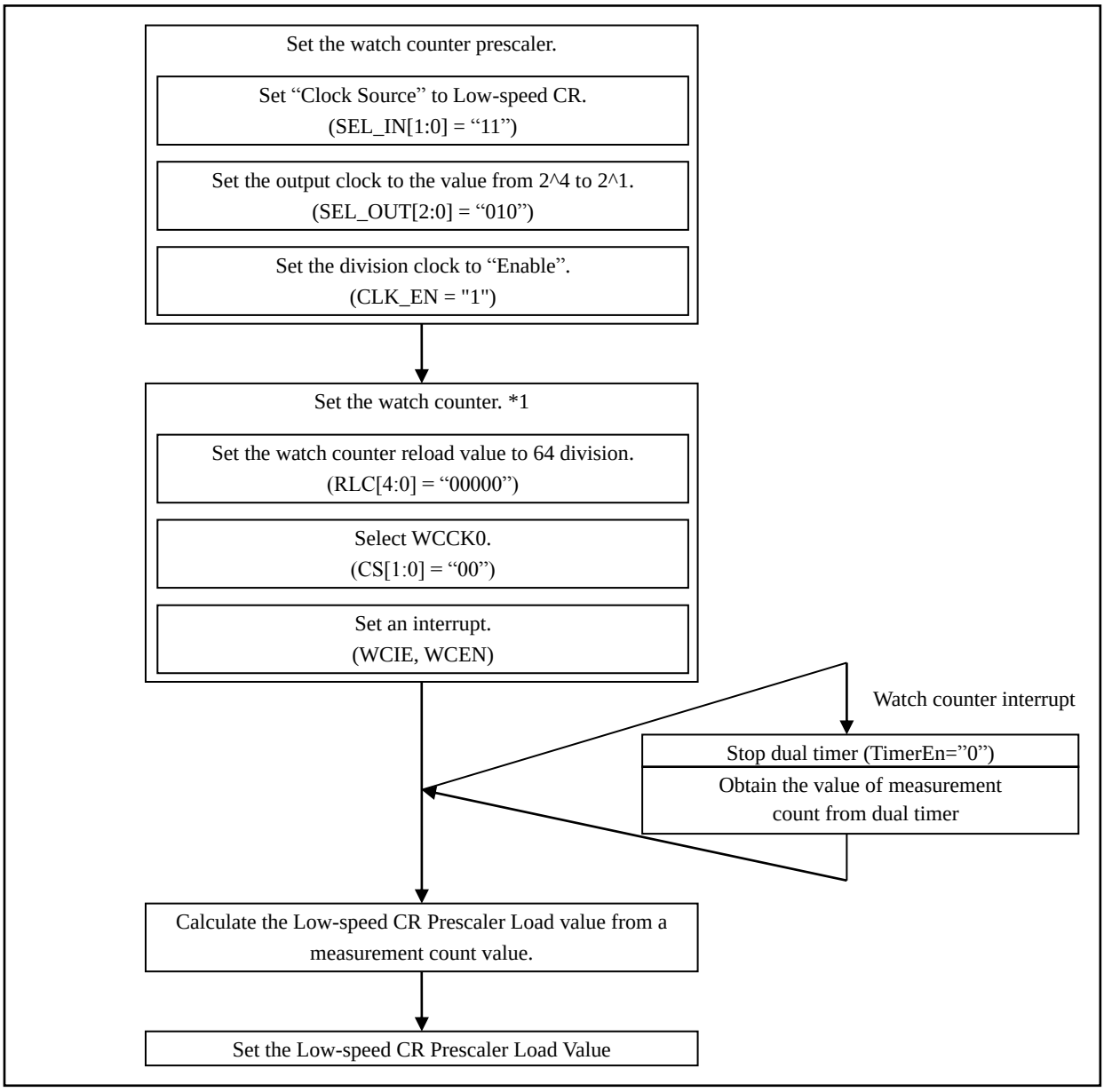
(2) Retrieves the value of the Low-speed CR Prescaler Load Register (LCR_PRSLD) at the moment the Low-speed CR Reload Counter indicates "0".

(3) Outputs the Low-speed CR (CLKLC) at the moment when the Low-speed CR Reload Counter underflow occurs.

3.3 Low-speed CR Correction Example

For the correction example of the Low-speed C, see Figure 3-2.

Figure 3-2 Low-speed CR Correction Example



*1 : Above is the example by using dual timer. It is possible to measure by using Base timer or MFT.

4. Low-speed CR Prescaler Register

This section shows the list of the Low-speed CR Prescaler Register.

Low-speed CR Prescaler Register

Table 4-1 List of Low-speed CR Prescaler Register

Abbreviation	Register name	Reference
LCR_PRSLD	Low-speed CR Prescaler Control Register	4.1

4.1 Low-speed CR Prescaler Control Register (LCR_PRSLD)

The Low-speed CR Prescaler Control Register is used to set the division ratio of low-speed CR.

bit	7	6	5	4	3	2	1	0
Field	Reserved		LCR_PRSLD[5:0]					
Attribute	-		R/W					
Initial Value	00		000000					

[bit7:6] Reserved: Reserved bits

Always “0” is read.

They have no effect in write mode.

[bit5:0] LCR_PRSLD: Low-speed CR Prescaler Load

At writing, sets the division ratio of the Low-speed CR Prescaler (the reload value of a reload counter) .

At reading, the set value is read.

Note:

- *This register is not initialized with software reset.*

CHAPTER 3: Clock Supervisor

This chapter explains the clock supervisor functions.

1. Overview
2. Configurations and Block Diagrams
3. Explanation of Operations
4. Setup Procedure Examples
5. Operation Examples
6. Registers
7. Usage Precautions

CODE: 9BFCSV-FM0-E03.0

1. Overview

This section provides an overview of the clock supervisor functions.

The clock supervisor includes the following two types of functions.

Clock Failure Detection (CSV: Clock Failure Detection by Clock Supervisor)

The clock failure detection monitors the main and sub clocks. If a rising edge of the monitored clock is not detected within the specified period, this function determines that the oscillator has failed, and outputs a system reset request.

Anomalous Frequency Detection (FCS: Anomalous Frequency Detection by Clock Supervisor)

The anomalous frequency detection monitors frequency of the main clock. Within the specified period between an edge and the next edge of the divided clock of high-speed CR, this function counts up the internal counter value using the main clock. If the count value reaches out of the set window range, the function determines that the main clock frequency is anomalous, and outputs an interrupt request to the CPU or a system reset request.

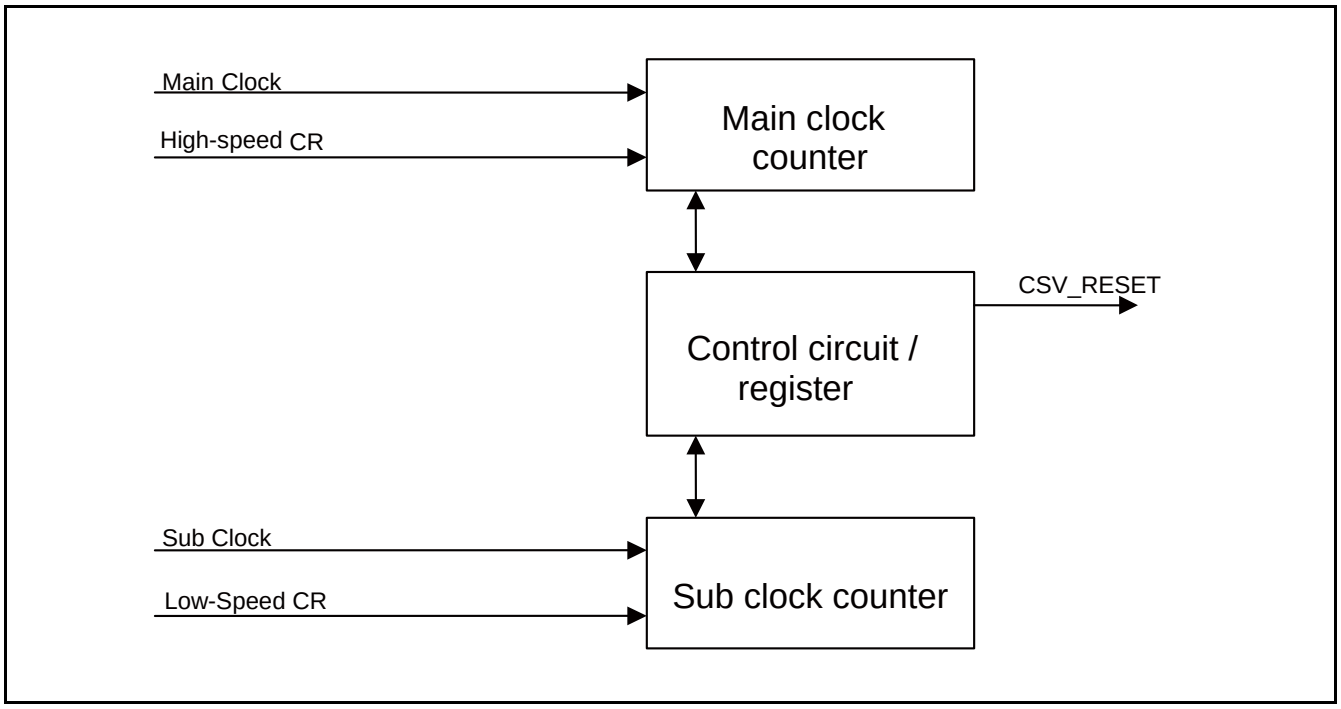
2. Configurations and Block Diagrams

This section explains the block diagrams of the clock supervisor functions.

2.1 Clock Failure Detection

Figure 2-1 shows the block diagram of the clock failure detection.

Figure 2-1 Clock Failure Detection Block Diagram



The clock failure detection consists of the following three types of blocks.

Control Circuit/Register

- This block includes a circuit controlling the clock failure detection,
- Also includes setup registers enabling/disabling the clock failure detection.

Main Clock Counter

A counter that monitors the main clock with the high-speed CR clock.

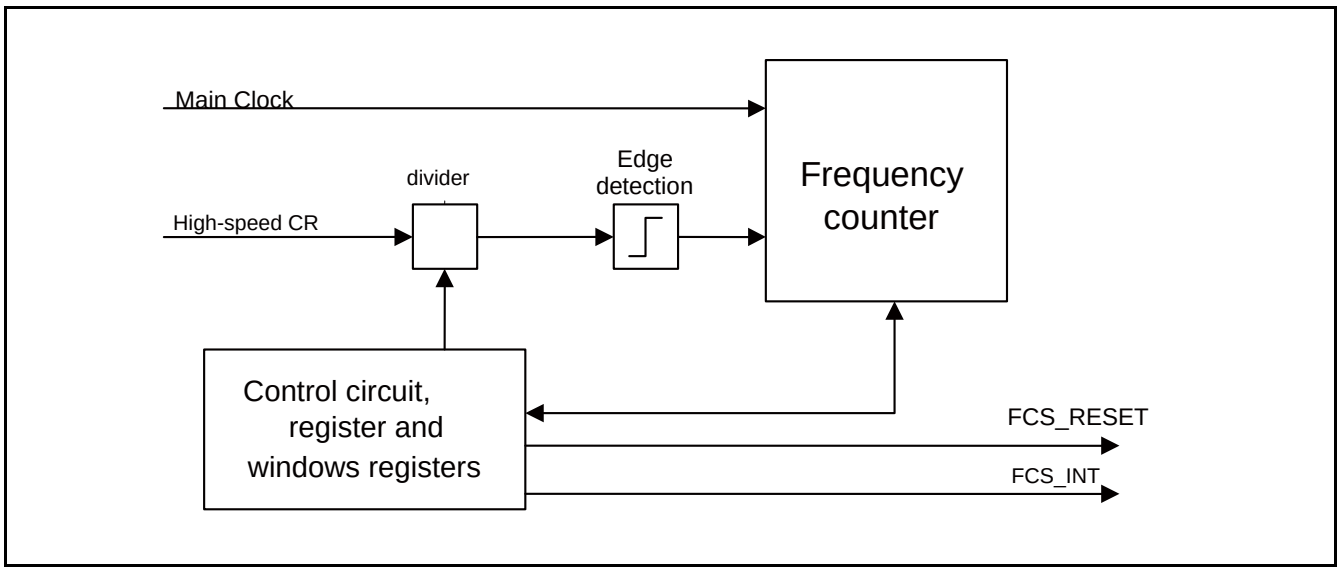
Sub Clock Counter

A counter that monitors the sub clock with the low-speed CR clock.

2.2 Anomalous Frequency Detection

Figure 2-2 shows the block diagram of the anomalous frequency detection.

Figure 2-2 Anomalous Frequency Detection Block Diagram



The anomalous frequency detection consists of the following three types of blocks.

Control Circuit/Register and Window Registers

- This block includes a circuit controlling the anomalous frequency detection.
- Also includes setup registers enabling/disabling the anomalous frequency detection.
- Also includes window registers defining the frequency range for measurements.

Frequency Counter

A counter based on the main clock.

Divider/Edge Detection

- This block divides the high-speed CR.
- Also detects rising edges of the divided clock of high-speed CR.

3. Explanation of Operations

This section explains the operations of the clock supervisor functions.

Clock Failure Detection Function

The clock failure detection function monitors the main and sub clocks. If a rising edge of the monitored clock is not detected within the specified period, this function determines that the oscillator has failed, and outputs a system reset request.

- This reset request is called as the CSV reset request.
- CSV function monitors each of the main and sub clocks independently.
- It stops monitoring when the main and sub oscillators stop oscillating.
- It stops monitoring while waiting for oscillation stabilization wait time.
- When the oscillation stabilization wait time of main and sub oscillators ends, CSV function is automatically enabled.

Notes:

- *Each of the main and sub clock failure detection function can be enabled/disabled independently using the CSV control register (CSV_CTL).*
- *The main clock is monitored with the high-speed CR clock, and the sub clock is monitored with the low-speed CR clock. When a rising edge is not detected within 32 clocks of high-speed CR for the main clock, or within 32 clocks of low-speed CR for the sub clock, this function determines that the oscillator has failed.*

Anomalous Frequency Detection Function

The anomalous frequency detection function monitors the main clock.

Within the specified period between a rising edge and the next rising edge of the divided clock of high-speed CR, this function counts up the internal counter using the main clock. If the count value reaches out of the set window range, the function determines that the main clock frequency is anomalous, and outputs an interrupt request or a system reset request to the CPU.

- This interrupt request is called as the FCS interrupt request, and reset request is called as the FCS reset request.
- The FCS function only monitors frequency of the main clock.
- It stops monitoring when the main oscillator stops oscillating.
- It stops monitoring while waiting for oscillation stabilization wait time.
- The FCS function is started with software, a user program.

Notes:

- *If the FCS reset is enabled:*

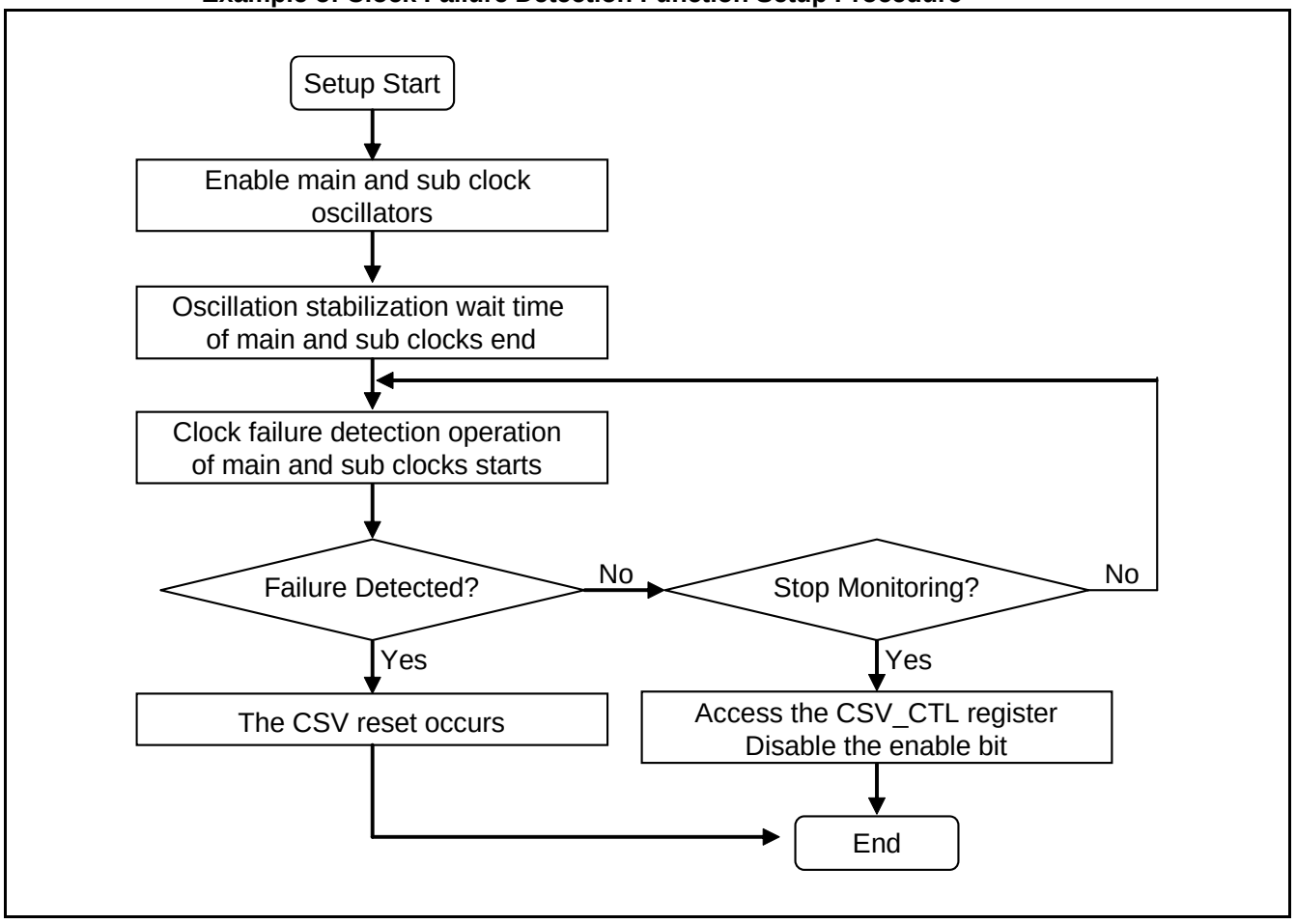
  An interrupt request occurs the first time a counter value deviates from the set window. If the interrupt request has not been cleared, and the counter value falls out of the specified window, a system reset request is output.
  If the FCS reset is not enabled, the reset request is masked.

- *The counter value, if it goes out of the specified window, is stored in the frequency detection counter register (FCSWD_CTL).*

4. Setup Procedure Examples

This section explains examples of setting up the clock supervisor functions.

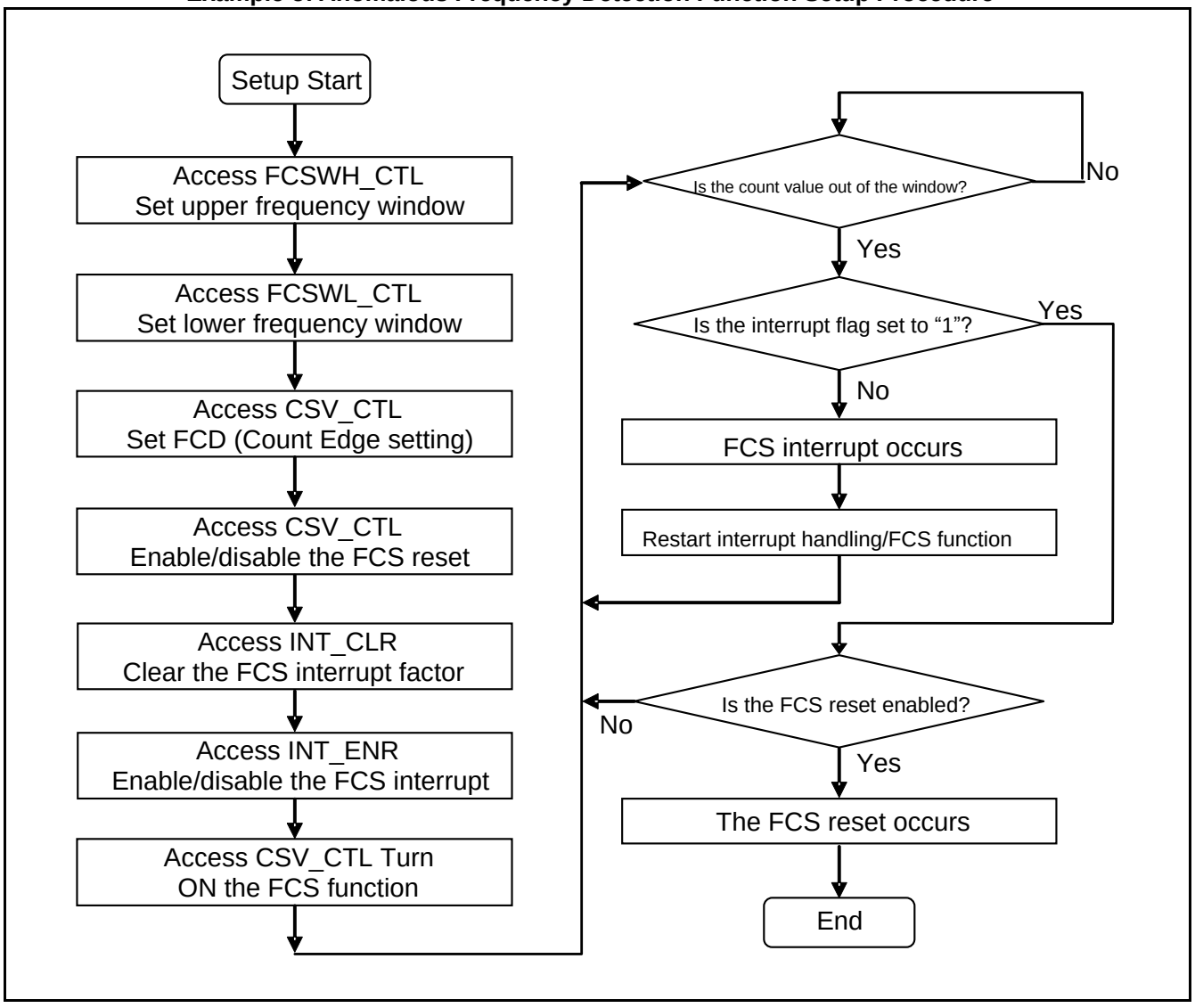
Example of Clock Failure Detection Function Setup Procedure



Notes:

- *When 32 kHz oscillation clock control linkage bit of VBAT register Sub oscillation control register (WTOSCCNT.SOSCNTL) is changed from 1 to 0, write a register value after the sub clock oscillation stabilization wait completion.*
- *To operate only VBAT domain with turning off the power on CHIP side, set WTOSCCNT.SOSCNTL=0 and then turn off the power on CHIP side. Moreover, after the power on CHIP side is turned off, sub clock supervisor function does not operate.*
- *For details on VBAT, see Chapter VBAT Domain.*

Example of Anomalous Frequency Detection Function Setup Procedure



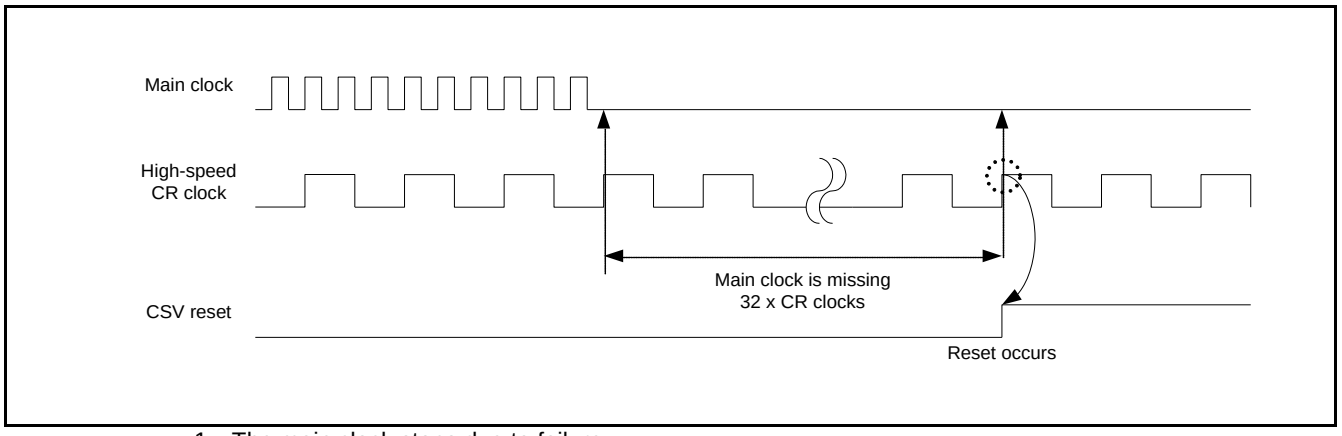
5. Operation Examples

This section explains examples of clock supervisor operations.

5.1 Clock Failure Detection

Figure 5-1 provides an example of clock failure detection operation.

Figure 5-1 Example of Clock Failure Detection Operation



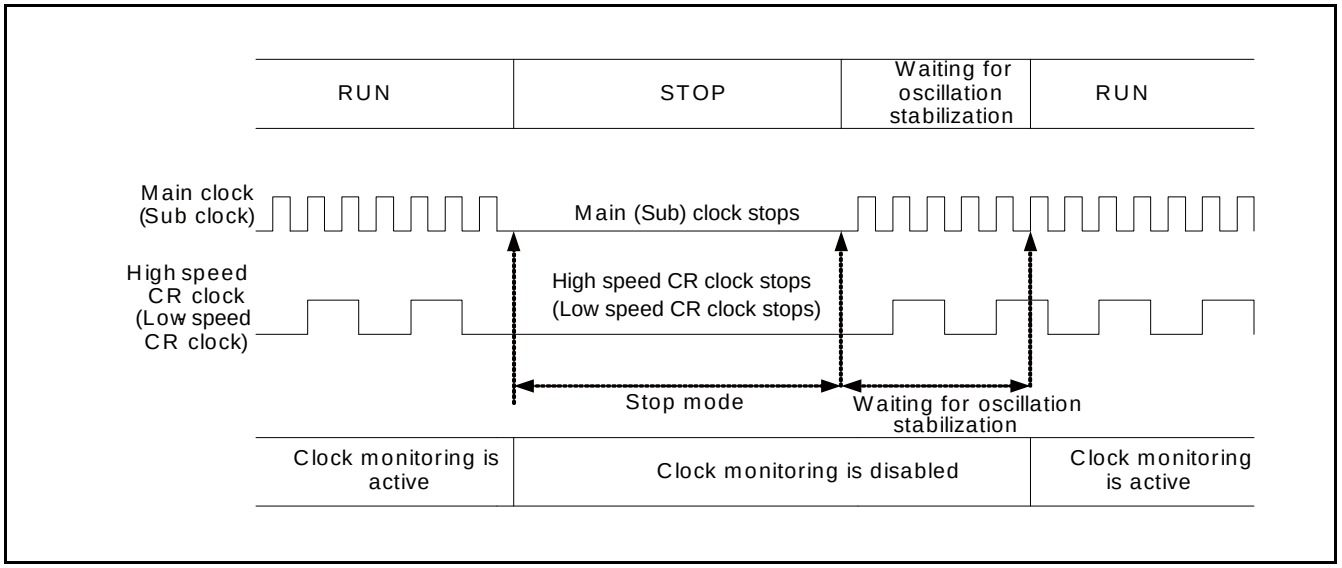
1. The main clock stops due to failure.
2. The function counts up clocks using the high-speed CR clock.
3. If the main clock keeps stopping during 32 clocks of high-speed CR, the function determines that the clock has failed and issues the CSV reset.

Note:

- *In case of the sub clock monitoring, the function determines that the sub clock has failed if it keeps stopping during 32 clocks of low-speed CR.*

Figure 5-2 provides an example of the clock failure detection operation in stop mode.

Figure 5-2 Example of Clock Failure Detection Operation in Stop Mode



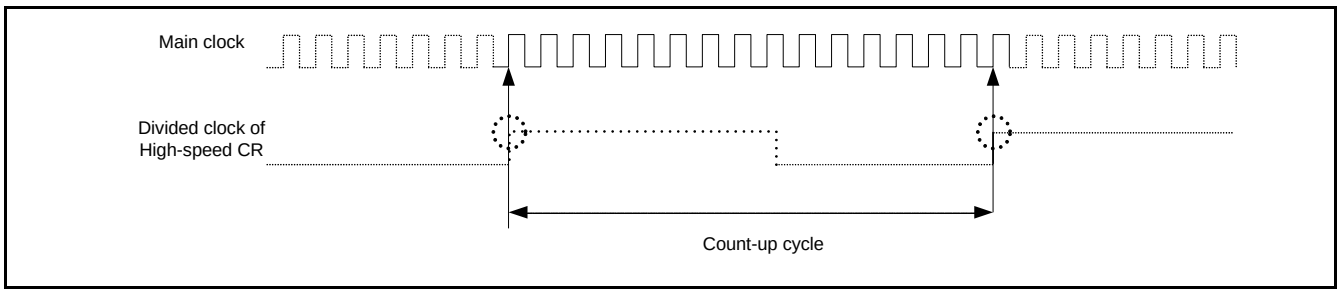
1. In stop mode, the main clock and high-speed CR clock stop. Meanwhile, the clock monitoring function also stops.

2. Upon the release of stop mode, oscillation of main clock and high-speed CR clock restart, waiting for oscillation stabilization. Meanwhile, the clock monitoring function keeps stopping.
3. When the oscillation stabilization wait time ends, the clock monitoring restarts.

5.2 Anomalous Frequency Detection

Figure 5-3 provides an example of anomalous frequency detection function operation.

Figure 5-3 Example of Anomalous Frequency Detection Function Operation



1. This function detects rising edges of the divided clock of high-speed CR.
2. After detecting edges, it counts up clocks using the main clock.
3. It keeps counting up until it detects the next rising edge of the divided clock of high-speed CR.
4. Let "α" be the count value with the main clock.
   Also let A denote the lower window value, and B the upper window value. Compare the count value α with those window values and if expression
   $A \leq \alpha \leq B$
   holds true, then the frequency is considered to be normal.
   If the count value α is out of the range, i.e., either
   $\alpha < A$, or $B < \alpha$
   is true, then the frequency is considered to be anomalous, and an interrupt occurs.
   If the interrupt flag has not been cleared after the interrupt and an anomalous frequency is detected again, then the function issues a reset depending on the setting.

5.3 Example of Anomalous Frequency Detection Function Window Setting

The anomalous frequency detection counts up between edges of the divided clock of high-speed CR. The measurement interval is also affected by the accuracy of CR. When you configure the window register value, therefore, the CR accuracy must be considered for the value.
For frequency accuracy of the CR oscillator, check the relevant "Data Sheet".

Calculation Method

The count value range of anomalous frequency detection must be added the CR accuracy, then, the window register value is set. The count range expression must be used as follows.

$$\text{Count value} = \left[\frac{1}{\text{Frequency of divided clock of CR} \times \left(1 \pm \frac{\text{CR accuracy}}{100} \right)} \right] \times \text{Frequency of main clock}$$

The count value by main clock of frequency L [Hz] can be calculated using the divide-by-Y CR oscillator clock of ±Z% accuracy with frequency K [Hz].

Count value A (positive CR frequency accuracy) = 1/[(K/Y) × (1 + Z/100)] × L
Count value B (negative CR frequency accuracy) = 1/[(K/Y) × (1 - Z/100)] × L

Those expressions lead the count value within the range A to B added internal CR accuracy.
Set the value smaller than count value A for the lower limit of the window, and larger than count value B for the upper limit.

The window setting is determined by the value allowed for frequency fluctuation of main oscillation defined by the user.

Example Calculation

The count value by main clock of frequency 4 MHz is calculated using the divide-by-1024 CR oscillator clock of ±5% accuracy with frequency 4 MHz.

Count value A (positive CR frequency accuracy)

$$\text{Count value A} = \left[\frac{1}{\frac{4\times10^6}{1024} \times \left(1 + \frac{5}{100} \right)} \right] \times 4\times10^6 \approx 975$$

Count value B (negative CR frequency accuracy)

$$\text{Count value B} = \left[\frac{1}{\frac{4\times10^6}{1024} \times \left(1 - \frac{5}{100} \right)} \right] \times 4\times10^6 \approx 1078$$

Those expressions yield the count value within the range 975 to 1078 including the high-speed CR error. If the window setting value is 5%, window setting value is as follows.

Window lower limit = 975 × 0.95(-5%) = 926.25 ≈ 3.43 MHz
Window upper limit = 1078 × 1.05(+5%) = 1131.9 ≈ 4.64 MHz

Thus, you can recognize that a main clock frequency out of the 3.4 MHz to 4.6 MHz range is anomalous. Table 5-1 provides an example of the window settings.

Table 5-1 Example of Window Settings

Divided Clock of High-Speed CR	Main Clock	High-Speed CR Error	Count Value Including High-Speed CR Error	Lower Limit of Window Set Value	Upper Limit of Window Set Value
Divide-by-1024 clocks of CR:4 MHz	4 MHz	±5%	975 (≈ 3.61 MHz) - 1078 (≈ 4.42 MHz)	926 (≈ 3.43 MHz)	1131 (≈ 4.64 MHz)

6. Registers

This section explains the register list of the clock supervisor functions.

Register List

Table 6-1 shows the register list.

Table 6-1 Register List

Abbreviation	Register Name	Reference
CSV_CTL	CSV control register	6.1
CSV_STR	CSV status register	6.2
FCSWH_CTL	Frequency detection window setting register (Upper)	6.3
FCSWL_CTL	Frequency detection window setting register (Lower)	6.4
FCSWD_CTL	Frequency detection counter register	6.5

6.1 CSV Control Register (CSV_CTL)

The CSV_CTL register configures the control of CSV function.

Register Configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved		FCD		Reserved		FCSRE	FCSDE
Attribute	-		R/W		-		R/W	R/W
Initial value	0		111		00		0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved						SCSVE	MCSVE
Attribute	-						R/W	R/W
Initial value	000000						1	1

Register Functions

[bit15] Reserved: Reserved bit

0 is read from this bit.

Set this bit to 0 when writing.

[bit14:12] FCD: FCS count cycle setting bits

bit14:12	Description
When 000 is written	Setting is prohibited
When 001 is written	
When 010 is written	
When 011 is written	
When 100 is written	
When 101 is written	1/256 frequency of high-speed CR oscillation
When 110 is written	1/512 frequency of high-speed CR oscillation
When 111 is written	1/1024 frequency of high-speed CR oscillation [Initial value]
When read	The register value is read.

[bit11:10] Reserved: Reserved bits

0b00 is read from these bits.

Set these bits to 0b00 when writing.

[bit9] FCSRE: FCS reset output enable bit

Bit	Description
When 0 is written	The FCS reset is disabled [Initial value]
When 1 is written	The FCS reset is enabled
When read	The register value is read.

[bit8] FCSDE: FCS function enable bit

Bit	Description
When 0 is written	The FCS function is disabled [Initial value]
When 1 is written	The FCS function is enabled.
When read	The register value is read.

[bit7:2] Reserved: Reserved bits

0b000000 is read from these bits.
Set these bits to 0b000000 when writing.

[bit1] SCSVE: Sub CSV function enable bit

Bit	Description
When 0 is written	The sub CSV function is disabled
When 1 is written	The sub CSV function is enabled. [Initial value]
When read	The register value is read.

[bit0] MCSVE: Main CSV function enable bit

Bit	Description
When 0 is written	The main CSV function is disabled
When 1 is written	The main CSV function is enabled. [Initial value]
When read	The register value is read.

Note:

- *This register is not initialized by software reset.*
- *To enable sub clock supervisor function, set sub clock oscillation enable setting of system clock mode control register (SCM_CTL.SOSCE) to "1" and wait until the sub clock oscillation stabilization bit of system clock mode control register (SCM_STR.SORDY) becomes "1" by stabilized.*
- *The sub clock supervisor function does not operate only by enabling sub clock oscillation of VBAT RTC (WTOSCCNT.SOSCEX=0).*
- *The following setting combination is prohibited:*
  32 kHz oscillation clock control linkage bit of VBAT RTC sub clock oscillation control register (WTOSCCNT.SOSCNTL) is "0".
  32 kHz oscillation enable bit of VBAT RTC sub clock oscillation control register (WTOSCCNT.SOSCEX) is "1".
  Sub clock oscillation enable setting bit of system clock mode control register (SCM_CTL.SOSCE) is "1".
  Sub CSV function enable bit of CSV control register (CSV_CTL.SCSVE) = "1".
- *For details on VBAT RTC, see "VBAT Domain".*

6.2 CSV Status Register (CSV_STR)

The CSV_STR register indicates the status of CSV function.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						SCMF	MCMF
Attribute	-						R	R
Initial value	000000						0	0

Register Functions

[bit7:2] Reserved: Reserved bits

0b000000 is read from these bits.

Set these bits to 0b000000 when writing.

[bit1] SCMF: Sub clock failure detection flag

Bit	Description
When written	No effect
When 0 is read	No sub clock failure has been detected. [Initial value]
When 1 is read	A sub clock failure has been detected.

[bit0] MCMF: Main clock failure detection flag

Bit	Description
When written	No effect
When 0 is read	No main clock failure has been detected. [Initial value]
When 1 is read	A main clock failure has been detected.

Note:

- *This register is cleared when being read.*

6.3 Frequency Detection Window Setting Register (Upper) (FCSWH_CTL)

The FCSWH_CTL register configures the frequency detection window setting register (Upper).

Register Configuration

bit	15	0
Field	FWH	
Attribute	R/W	
Initial value	0xFFFF	

Register Functions

[bit15:0] FWH: Frequency detection window setting bits (Upper)

bit15:0	Description
When written	Any value can be written to these bits.
When read	The register value is read.

Notes:

- *Set a value larger than the value set in FCSWL_CTL (Frequency detection window setting register (Lower)).*
- *This register is not initialized by software reset.*

6.4 Frequency Detection Window Setting Register (Lower) (FCSWL_CTL)

The FCSWL_CTL register configures the frequency detection window setting register (Lower).

Register Configuration

bit	15	0
Field	FWL	
Attribute	R/W	
Initial value	0x0000	

Register Functions

[bit15:0] FWL: Frequency detection window setting bits (Lower)

bit15:0	Description
When written	Any value can be written to these bits.
When read	The register value is read.

Notes:

- *Set a value smaller than the value set in FCSWH_CTL (Frequency detection window setting register (Upper)).*
- *This register is not initialized by software reset.*

6.5 Frequency Detection Counter Register (FCSWD_CTL)

The FCSWD_CTL register indicates the counter value of frequency detection using the main clock.

Register Configuration

bit	15															0
Field	FWD															
Attribute	R															
Initial value	0x0000															

Register Functions

[bit15:0] FWD: Frequency detection count data

bit15:0	Description
When written	No effect on operation
When read	The count value is read.

Notes:

- *This register retains the count value when detecting an error.*
- *This register is not initialized by software reset.*

7. Usage Precautions

This section explains the precautions for using the clock supervisor functions.

- For details on enabling and clearing the frequency detection interrupt sources, see Chapter "Clock".

- For details on clock failure detection and anomalous frequency detection reset sources, see Chapter "Resets".

- Operation after the occurrence of a reset

  After the occurrence of a reset triggered by clock failure detection, clock mode returns to high-speed CR.

  Do not select the faulty clock again.

- The high-speed CR clock for use of the frequency detection

  The frequency failure detection is affected by the frequency accuracy of high-speed CR itself. When you configure frequency window, therefore, the accuracy of high-speed CR must be considered for the window value. Do not trim the high-speed CR clock after the anomalous frequency detection has been enabled.

- The order of the anomalous frequency detection settings before using

  Before enabling FCS (FCSDE=1), specify the count cycle (FCD), reset enable (FCSRE), and frequency window (FWH/FWL) settings.

  If you want to change any of FCD/FCSRE/FWH/FWL after FCS has been enabled, stop the FCS function before changing the setting. Do not change the setting while FCS is enabled.

- The enable settings of the anomalous frequency detection before using

  Depending on the setting of the FCSRE bit in the CSV control register (CSV_CTL), operation during anomalous frequency detection varies. Table 7-1 shows the setting list.

Table 7-1 List of the FCS Function and FCSRE Bit Settings

	FCSRE=0	FCSRE=1
FCSDE=0	Stops FCS function	Stops FCS function
FCSDE=1	Enables FCS function Generates an interrupt upon error detection	Enables FCS function An interrupt occurs upon the first error detection A reset occurs upon the second error detection

- Interrupt settings for the frequency detection and main timer mode

  The internal bus clock stops while the clock mode is in main timer mode. In this mode, an interrupt does not occur even if an error is detected while FCSRE is set to 0.

  In main timer mode, therefore, do not set FCSRE bit to "0". If FCSRE bit is set to 1, a reset occurs upon the second error detection.

- The settings for CSV OFF and external reset.

  When CSV function is set to OFF, the CSV reset is not generated if the clock failure occurs. The external reset (INITX) is not also accepted if the clock failure occurs. So, it is recommended not to turn OFF the CSV function, if you do not have special reason.

CHAPTER 4: Resets

This chapter explains the function and operation of the resets.

1. Overview
2. Configuration
3. Explanation of Operations
4. Register

CODE: 9AFRESET-FM0-E03.0

1. Overview

This family has the following reset factors and issues a reset to initialize a device upon accepting a reset factor.

- Power-on reset
- INITX pin input
- External power supply/low-voltage detection reset
- Software watchdog reset
- Hardware watchdog reset
- Clock failure detection reset
- Anomalous frequency detection reset
- Software reset
- Deep standby transition reset

VBAT domain cannot be initialized with reset factors in this section.

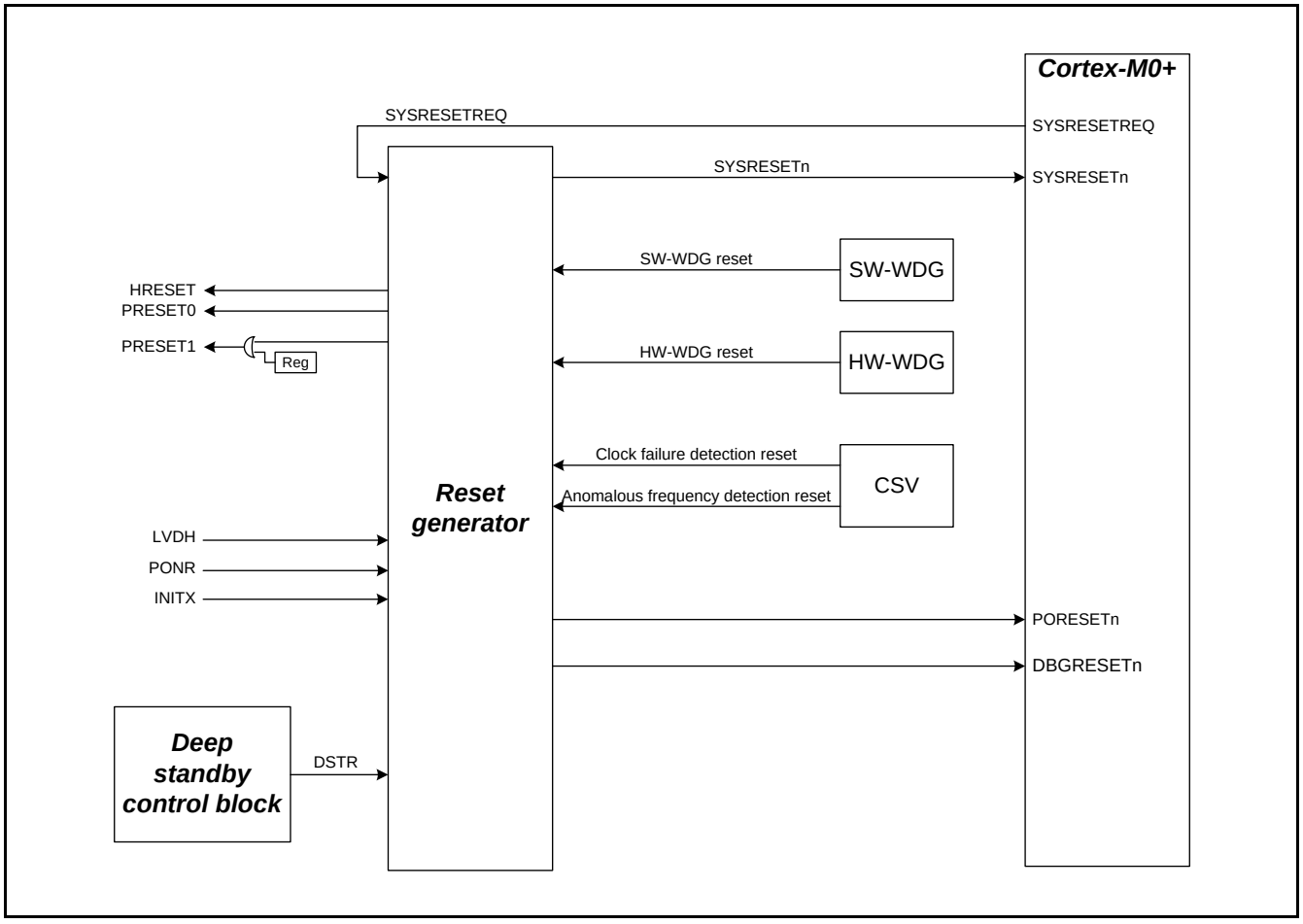
For the reset of VBAT domain, see Chapter "VBAT Domain".

2. Configuration

This section explains the configuration of the reset circuit.

Block Diagram of Resets

Figure 2-1 Block Diagram of Resets



PONR:	Power-on reset
INITX:	INITX pin input reset
LVDH:	Low-voltage detection reset
HRESET:	AHB bus reset (a bus reset issued by all reset factors)
PRESET0, 1:	APB0, APB1 bus resets (bus resets issued by all reset factors)
SW-WDG reset:	Software watchdog reset
HW-WDG reset:	Hardware watchdog reset
CSV reset:	Clock failure detection reset
FCS reset:	Anomalous frequency detection reset
PORESETn:	Power-on reset that is input to Cortex-M0+

SYSRESETn:	System reset that is input to Cortex-M0+
SYSRESETREQ:	"SYSRESETREQ bit" signal of Cortex-M0+ internal reset control register
DBGRESETn:	SW-DP reset
DSTR:	Deep standby transition reset

3. Explanation of Operations

This section explains the operations of the resets of this family.

3.1 Reset Factors

3.2 Resetting Inside Device

3.3 Reset Sequence

3.4 Operations after Resets are Cleared

3.1 Reset Factors

This section explains reset factors.

Power-On Reset (PONR)

A reset that is generated at power-up

Generated by	This signal is generated by detecting a rising edge of the power supply.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware.
Flag	bit0 (PONR) of reset factor register (RST_STR) = 1

INITX Pin Input Reset (INITX)

A reset that is externally input from a device

Generated by	This signal is generated by inputting a low level to INITX pin.
Cleared by	This signal is cleared by inputting a high level to INITX pin.
Initialization target	Initializes all register settings and hardware except the debug circuit, deep standby control block, and RTC (some registers). Note: The following registers are not initialized. − Reset factor register (RST_STR) − bit15 to bit8 of low-voltage detection voltage setting register − Deep standby return factor register 1 and 2 (WRFSR, WIFSR) − Deep Standby RAM Retention Register (DSRAMR) − Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit1 (INITX) of reset factor register (RST_STR) = 1

* The content of the on-chip SRAM is retained if a reset is asynchronously input from the INITX pin.

Low-Voltage Detection Reset, External Voltage Monitoring (LVDH)

A reset that is input from a low-voltage detection circuit when a decrease in the external voltage is detected

Generated by	This signal is generated when an external voltage is lowered than a specified level.
Cleared by	This signal is cleared when an external voltage is more than a specified level.
Initialization target	Initializes all register settings and hardware.
Flag	bit0 (PONR) of reset factor register (RST_STR) = 1

Software Watchdog Reset (SWDGR)

A reset that is input from the software watchdog timer.

Generated by	This signal is generated when the software watchdog timer underflows.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit and hardware watchdog timer (including control registers) and deep standby control block. Note: The following registers are not initialized. − Reset factor register (RST_STR) − bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) − Deep standby return factor register 1 and 2 (WRFSR, WIFSR) − Deep Standby RAM Retention Register (DSRAMR) − Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit4 (SWDT) of reset factor register (RST_STR)= 1

Hardware Watchdog Reset (HWDGR)

A reset that is input from the hardware watchdog timer.

Generated by	This signal is generated when the hardware watchdog timer underflows.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit, deep standby control block, and RTC (some registers). Note: The following registers are not initialized. − Reset factor register (RST_STR) − bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) − Deep standby return factor register 1 and 2 (WRFSR, WIFSR) − Deep Standby RAM Retention Register (DSRAMR) − Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit5 (HWDT) of reset factor register (RST_STR) = 1

Clock Failure Detection Reset (CSVR)

A reset that is input when the main or sub crystal oscillator being monitored fails.

Generated by	This signal is generated when a clock failure is detected in the main or sub crystal oscillator.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit and clock failure detection circuit (some registers) , deep standby control block, and RTC (some registers). Note: The following registers are not initialized. − Reset factor register (RST_STR) − bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) − Deep standby return factor register 1 and 2 (WRFSR, WIFSR) − Deep Standby RAM Retention Register (DSRAMR) − Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit6 (CSVR) of reset factor register (RST_STR) = 1 bit1 (SCMF) or bit0 (MCMF) of CSV status register (CSV_STR) = 1 Note: For details on the CSV_STR, see Chapter "Clock supervisor".

Anomalous Frequency Detection Reset (FCSR)

A reset that is input when an anomalous frequency is detected in the main crystal oscillator.

Generated by	This signal is generated when the frequency of the main crystal oscillator is outside of any given setting.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit, anomalous frequency detection (some registers), deep standby control block, and RTC (some registers). Note: The following registers are not initialized. − Reset factor register (RST_STR) − bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) − Deep standby return factor register 1 and 2 (WRFSR, WIFSR) − Deep Standby RAM Retention Register (DSRAMR) − Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit7 (FCSR) of reset factor register (RST_STR) = 1

Software Reset (SRST)

A reset that is generated when an access to the reset control register occurs.

Generated by	This signal is generated by a write to the Cortex-M0+ internal reset control register (SYSRESETREQ bit).
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the following: Functions and registers that are not initialized by a software reset − Debug circuit − Deep standby control block − Some registers of RTC − Some registers related to clock control (Peripheral clock stop register can be initialized.) − Part of registers that control software and hardware watchdog timers − Part of registers in the clock failure detection circuit − Part of registers that detect an anomalous frequency − Part of registers for CR trimming − Reset factor register (RST_STR) − bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) − RTC mode control register (PMD_CTL) − Deep standby return factor register 1 and 2 (WRFSR, WIFSR) − Deep Standby RAM Retention Register (DSRAMR) − Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit8 (SRST) of reset factor register (RST_STR) = 1

Deep standby transition reset (DSTR)

This reset occurs when transiting to deep standby mode.

Generated by	This signal is generated by transiting to deep standby mode
Cleared by	This signal is cleared by returning from deep standby mode
Initialization target	Initializes all register settings and hardware except the following: Functions and registers that are not initialized by a deep standby transition reset. − Deep standby control block − Some registers of RTC − HDMI-CEC/ Remote Control Reception − Some registers of GPIO (for detail, see chapter of “I/O port”) − Low-voltage detection circuit register − RTC mode control register (PMD_CTL) − Deep standby return factor register 1 and 2 (WRFSR, WIFSR) − Deep standby return permit register (WIER) − WKUP pin input level register (WILVR) − Deep Standby RAM Retention Register (DSRAMR) − Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	The bit of either deep standby return factor register 1 or 2 (WRFSR, WIFSR) is "1". Note: The bit that becomes "1" differs by return factors.

Notes:

- *For Cortex-M0+ internal reset control register (SYSRESETREQ) that controls the software reset, see "Chapter B3, System Address Map", in "Armv6-M Architecture Reference Manual".*
- *The reset factor register that can determine the occurrence of each reset factor is initialized only by power-on reset.*

3.2 Resetting Inside Device

This section explains the internal reset signals of this device.

Resets that are internally connected to the device are divided into resets that are input to the Cortex-M0+ core and resets that are input to peripheral circuits.

3.2.1 Resets to Cortex-M0+

3.2.2 Resets to Peripheral Circuit

3.2.1 Resets to Cortex-M0+

The device has three reset inputs to the Cortex-M0+ are PORESETn, SYSRESETn, and DBGRESETn. The following provides reset factors for these three reset inputs.

Power-on Reset PORESETn

Reset factors	– Power-on reset (PONR) – Low-voltage detection reset (LVDH) – Deep standby transition reset (DSTR)

System Reset SYSRESETn

Reset factors	– Power-on reset (PONR) – Low-voltage detection reset (LVDH) – INITX pin input (INITX) – Software watchdog reset (SWDGR) – Hardware watchdog reset (HWDGR) – Clock Failure Detection reset (CSVR) – Anomalous frequency detection reset (FCSR) – Software reset (SRST) – Deep standby transition reset (DSTR)

SW-DP Reset DBGRESETn

Reset factors	– Power-on reset (PONR) – Low-voltage detection reset (LVDH) – Deep standby transition reset (DSTR)

3.2.2 Resets to Peripheral Circuit

The bus resets (HRESET, PRESET0 and PRESET1) that are input to the peripheral circuit are basically generated by all reset factors. Resetting of PRESET1 can be controlled by register settings.

The following provides reset factors for the bus resets.

Resets to Peripheral Circuit

■ HRESET and PRESET0

Reset factors	– Power-on reset (PONR) – Low-voltage detection reset (LVDH) – INITX pin input (INITX) – Software watchdog reset (SWDGR) – Hardware watchdog reset (HWDGR) – Clock Failure Detection reset (CSVR) – Anomalous frequency detection reset (FCSR) – Software reset (SRST) – Deep standby transition reset (DSTR)

■ PRESET1

Reset factors	– Power-on reset (PONR) – Low-voltage detection reset (LVDH) – INITX pin input (INITX) – Software watchdog reset (SWDGR) – Hardware watchdog reset (HWDGR) – Clock Failure Detection reset (CSVR) – Anomalous frequency detection reset (FCSR) – Software reset (SRST) – APB bus resets (APBC1_PSR) – Deep standby transition reset (DSTR)

Notes:

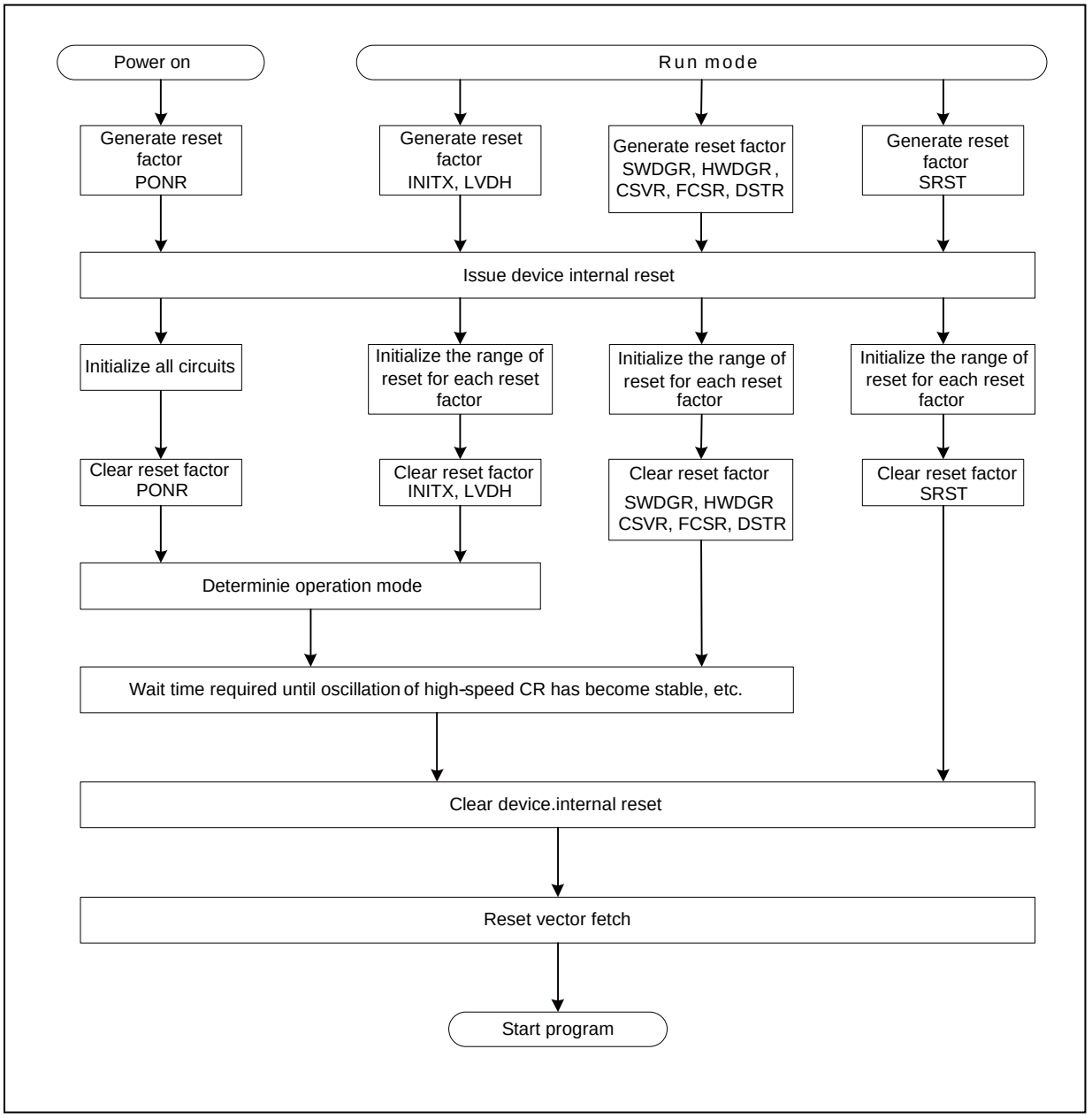
- *The peripheral circuit is essentially initialized with all reset factors. Depending on the specifications of the peripheral circuit, there are registers that are initialized only with specific causes. For the initialization conditions for registers, see the initialization conditions for the registers described in the relevant chapter.*
- *For details on APB bus resets (APBC1_PSR), see Chapter "Clock".*

3.3 Reset Sequence

This family initiates the program and hardware operations starting with the initial state when a reset factor is cleared.
This family of operations starting with the reset and ending with the initiation of the operations is called a reset sequence.
The following explains a reset sequence.

State Transition Diagram for Resets

The following diagram shows a transition of reset states. The detailed operations are given in the following sections "3.4 Operations after Resets are Cleared".



1. Capturing reset factors

   Reset factors are captured and retained until a reset is issued to the device.

2. Issuing resets

   When a reset is ready to be issued, a device internal reset is issued.

3. Clearing resets

   When a reset factor is cleared, a device internal reset is extended for the amount of time required to clear the reset (for example, a wait time required until oscillation of a high-speed CR has become stable). When the extended period of time has expired, the reset is cleared.

4. Determining operation mode

   The operation mode defined by MD0 and MD1 is determined as PONR, LVDH or INITX is cleared and notified to each piece of the hardware. Any other reset factors do not cause the operation mode to change.

5. Reset vector fetch

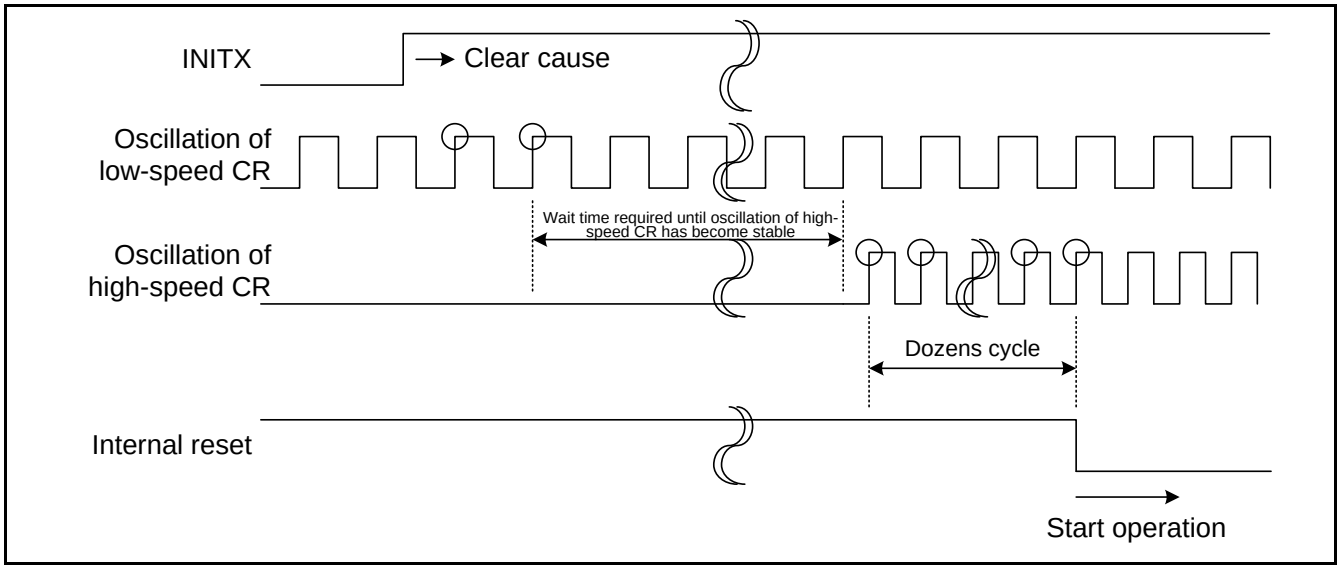
   After a device internal reset is cleared, the CPU starts fetching a reset vector. The CPU fetches the obtained reset vector into the program counter and starts programmed operations.

3.4 Operations after Resets are Cleared

PONR, LVDH, INITX, HWDGR, SWDGR, CSVR, FCSR, DSTR

Figure 3-1 provides an example of the operation waveform after a cause of INITX pin input reset has been cleared.

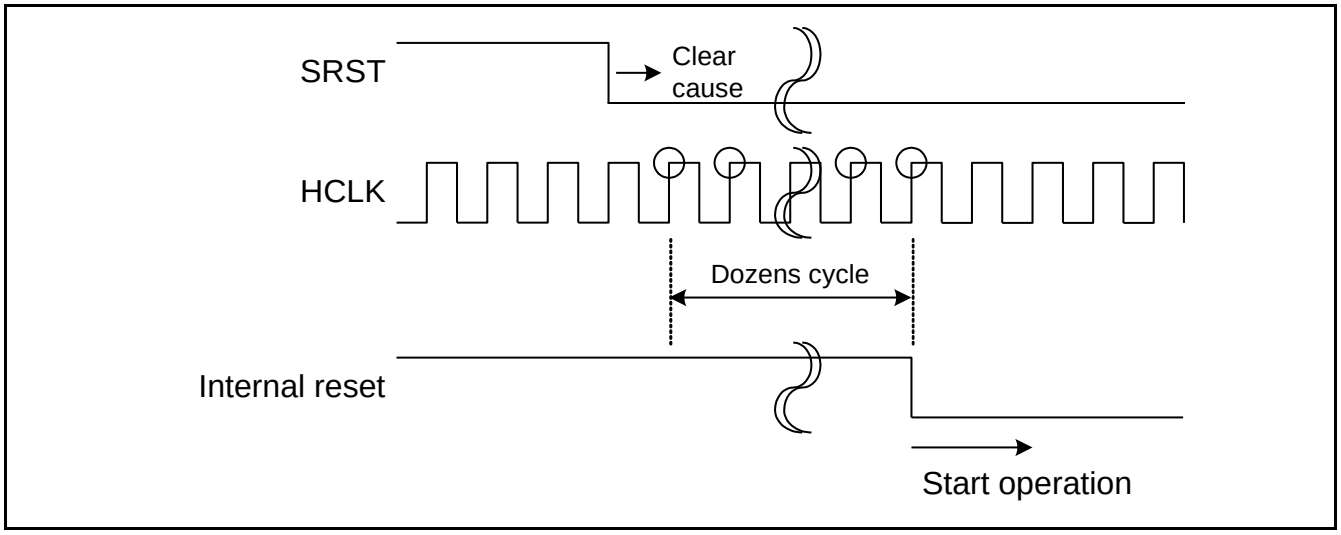
Figure 3-1 Operation Waveform Example after INITX Pin Input Reset has been Cleared



SRST

Figure 3-2 shows an example of the operation waveform after a software reset has been cleared.

Figure 3-2 Operation Waveform Example after a Software Reset has been Cleared



4. Register

This section describes the register for resets.

List of Register for Resets

Table 4-1 List of Register for Resets

Abbreviation	Register Name	Reference
RST_STR	Reset Factor Register	4.1

4.1 Reset Factor Register (RST_STR)

The Reset Factor Register (RST_STR) shows the factors of resets that have just occurred. All bits of the RST_STR are initialized by a power-on reset, a low-voltage detection reset or a deep standby reset. It is not initialized by any other reset. All bits of the RST_STR are cleared to “0” by reading this register. After initializing, until it has been read, this register stores all reset factors that have been generated.

Register Configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							SRST
Attribute				-				R
Initial value				-				0

Bit	7	6	5	4	3	2	1	0
Field	FCSR	CSVR	HWDT	SWDT	Reserved		INITX	PONR
Attribute	R	R	R	R	-		R	R
Initial value	0	0	0	0	-		0	1

Note: The initial value shows the value is initialized by a power-on reset, a low-voltage detection reset or a deep standby reset.

Register Functions

[bit15:9] Reserved: Reserved bits

The read value is undefined. These bits have no effect when written.

[bit8] SRST: Software reset flag

Indicates a reset that is generated by writing "1" to Cortex-M0+ internal reset control register (SYSRESETREQ bit).

When a software reset is generated, SRST is set (SRST = 1). This bit is cleared to 0 by reading this register.

Bit	Description
0	A software reset has not been issued.
1	A software reset has been issued.

[bit7] FCSR: Flag for anomalous frequency detection reset

Indicates a reset when an anomalous frequency is detected in the main oscillation.

When the frequency of the main oscillation is outside of a given setting, a reset is issued and FCSR is set (FCSR = 1). This bit is cleared to 0 by reading this register.

Bit	Description
0	An anomalous frequency detection reset has not been issued.
1	An anomalous frequency detection reset has been issued.

[bit6] CSVR: Clock failure detection reset flag

Indicates a reset when a failure is detected in the main or sub oscillation.

If a stop is detected, a reset is issued and CSVR is set (CSVR = 1). This bit is cleared to 0 by reading this register.

Bit	Description
0	A clock failure detection reset has not been issued.
1	A clock failure detection reset has been issued.

Note: Please refer to another chapter "Clock supervisor" for the method of judging whether the main oscillation or the sub oscillation broke down.

[bit5] HWDT: Hardware watchdog reset flag

Indicates a reset from the hardware watchdog timer.

If the timer underflows, a reset is issued and HWDT is set (HWDT = 1). This bit is cleared to 0 by reading this register.

Bit	Description
0	A hardware watchdog reset has not been issued.
1	A hardware watchdog reset has been issued.

[bit4] SWDT: Software watchdog reset flag

Indicates a reset from the software watchdog timer.

If the timer overflows, a reset is issued and SWDT is set (SWDT = 1). This bit is cleared to 0 by reading this register.

Bit	Description
0	A software watchdog reset has not been issued.
1	A software watchdog reset has been issued.

[bit3:2] Reserved: Reserved bits

The read value is undefined. These bits have no effect when written.

[bit1] INITX: INITX pin input reset flag

Indicates a reset that is externally input.

If a reset is externally input, INITX is set (INITX = 1). This bit is cleared to 0 by reading this register.

Bit	Description
0	An INITX pin input reset has not been issued.
1	An INITX pin input reset has been issued.

[bit0] PONR: Power-on reset flag

Indicates a reset that is power-on reset, low voltage detection reset or deep standby reset.

If a rising edge of power supply, a low-voltage or deep standby transition is detected, a reset is issued and the PONR is initialized to 1 (PONR = 1). This bit is cleared to 0 by reading this register.

Bit	Description
0	A power-on reset, low-voltage detection reset and deep standby reset has not been issued.
1	A power-on reset, low-voltage detection reset, or deep standby reset has been issued.

Notes:

- *Determine whether it is a return from deep standby mode or not by deep standby return factor registers 1 and 2 (WRFSR and WIFSR). For details, see "8.5 Deep standby return factor register 1(WRFSR)" and "8.6 Deep standby return factor register 2(WIFSR)" in Chapter "Low Power Consumption Mode".*

CHAPTER 5-1: Low-voltage Detection Overview

This chapter explains the functions and operations of the Low-voltage Detection Circuit.

1. Overview

CODE: 9AFLVD-FM0T0-E03.0

1. Overview

The Low-voltage Detection Circuit monitors the power supply voltage, and generates reset and interrupt signals when the power supply voltage falls below the detection voltage.

Overview of Low-Voltage Detection Circuit

- Operations of Low-voltage Reset Circuit
  - This circuit monitors the power supply voltage (VCC) and generates a reset signal when the power supply voltage falls below the specified value.
  - This circuit allows selection of whether to enable or stop operations. The initial state is operating.
  - This circuit allows specification of the detection voltage (depends on the type of LVD). However, when the low-voltage reset is generated, the set value is initialized. So, the release voltage becomes the initial value. If the power supply voltage is higher than the release voltage, the reset is released.
  - This circuit monitors the power supply voltage even in standby modes and deep standby modes.
  - This circuit generates a reset signal when the reduction of the power supply voltage is detected in standby modes and deep standby modes.

- Operations of Low-voltage Interrupt Circuit
  - This circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified value.
  - This circuit allows selection of whether to enable or stop operations. The initial state is set to disable.
  - This circuit allows specification of the detection voltage.
  - This circuit can monitor the power supply voltage even in standby modes and deep standby modes.
  - This circuit returns from standby mode and deep standby modes when the reduction of the power supply voltage is detected in the standby mode and deep standby modes.

Notes:

- *If a low-voltage detection interrupt is enabled or the detection voltage is specified for a low-voltage detection interrupt, this circuit starts VCC voltage monitoring after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.*
  For the stabilization wait time of the Low-voltage Detection Circuit, refer to "Data Sheet" of the product used.
- *This circuit does not conduct monitoring the power supply voltage if PCLK1 is stopped by TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, Deep standby STOP mode, or APB1 Prescaler Register (APBC1_PSR) while waiting for the stabilization of the Low-voltage Detection Circuit. After the status flag is read and the stabilization wait time has lapsed, change to the desired mode.*
- *The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected to prevent a writing error. To release write protection mode, write 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR).*

■ The TYPEs of LVD

There are 3 types of LVD mounted in different product. For the details of each setting, refer to the each chapter as following Table 1-1.

Table 1-1 Correspondence Table for LVD Chapter

Product TYPE	Reference
TYPE1-M0+	Chapter "Low Voltage Detection (TYPE1)"
TYPE2-M0+	Chapter "Low Voltage Detection (TYPE2)"
TYPE3-M0+	Chapter "Low Voltage Detection (TYPE3)"

CHAPTER 5-2: Low-voltage Detection (TYPE1)



This chapter explains the functions and operations of the Low-voltage Detection Circuit.

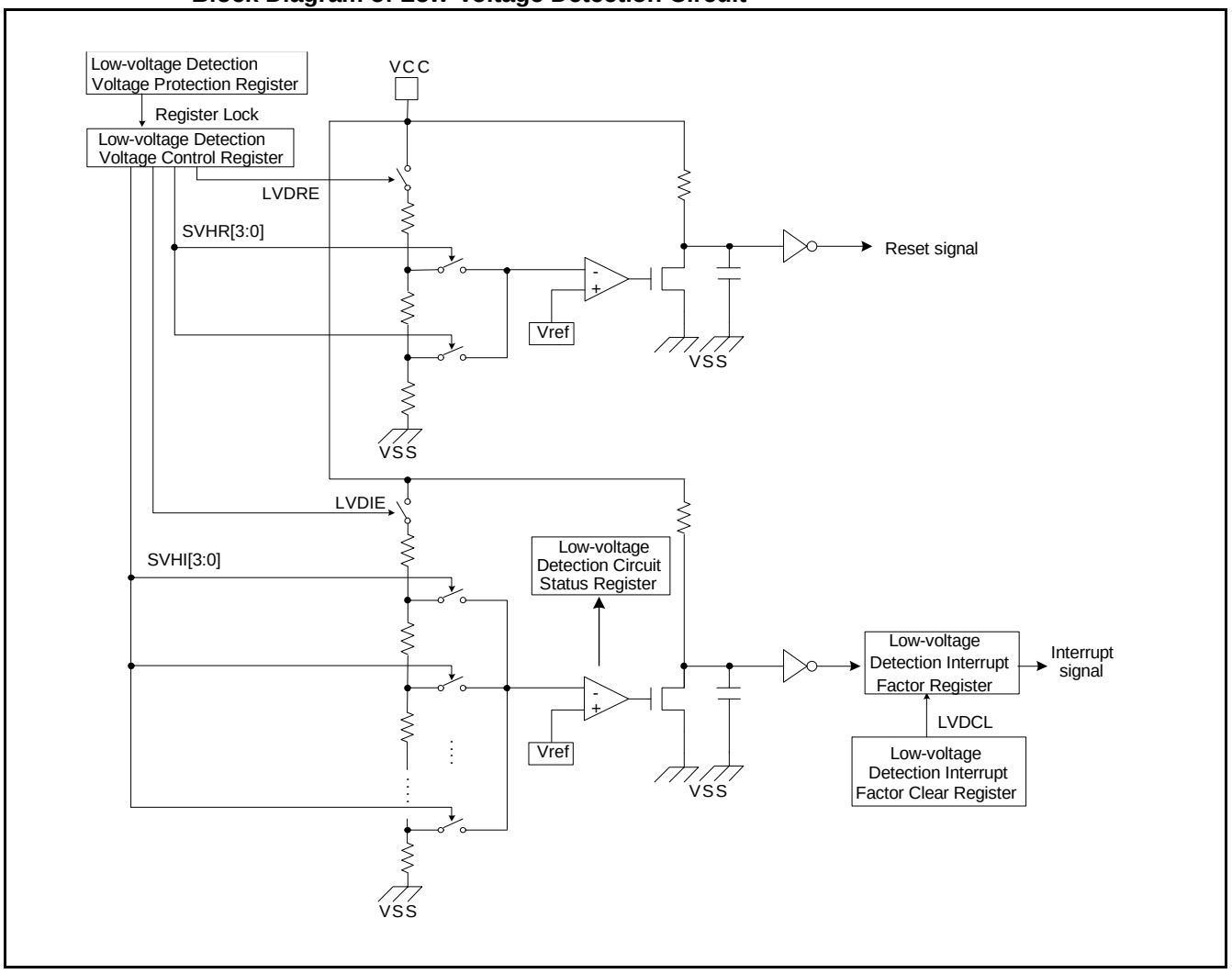
1. Configuration
2. Operations
3. Setup Procedure Examples
4. Registers
5. Usage Precautions

CODE: 9AFLVD-FM0T1-E01.0

1. Configuration

This section shows the block diagram of the Low-voltage Detection Circuit.

Block Diagram of Low-Voltage Detection Circuit



- Low-voltage Detection Voltage Control Register (LVD_CTL)
  This register controls whether to enable monitoring the power supply voltage for a low-voltage detection reset and low-voltage detection interrupt, and specifies the detection voltage for a low-voltage detection reset and low-voltage detection interrupt.
- Low-voltage Detection Voltage Protection Register (LVD_RLR)
  This register write-protects the Low-voltage Detection Voltage Control Register.
- Low-voltage Detection Interrupt Factor Register (LVD_STR)
  This register holds a low-voltage detection interrupt factor.
- Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR)
  This register clears a low-voltage detection interrupt factor.

- Low-voltage Detection Circuit Status Register (LVD_STR2)

This register checks the operation status of a low-voltage detection interrupt circuit.

Pins of Low-Voltage Detection Circuit

The following shows the pins used in the Low-voltage Detection Circuit.

- VCC pin

  The Low-voltage Detection Circuit monitors the power supply voltage of this pin.

- VSS pin

  This pin is a GND pin used as a basis to detect the power supply.

2. Operations

This section explains the operations of the Low-Voltage Detection Reset Circuit and the Low-voltage Detection Interrupt Circuit.

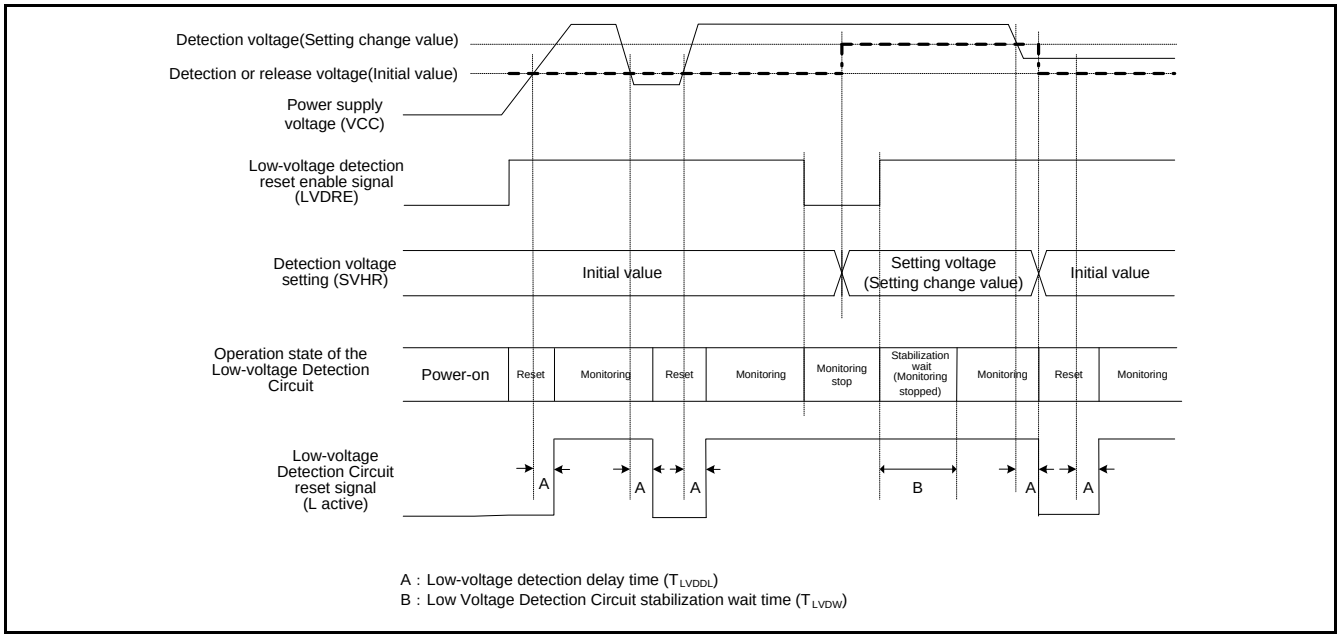
Operations of Low-Voltage Detection Reset Circuit

■ Explanation of Circuit Operation

The Low-Voltage Detection Reset Circuit enters a monitoring state after power-on. This circuit generates a reset signal when the specified power supply voltage (VCC) falls below the detection voltage. A reset is released when the power supply voltage exceeds the release voltage.

Reset operation is valid only when LVDRE bit of Low-voltage Detection Voltage Control Register (LVD_CTL) is "1". Detection voltage of the reset can be set by SVHR bit of Low-voltage Detection Voltage Control Register (LVD_CTL). However, SVHR bit is initialized by the low-voltage detection reset. So, the release voltage becomes the initial value. If the power supply voltage is higher than the release voltage, the reset is released. When reset enable and reset detection voltage are set, Low-voltage detection reset status flag (LVDRRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) becomes "1" and starts power supply voltage monitoring after the stabilization wait period for the low-voltage detection circuit lapses.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode).



A : Low-voltage detection delay time (T_{LVDDL})
B : Low Voltage Detection Circuit stabilization wait time (T_{LVDW})

Operations of Low-Voltage Detection Interrupt Circuit

■ Explanation of Circuit Operation

The Low-voltage Detection Interrupt Circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified value.

An interrupt request is enabled when "1" is set to the LVDIE bit of the Low-voltage Detection Voltage Control Register (LVD_CTL). The initial value is set to Not Enable ("0"). The interrupt detection voltage can be set by the SVHI bit of the Low-voltage Detection Voltage Control Register (LVD_CTL). When an interrupt request is enabled and the interrupt detection voltage is specified, Low-voltage detection interrupt status flag (LVDIRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1" and this circuit starts monitoring the power supply voltage after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode). It is also applicable when the CPU returns from those modes.

■ Low-voltage detection interrupt request

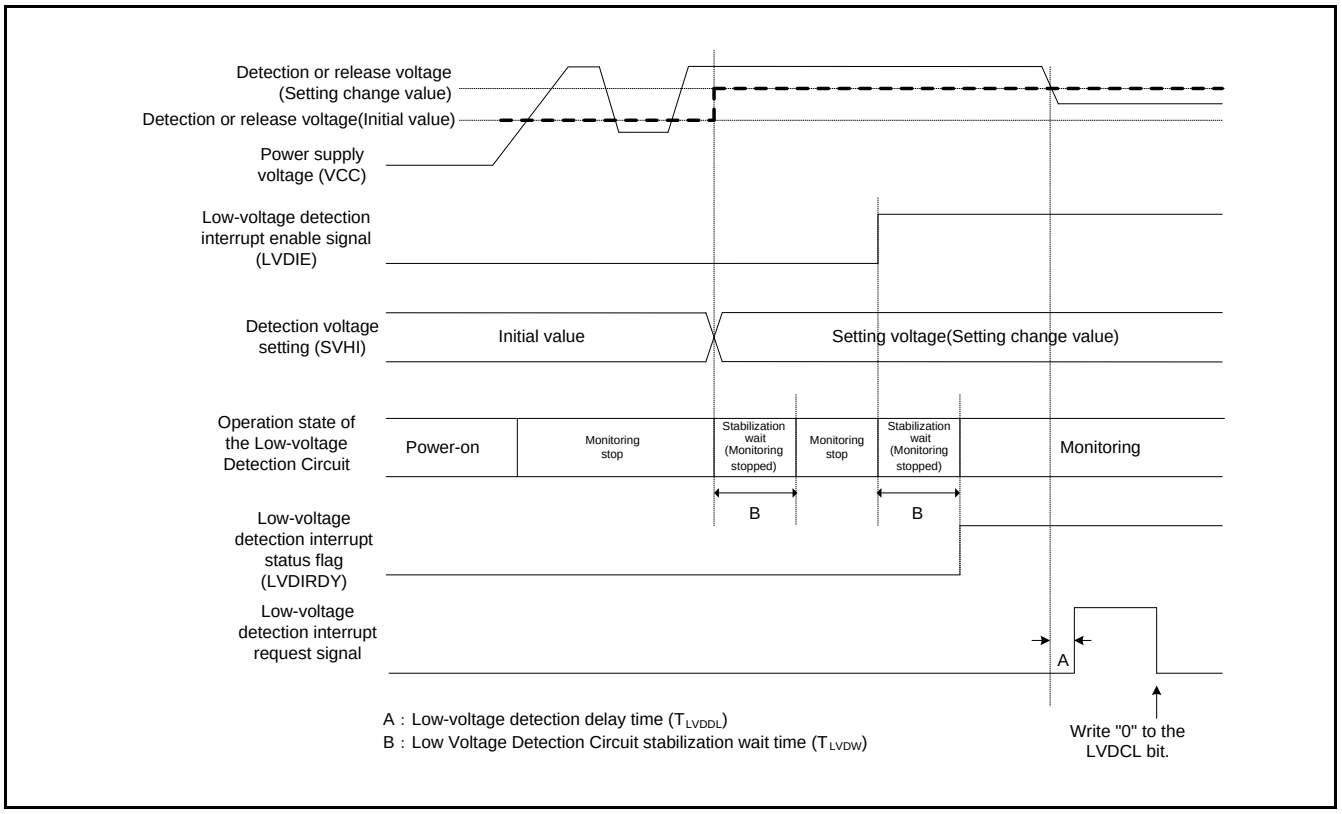
When the power supply voltage (VCC) falls below the specified voltage while a low-voltage detection interrupt is effective, "1" is set to the LVDIR bit of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to generate an interrupt request signal.

An interrupt request can be checked by reading the LVDIR bit.

■ Canceling a low-voltage detection interrupt request

To cancel a low-voltage detection interrupt request, write "0" to the LVDCL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR). This clears a low-voltage detection interrupt factor and cancels a low-voltage detection interrupt request.

Even when the power supply voltage is below the specified detection voltage, an interrupt request is canceled when "0" is written to the LVDCL bit.



Note:

- *This circuit does not conduct monitoring the power supply voltage if PCLK1 is stopped by TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, Deep standby STOP mode or APB1 Prescaler Register (APBC1_PSR) while waiting for the stabilization of the Low-voltage Detection Circuit. After checking that Low-voltage detection interrupt status flag (LVDIRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1", change to the desired mode.*

3. Setup Procedure Examples

This section explains the procedures to set up the Low-voltage Detection Circuit, giving examples.

Figure 3-1 Setting Procedure Example for Low-Voltage Detection Reset

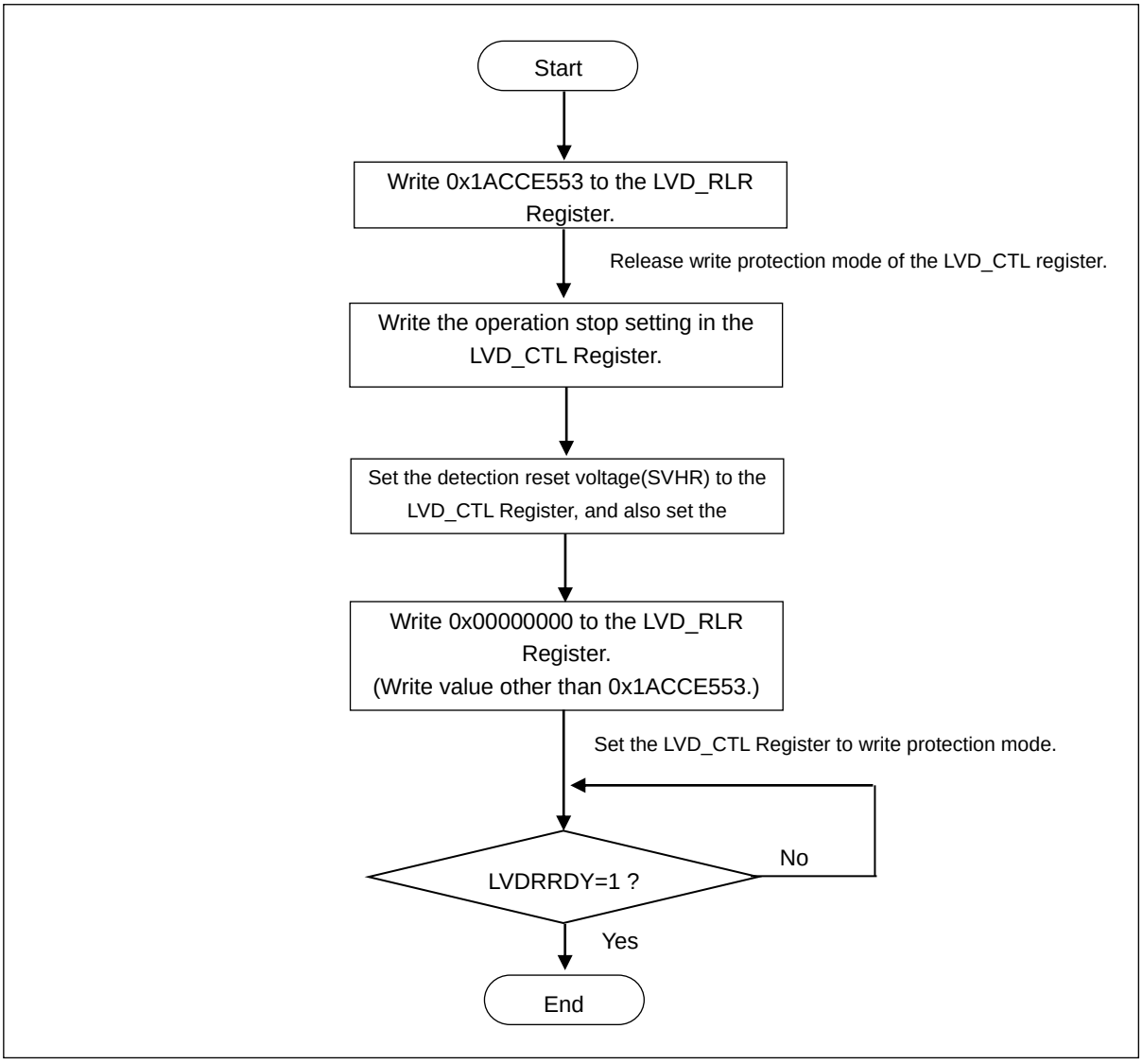
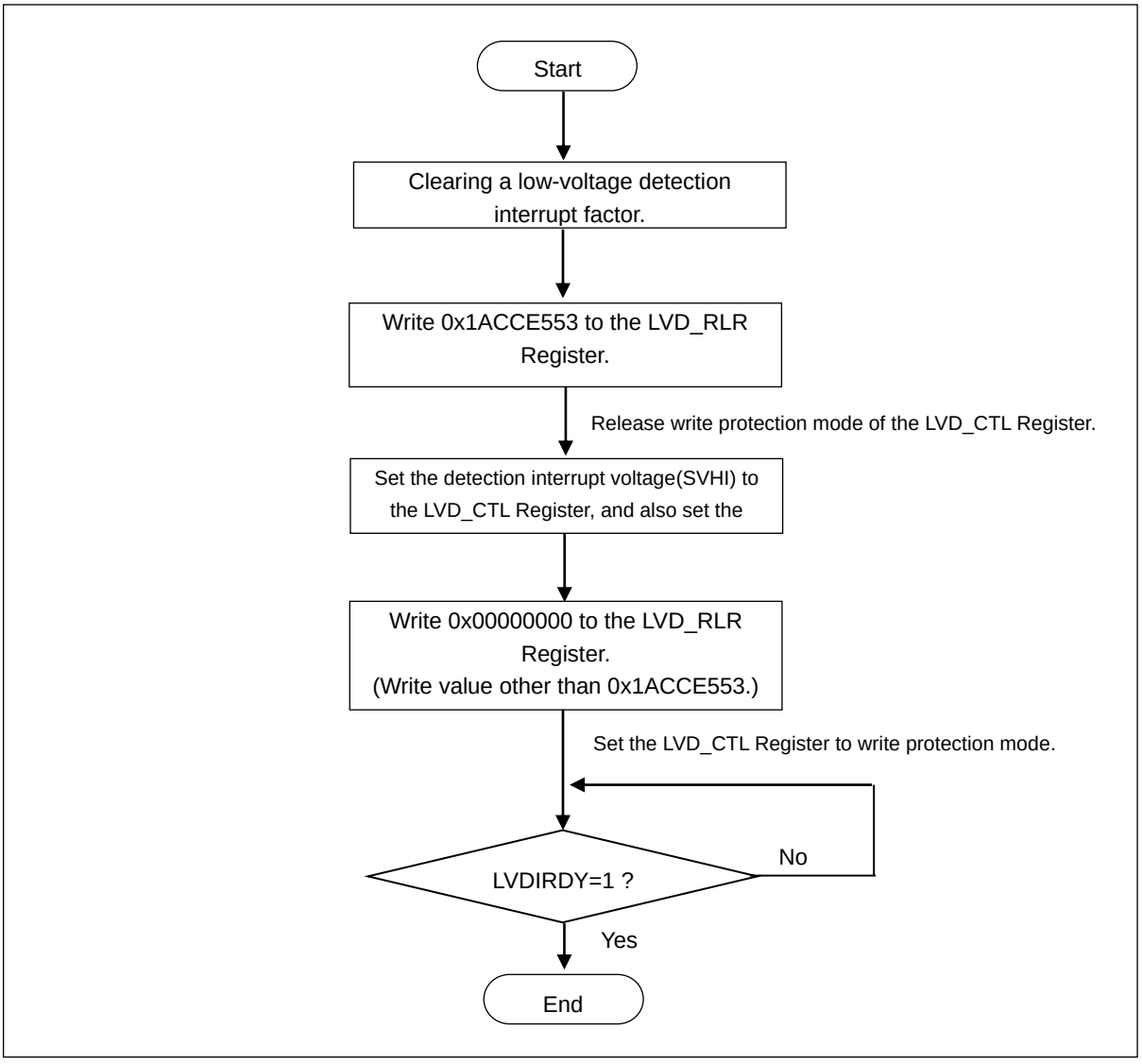


Figure 3-2 Setting Procedure Example for Low-Voltage Detection Interrupt



4. Registers

This section explains the configuration and functions of the registers used in the Low-voltage Detection Circuit.

List of Low-Voltage Detection Circuit Registers

Table 4-1 List of Low-Voltage Detection Circuit Registers

Abbreviation	Register Name	Reference
LVD_CTL	Low-voltage Detection Voltage Control Register	4.1
LVD_STR	Low-voltage Detection Interrupt Factor Register	4.2
LVD_CLR	Low-voltage Detection Interrupt Factor Clear Register	4.3
LVD_RLR	Low-voltage Detection Voltage Protection Register	4.4
LVD_STR2	Low-voltage Detection Circuit Status Register	4.5

4.1 Low-Voltage Detection Voltage Control Register (LVD_CTL)

The Low-voltage Detection Voltage Control Register (LVD_CTL) controls whether to enable monitoring the power supply voltage for low-voltage detection reset and a low-voltage detection interrupt and specifies the detection voltage for low-voltage detection reset and a low-voltage detection interrupt.

Register Configuration

bit	15	14	13	12	11	10	9	8
Field	LVDRE	SVHR					Reserved	
Attribute	R/W	R/W					-	
Initial value	1	00000					00	

bit	7	6	5	4	3	2	1	0
Field	LVDIE	SVHI					Reserved	
Attribute	R/W	R/W					-	
Initial value	0	00011					00	

Register Functions

[bit15] LVDRE: Low-voltage detection reset operation enable bit

This bit enables operation of power supply voltage monitoring of low-voltage detection reset. If it is not enabled, the low-voltage detection reset circuit stops operating.

Bit	Description
0	Generation of low-voltage detection reset is not enabled.
1	Generation of low-voltage detection reset is enabled. [initial value]

[bit14:10] SVHR: Low-voltage detection reset voltage setting bits

These bits set detection voltage of low-voltage detection reset.

bit14:10	Description
00000	Set the low-voltage detection reset voltage in the vicinity of 2.45 V. [Initial value]
00001	Set the low-voltage detection reset voltage in the vicinity of 2.60 V.
00010	Set the low-voltage detection reset voltage in the vicinity of 2.70 V.
00011	Set the low-voltage detection reset voltage in the vicinity of 2.80 V.
00100	Set the low-voltage detection reset voltage in the vicinity of 3.00 V.
00101	Set the low-voltage detection reset voltage in the vicinity of 3.20 V.
00110	Set the low-voltage detection reset voltage in the vicinity of 3.60 V.
00111	Set the low-voltage detection reset voltage in the vicinity of 3.70 V.
01000	Set the low-voltage detection reset voltage in the vicinity of 4.00 V.
01001	Set the low-voltage detection reset voltage in the vicinity of 4.10 V.
01010	Set the low-voltage detection reset voltage in the vicinity of 4.20 V.
Others	Setting is prohibited.

[bit9:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVDIE: Low-voltage detection interrupt enable bit

This bit is used to enable monitoring the power supply voltage of a low-voltage detection interrupt. When not enabling monitoring the power supply voltage, the Low-voltage Detection Interrupt Circuit is stopped.

Bit	Description
0	Does not enable the generation of a low-voltage detection interrupt. [Initial value]
1	Enables the generation of a low-voltage detection interrupt.

[bit6:2] SVHI: Low-voltage detection interrupt voltage setting bits

These bits are used to specify the detection voltage of a low-voltage detection interrupt.

bit6:2	Description
00011	Set the low-voltage detection interrupt voltage in the vicinity of 2.80 V. [Initial value]
00100	Set the low-voltage detection interrupt voltage in the vicinity of 3.00 V.
00101	Set the low-voltage detection interrupt voltage in the vicinity of 3.20 V.
00110	Set the low-voltage detection interrupt voltage in the vicinity of 3.60 V.
00111	Set the low-voltage detection interrupt voltage in the vicinity of 3.70 V.
01000	Set the low-voltage detection interrupt voltage in the vicinity of 4.00 V.
01001	Set the low-voltage detection interrupt voltage in the vicinity of 4.10 V.
01010	Set the low-voltage detection interrupt voltage in the vicinity of 4.20 V.
Others	Setting is prohibited.

[bit1:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Notes:

- *The low-voltage detection interrupt enable bit (LVDIE) must be enabled after "0" was written to the LVDCL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) to clear the low-voltage detection interrupt factor bit (LVDIR).*
- *When the low-voltage detection interrupt enable bit (LVDIE) is not enabled, the Low-voltage Detection Circuit for detecting a low-voltage interrupt is stopped. Therefore, the low-voltage detection interrupt factor bit (LVDIR) is not set.*
- *The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the Low-voltage Detection Voltage Control Register (LVD_CTL), set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by reset factor other than these resets.*

- *After the setting value of low-voltage detection reset voltage is changed, as the setting value is initialized by the low-voltage detection reset, the released voltage becomes the initial value. At that time, if the power supply voltage is higher than the released voltage, the reset is released. For the initial values of the detection voltage and the released voltage, see the data sheet of the product used.*
- *This register is not initialized by deep standby transition reset.*

4.2 Low-Voltage Detection Interrupt Factor Register (LVD_STR)

The Low-voltage Detection Interrupt Factor Register (LVD_STR) holds a low-voltage detection interrupt factor.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	LVDIR	Reserved						
Attribute	R	-						
Initial value	0	0000000						

Register Functions

[bit7] LVDIR: Low-voltage detection interrupt factor bit

Bit	Description
0	A low-voltage detection interrupt request is not detected. [Initial value]
1	A low-voltage detection interrupt request has been detected.

[bit6:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

4.3 Low-Voltage Detection Interrupt Factor Clear Register (LVD_CLR)

The Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) clears a low-voltage detection interrupt factor.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	LVDCL	Reserved						
Attribute	R/W	-						
Initial value	1	0000000						

Register Functions

[bit7] LVDCL: Low-voltage detection interrupt factor clear bit

Bit	Description
0	Clears the low-voltage detection interrupt factor bit (LVDIR) of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to "0".
1	Has no effect on the operation in write mode. [Initial value]

"1" is always set in read mode.

[bit6:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

4.4 Low-Voltage Detection Voltage Protection Register (LVD_RLR)

The Low-voltage Detection Voltage Protection Register (LVD_RLR) write-protects the Low-voltage Detection Voltage Control Register (LVD_CTL).

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	LVDLCK[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LVDLCK[15:0]															
Attribute	R/W															
Initial value	0x0001															

Register Functions

[bit31:0] LVDLCK[31:0]: Low-voltage Detection Voltage Control Register protection bits

- Setting 0x1ACCE553 enables writing the Low-voltage Detection Voltage Control Register (LVD_CTL) (releases write protection mode).
- Setting a value other than 0x1ACCE553 disables writing the Low-voltage Detection Voltage Control Register (LVD_CTL) (enables write protection mode).
- When the Low-voltage Detection Voltage Control Register (LVD_CTL) is not set in write protection mode, 0x00000000 is read.
- When the Low-voltage Detection Voltage Control Register (LVD_CTL) is set in write protection mode, 0x00000001 is read.

Notes:

- *The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state. To write the LVD_CTL Register, set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *To enable write protection mode of the LVD_CTL register, set a value other than 0x1ACCE553 to the LVD_RLR register.*
- *Once write protection mode is released for the LVD_CTL Register, it remains released until a value other than 0x1ACCE553 is written to the LVD_RLR Register.*
- *This register is not initialized by deep standby transition reset.*

4.5 Low-Voltage Detection Circuit Status Register (LVD_STR2)

The Low-voltage Detection Circuit Status Register (LVD_STR2) checks the operation status of a low-voltage detection interrupt.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	LVDIRDY	LVDRRDY	Reserved					
Attribute	R	R	-					
Initial value	0	1	000000					

Register Functions

[bit7] LVDIRDY: Low-voltage detection interrupt status flag

Bit	Description
0	Stabilization wait state or monitoring stop state [Initial value]
1	Monitoring state

This bit has no effect on the operation in write mode.

[bit6] LVDRRDY: Low-voltage detection reset status flag

Bit	Description
0	Stabilization wait state or monitoring stop state
1	Monitoring state [Initial value]

This bit has no effect on the operation in write mode.

[bit5:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

5. Usage Precautions

This section explains the precautions for using Low-voltage Detection Circuit.

■ Low-voltage detection interrupt factor bit at STOP mode transition

Even if the power supply voltage is not greater than the detection voltage after clearing the Low-voltage Detection Interrupt factor bit (LVD_STRLVDIR), the interrupt factor will not be enabled again unless the power supply voltage becomes greater than the released voltage once.
But, the voltage comparison is executed by the Low-voltage Detection Circuit without fail when the transition to the STOP mode is executed while the power supply voltage is not greater than the detection voltage after clearing LVDIR.
So, the Low-voltage detection interrupt factor is set again by the transition to STOP mode and the process could go to the interrupt routine.

For example, in case of the transition to STOP mode in the interrupt routine of the Low-voltage detection, after clearing the interrupt factor and finishing the interrupt routine, the interrupt factor bit is set soon again and the interrupt routine might be repeated.

To prevent the occurrence of repeated interrupts when the voltage is not greater than the detection voltage after the detection of the low-voltage interrupt, disable the Low-voltage detection interrupt enable bit (LVDIE) and get out of the routine.

■ Setup of Detection Voltage of Low-voltage Detection Reset

When the Low-voltage detection reset is generated after setting the detection voltage of Low-voltage detection reset voltage, the detection voltage setting value is initialized. When the power supply voltage is higher than the initial setup value as in the case where the power supply voltage lowers slowly, the reset is released. But, as the program is returned to the beginning by the reset, set the detection voltage again. As the power supply voltage is already lower than the detection voltage, the Low-voltage detection reset is set again.

That is to say, the loop of the detection voltage change, reset, initialization (returned to the beginning), the detection voltage change, and reset could be repeated according to the power supply voltage change and program description.

The following measures can be taken to prevent the loop of the Low-voltage detection reset when the power supply voltage is not greater than the detection voltage:

- For the setting value of the Low-voltage detection reset, only the initial value is used.
- Set the Low-voltage detection interrupt before the Low-voltage detection reset. Confirm whether an interrupt flag is set at the beginning of the program and change the setting value of the Low-voltage reset detection voltage, if required.

CHAPTER 5-3: Low-voltage Detection (TYPE2)



This chapter explains the functions and operations of the Low-voltage Detection Circuit.

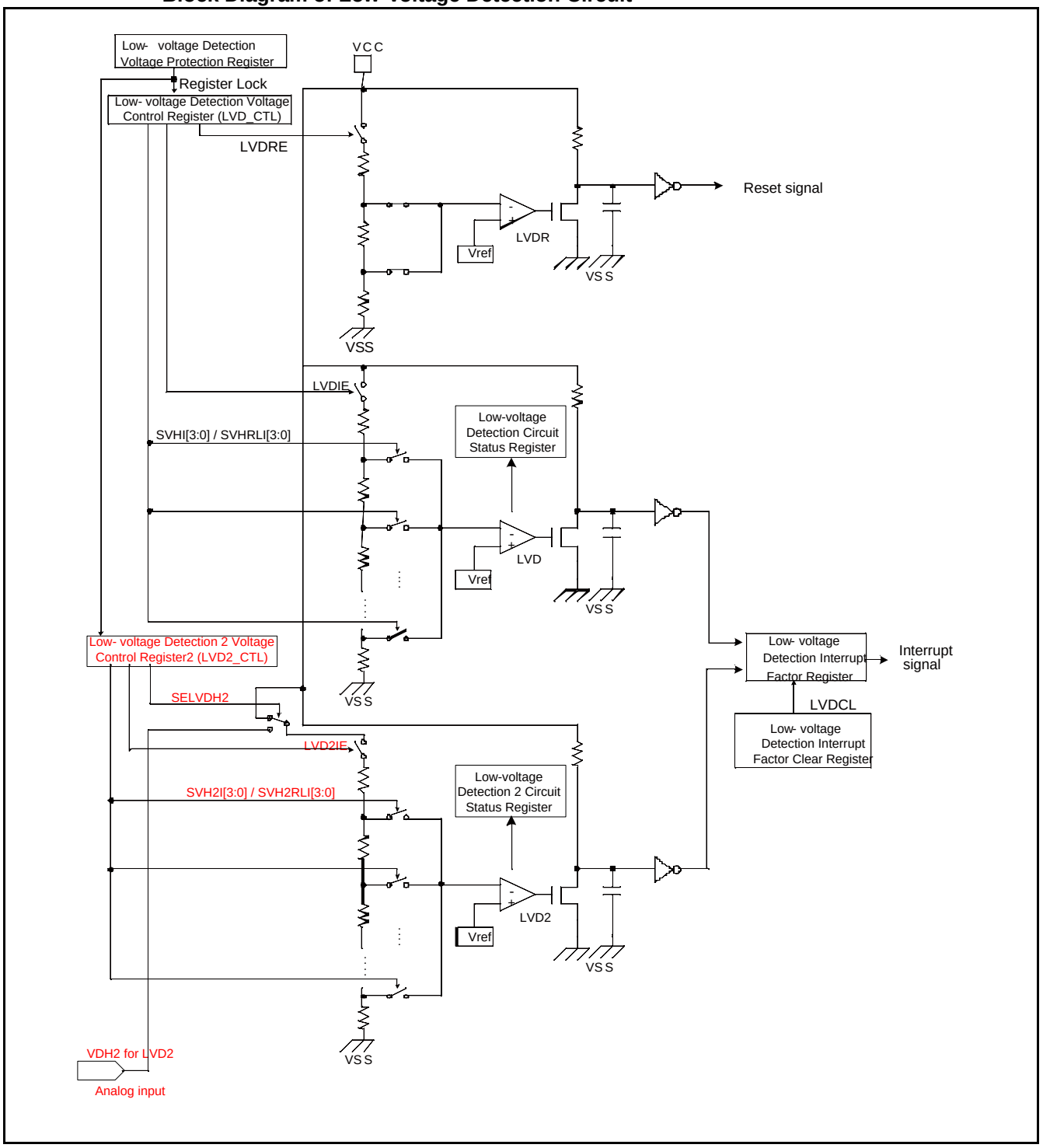
1. Configuration
2. Operations
3. Setup Procedure Examples
4. Registers
5. Usage Precautions

CODE: 9AFLVD-FM0T1-E01.0

1. Configuration

This section shows the block diagram of the Low-voltage Detection Circuit.

Block Diagram of Low-Voltage Detection Circuit



- Low-voltage Detection Voltage Control Register (LVD_CTL)
  This register controls whether to enable monitoring the power supply voltage for a low-voltage detection reset and low-voltage detection interrupt, and specifies the detection voltage for a low-voltage detection interrupt.
- Low-voltage Detection 2 Voltage Control Register (LVD2_CTL)
  This register controls whether to enable monitoring the power supply voltage for a low-voltage detection interrupt, and specifies the detection voltage for a low-voltage detection interrupt.
- Low-voltage Detection Voltage Protection Register (LVD_RLR)
  This register write-protects the Low-voltage Detection Voltage Control Register.
- Low-voltage Detection Interrupt Factor Register (LVD_STR)
  This register holds a low-voltage detection interrupt factor.
- Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR)
  This register clears a low-voltage detection interrupt factor.
- Low-voltage Detection Circuit Status Register (LVD_STR2)
  This register checks the operation status of a low-voltage detection interrupt circuit.
- Vref Calibration Control Register (CAL_CTL)
  This register controls Vref calibration.
- Vref Calibration Value Set Register (CAL_SET)
  This register holds Vref calibration value.
- Vref Calibration Security Key Register (CAL_KEY)
  This register holds a security key to unlock CAL_CTL and CAL_SET register.

Pins of Low-voltage Detection Circuit

The following shows the pins used in the Low-voltage Detection Circuit.

- VCC pin
  The Low-voltage Detection Circuit monitors the power supply voltage of this pin.
- VDH2 pin
  The Low-voltage Detection Circuit monitors the external voltage of this pin.
- VSS pin
  This pin is a GND pin used as a basis to detect the power supply.

2. Operations

This section explains the operations of the Low-Voltage Detection Reset Circuit and the Low-voltage Detection Interrupt Circuit.

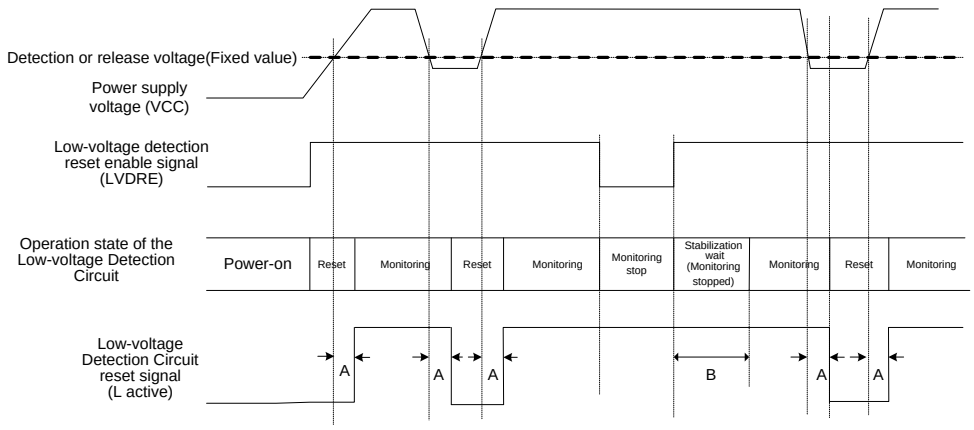
Operations of Low-Voltage Detection Reset Circuit

■ Explanation of Circuit Operation

The Low-Voltage Detection Reset Circuit enters a monitoring state after power-on. This circuit generates a reset signal when the specified power supply voltage (VCC) falls below the detection voltage. A reset is released when the power supply voltage exceeds the release voltage.

Reset operation is valid only when LVDRE bit of Low-voltage Detection Voltage Control Register (LVD_CTL) is "1". Detection voltage of the reset is fixed and cannot be changed. If the power supply voltage is higher than the release voltage, the reset is released. When reset enable is set, Low-voltage detection reset status flag (LVDRRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) becomes "1" and starts power supply voltage monitoring after the stabilization wait period for the low-voltage detection circuit lapses.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode).



A : Low-voltage detection delay time (T_{LVDDL})
B : Low Voltage Detection Circuit stabilization wait time (T_{LVDW})

Operations of Low-voltage Detection Interrupt Circuit

■ Explanation of Circuit Operation

The Low-voltage Detection Interrupt Circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified value.

An interrupt request is enabled when "1" is set to the LVDIE bit of the Low-voltage Detection Voltage Control Register (LVD_CTL). The initial value is set to Not Enable ("0"). The interrupt detection voltage can be set by the SVHI bit of the Low-voltage Detection Voltage Control Register (LVD_CTL), and the release voltage can be set by the SVHRLI bit of Low-voltage Detection release voltage Control Register (LVD_CTL2). When an interrupt request is enabled and the interrupt detection/release voltage is specified, Low-voltage detection interrupt status flag (LVDIRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1" and this circuit starts monitoring the power supply voltage after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode). It is also applicable when the CPU returns from those modes.

■ Low-voltage detection interrupt request

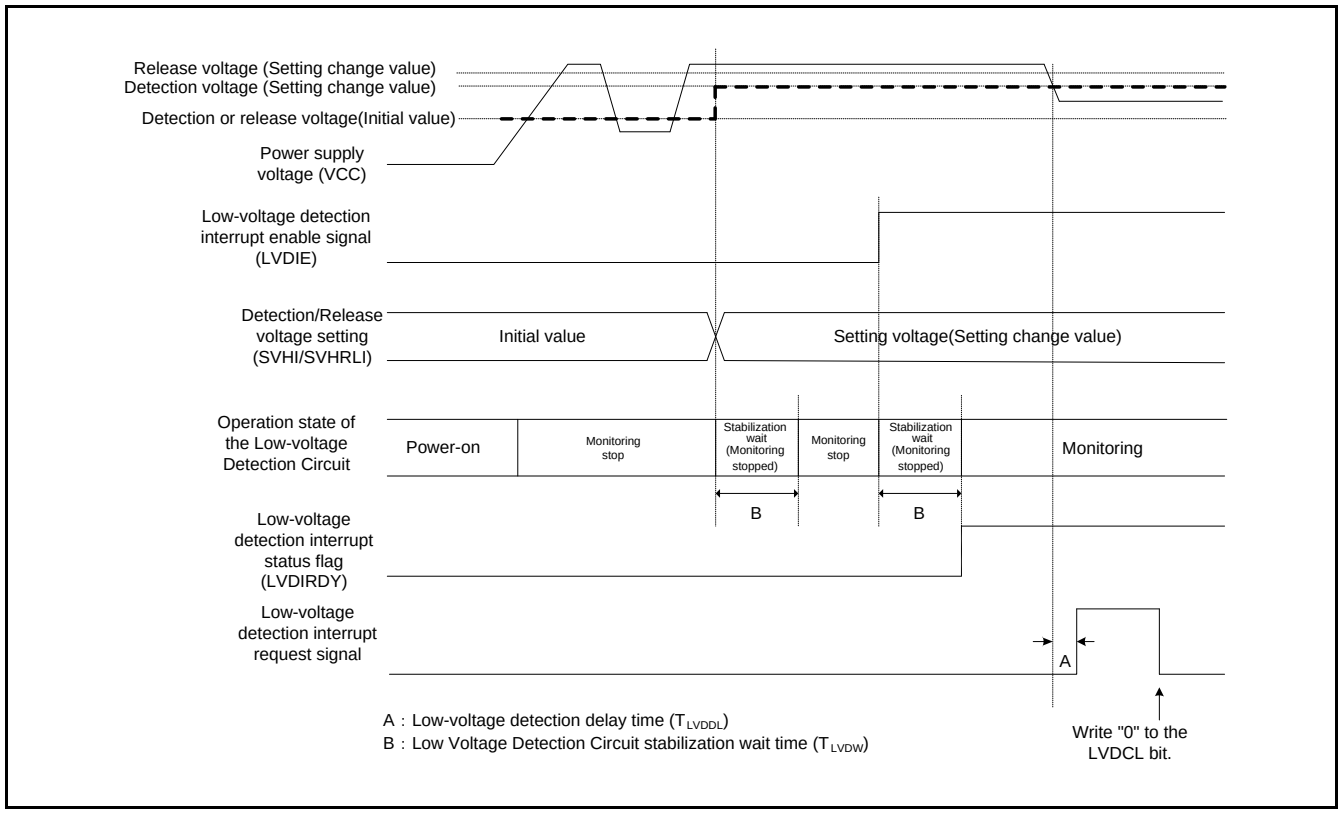
When the power supply voltage (VCC) falls below the specified voltage while a low-voltage detection interrupt is effective, "1" is set to the LVDIR bit of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to generate an interrupt request signal.

An interrupt request can be checked by reading the LVDIR bit.

■ Canceling a low-voltage detection interrupt request

To cancel a low-voltage detection interrupt request, write "0" to the LVDCL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR). This clears a low-voltage detection interrupt factor and cancels a low-voltage detection interrupt request.

Even when the power supply voltage is below the specified detection voltage, an interrupt request is canceled when "0" is written to the LVDCL bit.



Note:

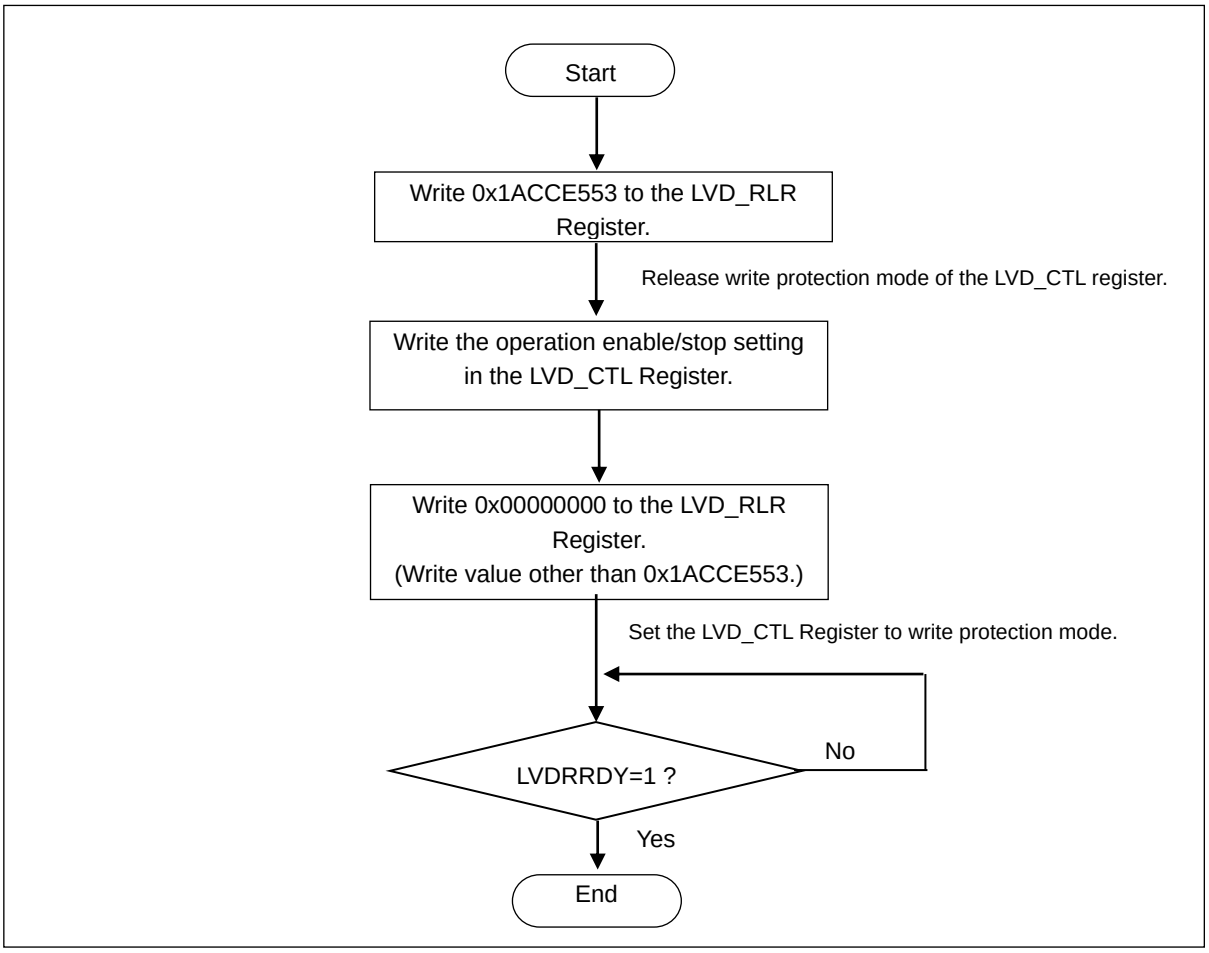
- *This circuit does not conduct monitoring the power supply voltage if PCLK1 is stopped by TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, Deep standby STOP mode or APB1 Prescaler Register (APBC1_PSR) while waiting for the stabilization of the Low-voltage Detection Circuit. After checking that Low-voltage detection interrupt status flag (LVDIRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1", change to the desired mode.*

3. Setup Procedure Examples

This section explains the procedures to set up the Low-voltage Detection Circuit, giving examples.

3.1 Example for Low-voltage detection reset

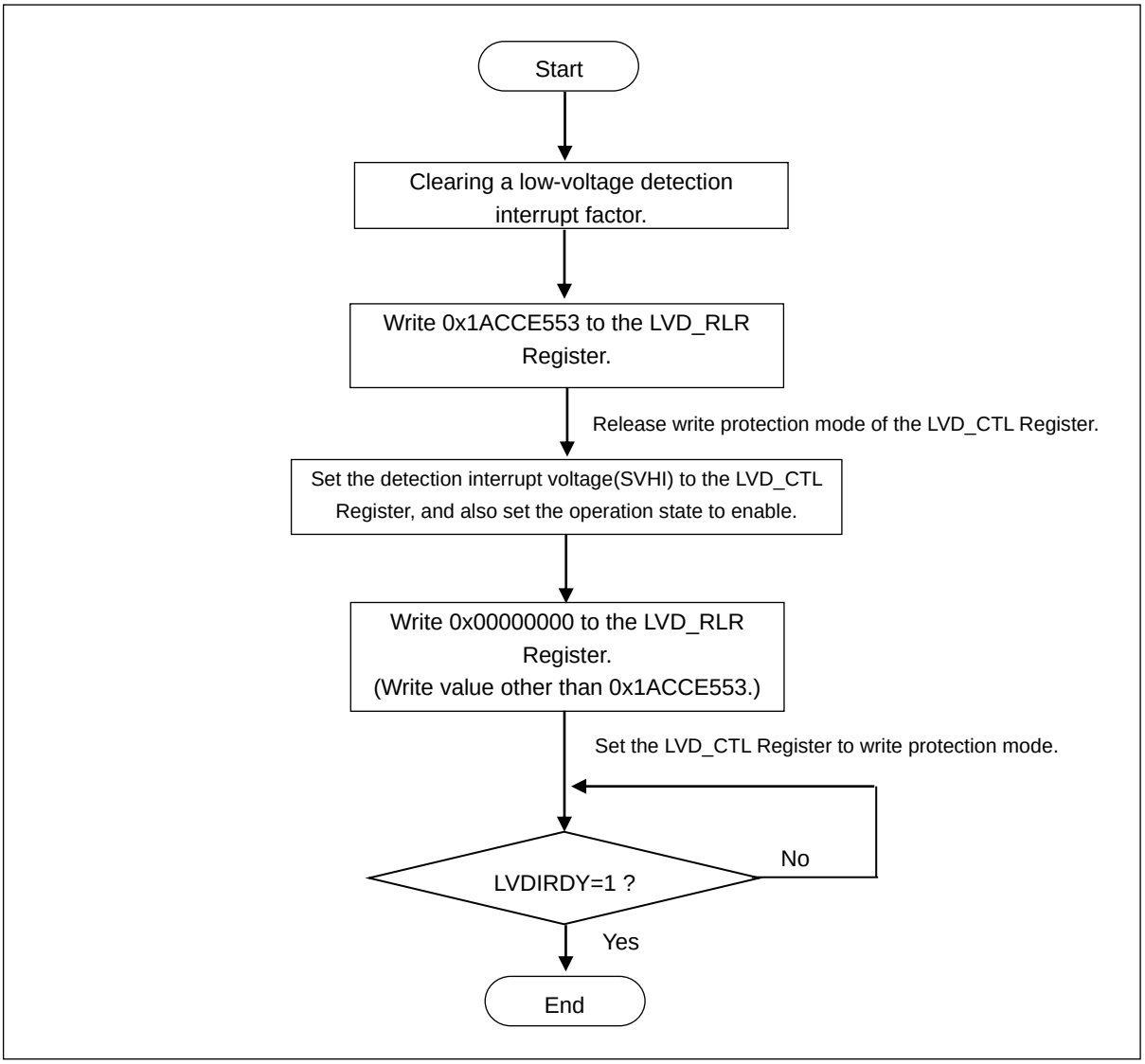
Figure 3-1 Setting procedure example for low-voltage detection reset



3.2 Example for Low-voltage detection interrupt (LVDRLIE = 0)

This section explains the procedures to set up the Low-voltage Detection Circuit without setting release voltage separately.

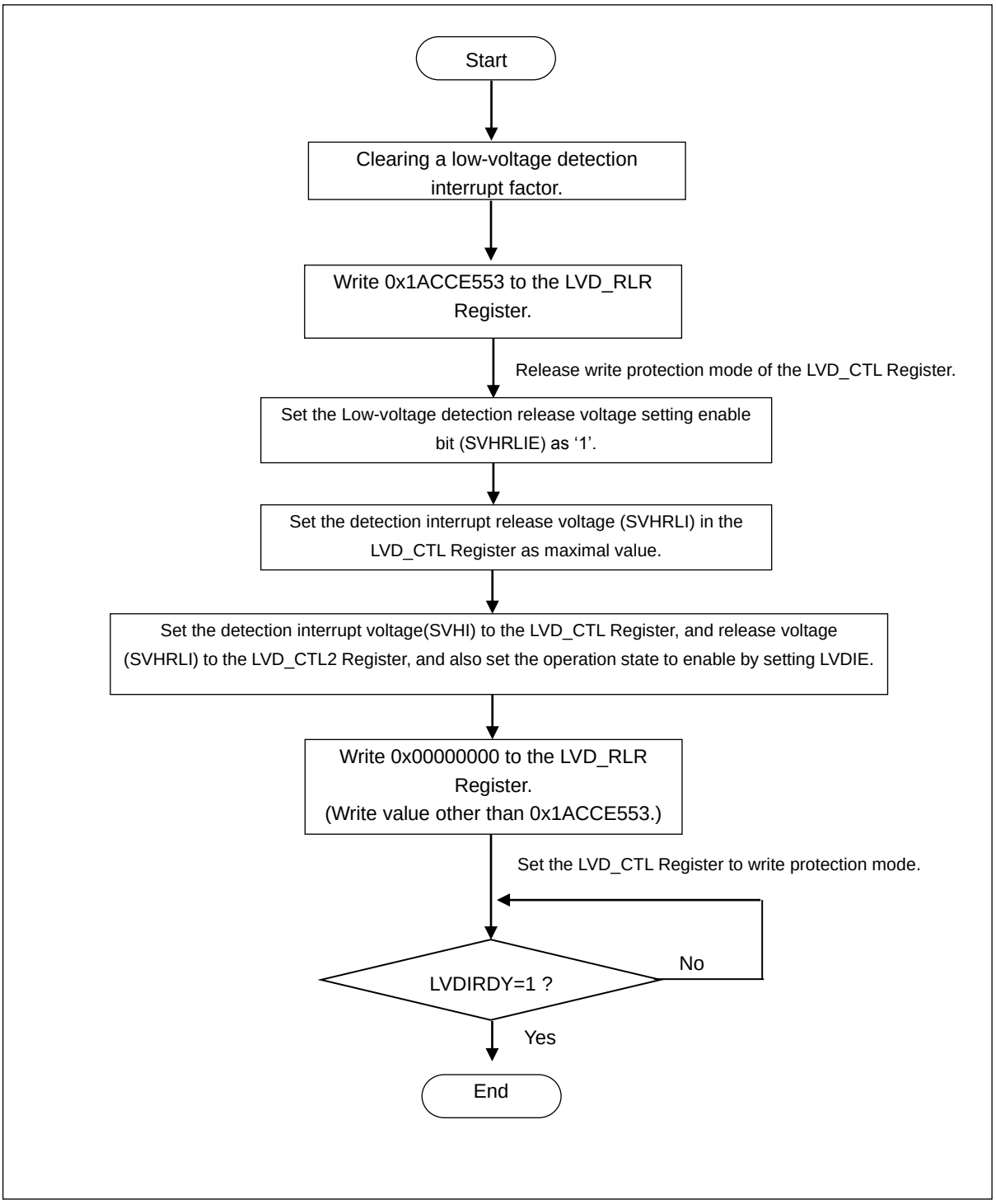
Figure 3-2 Setting procedure example for low-voltage detection interrupt (LVDRLIE = 0).



3.3 Example for Low-voltage detection interrupt (LVDRLIE = 1)

This section explains the procedures to set up the Low-voltage Detection Circuit by setting the detection voltage and release voltage separately.

Figure 3-3 Setting procedure example for low-voltage detection interrupt(LVDRLIE = 1)

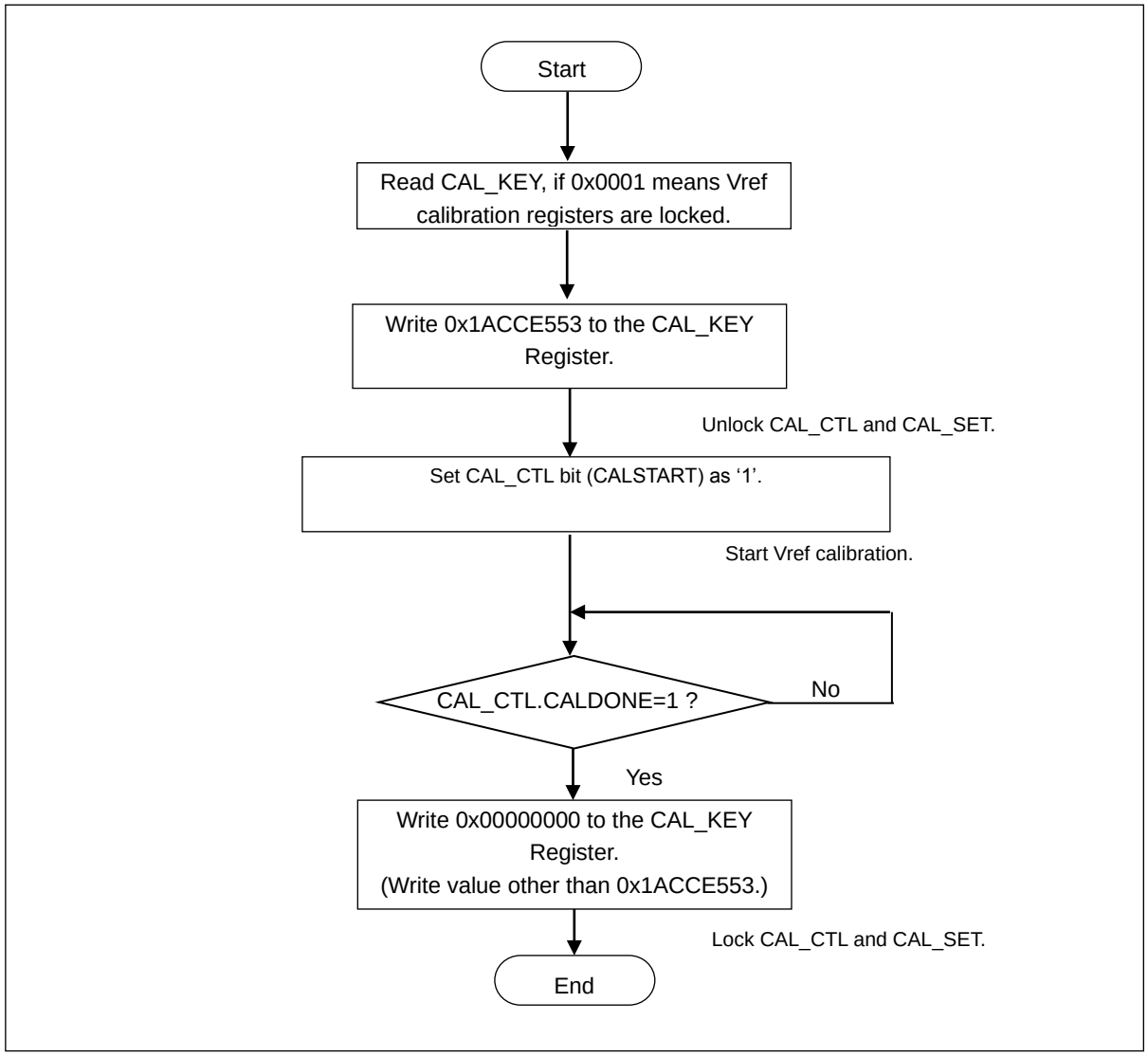


3.4 Example for Vref calibration

This section explains the procedures to start Vref calibration.

There’re 2 Vref in product, one is bipolar Vref, it’s accurate with higher current; the other is MOS Vref, it’s low power and can be calibrated with bipolar Vref. In this part, we’ll explain how to do MOS Vref calibration and how to select it.

Figure 3-4 Setting procedure example for Vref calibration



4. Registers

This section explains the configuration and functions of the registers used in the Low-voltage Detection Circuit.

List of low-voltage detection circuit registers

Table 4-1 List of low-voltage detection circuit registers

Abbreviation	Register name	Reference
LVD_CTL	Low-voltage Detection Voltage Control Register	4.1
LVD_STR	Low-voltage Detection Interrupt Factor Register	4.5
LVD_CLR	Low-voltage Detection Interrupt Factor Clear Register	4.6
LVD_RLR	Low-voltage Detection Voltage Protection Register	4.7
LVD_STR2	Low-voltage Detection Circuit Status Register	4.8
LVD_CTL2	Low-voltage Detection Voltage Control Register 2	4.2
LVD2_CTL	Low-voltage Detection (ch.2) Voltage Control Register	4.3
LVD2_CTL2	Low-voltage Detection (ch.2) Voltage Control Register 2	4.4
CAL_CTL	Vref Calibration Control Register	4.9
CAL_SET	Vref Calibration Value Register.	4.10
CAL_KEY	Vref Calibration Security Key Register.	4.11

4.1 Low-voltage Detection Voltage Control Register (LVD_CTL)

The Low-voltage Detection Voltage Control Register (LVD_CTL) controls whether to enable monitoring the power supply voltage for low-voltage detection reset and a low-voltage detection interrupt, and specifies the detection voltage for a low-voltage detection interrupt.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	LVDRE	Reserved					Reserved	
Attribute	R/W	-					-	
Initial value	1	00000					00	

bit	7	6	5	4	3	2	1	0
Field	LVDIE	SVHI					Reserved	
Attribute	R/W	R/W					-	
Initial value	0	00011					00	

Register functions

[bit15] LVDRE: Low-voltage detection reset operation enable bit

This bit enables operation of power supply voltage monitoring of low-voltage detection reset. If it is not enabled, the low-voltage detection reset circuit stops operating.

Bit	Description
0	Generation of low-voltage detection reset is not enabled.
1	Generation of low-voltage detection reset is enabled. [initial value]

[bit14:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVDIE: Low-voltage detection interrupt enable bit

This bit is used to enable monitoring the power supply voltage of a low-voltage detection interrupt. When not enabling monitoring the power supply voltage, the Low-voltage Detection Interrupt Circuit is stopped.

Bit	Description
0	Does not enable the generation of a low-voltage detection interrupt. [Initial value]
1	Enables the generation of a low-voltage detection interrupt.

[bit6:2] SVHI: Low-voltage detection interrupt voltage setting bits

These bits are used to specify the detection voltage of a low-voltage detection interrupt.

bit6:2	Description
00000	Set the low-voltage detection interrupt voltage in the vicinity of 1.50 V.
00001	Set the low-voltage detection interrupt voltage in the vicinity of 1.55 V.
00010	Set the low-voltage detection interrupt voltage in the vicinity of 1.60 V.

bit6:2	Description
00011	Set the low-voltage detection interrupt voltage in the vicinity of 1.65 V. [Initial value]
00100	Set the low-voltage detection interrupt voltage in the vicinity of 1.70 V.
00101	Set the low-voltage detection interrupt voltage in the vicinity of 1.75 V.
00110	Set the low-voltage detection interrupt voltage in the vicinity of 1.80 V.
00111	Set the low-voltage detection interrupt voltage in the vicinity of 1.85 V.
01000	Set the low-voltage detection interrupt voltage in the vicinity of 1.90 V.
01001	Set the low-voltage detection interrupt voltage in the vicinity of 1.95 V.
01010	Set the low-voltage detection interrupt voltage in the vicinity of 2.00 V.
01011	Set the low-voltage detection interrupt voltage in the vicinity of 2.05 V.
01100	Set the low-voltage detection interrupt voltage in the vicinity of 2.50 V.
01101	Set the low-voltage detection interrupt voltage in the vicinity of 2.60 V.
01110	Set the low-voltage detection interrupt voltage in the vicinity of 2.70 V.
01111	Set the low-voltage detection interrupt voltage in the vicinity of 2.80 V.
10000	Set the low-voltage detection interrupt voltage in the vicinity of 2.90 V.
10001	Set the low-voltage detection interrupt voltage in the vicinity of 3.00 V.
10010	Set the low-voltage detection interrupt voltage in the vicinity of 3.10 V.
10011	Set the low-voltage detection interrupt voltage in the vicinity of 3.20 V.
Others	Setting is prohibited.

[bit1:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Notes:

- *The low-voltage detection interrupt enable bit (LVDIE) must be enabled after "0" was written to the LVDCL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) to clear the low-voltage detection interrupt factor bit (LVDIR).*
- *When the low-voltage detection interrupt enable bit (LVDIE) is not enabled, the Low-voltage Detection Circuit for detecting a low-voltage interrupt is stopped. Therefore, the low-voltage detection interrupt factor bit (LVDIR) is not set.*
- *The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the Low-voltage Detection Voltage Control Register (LVD_CTL), set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by reset factor other than these resets.*
- *The low-voltage detection reset voltage is a fixed value. Please check the datasheet for detail.*
- *This register is not initialized by deep standby transition reset.*

4.2 Low-voltage detection release voltage Control Register (LVD_CTL2)

The Low-voltage detection release voltage Control Register (LVD_CTL2) controls whether to set the detection voltage and release voltage separately for a low-voltage detection interrupt, and specifies the release voltage for a low-voltage detection interrupt. Moreover, the polarity of the LVD can be set.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							LVDPOL
Attribute	-							R/W
Initial value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	LVDRLIE	SVHRLI					Reserved	
Attribute	R/W	R/W					-	
Initial value	0	00011					00	

Register functions

[bit15:9] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit8] LVDPOL: Low-voltage detection polarity setting bit

This bit can reverse the polarity of the LVD.

For detail, please refer to section “Operations”

Bit	Description
0	Don’t reverse the polarity of LVD.[Initial value].
1	Reverse the polarity of LVD.

[bit7] LVDRLIE: Low-voltage detection release voltage setting enable bit

This bit is used to determined that whether the detection voltage and release voltage of LVD can be set separately.

Bit	Description
0	The release voltage cannot be set. The value of SVHRLI always equal to SVHI. [Initial value]
1	The release voltage can be set separately.

[bit6:2] SVHRLI: Low-voltage detection release voltage setting bits

These bits are used to specify the detection voltage of a low-voltage detection interrupt.

bit6:2	Description
00000	Set the low-voltage detection release voltage in the vicinity of 1.55 V.
00001	Set the low-voltage detection release voltage in the vicinity of 1.60 V.
00010	Set the low-voltage detection release voltage in the vicinity of 1.65 V.

bit6:2	Description
00011	Set the low-voltage detection release voltage in the vicinity of 1.70 V. [Initial value]
00100	Set the low-voltage detection release voltage in the vicinity of 1.75 V.
00101	Set the low-voltage detection release voltage in the vicinity of 1.80 V.
00110	Set the low-voltage detection release voltage in the vicinity of 1.85 V.
00111	Set the low-voltage detection release voltage in the vicinity of 1.90 V.
01000	Set the low-voltage detection release voltage in the vicinity of 1.95 V.
01001	Set the low-voltage detection release voltage in the vicinity of 2.00 V.
01010	Set the low-voltage detection release voltage in the vicinity of 2.05 V.
01011	Set the low-voltage detection release voltage in the vicinity of 2.10 V.
01100	Set the low-voltage detection release voltage in the vicinity of 2.60 V.
01101	Set the low-voltage detection release voltage in the vicinity of 2.70 V.
01110	Set the low-voltage detection release voltage in the vicinity of 2.80 V.
01111	Set the low-voltage detection release voltage in the vicinity of 2.90 V.
10000	Set the low-voltage detection release voltage in the vicinity of 3.00 V.
10001	Set the low-voltage detection release voltage in the vicinity of 3.10 V.
10010	Set the low-voltage detection release voltage in the vicinity of 3.20 V.
10011	Set the low-voltage detection release voltage in the vicinity of 3.30 V.
Others	Setting is prohibited.

[bit1:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Notes:

- *The low-voltage detection interrupt release voltage setting enable bit(LVDRLIE) must be set to 0 before LVDIE is enabled.*
- *The Low-voltage Detection Voltage Control Register (LVD_CTL2) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the Low-voltage Detection Voltage Control Register (LVD_CTL2), set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by reset factor other than these resets.*
- *When LVDRLIE is set to '0', the value of SVHRLI is always equal to the value of SVHI.*
- *In case of LVDRLIE=1, the write access to SVHRLI is denied if attempt writing a value to SVHRLI which is smaller than the value in SVHI.*
- *This register is not initialized by deep standby transition reset.*

4.3 Low-voltage Detection (ch.2) Voltage Control Register (LVD2_CTL)

The Low-voltage Detection (ch.2) Voltage Control Register (LVD2_CTL) controls whether to enable monitoring the power supply/external voltage for another low-voltage detection interrupt, and specifies the detection voltage for a low-voltage detection interrupt.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	LVD2IE	SVH2I					Reserved	
Attribute	R/W	R/W					-	
Initial value	0	00011					00	

Register functions

[bit15:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVD2IE: Low-voltage detection 2 interrupt enable bit

This bit is used to enable monitoring the power supply voltage/external voltage of a low-voltage detection interrupt (channel 2). When not enabling monitoring the power supply voltage/external voltage, the Low-voltage Detection Interrupt Circuit is stopped.

Bit	Description
0	Does not enable the generation of a low-voltage detection interrupt. [Initial value]
1	Enables the generation of a low-voltage detection interrupt.

[bit6:2] SVH2I: Low-voltage detection 2 interrupt voltage setting bits

These bits are used to specify the detection voltage of a low-voltage detection interrupt (channel 2).

bit6:2	Description
00000	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.50 V.
00001	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.55 V.
00010	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.60 V.
00011	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.65 V. [Initial value]
00100	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.70 V.
00101	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.75 V.
00110	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.80 V.
00111	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.85 V.
01000	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.90 V.
01001	Set the low-voltage detection 2 interrupt voltage in the vicinity of 1.95 V.

bit6:2	Description
01010	Set the low-voltage detection 2 interrupt voltage in the vicinity of 2.00 V.
01011	Set the low-voltage detection 2 interrupt voltage in the vicinity of 2.05 V.
01100	Set the low-voltage detection 2 interrupt voltage in the vicinity of 2.50 V.
01101	Set the low-voltage detection 2 interrupt voltage in the vicinity of 2.60 V.
01110	Set the low-voltage detection 2 interrupt voltage in the vicinity of 2.70 V.
01111	Set the low-voltage detection 2 interrupt voltage in the vicinity of 2.80 V.
10000	Set the low-voltage detection 2 interrupt voltage in the vicinity of 2.90 V.
10001	Set the low-voltage detection 2 interrupt voltage in the vicinity of 3.00 V.
10010	Set the low-voltage detection 2 interrupt voltage in the vicinity of 3.10 V.
10011	Set the low-voltage detection 2 interrupt voltage in the vicinity of 3.20 V.
Others	Setting is prohibited.

[bit1:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Notes:

- *The low-voltage detection 2 interrupt enable bit (LVD2IE) must be enabled after "0" was written to the LVD2CL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) to clear the low-voltage detection interrupt factor bit (LVD2IR).*
- *When the low-voltage detection 2 interrupt enable bit (LVD2IE) is not enabled, the Low-voltage Detection Circuit (channel 2) for detecting a low-voltage interrupt is stopped. Therefore, the low-voltage detection 2 interrupt factor bit (LVD2IR) is not set.*
- *The Low-voltage Detection Voltage Control Register (LVD2_CTL) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the Low-voltage Detection Voltage Control Register (LVD2_CTL), set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by reset factor other than these resets.*
- *This register is not initialized by deep standby transition reset.*

4.4 Low-voltage Detection (ch.2) release voltage Control Register (LVD2_CTL2)

The Low-voltage detection 2 release voltage Control Register (LVD2_CTL2) controls whether to set the detection voltage and release voltage separately for a low-voltage detection interrupt (ch.2), and specifies the release voltage for a low-voltage detection interrupt. Moreover, the polarity of the LVD2 can be set.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved						LVD2POL
Attribute	R/W	-						R/W
Initial value	0	000000						0

bit	7	6	5	4	3	2	1	0
Field	LVD2RLIE	SVH2RLI					Reserved	
Attribute	R/W	R/W					-	
Initial value	0	00011					00	

Register functions

[bit15:9] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit8] LVD2POL: Low-voltage detection 2 polarity setting bit

This bit can reverse the polarity of the LVD2.

Bit	Description
0	Don't reverse the polarity of LVD2.[Initial value].
1	Reverse the polarity of LVD2.

[bit7] LVD2RLIE: Low-voltage detection 2 release voltage setting enable bit

This bit is used to determined that whether the detection voltage and release voltage of LVD can be set separately.

Bit	Description
0	The release voltage cannot be set. The value of SVH2RLI always equal to SVH2I. [Initial value]
1	The release voltage can be set separately.

[bit6:2] SVH2RLI: Low-voltage detection release voltage setting bits

These bits are used to specify the detection voltage of a low-voltage detection interrupt.

bit6:2	Description
00000	Set the low-voltage detection 2 release voltage in the vicinity of 1.55 V.
00001	Set the low-voltage detection 2 release voltage in the vicinity of 1.60 V.
00010	Set the low-voltage detection 2 release voltage in the vicinity of 1.65 V.
00011	Set the low-voltage detection 2 release voltage in the vicinity of 1.70 V. [Initial value]
00100	Set the low-voltage detection 2 release voltage in the vicinity of 1.75 V.
00101	Set the low-voltage detection 2 release voltage in the vicinity of 1.80 V.
00110	Set the low-voltage detection 2 release voltage in the vicinity of 1.85 V.
00111	Set the low-voltage detection 2 release voltage in the vicinity of 1.90 V.
01000	Set the low-voltage detection 2 release voltage in the vicinity of 1.95 V.
01001	Set the low-voltage detection 2 release voltage in the vicinity of 2.00 V.
01010	Set the low-voltage detection 2 release voltage in the vicinity of 2.05 V.
01011	Set the low-voltage detection 2 release voltage in the vicinity of 2.10 V.
01100	Set the low-voltage detection 2 release voltage in the vicinity of 2.60 V.
01101	Set the low-voltage detection 2 release voltage in the vicinity of 2.70 V.
01110	Set the low-voltage detection 2 release voltage in the vicinity of 2.80 V.
01111	Set the low-voltage detection 2 release voltage in the vicinity of 2.90 V.
10000	Set the low-voltage detection 2 release voltage in the vicinity of 3.00 V.
10001	Set the low-voltage detection 2 release voltage in the vicinity of 3.10 V.
10010	Set the low-voltage detection 2 release voltage in the vicinity of 3.20 V.
10011	Set the low-voltage detection 2 release voltage in the vicinity of 3.30 V.
Others	Setting is prohibited.

[bit1:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Notes:

- *The low-voltage detection interrupt release voltage setting enable bit(LVDRLIE) must be set to 0 before LVDIE is enabled.*
- *When LVDRLIE is set to '0', the value of SVHRLI is always equal to the value of SVHI.*
- *The Low-voltage Detection Voltage Control Register (LVD_CTL2) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the Low-voltage Detection Voltage Control Register (LVD_CTL2), set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by reset factor other than these resets.*
- *The write access to SVH2RLI is denied if attempt writing a value to SVH2RLI which is smaller than the value in SVH2I.*
- *This register is not initialized by deep standby transition reset.*

4.5 Low-voltage Detection Interrupt Factor Register (LVD_STR)

The Low-voltage Detection Interrupt Factor Register (LVD_STR) holds a low-voltage detection interrupt factor.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	LVD2IR	Reserved						LVD2HSTR
Attribute	R	-						R
Initial value	0	000000						0

bit	7	6	5	4	3	2	1	0
Field	LVDIR	Reserved						LVDHSTR
Attribute	R	-						R
Initial value	0	000000						0

Register functions

[bit15] LVD2IR: Low-voltage detection 2 interrupt factor bit

Bit	Description
0	A low-voltage detection 2 interrupt request is not detected. [Initial value]
1	A low-voltage detection 2 interrupt request has been detected.

[bit14:9] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit8] LVD2HSTR:

Indicate the original LVD2's output. It is not affected by the setting of "LVD2POL".

Bit	Description
0	The voltage of VCC is lower than the setting of SVH2I.
1	The voltage of VCC is higher than the setting of SVH2RLI. [Initial value]

[bit7] LVDIR: Low-voltage detection interrupt factor bit

Bit	Description
0	A low-voltage detection interrupt request is not detected. [Initial value]
1	A low-voltage detection interrupt request has been detected.

[bit6:1] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit0] LVDHSTR:

Indicate the original LVDH output. It is not affected by the setting of "LVDPOL".

Bit	Description
0	The voltage of VCC is lower than the setting of SVHI.
1	The voltage of VCC is higher than the setting of SVHRLI. [Initial value]

Note:

- *This register is not initialized by deep standby transition reset.*

4.6 Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR)

The Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) clears a low-voltage detection interrupt factor.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	LVD2CL	Reserved						
Attribute	R1/W0	-						
Initial value	1	0000000						

bit	7	6	5	4	3	2	1	0
Field	LVDCL	Reserved						
Attribute	R1/W0	-						
Initial value	1	0000000						

Register functions

[bit15] LVD2CL: Low-voltage detection 2 interrupt factor clear bit

Bit	Description
0	Clears the low-voltage detection 2 interrupt factor bit (LVD2IR) of the Low-voltage Detection 2 Interrupt Factor Register (LVD_STR) to "0".
1	Has no effect on the operation in write mode. [Initial value]

"1" is always set in read mode.

[bit14:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVDCL: Low-voltage detection interrupt factor clear bit

Bit	Description
0	Clears the low-voltage detection interrupt factor bit (LVDIR) of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to "0".
1	Has no effect on the operation in write mode. [Initial value]

"1" is always set in read mode.

[bit6:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

4.7 Low-voltage Detection Voltage Protection Register (LVD_RLR)

The Low-voltage Detection Voltage Protection Register (LVD_RLR) write-protects the Low-voltage Detection Voltage Control Register (LVD_CTL), .

Register configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	LVDLCK[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LVDLCK[15:0]															
Attribute	R/W															
Initial value	0x0001															

Register functions

[bit31:0] LVDLCK[31:0]: Low-voltage Detection Voltage Control Register protection bits

- Setting 0x1ACCE553 enables writing the Low-voltage Detection Voltage Control Register (LVD_CTL) (releases write protection mode).
- Setting a value other than 0x1ACCE553 disables writing the Low-voltage Detection Voltage Control Register (LVD_CTL) (enables write protection mode).
- When the Low-voltage Detection Voltage Control Register (LVD_CTL) is not set in write protection mode, 0x00000000 is read.
- When the Low-voltage Detection Voltage Control Register (LVD_CTL) is set in write protection mode, 0x00000001 is read.

Notes:

- *The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state. To write the LVD_CTL Register, set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *To enable write protection mode of the LVD_CTL register, set a value other than 0x1ACCE553 to the LVD_RLR register.*
- *Once write protection mode is released for the LVD_CTL Register, it remains released until a value other than 0x1ACCE553 is written to the LVD_RLR Register.*
- *This register is not initialized by deep standby transition reset.*

4.8 Low-voltage Detection Circuit Status Register (LVD_STR2)

The Low-voltage Detection Circuit Status Register (LVD_STR2) checks the operation status of a low-voltage detection interrupt.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	LVD2IRDY	Reserved						
Attribute	R	-						
Initial value	0	0000000						

bit	7	6	5	4	3	2	1	0
Field	LVDIRDY	LVDRRDY	Reserved					
Attribute	R	R	-					
Initial value	0	1	000000					

Register functions

[bit15] LVD2IRDY: Low-voltage detection 2 interrupt status flag

Bit	Description
0	Stabilization wait state or monitoring stop state
1	Monitoring state [Initial value]

This bit has no effect on the operation in write mode.

[bit14:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVDIRDY: Low-voltage detection interrupt status flag

Bit	Description
0	Stabilization wait state or monitoring stop state [Initial value]
1	Monitoring state

This bit has no effect on the operation in write mode.

[bit6] LVDRRDY: Low-voltage detection reset status flag

Bit	Description
0	Stabilization wait state or monitoring stop state
1	Monitoring state [Initial value]

This bit has no effect on the operation in write mode.

[bit5:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

4.9 Vref Calibration Control Register (CAL_CTL)

The Vref Calibration Control Register (CAL_CTL) controls Vref calibration.

There’re 2 Vref in product, one is bipolar Vref, it’s accurate with higher current; the other is MOS Vref, it’s low power and can be calibrated with bipolar Vref. In this part, we’ll explain how to do low per Vref calibration and how to select it.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved				CALDONE	BGRSEL	Reserved	CALSTART
Attribute	-				R	R/W	-	R/W
Initial value	0000				1	0	0	0

Register functions

[bit3] CALDONE: Vref Calibration Done Flag

Bit	Description
0	Vref calibration is on-going or not start. [Initial value]
1	Vref calibration is finished.

This bit has no effect on the operation in write mode.

[bit2] BGRSEL: Vref Select Bit

Bit	Description
0	Select Bipolar Vref [Initial value]
1	Select MOS Vref. When finish Vref calibration, will select MOS Vref automatically.

[bit0] CALSTART: Vref Calibration Start Bit

Bit	Description
0	Vref calibration is not start. [Initial value]
1	Write 1 to start Vref calibration. If calibration done, this bit is clear to 0 automatically.

[bit15:4][bit1] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

4.10 Vref Calibration Value Register (CAL_SET)

The Vref Calibration Valuel Register (CAL_set) stores Vref calibration value.

This calibration value only valid when MOS Verf is selected.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved			DSTB[4:0]				
Attribute	-			R/W				
Initial value	000			10001				

[bit4:0] DSTB: Vref Calibration Value

Bit4:0	Description
xxxxx	Vref calibration value. [Initial value:10001, near 1.2000V] DSTB[4:0] will be valid if set CAL_CTL.BGRSEL=1.

[bit15:5] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

4.11 Vref Calibration Security Key Register (CAL_KEY)

The Vref Calibration Security Key is used to lock/unlock CAL_CTL and CAL_SET writing.

If locked, user cannot change CAL_CTL and CAL_SET register.

Only support word access.

Register configuration

bit	31	30	29	28	27	26	25	24
Field	CAL_KEY[31:24]							
Attribute	R0/W							
Initial value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CAL_KEY[23:16]							
Attribute	R0/W							
Initial value	00000000							

bit	15	14	13	12	11	10	9	8
Field	CAL_KEY[15:8]							
Attribute	R0/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	CAL_KEY[7:0]							
Attribute	R0/W							
Initial value	00000001							

[bit31:0] CAL_KEY: Vref Calibration Security Key Value

Bit31:0	Description
xxxxxxxx	Vref calibration security key value. [Initial value:0001] Write 32'h1ACCE553 to unlock the register CAL_CTL and CAL_SET. Write other value to lock above registers. Read 0x0001 means above registers are under protection. Read 0x0000 means above registers are not under protection.

5. Usage Precautions

This section explains the precautions for using Low-voltage Detection Circuit.

■ Low-voltage detection interrupt factor bit at STOP mode transition

Even if the power supply voltage is not greater than the detection voltage after clearing the Low-voltage Detection Interrupt factor bit (LVD_STRLVDIR), the interrupt factor will not be enabled again unless the power supply voltage becomes greater than the released voltage once.
But, the voltage comparison is executed by the Low-voltage Detection Circuit without fail when the transition to the STOP mode is executed while the power supply voltage is not greater than the detection voltage after clearing LVDIR.
So, the Low-voltage detection interrupt factor is set again by the transition to STOP mode and the process could go to the interrupt routine.

For example, in case of the transition to STOP mode in the interrupt routine of the Low-voltage detection, after clearing the interrupt factor and finishing the interrupt routine, the interrupt factor bit is set soon again and the interrupt routine might be repeated.

To prevent the occurrence of repeated interrupts when the voltage is not greater than the detection voltage after the detection of the low-voltage interrupt, disable the Low-voltage detection interrupt enable bit (LVDIE) and get out of the routine.

■ Setup of Detection Voltage of Low-voltage Detection Reset

When the Low-voltage detection reset is generated after setting the detection voltage of Low-voltage detection reset voltage, the detection voltage setting value is initialized. When the power supply voltage is higher than the initial setup value as in the case where the power supply voltage lowers slowly, the reset is released. But, as the program is returned to the beginning by the reset, set the detection voltage again. As the power supply voltage is already lower than the detection voltage, the Low-voltage detection reset is set again.

That is to say, the loop of the detection voltage change, reset, initialization (returned to the beginning), the detection voltage change, and reset could be repeated according to the power supply voltage change and program description.

The following measures can be taken to prevent the loop of the Low-voltage detection reset when the power supply voltage is not greater than the detection voltage:

- For the setting value of the Low-voltage detection reset, only the initial value is used.
- Set the Low-voltage detection interrupt before the Low-voltage detection reset. Confirm whether an interrupt flag is set at the beginning of the program and change the setting value of the Low-voltage reset detection voltage, if required.

CHAPTER 5-4: Low-voltage Detection (TYPE3)

This chapter explains the functions and operations of the Low-voltage Detection Circuit.

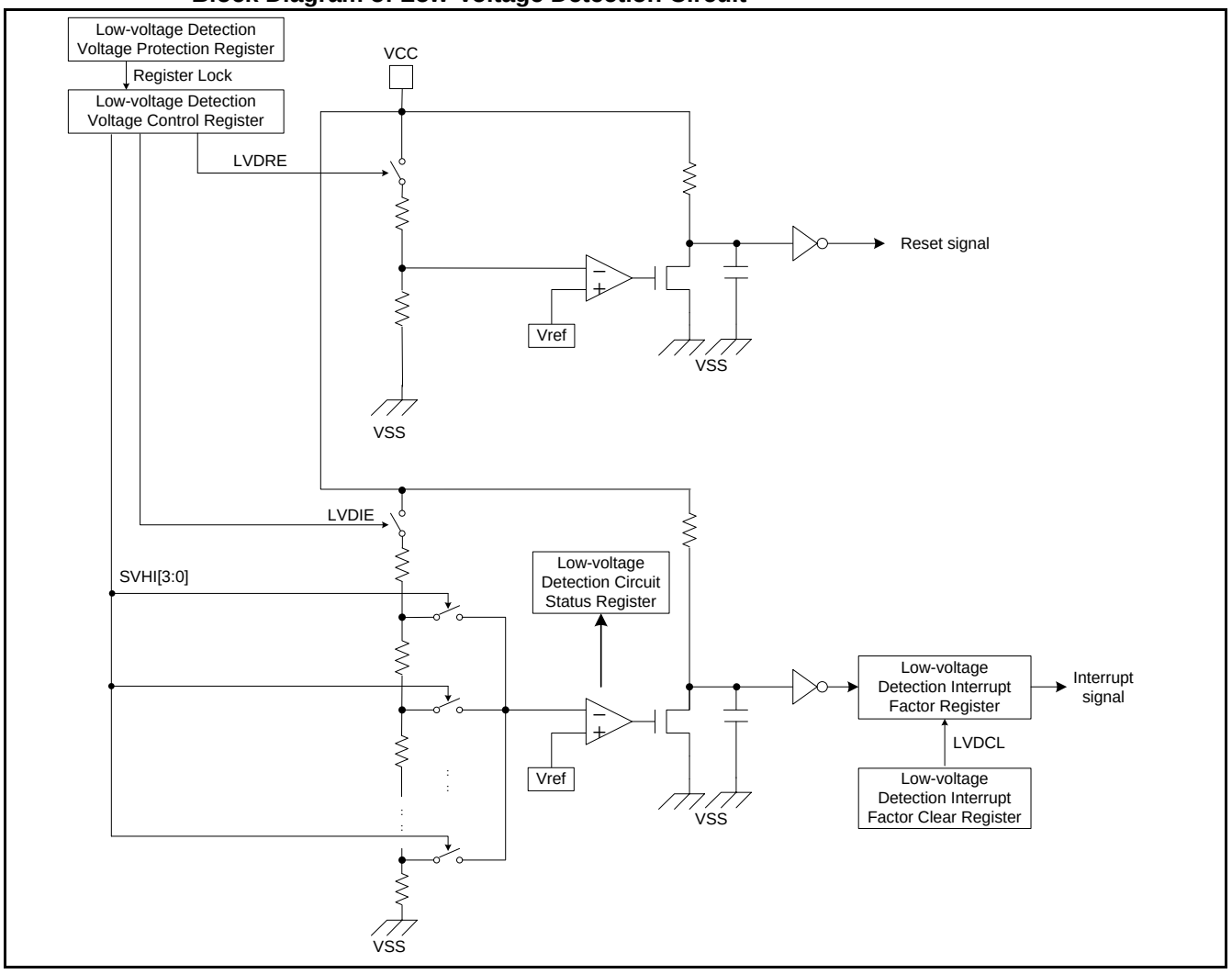
1. Configuration
2. Operations
3. Setup Procedure Examples
4. Registers
5. Usage Precautions

CODE: 9AFLVD-FM0T3-E03.0

1. Configuration

This section shows the block diagram of the Low-voltage Detection Circuit.

Block Diagram of Low-Voltage Detection Circuit



- Low-voltage Detection Voltage Control Register (LVD_CTL)
  This register controls whether to enable monitoring the power supply voltage for a low-voltage detection reset and low-voltage detection interrupt, and specifies the detection voltage for a low-voltage detection interrupt.
- Low-voltage Detection Voltage Protection Register (LVD_RLR)
  This register write-protects the Low-voltage Detection Voltage Control Register.
- Low-voltage Detection Interrupt Factor Register (LVD_STR)
  This register holds a low-voltage detection interrupt factor.
- Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR)
  This register clears a low-voltage detection interrupt factor.

- Low-voltage Detection Circuit Status Register (LVD_STR2)
  This register checks the operation status of a low-voltage detection interrupt circuit.
- Reference Voltage generation circuit (Vref)
  Vref is a circuit that generates reference voltage. In this product, two types of Vrefs are equipped, Bipolar Vref and MOS Vref. In initial condition, Bipolar Vref is used as a reference voltage generator for Low-Voltage-Detection Reset/Interrupt.

Pins of Low-voltage Detection Circuit

The following shows the pins used in the Low-voltage Detection Circuit.

- VCC pin
  The Low-voltage Detection Circuit monitors the power supply voltage of this pin.
- VSS pin
  This pin is a GND pin used as a basis to detect the power supply.

2. Operations

This section explains the operations of the Low-Voltage Detection Reset Circuit and the Low-voltage Detection Interrupt Circuit.

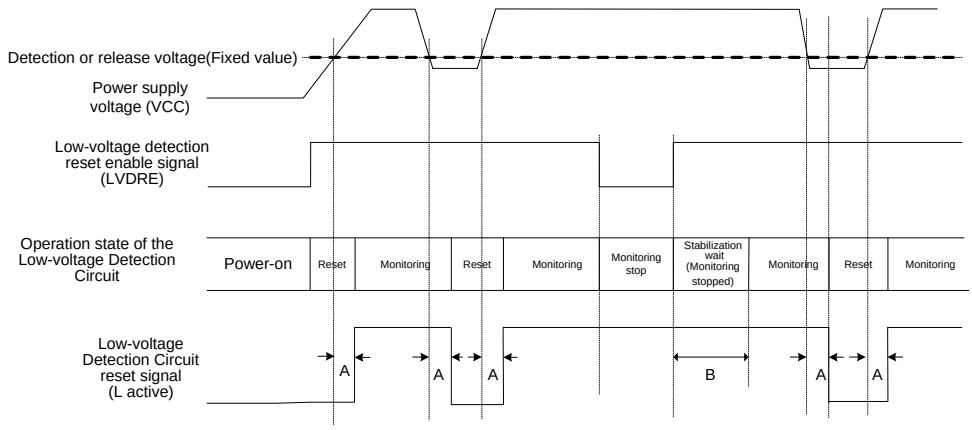
Operations of Low-Voltage Detection Reset Circuit

■ Explanation of Circuit Operation

The Low-Voltage Detection Reset Circuit enters a monitoring state after power-on. This circuit generates a reset signal when the power supply voltage (VCC) falls below the detection voltage specified. A reset is released when the power supply voltage exceeds the release voltage.

Reset operation is valid only when LVDRE bit of Low-voltage Detection Voltage Control Register (LVD_CTL) is "1". Detection voltage of the reset is fixed and cannot be changed. If the power supply voltage is higher than the release voltage, the reset is released. When reset enable is set, Low-voltage detection reset status flag (LVDRRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) becomes "1" and starts power supply voltage monitoring after the stabilization wait period for the low-voltage detection circuit lapses.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode).



A : Low-voltage detection delay time (T_{LVDDL})
B : Low Voltage Detection Circuit stabilization wait time (T_{LVDW})

Operations of Low-voltage Detection Interrupt Circuit

■ Explanation of Circuit Operation

The Low-voltage Detection Interrupt Circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified value.

An interrupt request is enabled when "1" is set to the LVDIE bit of the Low-voltage Detection Voltage Control Register (LVD_CTL). The initial value is set to Not Enable ("0"). The interrupt detection voltage can be set by the SVHI bit of the Low-voltage Detection Voltage Control Register (LVD_CTL). When an interrupt request is enabled and the interrupt detection voltage is specified, Low-voltage detection interrupt status flag (LVDIRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1" and this circuit starts monitoring the power supply voltage after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode). It is also applicable when the CPU returns from those modes.

■ Low-voltage detection interrupt request

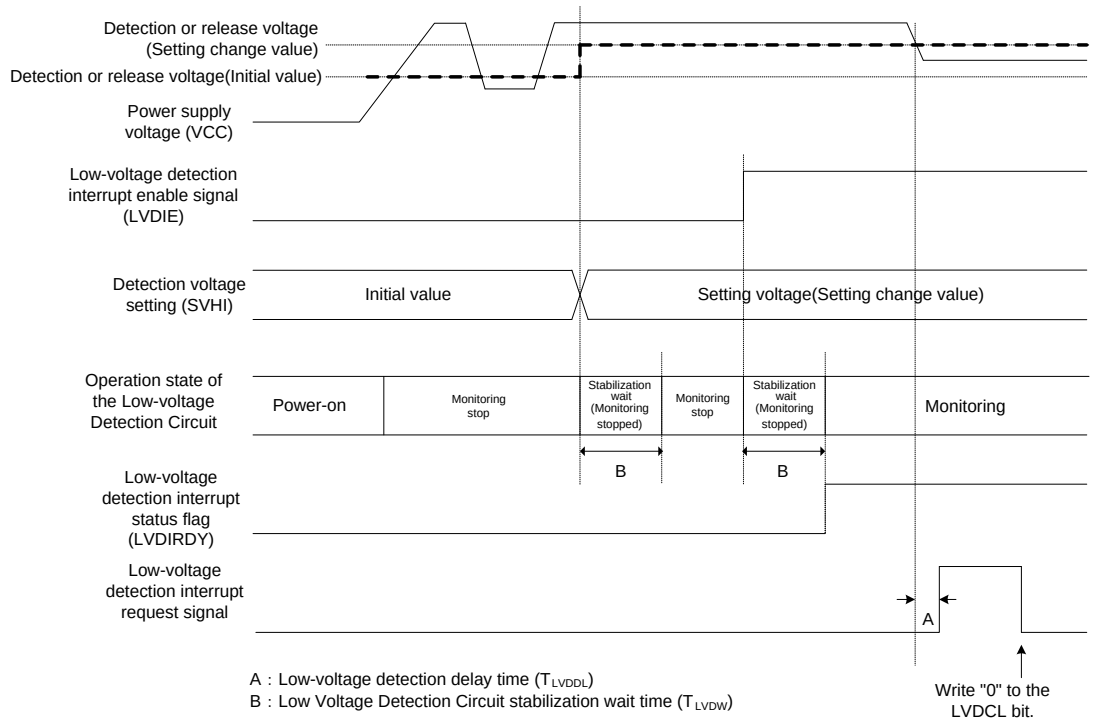
When the power supply voltage (VCC) falls below the specified voltage while a low-voltage detection interrupt is effective, "1" is set to the LVDIR bit of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to generate an interrupt request signal.

An interrupt request can be checked by reading the LVDIR bit.

■ Canceling a low-voltage detection interrupt request

To cancel a low-voltage detection interrupt request, write "0" to the LVDCL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR). This clears a low-voltage detection interrupt factor and cancels a low-voltage detection interrupt request.

Even when the power supply voltage is below the specified detection voltage, an interrupt request is canceled when "0" is written to the LVDCL bit.



Note:

- *This circuit does not conduct monitoring the power supply voltage if PCLK1 is stopped by TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, Deep standby STOP mode or APB1 Prescaler Register (APBC1_PSR) while waiting for the stabilization of the Low-voltage Detection Circuit. After checking that Low-voltage detection interrupt status flag (LVDIRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1", change to the desired mode.*

Operations of Vref Calibration

■ Purpose of Vref Calibration

Typically, Low Voltage Detection Reset/Interrupt circuit needs accurate reference voltage.

In this product, two types of Vref are equipped. One is a Bipolar Vref, which dissipates high current but is accurate. The other is a MOS Vref, which dissipates lower current and is accurate if it’s calibrated with output voltage of Bipolar Vref.

Usually, MOS Vref is used in view of low power consumption. Therefore, calibration is necessary ahead of using MOS Vref, especially for example when Deep standby mode is used.

■ Starting Vref Calibration

Writing “1” to CALSTART bit of CAL_CTL register starts Calibration. By monitoring CALDONE bits of CAL_CTL register, user understand whether the calibration was finished.

When the calibration finishes, Vref switches from Bipolar Vref to MOS Vref, automatically.

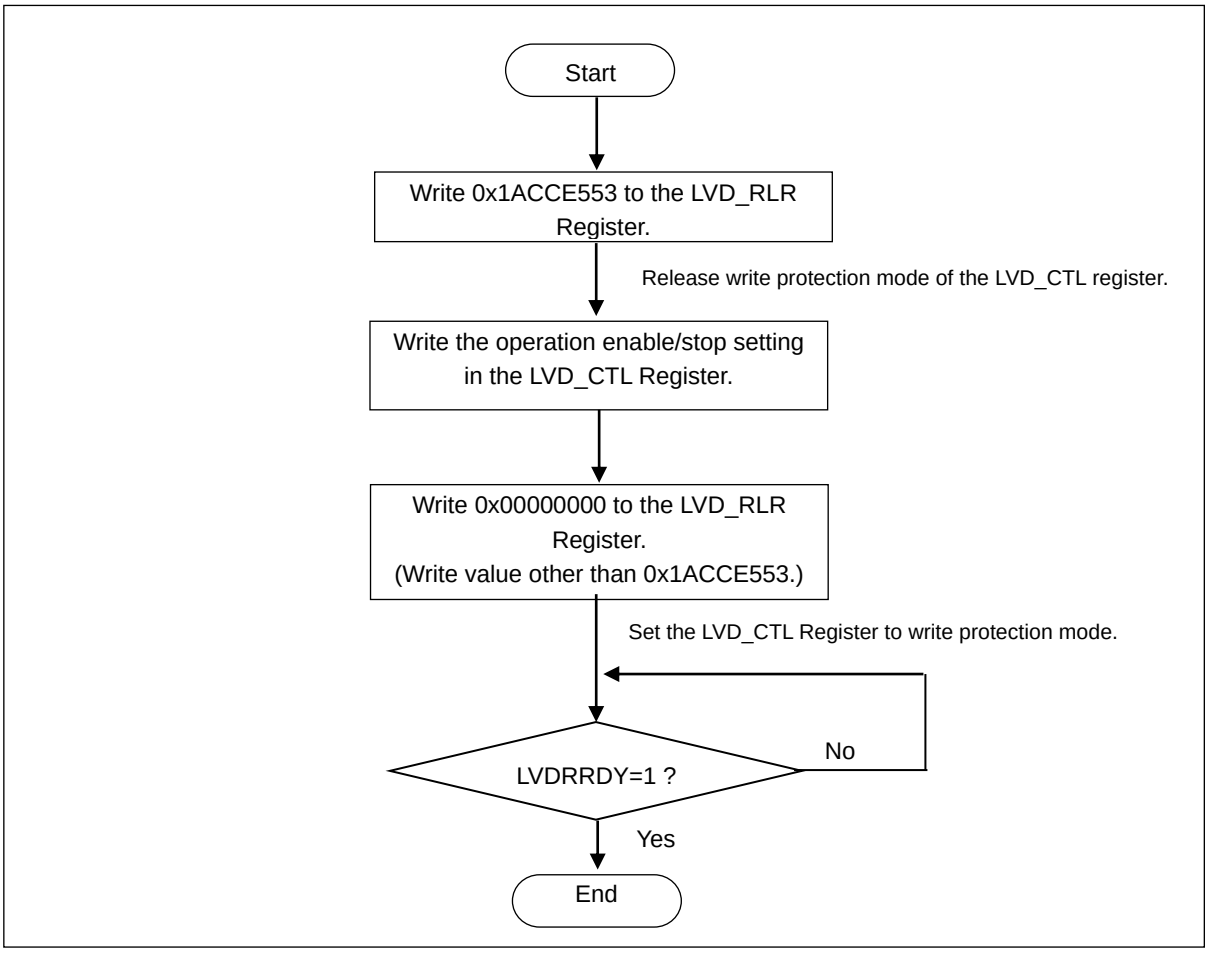
In order to use(read/write) CAL_CTL register, it has to be unlocked with CAL_KEY register. See “3.3 Example for Vref calibration” for detail.

3. Setup Procedure Examples

This section explains the procedures to set up the Low-voltage Detection Circuit, giving examples.

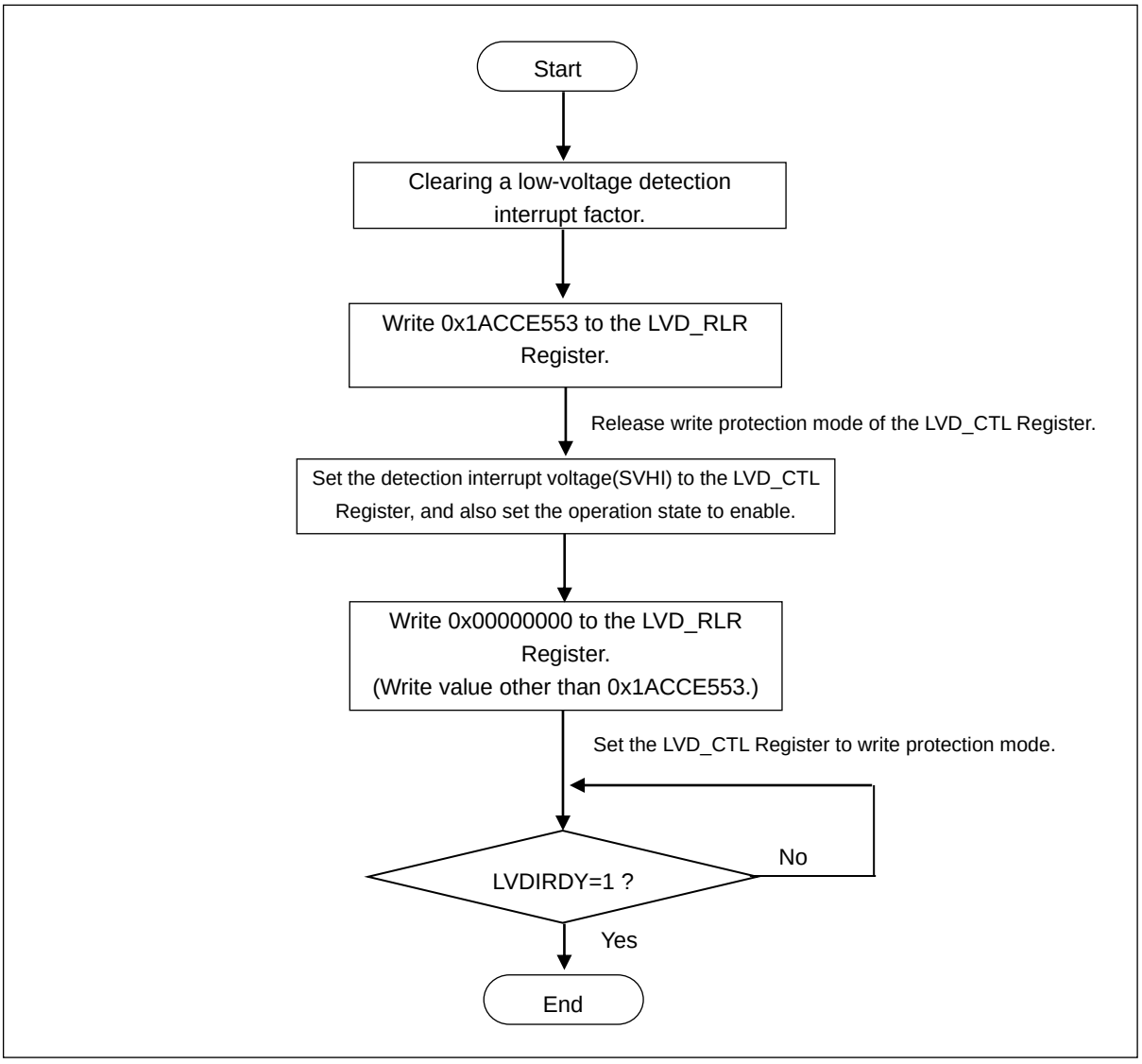
3.1 Example for Low-voltage detection reset

Figure 3-1 Setting procedure example for low-voltage detection reset



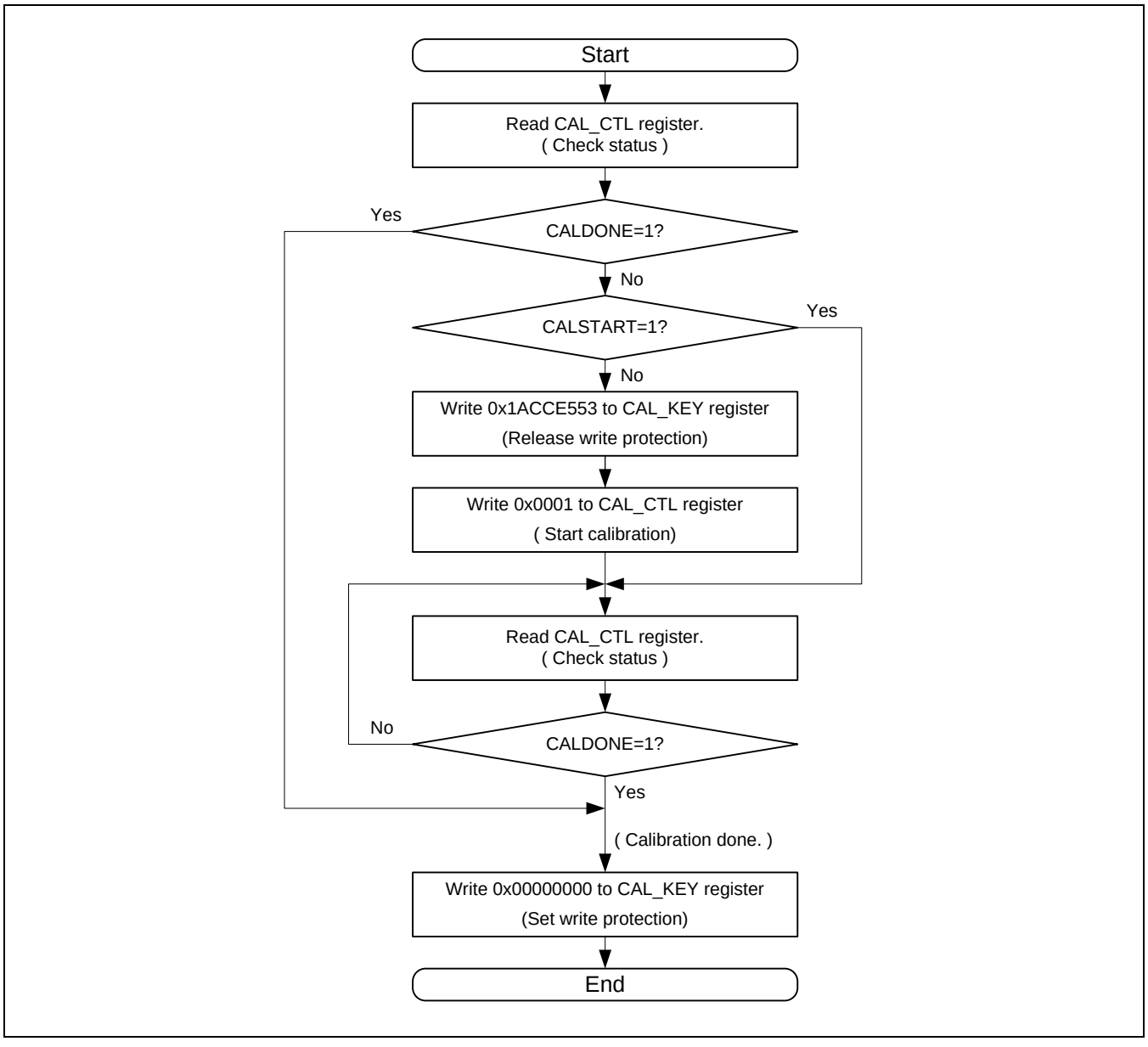
3.2 Example for Low-voltage detection interrupt

Figure 3-2 Setting procedure example for low-voltage detection interrupt.



3.3 Example for Vref calibration

Figure 3-3 Setting procedure example for Vref calibration



4. Registers

This section explains the configuration and functions of the registers used in the Low-voltage Detection Circuit.

List of low-voltage detection circuit registers

Table 4-1 List of low-voltage detection circuit registers

Abbreviation	Register name	Reference
LVD_CTL	Low-voltage Detection Voltage Control Register	4.1
LVD_STR	Low-voltage Detection Interrupt Factor Register	4.2
LVD_CLR	Low-voltage Detection Interrupt Factor Clear Register	4.3
LVD_RLR	Low-voltage Detection Voltage Protection Register	4.4
LVD_STR2	Low-voltage Detection Circuit Status Register	4.5
CAL_CTL	Vref Calibration Control Register	4.6
CAL_KEY	Vref Calibration Security Key Register.	4.7

4.1 Low-voltage Detection Voltage Control Register (LVD_CTL)

The Low-voltage Detection Voltage Control Register (LVD_CTL) controls whether to enable monitoring the power supply voltage for low-voltage detection reset and a low-voltage detection interrupt, and specifies the detection voltage for a low-voltage detection interrupt.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	LVDRE	Reserved						
Attribute	R/W	-						
Initial value	1	0000000						

bit	7	6	5	4	3	2	1	0
Field	LVDIE	SVHI					Reserved	
Attribute	R/W	R/W					-	
Initial value	0	00011					00	

Register functions

[bit15] LVDRE: Low-voltage detection reset operation enable bit

This bit enables operation of power supply voltage monitoring of low-voltage detection reset. If it is not enabled, the low-voltage detection reset circuit stops operating.

Bit	Description
0	Generation of low-voltage detection reset is not enabled.
1	Generation of low-voltage detection reset is enabled. [initial value]

[bit14:8] Reserved: Reserved bits

The read value is always "0". Write always 0 when writing.

[bit7] LVDIE: Low-voltage detection interrupt enable bit

This bit is used to enable monitoring the power supply voltage of a low-voltage detection interrupt. When not enabling monitoring the power supply voltage, the Low-voltage Detection Interrupt Circuit is stopped.

Bit	Description
0	Does not enable the generation of a low-voltage detection interrupt. [Initial value]
1	Enables the generation of a low-voltage detection interrupt.

[bit6:2] SVHI: Low-voltage detection interrupt voltage setting bits

These bits are used to specify the detection voltage of a low-voltage detection interrupt.

bit6:2	Description
00000	Set the low-voltage detection interrupt voltage in the vicinity of 1.50 V.
00001	Set the low-voltage detection interrupt voltage in the vicinity of 1.55 V.
00010	Set the low-voltage detection interrupt voltage in the vicinity of 1.60 V.

bit6:2	Description
00011	Set the low-voltage detection interrupt voltage in the vicinity of 1.65 V. [Initial value]
00100	Set the low-voltage detection interrupt voltage in the vicinity of 1.70 V.
00101	Set the low-voltage detection interrupt voltage in the vicinity of 1.75 V.
00110	Set the low-voltage detection interrupt voltage in the vicinity of 1.80 V.
00111	Set the low-voltage detection interrupt voltage in the vicinity of 1.85 V.
01000	Set the low-voltage detection interrupt voltage in the vicinity of 1.90 V.
01001	Set the low-voltage detection interrupt voltage in the vicinity of 1.95 V.
01010	Set the low-voltage detection interrupt voltage in the vicinity of 2.00 V.
01011	Set the low-voltage detection interrupt voltage in the vicinity of 2.05 V.
01100	Set the low-voltage detection interrupt voltage in the vicinity of 2.50 V.
01101	Set the low-voltage detection interrupt voltage in the vicinity of 2.60 V.
01110	Set the low-voltage detection interrupt voltage in the vicinity of 2.70 V.
01111	Set the low-voltage detection interrupt voltage in the vicinity of 2.80 V.
10000	Set the low-voltage detection interrupt voltage in the vicinity of 2.90 V.
10001	Set the low-voltage detection interrupt voltage in the vicinity of 3.00 V.
10010	Set the low-voltage detection interrupt voltage in the vicinity of 3.10 V.
10011	Set the low-voltage detection interrupt voltage in the vicinity of 3.20 V.
Others	Setting is prohibited.

[bit1:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Notes:

- *The low-voltage detection interrupt enable bit (LVDIE) must be enabled after "0" was written to the LVDCL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) to clear the low-voltage detection interrupt factor bit (LVDIR).*
- *When the low-voltage detection interrupt enable bit (LVDIE) is not enabled, the Low-voltage Detection Circuit for detecting a low-voltage interrupt is stopped. Therefore, the low-voltage detection interrupt factor bit (LVDIR) is not set.*
- *The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the Low-voltage Detection Voltage Control Register (LVD_CTL), set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by reset factor other than these resets.*
- *The low-voltage detection reset voltage is a fixed value. Please check the datasheet for detail.*
- *This register is not initialized by deep standby transition reset.*

4.2 Low-voltage Detection Interrupt Factor Register (LVD_STR)

The Low-voltage Detection Interrupt Factor Register (LVD_STR) holds a low-voltage detection interrupt factor.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	LVDIR	Reserved						Reserve
Attribute	R	-						-
Initial value	0	000000						-

Register functions

[bit15:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVDIR: Low-voltage detection interrupt factor bit

Bit	Description
0	A low-voltage detection interrupt request is not detected. [Initial value]
1	A low-voltage detection interrupt request has been detected.

[bit6:1] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit0] Reserved: Reserved bits

The read value is either "0" or “1”. These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

4.3 Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR)

The Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) clears a low-voltage detection interrupt factor.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	LVDCL	Reserved						
Attribute	R/W	-						
Initial value	1	0000000						

Register functions

[bit15:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVDCL: Low-voltage detection interrupt factor clear bit

Bit		Description
Writing	0	Clears the low-voltage detection interrupt factor bit (LVDIR) of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to "0".
	1	Has no effect on the operation in write mode. [Initial value]
Reading	1	Read always

"1" is always set in read mode.

[bit6:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

4.4 Low-voltage Detection Voltage Protection Register (LVD_RLR)

The Low-voltage Detection Voltage Protection Register (LVD_RLR) write-protects the Low-voltage Detection Voltage Control Register (LVD_CTL), .

Register configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	LVDLCK[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LVDLCK[15:0]															
Attribute	R/W															
Initial value	0x0001															

Register functions

[bit31:0] LVDLCK[31:0]: Low-voltage Detection Voltage Control Register protection bits

- Setting 0x1ACCE553 enables writing the Low-voltage Detection Voltage Control Register (LVD_CTL) (releases write protection mode).
- Setting a value other than 0x1ACCE553 disables writing the Low-voltage Detection Voltage Control Register (LVD_CTL) (enables write protection mode).
- When the Low-voltage Detection Voltage Control Register (LVD_CTL) is not set in write protection mode, 0x00000000 is read.
- When the Low-voltage Detection Voltage Control Register (LVD_CTL) is set in write protection mode, 0x00000001 is read.

Notes:

- *The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state. To write the LVD_CTL Register, set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.*
- *To enable write protection mode of the LVD_CTL register, set a value other than 0x1ACCE553 to the LVD_RLR register.*
- *Once write protection mode is released for the LVD_CTL Register, it remains released until a value other than 0x1ACCE553 is written to the LVD_RLR Register.*
- *This register is not initialized by deep standby transition reset.*

4.5 Low-voltage Detection Circuit Status Register (LVD_STR2)

The Low-voltage Detection Circuit Status Register (LVD_STR2) checks the operation status of a low-voltage detection interrupt.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	LVDIRDY	LVDRRDY	Reserved					
Attribute	R	R	-					
Initial value	0	1	000000					

Register functions

[bit15:8] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit7] LVDIRDY: Low-voltage detection interrupt status flag

Bit	Description
0	Stabilization wait state or monitoring stop state [Initial value]
1	Monitoring state

This bit has no effect on the operation in write mode.

[bit6] LVDRRDY: Low-voltage detection reset status flag

Bit	Description
0	Stabilization wait state or monitoring stop state
1	Monitoring state [Initial value]

This bit has no effect on the operation in write mode.

[bit5:0] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

Note:

- *This register is not initialized by deep standby transition reset.*

4.6 Vref Calibration Control Register (CAL_CTL)

The Vref Calibration Control Register (CAL_CTL) controls Vref calibration.

Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved				CALDONE	Reserved	Reserved	CALSTART
Attribute	-				R	-	-	R/W
Initial value	0000				0	0	0	0

Register functions

[bit15:4] Reserved: Reserved bits

The read value is always "0". These bits have no effect on the operation when written.

[bit3] CALDONE: Vref Calibration Done Flag

This bit indicates whether the calibration is finished.

Write value to this bit is ignored.

Bit	Description
Reading 0	Vref calibration is not completed. [Initial value]
Reading 1	Vref calibration is completed.

[bit2:1] Reserved: Reserved bits

The read value is either "0" or “1”. When writing, write “0” to this bit. Writing “1” is prohibited.

[bit0] CALSTART: Vref Calibration Start Bit

This bit is used to start Vref calibration and to obtain its status.

Bit	Description
Writing 0	Writing 0 is prohibited.
Writing 1	Write 1 to start Vref calibration.
Reading 0	Vref calibration is not executing. [Initial value] (Calibration is either not stated or finished)
Reading 1	Vref calibration is executing.

Note:

- *When the calibration finishes, MOS Vref automatically becomes effective.*
- *CALDONE register is not initialized by Deep Standby Reset.*
- *Writing 1 to CALSTART (direction to calibration start) can be done only in the condition where both CALDONE=0 and CALSTART=0 are met.*

- *In the condition where CALDONE is 1 or CALSTART is 1, writing 1 to CALSTART is prohibited. Read the CAL_CTL register and confirm the state of the register before calibration start.*
- *For this register, Read-Modify-Write access is prohibited through Bit Band Alias area.*

4.7 Vref Calibration Security Key Register (CAL_KEY)

The Vref Calibration Security Key is used to lock/unlock CAL_CTL writing.

If locked, user cannot change CAL_CTL register.

Only support word access.

Register configuration

bit	31	30	29	28	27	26	25	24
Field	CAL_KEY[31:24]							
Attribute	R0/W							
Initial value	00000000							

bit	23	22	21	20	19	18	17	16
Field	CAL_KEY[23:16]							
Attribute	R0/W							
Initial value	00000000							

bit	15	14	13	12	11	10	9	8
Field	CAL_KEY[15:8]							
Attribute	R0/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	CAL_KEY[7:0]							
Attribute	R0/W							
Initial value	00000001							

Register functions

[bit31:0] CAL_KEY: Vref Calibration Security Key Bits

- Setting 0x1ACCE553 enables writing the Vref Calibration Control Register (CAL_CTL) (releases write protection mode).
- Setting a value other than 0x1ACCE553 disables writing the Vref Calibration Control Register (CAL_CTL) (enables write protection mode).
- When the Vref Calibration Control Register (CAL_CTL) is not set in write protection mode, 0x00000000 is read.
- When the Vref Calibration Control Register (CAL_CTL) is set in write protection mode, 0x00000001 is read.

Notes:

- *The Vref Calibration Control Register (CAL_CTL) is write-protected in the initial state. To write the CAL_CTL Register, set 0x1ACCE553 to the Vref security key Register (CAL_KEY) to release write protection mode.*
- *To enable write protection mode of the CAL_CTL register, set a value other than 0x1ACCE553 to the CAL_KEY register.*

- *Once write protection mode is released for the CAL_CTL Register, it remains released until a value other than 0x1ACCE553 is written to the CAL_KEY Register.*
- *This register is not initialized by deep standby transition reset.*

5. Usage Precautions

This section explains the precautions for using Low-voltage Detection Circuit.

■ Low-voltage detection interrupt factor bit at STOP mode transition

Even if the power supply voltage is not greater than the detection voltage after clearing the Low-voltage Detection Interrupt factor bit (LVD_STRLVDIR), the interrupt factor will not be enabled again unless the power supply voltage becomes greater than the released voltage once.
But, the voltage comparison is executed by the Low-voltage Detection Circuit without fail when the transition to the STOP mode is executed while the power supply voltage is not greater than the detection voltage after clearing LVDIR.
So, the Low-voltage detection interrupt factor is set again by the transition to STOP mode and the process could go to the interrupt routine.

For example, in case of the transition to STOP mode in the interrupt routine of the Low-voltage detection, after clearing the interrupt factor and finishing the interrupt routine, the interrupt factor bit is set soon again and the interrupt routine might be repeated.

To prevent the occurrence of repeated interrupts when the voltage is not greater than the detection voltage after the detection of the low-voltage interrupt, disable the Low-voltage detection interrupt enable bit (LVDIE) and get out of the routine.

■ Setup of Detection Voltage of Low-voltage Detection Reset

When the Low-voltage detection reset is generated after setting the detection voltage of Low-voltage detection reset voltage, the detection voltage setting value is initialized. When the power supply voltage is higher than the initial setup value as in the case where the power supply voltage lowers slowly, the reset is released. But, as the program is returned to the beginning by the reset, set the detection voltage again. As the power supply voltage is already lower than the detection voltage, the Low-voltage detection reset is set again.

That is to say, the loop of the detection voltage change, reset, initialization (returned to the beginning), the detection voltage change, and reset could be repeated according to the power supply voltage change and program description.

The following measures can be taken to prevent the loop of the Low-voltage detection reset when the power supply voltage is not greater than the detection voltage:

- For the setting value of the Low-voltage detection reset, only the initial value is used.
- Set the Low-voltage detection interrupt before the Low-voltage detection reset. Confirm whether an interrupt flag is set at the beginning of the program and change the setting value of the Low-voltage reset detection voltage, if required.

CHAPTER 6-1: Low Power Consumption Mode

This chapter explains the functions and operations of low power consumption mode.

1. Overview
2. Configuration of CPU Operation Modes
3. Operations of Standby Modes
4. Standby Mode Setting Procedure Examples
5. Description of Deep Standby Mode Operation
6. Deep Standby Mode Setting Procedure Examples
7. Deep Standby Return Factor Determination Procedure
8. Registers
9. Usage Precautions

CODE: 9AFLPMODE-FM0-E03.0

1. Overview

To reduce the power consumption, the system provides low power consumption mode, which enables the use of the standby mode of SLEEP, TIMER, RTC and STOP modes and the deep standby mode of deep standby RTC and deep standby STOP modes.

Overview of CPU Operation Modes

CPU operation modes are classified into the following types.

- Run modes
  - High-speed CR run mode
  - Main run mode
  - PLL run mode
  - Low-speed CR run mode
  - Sub run mode
- Standby modes
  - Sleep modes
    - High-speed CR sleep mode
    - Main sleep mode
    - PLL sleep mode
    - Low-speed CR sleep mode
    - Sub sleep mode
  - Timer modes
    - High-speed CR timer mode
    - Main timer mode
    - PLL timer mode
    - Low-speed CR timer mode
    - Sub timer mode
  - RTC mode
  - STOP mode
- Deep Standby modes
  - Deep Standby RTC mode
  - Deep Standby STOP mode

Table of Low Power Consumption Modes Equipped in Each TYPE

Table 1-1 Table of Low Power Consumption Mode

Mode	TYPE1-M0+	TYPE2-M0+ TYPE3-M0+
Run mode	O	O
Standby mode	○	O
Deep standby mode	-	O

Overview of RUN Mode

RUN mode is defined with a clock selected as a master clock. The base clocks, which are obtained by dividing the master clock frequency, are supplied to CPU clock, AHB bus clock, and APB bus clock to run the CPU, buses, and most peripherals.

The source clock frequency can be changed dynamically. When not using the main or sub oscillator, the source clock oscillator can be stopped.

RUN mode is divided into the following modes depending on the clock selected as a master clock.

- High-speed CR run mode

  In this mode, the high-speed CR oscillator clock is used as a master clock. When not using the main or sub oscillator, the respective oscillators can be stopped. The status of PLL Multiplier Circuit varies depending on the setting of the PLLE bit. The low-speed CR oscillator is always set to the active state. It changes to this mode after a reset has been released.

- Main run mode

  In this mode, the main oscillator clock is used as a master clock. The statuses of the high-speed CR oscillator PLL Multiplier Circuit and sub oscillator vary depending on the settings of the HCRE bit, PLLE bit and SOSCE bit respectively. The low-speed CR oscillator is always set to the active state.

- PLL run mode

  In this mode, the PLL clock obtained by multiplying the main oscillator clock or high-speed CR oscillator clock is used as a master clock. The low-speed CR oscillator is always set to the active state. The status of the sub oscillator varies depending on the setting of the SOSCE bit. The statuses of the high-speed CR oscillator and main oscillator vary depending on the setting of the PINC bit in the PSW_TMR Register.

- Low-speed CR run mode

  In this mode, the low-speed CR oscillator clock is used as a master clock. The status of the sub oscillator varies depending on the setting of the SOSCE bit. The main oscillator, high-speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

- Sub run mode

  In this mode, the sub oscillator clock is used as a master clock. The low-speed CR oscillator is always set to the active state. The main oscillator, high-speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

Overview of SLEEP Mode

SLEEP mode is classified as one of standby modes. SLEEP mode is used to stop CPU clocks. This causes the CPU to be stopped, reducing the power consumption. The resources connected to the AHB and APB bus clocks continue operations.

SLEEP mode is divided into the following modes depending on a master clock at the transition to SLEEP mode.

- High-speed CR sleep mode

  When the high-speed CR oscillator clock is selected as a master clock, the system changes to high-speed CR sleep mode if the transition to SLEEP mode is requested. In this mode, the status of the PLL Multiplier Circuit, main or sub oscillator varies depending on the setting of the PLLE, MOSCE or SOSCE bit. The low-speed CR oscillator is always set to the active state.

- Main sleep mode

When the main clock is selected as a master clock, the system changes to main sleep mode if the transition to SLEEP mode is requested. In this mode, the statuses of the high-speed CR oscillator PLL Multiplier Circuit and sub oscillator vary depending on the settings of the HCRE bit, PLLE bit and SOSCE bit respectively. The low-speed CR oscillator is always set to the active state.

- PLL sleep mode

  When the main PLL clock is selected as a master clock, the system changes to PLL sleep mode if the transition to SLEEP mode is requested. In this mode, the low-speed CR oscillator is always set to the active state. The status of the sub oscillator varies depending on the setting of the SOSCE bit. The statuses of the high-speed CR oscillator and main oscillator vary depending on the setting of the PINC bit in the PSW_TMR Register.

- Low-speed CR sleep mode

  When the low-speed CR clock is selected as a master clock, the system changes to low-speed CR sleep mode if the transition to SLEEP mode is requested. In this mode, the status of the sub oscillator varies depending on the setting of the SOSCE bit. The main oscillator, high-speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

- Sub sleep mode

  When the sub clock is selected as a master clock, the system changes to sub sleep mode if the transition to SLEEP mode is requested. In this mode, the low-speed CR oscillator is always set to the active state. The main oscillator, high-speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

Overview of TIMER Mode

TIMER mode is classified as one of standby modes. TIMER mode is used to stop supplying a base clock. This causes the CPU clock, AHB bus clock, and all APB bus clocks to be stopped, reducing the power consumption. In this case, all functions are stopped, excluding the oscillators, PLL, hardware watchdog timer, watch counter, RTC, clock failure detector, and Low Voltage Detection Circuit.

TIMER mode is divided into the following modes depending on a master clock at the transition to TIMER mode.

- High-speed CR timer mode

  When the high-speed CR oscillator clock is selected as a master clock, the system changes to high-speed CR timer mode if the transition to TIMER mode is requested. In this mode, the status of the PLL Multiplier Circuit, main or sub oscillator varies depending on the setting of the PLLE, MOSCE or SOSCE bit. The low-speed CR oscillator is always set to the active state.

- Main timer mode

  When the main clock is selected as a master clock, the system changes to main timer mode if the transition to TIMER mode is requested. In this mode, the statuses of the high-speed CR oscillator PLL Multiplier Circuit and sub oscillator vary depending on the settings of the HCRE bit, PLLE bit and SOSCE bit respectively. The low-speed CR oscillator is always set to the active state.

- PLL timer mode

  When the main PLL clock is selected as a master clock, the system changes to PLL timer mode if the transition to TIMER mode is requested. In this mode, the low-speed CR oscillator is always set to the active state. The status of the sub oscillator varies depending on the setting of the SOSCE bit. The statuses of the high-speed CR oscillator and main oscillator vary depending on the setting of the PINC bit in the PSW_TMR Register.

- Low-speed CR timer mode

  When the low-speed CR clock is selected as a master clock, the system changes to low-speed CR timer mode if the transition to TIMER mode is requested. In this mode, the status of the sub oscillator varies depending on the setting of the SOSCE bit. The main oscillator, high-speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

- Sub timer mode

  When the sub clock is selected as a master clock, the system changes to sub timer mode if the transition to TIMER mode is requested. In this mode, the sub oscillator and low-speed CR oscillator are always set to the active state. The main oscillator, high-speed CR oscillator, and PLL Multiplier Circuit are not available in this mode.

Overview of RTC Mode

RTC mode is classified as one of the standby modes. RTC mode stops oscillation other than that of the sub oscillator. All the functions except for watch counter, RTC, and low voltage detection circuit will be stopped.

Overview of STOP Mode

STOP mode is classified as one of standby modes. STOP mode is used to stop all oscillating operations. Enabling this mode stops all functions, excluding the Low Voltage Detection Circuit.

Overview of Deep Standby RTC Mode

Deep standby RTC mode is classified as one of the deep standby modes. Deep standby RTC mode stops oscillation other than that of the sub oscillator. All the functions except for the RTC and low voltage detection circuit will be stopped. It turns off RTC, low voltage detection circuit, CPUs other than GPIO, on-chip Flash memory, on-chip SRAM*, and peripheral functions inside the chip.

Overview of Deep Standby Stop Mode

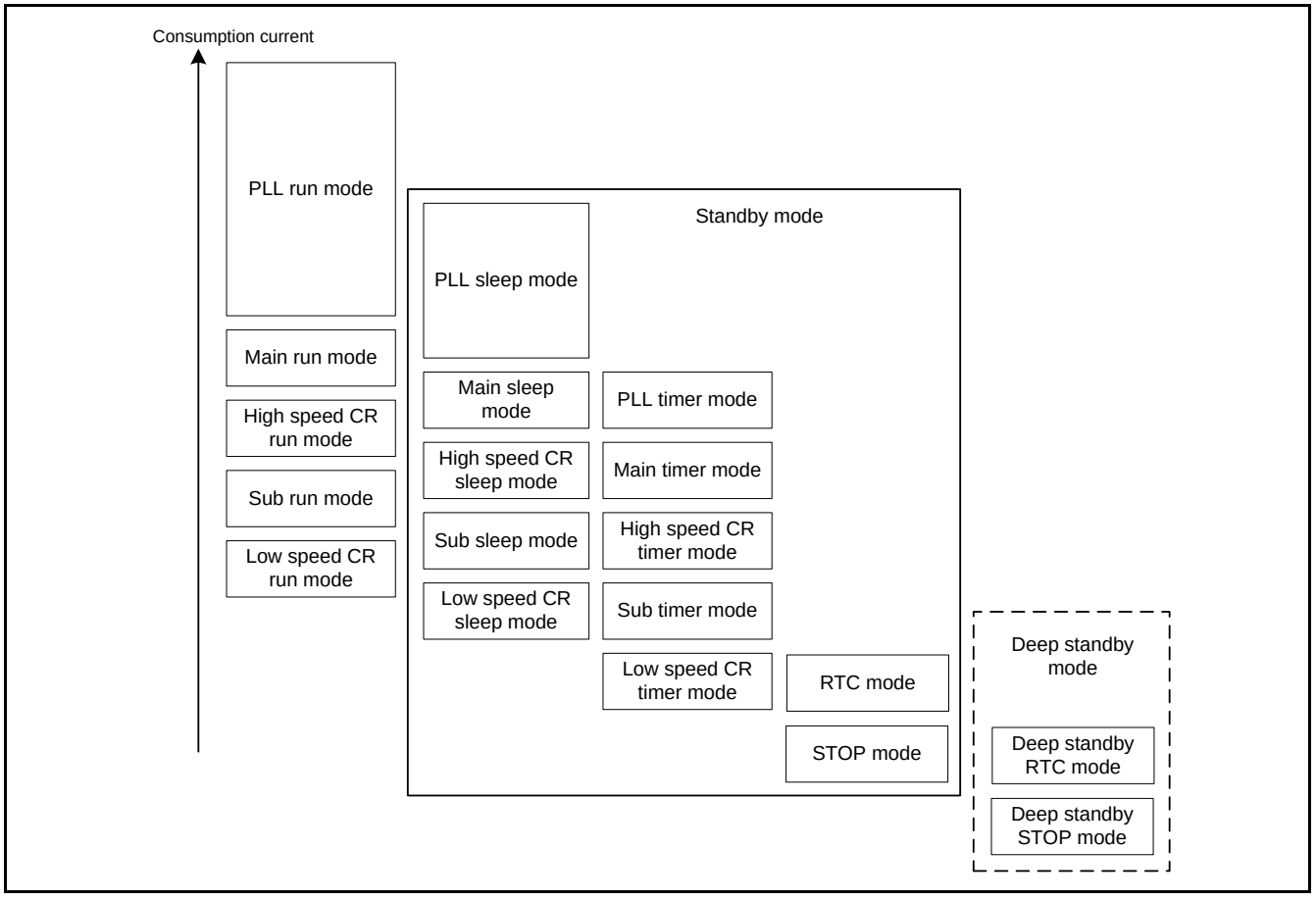
Deep standby stop mode is classified as one of the deep standby modes. Deep standby stop mode stops all oscillations. All the functions except for the low voltage detection circuit will be stopped. It turns off RTC, low voltage detection circuit, CPUs other than GPIO, on-chip Flash memory, on-chip SRAM*, and peripheral functions inside the chip.

*: Data in on-chip SRAM can be retained even in the deep standby modes.
In the setting to retain on-chip SRAM data, the power of on-chip SRAM is ON.

Relationships between CPU Operation Modes and Consumption Current Values

Figure 1-1 shows the relationships between CPU operation modes and consumption current values.

Figure 1-1 Relationships between CPU Operation Modes and Consumption Current Values



Note:

- *Figure 1-1 shows only an overview of the magnitude relationship among consumption currents of each mode. The actual consumption current values vary depending on the oscillator and PLL starting conditions in each mode or the clock configuration of the selected frequency and other elements.*

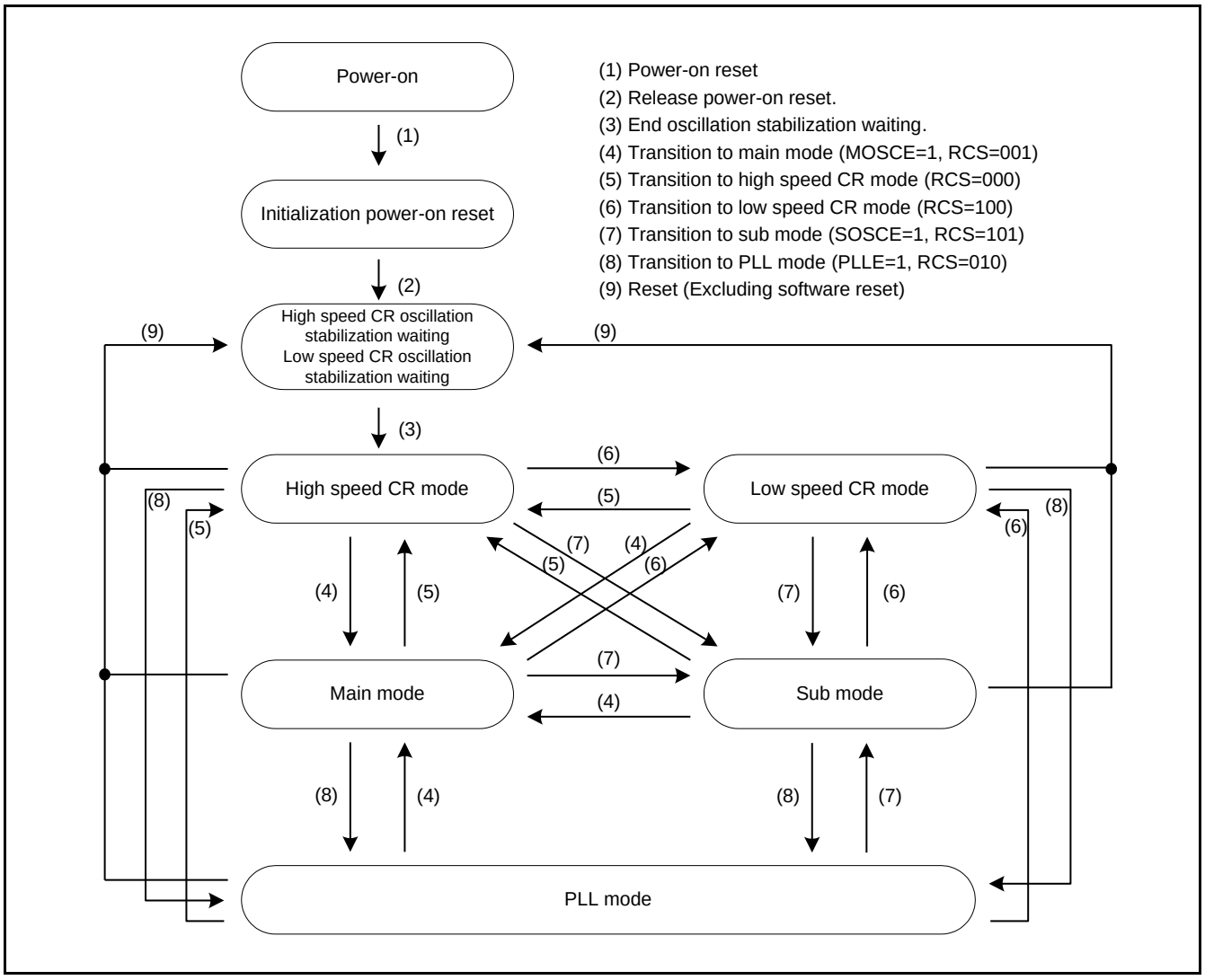
2. Configuration of CPU Operation Modes

This section explains the configuration of CPU operation modes.

CPU Operation Mode Transition Diagram

Figure 2-1 shows the CPU operation mode transition diagram.

Figure 2-1 CPU Operation Mode Transition Diagram



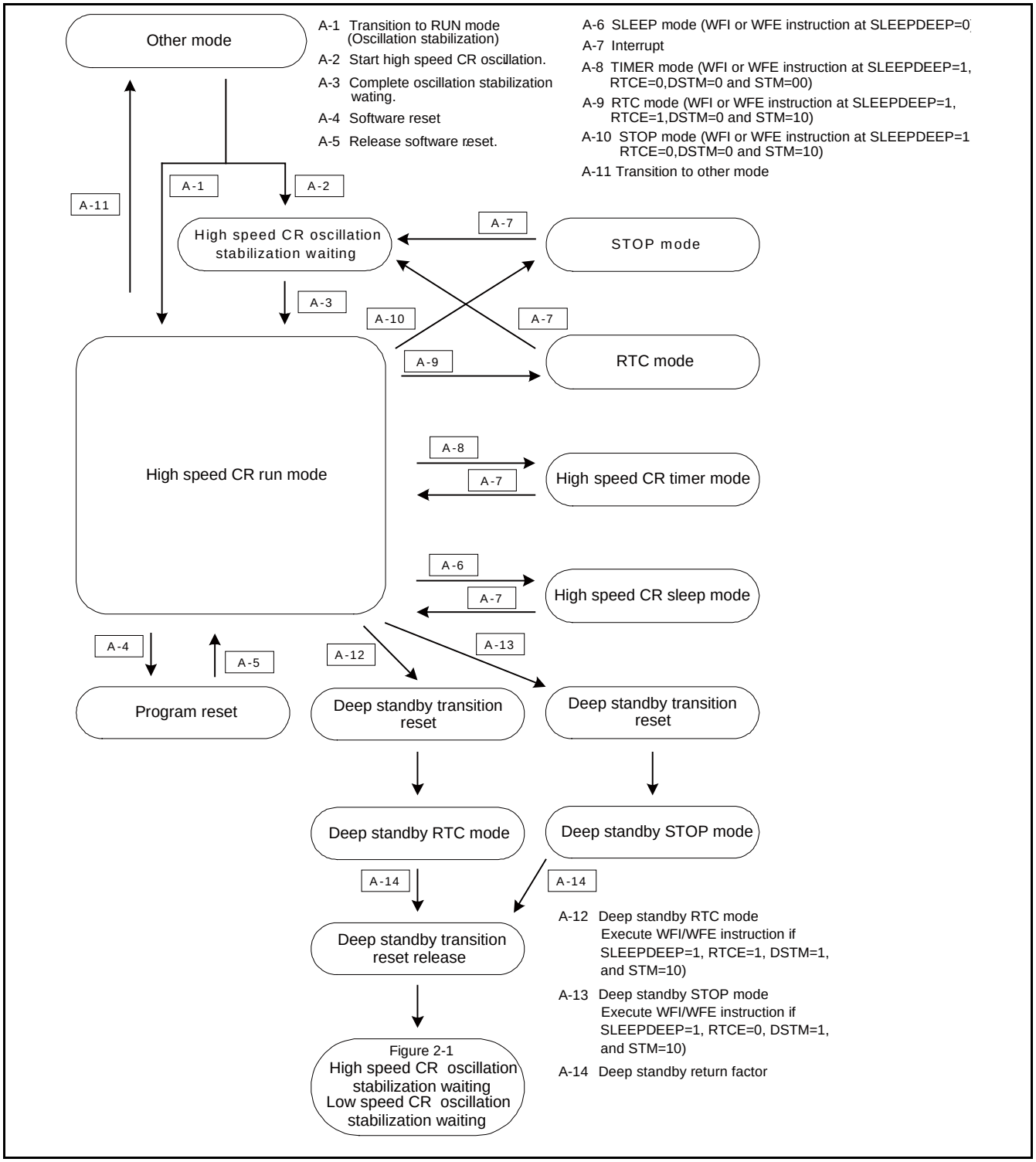
- High-speed CR mode
  In this mode, the high-speed CR oscillator clock is used as a master clock.
- Main mode
  In this mode, the main oscillator clock is used as a master clock.
- Low-speed CR mode
  In this mode, the low-speed CR oscillator clock is used as a master clock.

- Sub mode
  In this mode, the sub oscillator clock is used as a master clock.
- PLL mode
  In this mode, the PLL oscillator clock is used as a master clock.

■ High-speed CR mode transition diagram

In high-speed CR mode, the high-speed CR oscillator clock is used as a master clock.

Figure 2-2 High-Speed CR Mode Transition Diagram



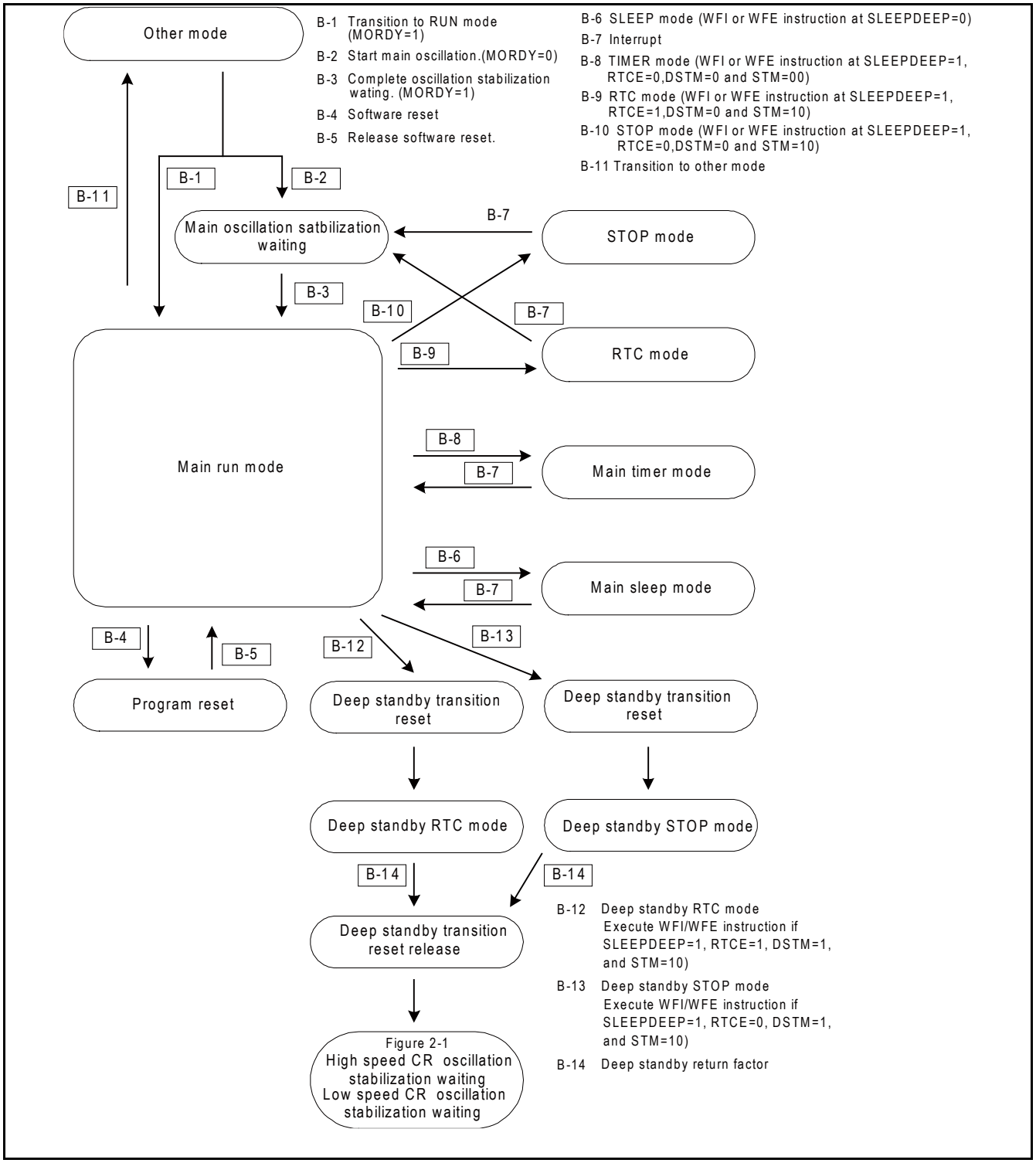
Note:

- *There are products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1 (table of low power consumption modes).*

Main Mode Transition Diagram

In main mode, the main oscillator clock is used as a master clock.

Figure 2-3 Main Mode Transition Diagram



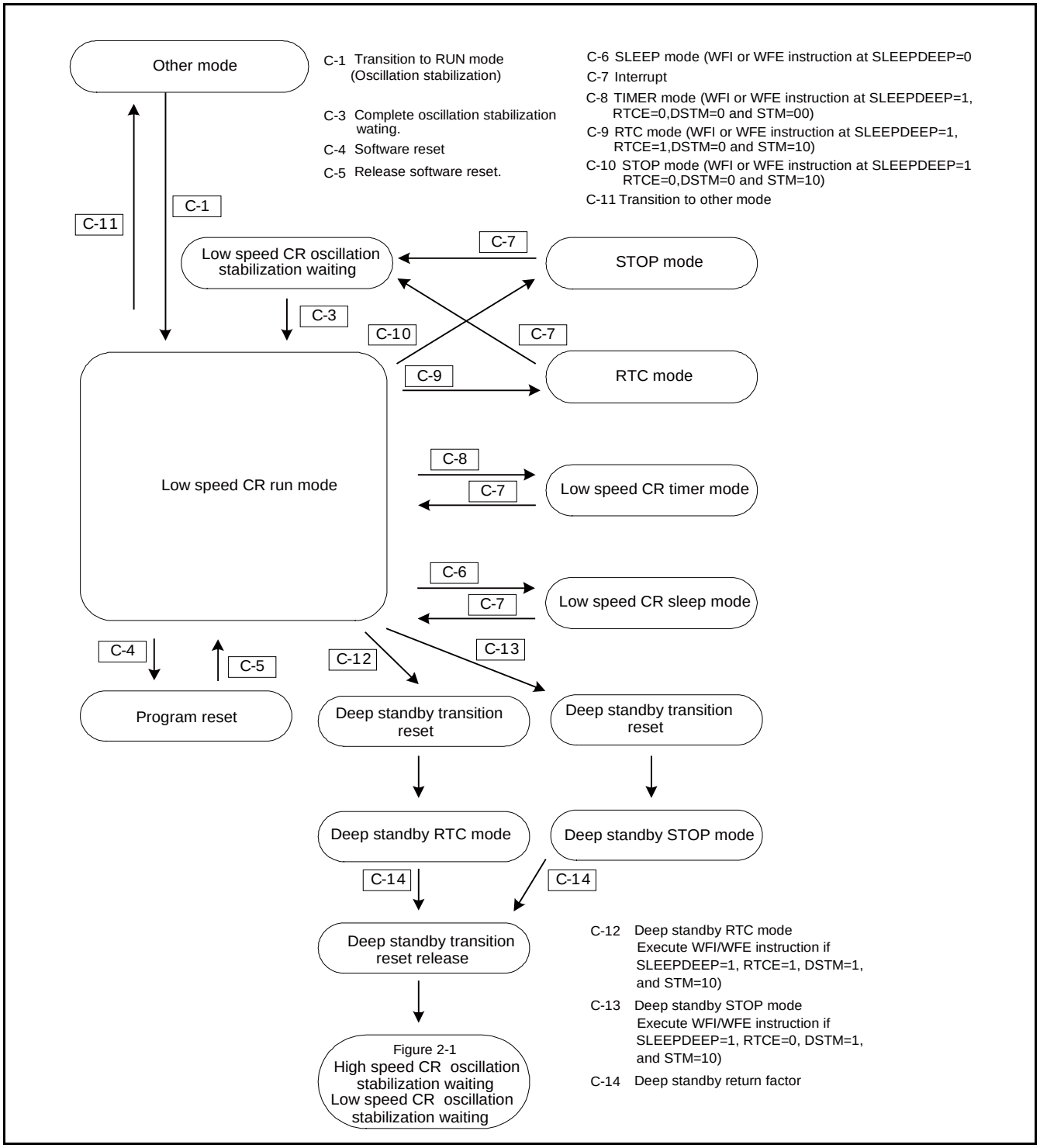
Note:

- *There are products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1 (table of low power consumption modes).*

Low-Speed CR Mode Transition Diagram

In low-speed CR mode, the low-speed CR oscillator clock is used as a master clock.

Figure 2-4 Low-Speed CR Mode Transition Diagram



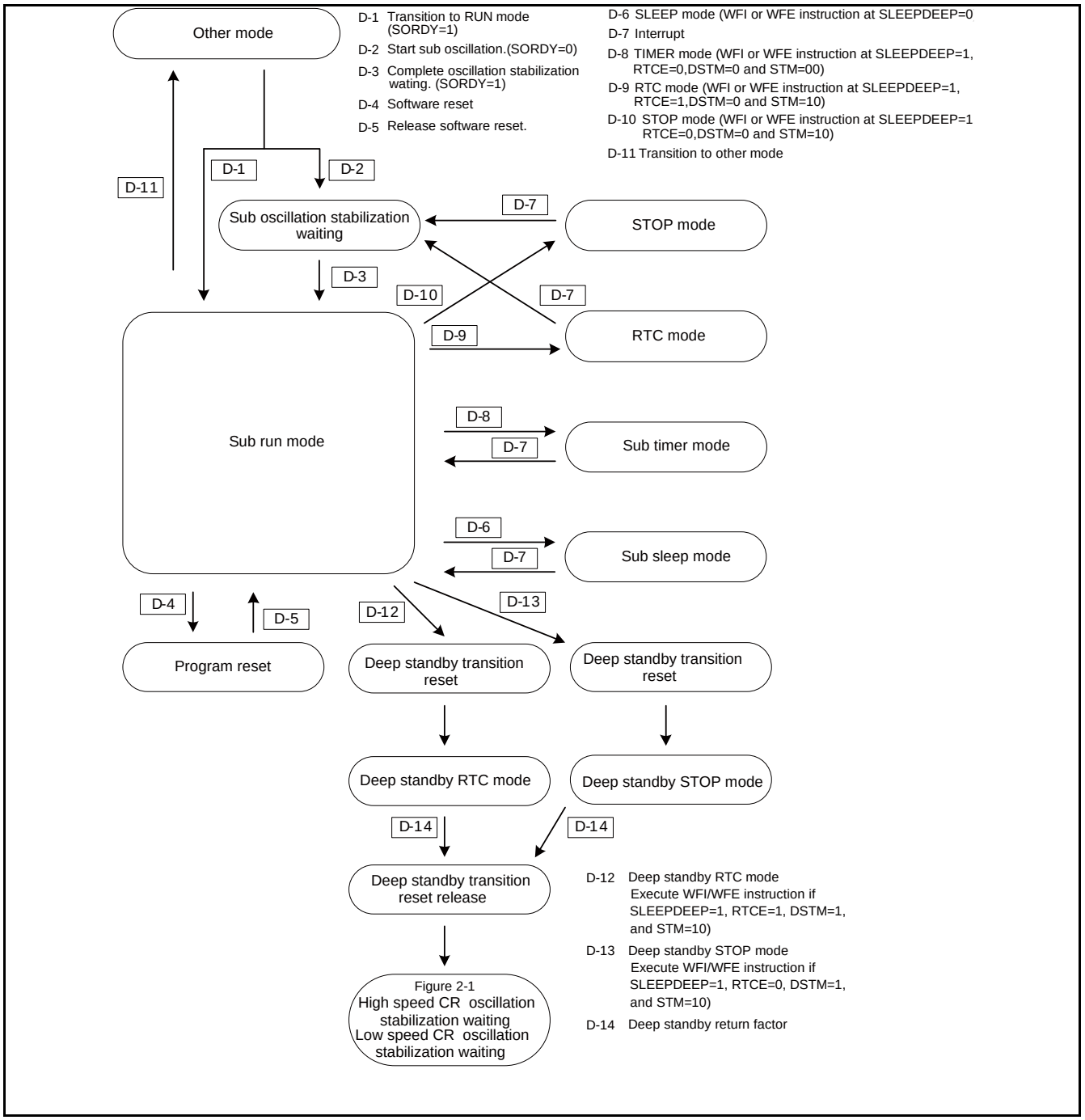
Note:

- *There are products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1 (table of low power consumption modes).*

Sub Mode Transition Diagram

In sub mode, the sub oscillator clock is used as a master clock.

Figure 2-5 Sub Mode Transition Diagram



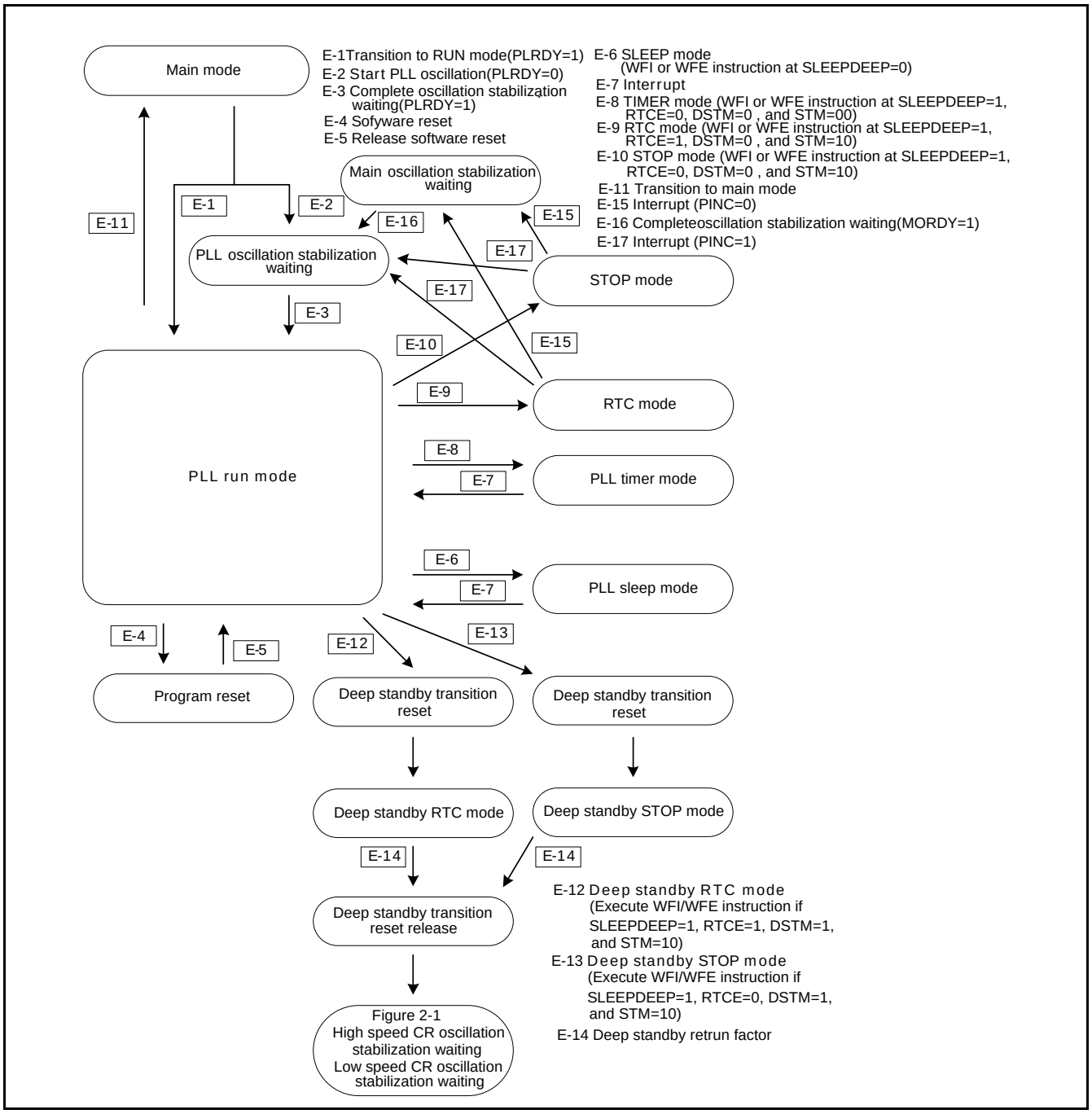
Note:

- *There are products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1 (table of low power consumption modes).*

PLL Mode Transition Diagram

In PLL mode, the main PLL clock is used as a master clock.

Figure 2-6 PLL Mode Transition Diagram



Note:

- *There are products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1 (table of low power consumption modes).*

MOSCE : MOSCE bit in System Clock Mode Control Register (SCM_CTL)
SOSCE : SOSCE bit in System Clock Mode Control Register (SCM_CTL)
PLLE : PLLE bit in System Clock Mode Control Register (SCM_CTL)
RCS : RCS bit in System Clock Mode Control Register (SCM_CTL)
MORDY : MORDY bit in System Clock Mode Status Register (SCM_STR)
SORDY : SORDY bit in System Clock Mode Status Register (SCM_STR)
PLRDY : PLRDY bit in System Clock Mode Status Register (SCM_STR)
PINC : PINC bit in PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)

*: For the SCM_CTL, SCM_STR and PSW_TMR Registers, refer to Chapter "Clock".

Note:

- *To return from low-speed CR timer mode, sub timer mode, RTC mode, STOP mode, Deep standby RTC mode, or Deep standby STOP mode, the voltage stabilization wait time (a few hundred μs) for the operation mode transition of the built-in regulator is ensured. After the wait time has lapsed, the system performs operations to return to each RUN mode.*

3. Operations of Standby Modes

This section explains operations of standby modes.
Standby modes are classified into four types: SLEEP modes (high-speed CR sleep, main sleep, PLL sleep, low-speed CR sleep, and sub sleep), TIMER modes (high-speed CR timer, main timer, PLL timer, low-speed CR timer, and sub timer), RTC mode and STOP mode.

Clock Operation States in Standby Modes

The table below shows the states of the oscillator clock, CPU clock, AHB bus clock, and APB bus clock in SLEEP, TIMER, RTC and STOP modes.

Table 3-1 Clock Operation States in SLEEP Modes

	SLEEP Modes				
	High-Speed CR Sleep Mode	**Main Sleep Mode**	**PLL Sleep Mode**	**Low-Speed CR Sleep Mode**	**Sub Sleep Mode**
High-speed CR clock	Operating	Varies depending on the setting of the HCRE bit,MCSVE bit and FCSDE bit.	Varies depending on the setting of the PINC bit, HCRE bit, MCSVE bit and FCSDE bit.	Stopped	Stopped
Main clock	Varies depending on the setting of the MOSCE bit.	Operating	Varies depending on the setting of the PINC bit and MOSCE bit.	Stopped	Stopped
Main PLL clock	Varies depending on the setting of the PLLE bit.	Varies depending on the setting of the PLLE bit.	Operating	Stopped	Stopped
Low-speed CR clock	Operating	Operating	Operating	Operating	Operating
Sub clock	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Operating
USB PLL clock	Varies depending on the setting of the UPLLEN bit.	Varies depending on the setting of the UPLLEN bit.	Varies depending on the setting of the UPLLEN bit.	Stopped	Stopped
CPU clock	Stopped				
AHB bus clock	High-speed CR clock	Main clock	PLL clock	Low-speed CR clock	Sub clock
APB0 bus clock	High-speed CR clock	Main clock	PLL clock	Low-speed CR clock	Sub clock
APB1 bus clock	High-speed CR clock	Main clock	PLL clock	Low-speed CR clock	Sub clock
	* Whether or not operation is enabled is determined depending on the setting of the APBC1EN bit.				

Table 3-2 Clock Operation States in TIMER Modes

	TIMER Modes				
	High-Speed CR Timer Mode	**Main Timer Mode**	**PLL Timer Mode**	**Low-Speed CR Timer Mode**	**Sub Timer Mode**
High-speed CR clock	Operating	Varies depending on the setting of the HCRE bit, MCSVE bit and FCSDE bit.	Varies depending on the setting of the PINC bit, HCRE bit, MCSVE bit and FCSDE bit.	Stopped	Stopped
Main clock	Varies depending on the setting of the MOSCE bit.	Operating	Varies depending on the setting of the PINC bit and MOSCE bit.	Stopped	Stopped
Main PLL clock	Varies depending on the setting of the PLLE bit.	Varies depending on the setting of the PLLE bit.	Operating	Stopped	Stopped
Low-speed CR clock	Operating	Operating	Operating	Operating	Operating
Sub clock	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Varies depending on the setting of the SOSCE bit.	Operating
USB PLL clock	Stopped				
CPU clock	Stopped				
AHB bus clock	Stopped				
APB0 bus clock	Stopped				
APB1 bus clock	Stopped				

Table 3-3 Clock Operation State in RTC Mode and STOP Mode

	RTC Mode	**STOP Mode**
High-speed CR clock	Stopped	Stopped
Main clock		
Main PLL clock		
Low-speed CR clock		
Sub clock	Operating	
USB PLL clock	Stopped	
CPU clock		
AHB bus clock		
APB0 bus clock		
APB1 bus clock		

MOSCE : MOSCE bit of System Clock Mode Control Register (SCM_CTL)
SOSCE : SOSCE bit of System Clock Mode Control Register (SCM_CTL)
PLLE : PLLE bit of System Clock Mode Control Register (SCM_CTL)

HCRE : HCRE bit of System Clock Mode Control Register (SCM_CTL)
MCSVE : MCSVE bit of CSV Control Register (CSV_CTL)
FCSDE : FCSDE bit of CSV Control Register (CSV_CTL)
PINC : PINC bit of PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)
APBC1EN : APBC1EN bit of Peripheral Bus Clock Frequency Division Register (APBC1_PSR)

*: For the SCM_CTL and APBC1_PSR Registers, refer to Chapter "Clock".

Return Factors from Standby Modes

The table below shows the factors by which the system returns from the SLEEP, TIMER, RTC and STOP modes.

Table 3-4 Return Factors from Standby Modes

	SLEEP Mode	TIMER Mode	RTC Mode	STOP Mode
Return factors by reset	− INITX pin input reset − Low-voltage detection reset − Software watchdog reset − Hardware watchdog reset − Clock failure detection reset − Anomalous frequency detection reset	− INITX pin input reset − Low-voltage detection reset − Hardware watchdog reset − Clock failure detection reset − Anomalous frequency detection reset(Main Timer Mode, PLL Timer Mode)	− INITX pin input reset − Low-voltage detection reset	− INITX pin input reset − Low-voltage detection reset
Return factors by interrupt	− Effective interrupt from each peripheral	− NMI interrupt − External interrupt − I2C Slave Interrupt − Hardware watchdog timer interrupt − USB wake up interrupt − Watch counter interrupt − RTC interrupt − HDMI-CEC/ Remote Control Reception interrupt − Low voltage detection interrupt	− NMI interrupt − External interruptt − I2C Slave Interrupt − USB wake up interrupt − RTC interrupt − HDMI-CEC/ Remote Control Reception interrupt − Low voltage detection interrupt	− NMI interrupt − External interrupt − I2C Slave Interrupt − USB wake up interrupt − Low voltage detection interrupt

3.1 Operations of SLEEP Modes (High-Speed CR Sleep, Main Sleep, PLL Sleep, Low-Speed CR Sleep, and Sub Sleep Modes)

SLEEP mode is classified as one of standby modes. Enabling SLEEP mode stops CPU clocks, reducing the power consumption.

Functions of SLEEP Mode

- CPU and on-chip memory

  In SLEEP mode, the clock supplied to the CPU is stopped. AHB bus clock continues to operate. On-chip memory keeps operating and retains the data.

- Peripherals

  The APB0 bus clock is still active in SLEEP mode. The state of the APB1 bus clock varies depending on the setting of the APBC1EN bit. Peripherals are operated in the state that is set at transition.

- Watch counter and RTC

  Watch counter and RTC remain unaffected by SLEEP mode. They continue to operate according to the setting before transiting to SLEEP mode.

- Oscillator clocks

  Table 3-1 shows the status of each oscillator clock.

- Reset and interrupt

  Reset and interrupt are available to return from SLEEP mode.

Status of Pin

All pin settings are held in SLEEP mode.

SLEEP Mode Setting Procedure

Execute the following steps to transit to SLEEP mode.

1. Set "0" to the SLEEPDEEP bit of the Cortex-M0+ System Control Register.
2. Execute the WFI or WFE instruction.
   The system transits to the appropriate SLEEP mode according to the current clock mode indicated in the RCM[2:0] bits of the System Clock Mode Status Register (SCM_STR).

For the System Clock Mode Status Register (SCM_STR), refer to Chapter "Clock".

Return from SLEEP Mode

The CPU returns from SLEEP mode in one of the following cases.

- Return by reset

If a reset (INITX pin input reset, low-voltage detection reset, software watchdog reset, hardware watchdog reset, clock supervisor reset, or anomalous frequency detection reset) occurs, the CPU changes to high-speed CR run mode regardless of clock mode.

- Return by interrupt

If an effective interrupt is received from a peripheral in SLEEP mode, the CPU returns from SLEEP mode and changes to RUN mode to fit clock mode indicated in the RCM[2:0] bits of the System Clock Mode Status Register (SCM_STR).

Table 3-5 Operation Modes after the CPU Returned from SLEEP Mode by Interrupt

	Status of Master Clock before Transition to SLEEP Mode				
	RCM=000 (High-Speed CR Oscillator)	**RCM=001 (Main Oscillator)**	**RCM=010 (PLL Oscillator)**	**RCM=100 (Low-Speed CR Oscillator)**	**RCM=101 (Sub Oscillator)**
Operation modes after return by interrupt	High-speed CR run mode	Main run mode	PLL run mode	Low-speed CR run mode	Sub run mode

RCM: RCM[2:0] bits of System Clock Mode Status Register (SCM_STR)

* For the SCM_CTL and SCM_STR Registers, refer to Chapter "Clock".

- Waiting for oscillation stabilization at return

When the CPU returns by reset, it waits for the stabilization of high and low-speed CR clock oscillations. If the CPU returns by interrupt, it does not need to wait for oscillation to stabilize.

3.2 Operations of TIMER Modes (High-Speed CR Timer, Main Timer, PLL Timer, Low-Speed CR Timer, and Sub Timer Modes)

TIMER mode is used to stop supplying a base clock. This causes the CPU clock, AHB bus clock, and all APB bus clocks to be stopped, leading to the further reduction of power consumption. In this case, all functions are stopped, excluding the oscillators, PLL, hardware watchdog timer, watch counter, RTC clock failure detector, and Low Voltage Detection Circuit.

Functions of TIMER Mode

- CPU and on-chip memory

  In TIMER mode, the CPU clock supplied to the CPU and AHB bus clock supplied to the on-chip memory and DMA controller are stopped. However, the contents of on-chip memory are retained. Also, the debug function is stopped.

- Peripherals

  In TIMER mode, all APB clocks are stopped, and all resources, excluding the hardware watchdog timer, watch counter, RTC, clock supervisor, and Low Voltage Detection Circuit, are stopped in the last state.

- Watch counter and RTC

  Watch counter and RTC remain unaffected by TIMER mode. They continue to operate according to the setting before transiting to TIMER mode.

- Oscillator clocks

  Table 3-2 shows the status of each oscillator clock.

- Reset and interrupt

  Reset and interrupt are available to return from TIMER mode.

- External bus

  The external bus is stopped in TIMER mode.

- Status of pin

  The system can control whether to retain the state just before the external pin changes to TIMER mode or change to high impedance status depending on the setting of the SPL bit in the Standby Mode Control Register (STB_CTL).

TIMER Mode Setting Procedure

Execute the following steps to transit to TIMER mode.

1. Set "0" in RTCE bit of RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC" to the KEY bit and "0" to DSTM bit and "0b00" to the STM bit of the Standby Mode Control Register (STB_CTL). Use the SPL bit to set the status of each pin in TIMER mode.
3. Set "1" to the SLEEPDEEP bit of the Cortex-M0+ System Control Register.
4. Execute the WFI or WFE instruction.
   The system transits to the appropriate TIMER mode according to the current clock mode indicated in the RCM[2:0] bits of the System Clock Mode Status Register (SCM_STR).

Note:

- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1.*

Return from TIMER Mode

The CPU returns from TIMER mode in one of the following cases.

- Return by reset

  If a reset (INITX pin input reset, low-voltage detection reset, hardware watchdog reset, clock supervisor reset, or anomalous frequency detection reset (main timer mode, PLL, TIMER mode) occurs, the CPU changes to high-speed CR run mode regardless of clock mode.

  Software watchdog reset is not available in this mode; therefore, the CPU cannot return by this reset.

- Return by interrupt

  If an effective NMI interrupt, external interrupt, hardware watchdog timer interrupt, watch counter interrupt, RTC interrupt, or low voltage detection interrupt request is received in TIMER mode, the CPU returns from TIMER mode and changes to RUN mode to fit clock mode indicated in the RCM[2:0] bits of SCM_STR register.

Table 3-6 Operation Modes after the CPU Returned from TIMER Mode by Interrupt

	Status of Master Clock before Transition to TIMER Mode				
	RCM=000 (High-Speed CR Oscillator)	**RCM=001 (Main Oscillator)**	**RCM=010 (PLL Oscillator)**	**RCM=100 (Low-Speed CR Oscillator)**	**RCM=101 (Sub Oscillator)**
Operation modes after return by interrupt	High-speed CR run mode	Main run mode	PLL run mode	Low-speed CR run mode	Sub run mode

- Waiting for oscillation stabilization at return

  When the CPU returns by reset, it waits for the stabilization of high and low-speed CR clock oscillations. If the CPU returns by interrupt, it does not need to wait for oscillation to stabilize.

- Waiting for the stabilization of the built-in regulator voltage at return

  To return from low-speed CR timer mode or sub timer mode by reset or interrupt, the voltage stabilization wait time (a few hundred μs) for the operation mode transition of the built-in regulator is ensured automatically. After the wait time has lapsed, return operations are performed.

Notes:

- *When an interrupt priority used for return is not set at a level to return the CPU, clock will be returned with the interrupt but the CPU remains stop state without returning. In order to do this, be sure to set the interrupt priority at a level which CPU is able to return.*
- *Before transiting to the timer mode, ensure that causes returning from timer modes in Table 3-4 are not set. If these interrupt causes are set, clear them.*
- *If the transition to TIMER mode is made during debugging, as the clock to the CPU stops, a return to RUN mode cannot be performed by the ICE. Use a return by reset or interrupt.*
- *In the case of transiting to the Low-speed CR timer mode or Sub timer mode, ensure that the flash memory automatic algorithm is terminated before executing transition.*

3.3 Operation of RTC Mode

RTC mode stops oscillation other than that of the sub oscillator. All the functions except for the watch counter, RTC, and low voltage detection circuit will be stopped.

Functions of RTC Mode

- CPU and on-chip memory

In RTC mode, the CPU clock supplied to the CPU and AHB bus clock supplied to the on-chip memory and DMA controller are stopped. However, the contents of on-chip memory are retained. Also, the debug function is stopped.

- Peripheral functions

In RTC mode, all APB bus clocks are stopped, and all resources, excluding the watch counter, RTC, and Low-voltage Detection Circuit, are stopped keeping the last state.

- Watch counter and RTC

Watch counter remains unaffected by RTC mode. It continues to operate according to the setting before transiting to RTC mode and it cannot be returned from RTC mode by the Watch counter interrupt.
RTC remains unaffected by RTC mode. It continues to operate according to the setting before transiting to RTC mode

- Oscillation clocks

Table 3-3 shows the status of each oscillation clock.

- Reset and interrupt

Reset and interrupt can be used to return from RTC mode.

- External bus

The external bus is stopped in RTC mode.

- Status of pin

The system can control whether to retain the status just before the external pin changes to RTC mode or changes to the high impedance status depending on the setting of the SPL bit in the Standby Mode Control Register (STB_CTL).

RTC Mode Setting Procedure

Execute the following steps to transit to RTC mode.

1. Set "1" in RTCE bit of RTC mode control register (PMD_CTL) while SORDY bit of System Clock Mode Status Register (SCM_STR) is "1".
2. Write "0x1ACC" to the KEY bit, "0" to the DSTM bit, and "0b10" to the STM bit of the Standby Mode Control Register (STB_CTL). Use the SPL bit to set the status of each pin in RTC mode.
3. Set "1" to the SLEEPDEEP bit of the Cortex-M0+ System Control Register.
4. Execute the WFI or WFE instruction.

Note:

- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby stop mode. See Table 1-1.*

Return from RTC Mode

The CPU returns from RTC mode in any one of the following cases.

- Return by reset

  If a reset (INITX pin input reset, low-voltage detection reset) occurs, the CPU changes to high-speed CR run mode regardless of clock mode.

  Software watchdog reset, hardware watchdog reset, clock supervisor reset, and anomalous frequency detection reset do not operate; therefore, the CPU cannot return by those resets.

- Return by interrupt

  If an effective NMI interrupt, external interrupt, RTC interrupt, or low voltage detection interrupt request is received in RTC mode, the CPU returns from RTC mode and transits to RUN mode to fit clock mode indicated in the RCM[2:0] bits of SCM_STR register.

Table 3-7 Operation Modes after the CPU has Returned from RTC Mode by Interrupt.

	Status of Master Clock before Transition to RTC Mode				
	RCM = 000 (High-Speed CR Oscillator)	**RCM = 001 (Main Oscillator)**	**RCM = 010 (PLL Oscillator)**	**RCM = 100 (Low-Speed CR Oscillator)**	**RCM = 101 (Sub Oscillator)**
Operation modes after return by interrupt	High-speed CR run mode	Main run mode	PLL run mode	Low-speed CR run mode	Sub run mode

- Waiting for oscillation stabilization at return

  When the CPU returns by reset, it waits for the stabilization of high and low-speed CR clock oscillations.

  If the CPU returns by interrupt, the oscillation stabilization wait changes by the master clock before transition to the RTC mode as shown in Table 3-8.

Table 3-8 Oscillation Stabilization Wait when Returning by Interrupt from RTC Mode

		Status of Master Clock before Transition to RTC Mode				
		RCM = 000 (High-Speed CR Oscillator)	**RCM = 001 (Main Oscillator)**	**RCM = 010 (PLL Oscillator)**	**RCM = 100 (Low-Speed CR Oscillator)**	**RCM = 101 (Sub Oscillator)**
Oscillation stabilization wait after returning by interrupt	High-speed CR clock	ON	HCR="enable": ON*1 Others: OFF	PINC="1" or HCR="enable": ON*1 Others: OFF	OFF	OFF
	Main clock	MOSCE="0": OFF MOSCE="1": ON	ON	PINC="0" or MOSCE="1": ON Others: OFF	OFF	OFF
	Main PLL clock	PLLE="0": OFF PLLE="1": ON	PLLE="0": OFF PLLE="1": ON	ON	OFF	OFF
	Low-speed CR clock	ON	ON	ON	ON	ON
	Sub clock	OFF*2	OFF*2	OFF*2	OFF*2	OFF*2

*1: HCR is defined as enabled when HCRE ="1" or MCSVE="1" or FCSDE="1".

*2: TYPE1-M0+/TYPE2-M0+/TYPE3-M0+ products have the oscillation stabilization wait time, but the actual oscillation continues. CSW_TMR:SOWT bit is possible to set minimum wait time "1100".

■ Waiting for the stabilization of the built-in regulator voltage at return

To return from RTC mode, the voltage stabilization wait time (a few hundred μs) for the operation mode transition of the built-in regulator is ensured automatically. After the wait time has lapsed, return operations are performed.

Notes:

- *When an interrupt priority used for return is not set at a level to return the CPU, the clock will be returned by the interrupt but the CPU remains in stop state without returning. In order to do this, be sure to set the interrupt priority at a level which the CPU is able to return.*
- *Before transiting to the RTC mode, ensure that the causes of return from timer mode in Table 3-4 are not set. If these return causes are set, clear them.*
- *If the transition to RTC mode is made during debugging, as the clock to the CPU stops, a return to RUN mode cannot be performed by the ICE. Use a return by reset or interrupt.*
- *In case of transiting to the RTC mode, ensure that the flash memory automatic algorithm is terminated before executing the transition.*

3.4 Operations of STOP Mode

STOP mode is used to stop all oscillating operations. Enabling this mode stops all functions, excluding the Low Voltage Detection Circuit.

Functions of STOP Mode

- CPU and on-chip memory

  In STOP mode, the CPU clock supplied to the CPU and AHB bus clock supplied to the on-chip memory and DMA controller are stopped. However, the contents of on-chip memory are retained. The debug function is stopped.

- Peripheral functions

  All APB bus clocks are stopped, and all resources, excluding the Low Voltage Detection Circuit, are stopped in the last state.

- Oscillator clocks

  All oscillator clocks are stopped.

- Reset and interrupt

  Reset and interrupt are available to return from STOP mode.

- External bus

  The external bus is stopped in STOP mode.

- Status of pin

  The system can control whether to retain the state just before the external pin changes to STOP mode or change to high impedance status depending on the setting of the SPL bit in the Standby Mode Control Register (STB_CTL).

STOP Mode Setting Procedure

Execute the following steps to transit to STOP mode.

1. Set "0" in RTCE bit of RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC" to the KEY bit and "0" to DSTM bit and "0b10" to the STM bit of the Standby Mode Control Register (STB_CTL). Use the SPL bit to set the status of each pin in STOP mode.
3. Set "1" to the SLEEPDEEP bit of the Cortex-M0+ System Control Register.
4. Execute the WFI or WFE instruction.

Note:

- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1.*

Return from STOP Mode

The CPU returns from STOP mode in one of the following cases.

- Return by reset

  If a reset (INITX pin input reset or low-voltage detection reset) occurs, the CPU changes to the high-speed CR run mode regardless of clock mode.

  Software watchdog reset, hardware watch dog reset, clock supervisor reset, and anomalous frequency detection reset are not available in this mode; therefore, the CPU cannot return by those resets.

- Return by interrupt

  If an effective NMI interrupt, external interrupt, or low voltage detection interrupt request is received in STOP mode, the CPU returns from STOP mode and changes to RUN mode to fit clock mode indicated in the RCM[2:0] bits of SCM_STR register.

Table 3-9 Operation Modes after the CPU Returned from the STOP Mode by Interrupt

	Status of Master Clock before Changing to STOP Mode				
	RCM=000 (High-Speed CR Oscillator)	**RCM=001 (Main Oscillator)**	**RCM=010 (PLL Oscillator)**	**RCM=100 (Low-Speed CR Oscillator)**	**RCM=101 (Sub Oscillator)**
Operation modes after return by interrupt	High-speed CR run mode	Main run mode	PLL run mode	Low-speed CR run mode	Sub run mode

- Waiting for oscillation stabilization at return

  When the CPU returns by reset, it waits for the stabilization of high and low-speed CR clock oscillations. If the CPU returns by interrupt, the oscillation stabilization wait state varies depending on the master clock that is output before the CPU changes to STOP mode as shown in Table 3-10.

Table 3-10 Waiting for Oscillation to Stabilize at Return from STOP Mode by Interrupt

		Status of Master Clock before Changing to STOP Mode				
		RCM=000 (High-Speed CR Oscillator)	**RCM=001 (Main Oscillator)**	**RCM=010 (PLL Oscillator)**	**RCM=100 (Low-Speed CR Oscillator)**	**RCM=101 (Sub Oscillator)**
Oscillation stabilization waiting after return by interrupt	High-speed CR clock	ON	HCR="enable": ON* Others: OFF	PINC="1" or HCR="enable": ON* Others: OFF	OFF	OFF
	Main clock	MOSCE="0": OFF MOSCE="1": ON	ON	PINC="0" or MOSCE="1": ON Others: OFF	OFF	OFF
	Main PLL clock	PLLE="0": OFF PLLE="1": ON	PLLE="0": OFF PLLE="1": ON	ON	OFF	OFF
	Low-speed CR clock	ON	ON	ON	ON	ON
	Sub clock	SOSCE="0": OFF SOSCE="1": ON	SOSCE="0": OFF SOSCE="1": ON	SOSCE="0": OFF SOSCE="1": ON	SOSCE="0": OFF SOSCE="1": ON	ON

*: HCR is defined as enabled when HCRE ="1" or MCSVE="1" or FCSDE="1".

■ Waiting for the stabilization of the built-in regulator voltage at return

When the CPU returns from STOP mode, the voltage stabilization wait time (a few hundred μs) for the operation mode transition of the built-in regulator is ensured automatically. After the wait time has lapsed, return operations are performed.

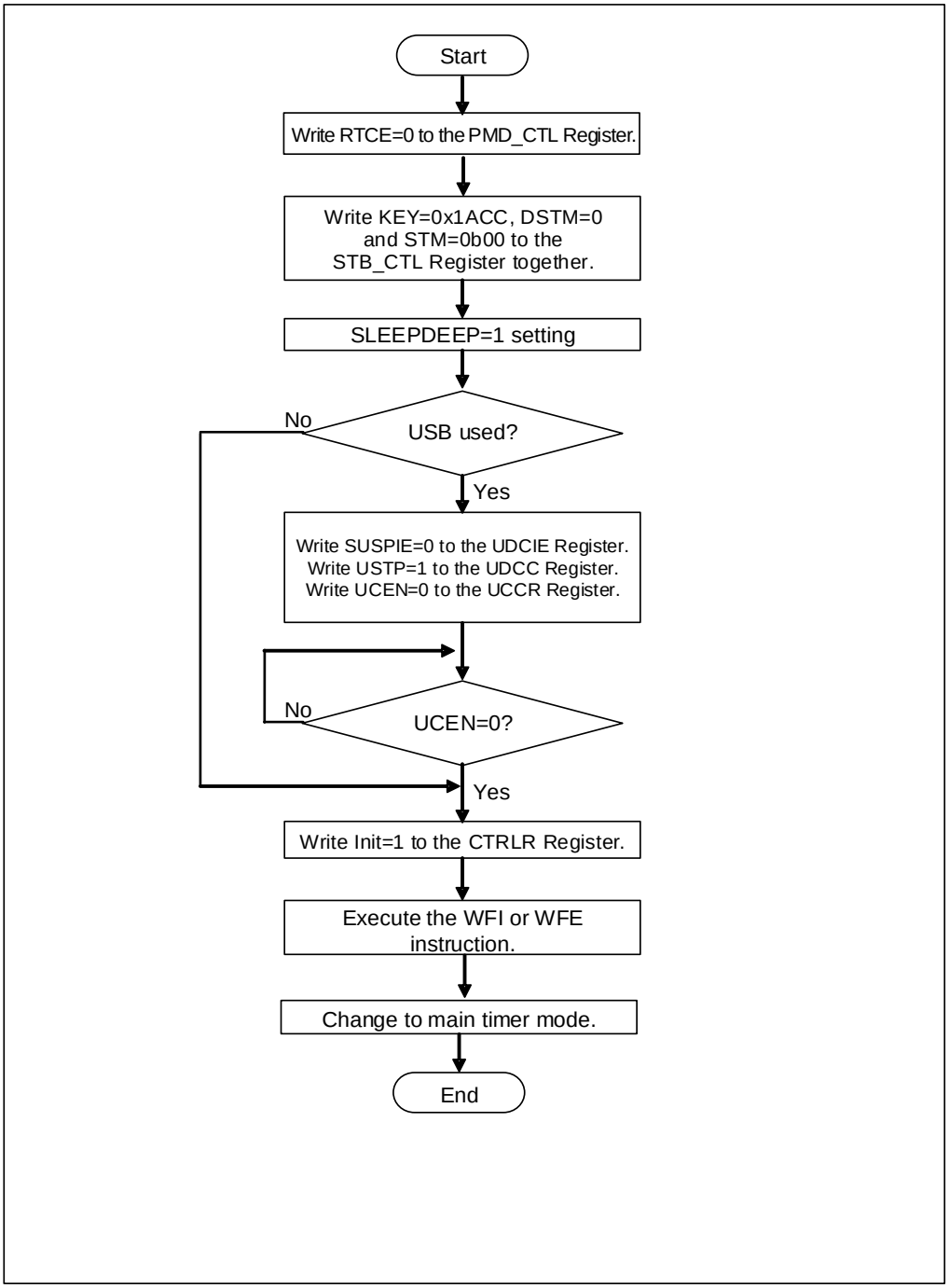
Notes:

- *When an interrupt priority used for return is not set at a level to return the CPU, clock will be returned with the interrupt but the CPU remains stop state without returning. In order to do this, be sure to set the interrupt priority at a level which CPU is able to return.*
- *Before transiting to the stop mode, ensure causes of return timer modes in Table 3-4 are not set. If these interrupt causes are set, clear them.*
- *If the transition to STOP mode is made during debugging, as the clock to the CPU stops, a return to RUN mode cannot be performed by the ICE. Use a return by reset or interrupt.*
- *In case of transiting to the STOP mode, ensure that the flash memory automatic algorithm is terminated before executing transition.*

4. Standby Mode Setting Procedure Examples

This section provides standby mode setting procedure examples.

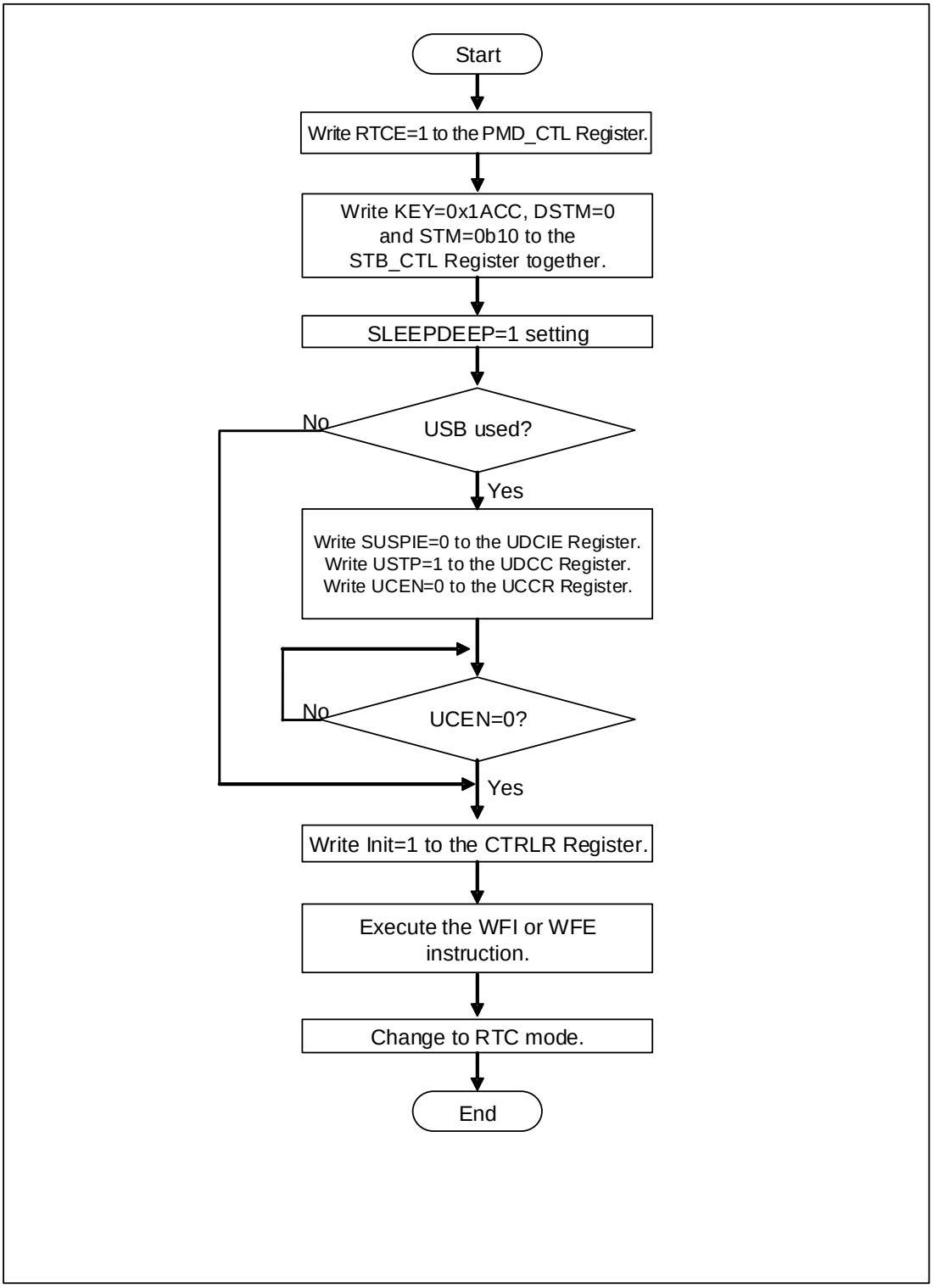
Figure 4-1 Main Timer Mode Setting Procedure Example



Note:

- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1.*

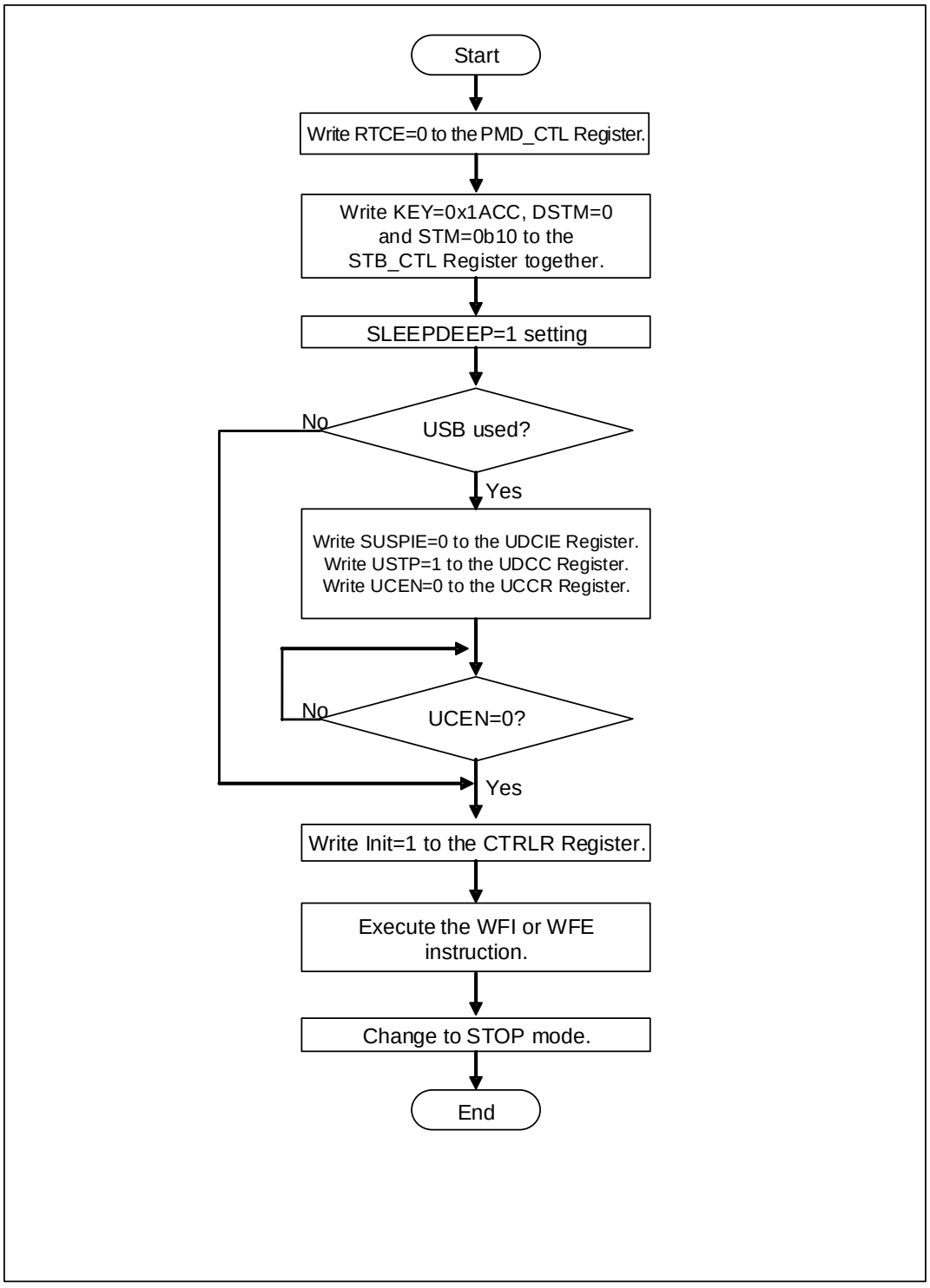
Figure 4-2 RTC Mode Setting Procedure Example (Main Clock is Selected as a Master Clock)



Notes:

- *In case of transiting to the RTC mode, ensure that the flash memory automatic algorithm is terminated before executing the transition.*
- *Writing "1" to RTCE bit of the RTC Mode Control Register (PMD_CTL) is valid only when the SORDY bit of the System Clock Mode Status Register (SCM_STR) is "1".*
- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1.*

Figure 4-3 STOP Mode Setting Procedure Example (Main Clock is Selected as a Master Clock)



Notes:

- *In case of transiting to the STOP mode, ensure that the flash memory automatic algorithm is terminated before executing the transition.*
- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1.*

5. Description of Deep Standby Mode Operation

This section describes the operation of deep standby mode.
Deep standby mode includes deep standby RTC mode and deep standby STOP mode.

5.1 Operation of Deep Standby Mode

5.1.1 Clock Operation Status in Deep Standby Mode

The following shows the status of the oscillation clock, CPU clock, AHB bus clock, and APB bus clock while in deep standby RTC mode and deep standby STOP mode.

Table 5-1 Clock Operation State in Deep Standby Mode

	Deep Standby RTC Mode	Deep Standby STOP Mode
High-speed CR clock	Stopped	Stopped
Main clock		
Main PLL clock		
Low-speed CR clock		
Sub clock	Operating	
USB PLL clock	Stopped	
CPU clock		
AHB bus clock		
APB0 bus clock		
APB1 bus clock		

5.1.2 Return Factors from Deep Standby Mode

The following shows the return factors from deep standby RTC mode and deep standby STOP mode.

Table 5-2 Return Factors from Deep Standby Mode

	Deep Standby RTC Mode	Deep Standby STOP Mode
Deep standby return factor	− INITX pin input reset − Low-voltage detection reset	− INITX pin input reset − Low-voltage detection reset
	− Low-voltage detection interrupt − RTC interrupt − HDMI-CEC/ Remote Control Reception interrupt	− Low-voltage detection interrupt
	− WKUP pin input	− WKUP pin input

Note:

- *Although each interrupt factor is retained after returning from deep standby mode, interrupt processing will not be executed since NVIC is initialized by deep standby transition reset.*

5.1.3 Internal Power Supply Status and Reset Status in Deep Standby Mode

The following shows the power supply status of each function in deep standby mode and initialization status in deep standby transition reset.

Table 5-3 Internal Power Supply Status and Initialization Status in Deep Standby Mode

	Product TYPE	Power Supply Status	Reset Status
CPU	TYPE2-M0+	Off	Initialized
On-chip Flash		Off	*1
On-chip SRAM		Off *2	*3
RTC		On	Not initialized
HDMI-CEC/ Remote Control Reception		On	Not initialized
Low-voltage detection circuit		On	Not initialized
GPIO		On	Partly initialized *4
Deep standby control block		On	Not initialized
Peripheral functions other than the above		Off	Initialized
CPU	TYPE3-M0+	Off	Initialized
On-chip Flash		Off	*1
On-chip SRAM		Off *2*5	*3
RTC		On	Not initialized
HDMI-CEC/ Remote Control Reception		On	Not initialized
Low-voltage detection circuit		On	Not initialized
GPIO		Off	Initialized
Deep standby control block		On	Not initialized
Peripheral functions other than the above		Off	Initialized

*1: The contents of on-chip Flash memory are retained.

*2: The contents of on-chip SRAM can be retained.
In the setting to retain on-chip SRAM data, the power of on-chip SRAM is ON.

*3: The contents of on-chip SRAM are not retained when the power is OFF.
In the setting to retain on-chip SRAM data, on-chip SRAM data is retained.

*4: PFRx registers excluding bit4:0 and CEC of PFR0 are initialized and others are not initialized.

*5: If the setting for retaining data in SRAM is done, data in address range of 0x2000_3000~0x2000_3FFF is retained.

5.2 Operation of Deep Standby RTC Mode

Deep standby RTC mode stops oscillation other than that of the sub oscillator. All the functions except for RTC, HDMI-CEC/Remote Control Reception and low voltage detection circuit will be stopped. It turns off CPUs, on-chip Flash memory, on-chip SRAM*, and peripheral functions inside the chip excluding RTC, HDMI-CEC/Remote Control Reception, the low-voltage detection circuit and GPIO.

Functions of Deep Standby RTC Mode

- CPU and on-chip memory

In deep standby RTC mode, the CPU clock supplied to CPU and AHB bus clock supplied to on-chip memory and DMA controller are stopped. It turns off the CPU, on-chip Flash memory, and on-chip SRAM*. The contents of the CPU register and on-chip SRAM are not retained*. The contents of on-chip flash memory are retained. Also, the debug function is stopped and turned off.

*: Data in on-chip SRAM can be retained.
In the setting to retain on-chip SRAM data, the power of on-chip SRAM is ON.

- Peripheral functions

All APB bus clocks are stopped, and RTC, HDMI-CEC/ Remote Control Reception, Low-voltage Detection Circuit, and all resources excluding GPIO, are turned off.

- RTC, HDMI-CEC/ Remote Control Reception

RTC and HDMI-CEC/ Remote Control Reception remains unaffected by deep standby RTC mode. It continues to operate according to the setting before transiting to deep standby RTC mode.

- Oscillation clock

The status of each oscillation clock is shown in Table 5-1.

- Reset, interrupt, and WKUP pin input

Reset, interrupt, and WKUP pin input can be used for returning from deep standby RTC mode.

- Status of pin

In deep standby RTC mode, the system can control whether the external pin switches to GPIO or to high impedance status by the SPL bit in the Standby Mode Control Register (STB_CTL).

Setting Procedure of Deep Standby RTC Mode

Execute the following steps to transit to deep standby RTC mode.

1. Set "1" in RTCE bit of the RTC Mode Control Register (PMD_CTL) while the SORDY bit of the System Clock Mode Status Register (SCM_STR) is "1".
2. Write "0x1ACC" to the KEY bit, "1" to the DSTM bit, and "0b10" to the STM bit of the Standby Mode Control Register (STB_CTL). Use the SPL bit to set the status of each pin in deep standby RTC mode.
3. Set "1" to the SLEEPDEEP bit of the Cortex-M0+ System Control Register.
4. Execute the WFI or WFE instruction.

Note:

- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby stop mode. See Table 1-1.*

Return from Deep Standby RTC Mode

CPU returns from deep standby RTC mode in any one of the following cases.

- Return by reset, interrupt, and WKUP pin input

If a reset (INITX pin input reset and low-voltage detection reset) occurs or if effective RTC interrupt, HDMI-CEC/ Remote Control Reception interrupt, low-voltage detection interrupt, and WKUP pin input request are received while in deep standby RTC mode, the CPU returns from deep standby RTC mode and changes to high-speed CR run mode regardless of clock mode by deep standby transition reset occurrence.

Software watchdog reset, hardware watchdog reset, clock supervisor reset, and anomalous frequency detection reset do not operate; therefore, the CPU cannot return by those resets.

- Waiting for oscillation stabilization at return

The oscillation stabilization wait for the high-speed CR clock and low-speed CR clock is executed regardless of return factor.

- Waiting for the stabilization of the built-in regulator voltage at return

To return from deep standby RTC mode, the voltage stabilization wait time (a few hundred μs) for the operation mode transition of the built-in regulator is ensured automatically. After the wait time has lapsed, return operations are performed.

Notes:

- *Before transiting to the deep standby RTC mode, ensure that the return factor from deep standby RTC mode in Table 5-2 is not set. If the factor is set, clear it.*
- *If the transition to deep standby RTC mode is made during debugging, a return to RUN mode cannot be performed by the ICE as the debug function is turned off. Use a return by reset, interrupt, or WKUP pin input.*
- *In the case of transiting to the deep standby RTC mode, ensure that the flash memory automatic algorithm is terminated before executing the transition.*

5.3 Operation of Deep Standby Stop Mode

Deep standby stop mode stops all oscillations. All the functions except for the low voltage detection circuit will be stopped. It turns off, CPUs, on-chip Flash memory, on-chip SRAM*, and peripheral functions, inside the chip excluding RTC, HDMI-CEC/Remote Control Reception, the low-voltage detection circuit and GPIO.

Functions of Deep Standby STOP Mode

- CPU and on-chip memory

In deep standby STOP mode, the CPU clock supplied to CPU and AHB bus clock supplied to on-chip memory and DMA controller are stopped. It turns off the CPU and on-chip Flash, on-chip SRAM*. The contents of the CPU register and on-chip SRAM are not retained*. The contents of on-chip flash memory are retained. Also, the debug function is stopped and turned off.

*: Data in on-chip SRAM can be retained.
In the setting to retain on-chip SRAM data, the power of on-chip SRAM is ON.

- Peripherals

All APB bus clocks are stopped, and RTC, HDMI-CEC/ Remote Control Reception, Low-voltage Detection Circuit, and all resources excluding GPIO, are turned off.

- Oscillation clock

All oscillations are stopped.

- Reset and WKUP pin input

Reset and WKUP pin input can be used for returning from deep standby STOP mode.

- Status of pin

The system can control whether the external pin switches to GPIO in deep standby STOP mode or to high impedance status by the SPL bit in the Standby Mode Control Register (STB_CTL).

Setting Procedure of Deep Standby STOP Mode

Execute the following steps to transit to deep standby STOP mode.

1. Set "0" in RTCE bit of the RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC" to the KEY bit, "1" to the DSTM bit, and "0b10" to the STM bit of the Standby Mode Control Register (STB_CTL). Use the SPL bit to set the status of each pin in deep standby STOP mode.
3. Set "1" to the SLEEPDEEP bit of the Cortex-M0+ System Control Register.
4. Execute the WFI or WFE instruction.

Note:

- *DSTM bit is not in the products that do not equip deep standby RTC mode, and deep standby STOP mode. See Table 1-1.*

Return from Deep Standby STOP Mode

CPU returns from deep standby STOP mode in any one of the following cases.

- Return by reset, interrupt, and WKUP pin input

  If a reset (INITX pin input reset and low-voltage detection reset) occurs, or if effective low-voltage detection interrupt, or WKUP pin input request is received while in deep standby STOP mode, the CPU returns from deep standby STOP mode and changes to high-speed CR run mode regardless of clock mode by deep standby transition reset.

  Software watchdog reset, hardware watchdog reset, clock supervisor reset, and anomalous frequency detection reset do not operate; therefore, the CPU cannot return by those resets.

- Waiting for oscillation stabilization at return

  The oscillation stabilization wait for the high-speed CR clock and low-speed CR clock is executed regardless of return factor.

- Waiting for the stabilization of the built-in regulator voltage at return

  To return from deep standby STOP mode, the voltage stabilization wait time (a few hundred μs) for the operation mode transition of the built-in regulator is ensured automatically. After the wait time has lapsed, return operations are performed.

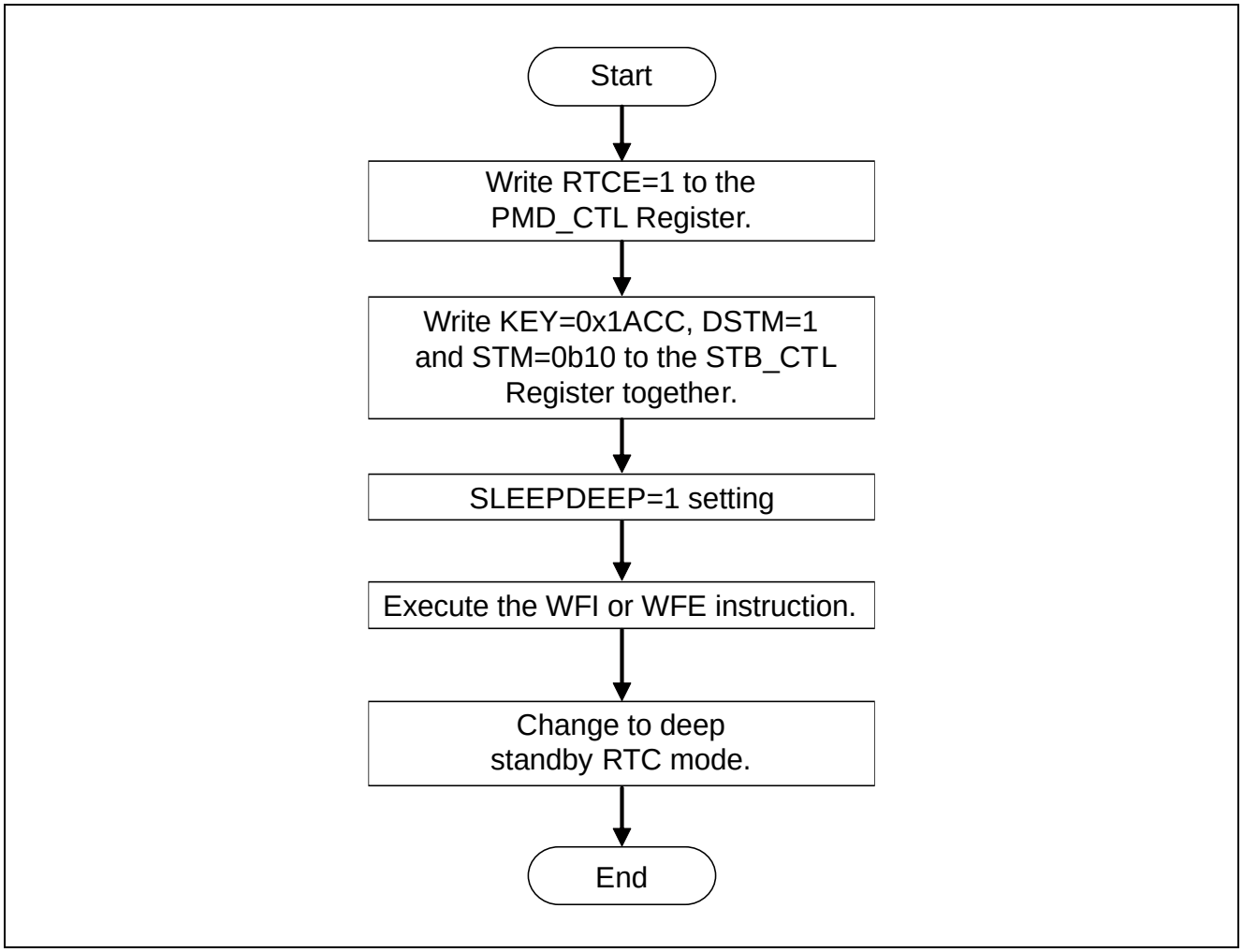
Notes:

- *Before transiting to the deep standby RTC mode, ensure that the return factor from deep standby RTC mode in Table 5-2 is not set. If the factor is set, clear it.*
- *If the transition to deep standby stop mode is made during debugging, a return to RUN mode cannot be performed by the ICE as the debug function turns off. Use a return by reset, interrupt, or WKUP pin input.*
- *In the case of transiting to the deep standby STOP mode, ensure that the flash memory automatic algorithm is terminated before executing the transition.*

6. Deep Standby Mode Setting Procedure Examples

This section explains the deep standby mode setting procedure examples.

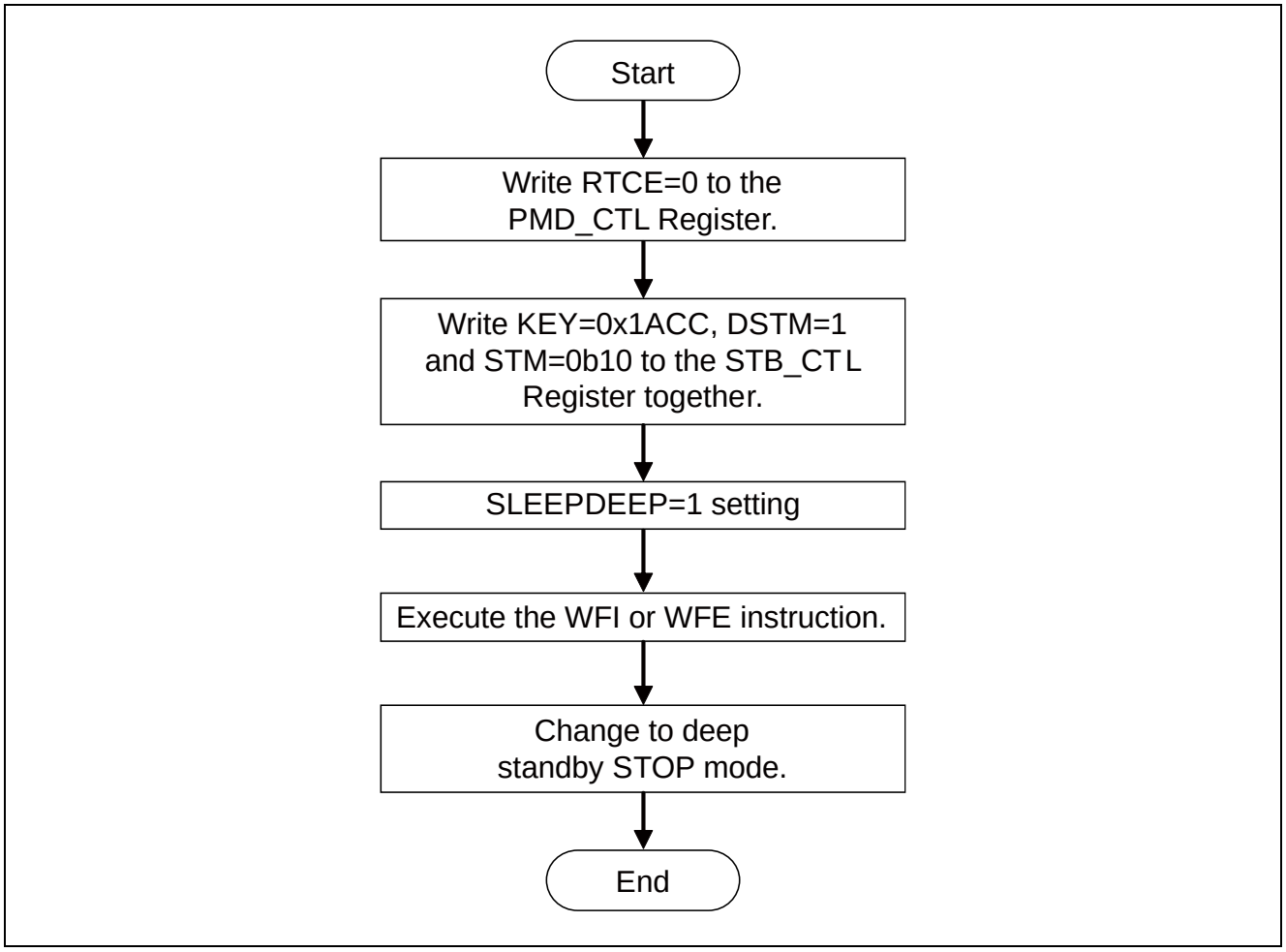
Figure 6-1 Setting Procedure Example for Deep Standby RTC Mode



Notes:

- *In case of transiting to the deep standby RTC mode, ensure that the flash memory automatic algorithm is terminated before executing the transition.*
- *Writing "1" to RTCE bit of the RTC Mode Control Register (PMD_CTL) is valid only when the SORDY bit of the System Clock Mode Status Register (SCM_STR) is "1".*

Figure 6-2 Setting Procedure Example for Deep Standby STOP Mode



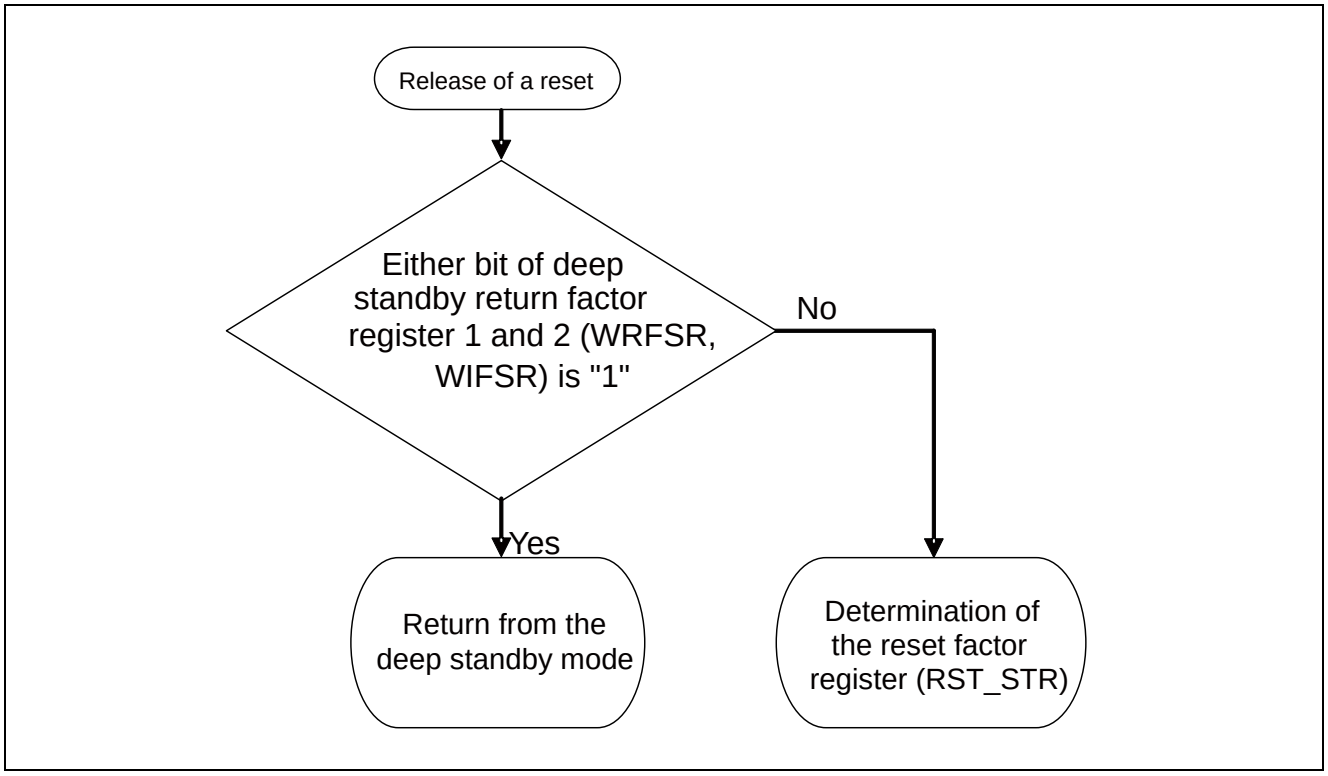
Note:

- *In case of transiting to the deep standby STOP mode, ensure that the flash memory automatic algorithm is terminated before executing the transition.*

7. Deep Standby Return Factor Determination Procedure

Figure 7-1 shows a procedure example to determine a return from deep standby mode.

Figure 7-1 Procedure Example for Deep Standby Return Factor Determination



Note:

- *At the transition to deep standby mode, the power supply of the CPU is turned off after deep standby transition reset. Therefore, the value of the reset factor register (RST_STR) is invalid when returning from deep standby mode.*

8. Registers

This section describes the registers used in low power consumption mode.

List of Low Power Consumption Mode Register

Table 8-1 List of Low Power Consumption Mode Register

Abbreviation	Register Name	Reference
STB_CTL	Standby Mode Control Register	8.1

■ List of registers of deep standby control block

Table 8-2 List of Registers of Deep Standby Control Block

Abbreviation	Register Name	Reference
REG_CTL	Sub Oscillation Circuit Power Supply Control Register	8.2
RCK_CTL	Sub Clock Control Register	8.3
PMD_CTL	RTC Mode Control Register	8.4
WRFSR	Deep Standby Return Factor Register 1	8.5
WIFSR	Deep Standby Return Factor Register 2	8.6
WIER	Deep Standby Return Enable Register	8.7
WILVR	WKUP Pin Input Level Register	8.8
DSRAMR	Deep Standby RAM Retention Register	8.9
BUR01 to 16	Backup Registers 01 to 16	8.10
MOSC_CTL	Mainosc Driving Level Control Register	8.11
WIOLC_CTL	IO State control Register	8.12
SUBOSC_CTL	Sub Oscillator IO Control Register	8.13
CEC_CTL	CEC IO Control Register	8.14
DEBUG_SW_CTL	Serial Wire Debug Control Register	8.15

■ Table of registers for low power consumption modes equipped in each TYPE

Table 8-3 Table of Registers for Low Power Consumption Mode

Register Name	TYPE1-M0+	TYPE2-M0+	TYPE3-M0+
STB_CTL	○	○	○
REG_CTL	○	-	○
RCK_CTL	○	○	○
PMD_CTL	○	○	○
WRFSR	-	○	○
WIFSR	-	○	○
WIER	-	○	○
WILVR	○	○	○
DSRAMR	-	○	○
BUR01 to 16	-	○	○
MOSC_CTL	-	○	○
WIOLC_CTL	-	-	○
SUBOSC_CTL	-	-	○
CEC_CTL	-	-	○
DEBUG_SW_CTL	-	-	○

Note:

- *For the System Clock Mode Control Register (SCM_CTL), refer to Chapter "Clock".*
  Registers of the deep standby control block are not turned off in deep standby mode.

8.1 Standby Mode Control Register (STB_CTL)

The Standby Mode Control Register (STB_CTL) controls standby mode and deep standby mode. The value written to the SPL, DSTM or STM bit is effective only when 0x1ACC is simultaneously written to the KEY bit.

Register Configuration

bit	31															16	15							8
Field	KEY																Reserved							
Attribute	R/W																-							
Initial value	0x0000																0x00							

bit	7	6	5	4	3	2	1	0
Field	Reserved			SPL	Reserved	DSTM	STM	
Attribute	-			R/W	-	R/W	R/W	
Initial value	000			0	0	0	00	

Register Functions

[bit31:16] KEY: Standby mode control write control bits

These bits release the SPL bit, DSTM bit or STM bit writing control.

- The value written to the SPL bit, DSTM bit or STM bit is effective only when 0x1ACC is written to the KEY bit.
- If a value other than 0x1ACC is written to the KEY bit, the value written to the SPL bit, DSTM bit or STM bit is not effective.
- 0x0000 is always read in read mode.

[bit15:5] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit4] SPL: Standby pin level setting bit

This bit sets the status of pin in TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, or Deep standby STOP mode.

Bit	Description
0	Retains status of each pin in TIMER mode, RTC mode, and STOP mode and switches to GPIO in deep standby RTC mode and deep standby stop mode. [Initial value]
1	Sets the status of each pin to high impedance in TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, or Deep standby STOP mode.

[bit3] Reserved: Reserved bit

The read value is always "0". Writing has no effect on the operation.

[bit2] DSTM: Deep standby mode select bit

This bit selects transiting to either standby mode or deep standby mode.

[bit1:0] STM: Standby mode select bits

These bits are a combination of DSTM bit and RTCE bit of the RTC Mode Control Register (PMD_CTL) and select transiting to any one of the following: TIMER mode, RTC mode, STOP mode, deep standby RTC mode, and deep standby STOP mode.

DSTM	STM		PMD_CTL: RTCE	Description
	bit1	bit0		
0	0	0	0	TIMER mode [initial value]
0	0	0	1	Setting is prohibited.
0	0	1	0	Setting is prohibited
0	0	1	1	Setting is prohibited
0	1	0	0	STOP mode
0	1	0	1	RTC mode
0	1	1	0	Setting is prohibited
0	1	1	1	Setting is prohibited
1	0	0	0	Setting is prohibited
1	0	0	1	Setting is prohibited
1	0	1	0	Setting is prohibited
1	0	1	1	Setting is prohibited
1	1	0	0	Deep standby STOP mode
1	1	0	1	Deep standby RTC mode
1	1	1	0	Setting is prohibited
1	1	1	1	Setting is prohibited

Notes:

- *The written value to SPL bit, DSTM bit, STM bit in the Standby Mode Control Register (STB_CTL) is valid only when "0x1ACC" is written to KEY bit at the same time. If a value other than "0x1ACC" is written to KEY bit, writing to SPL bit, DSTM bit, and STM bit becomes invalid.*

8.2 Sub Oscillation Circuit Power Supply Control Register (REG_CTL)

The Sub Oscillation Circuit Power Supply Control Register (REG_CTL) controls the power supply for sub oscillation circuit.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					ISUBSEL		Reserved
Attribute			-			R/W		-
Initial value			00000			10		0

Register Functions

[bit7:3] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit2:1] ISUBSEL: Sub oscillation circuit current setting bits

These bits set the current to sub oscillation circuit.

bit1	bit0	Description
0	0	Setting is prohibited
0	1	Setting is prohibited
1	0	360nA [initial value]
1	1	Setting is prohibited

[bit0] Reserved: Reserved bit

The read value is always "0". Writing has no effect on the operation.

Notes:

- *This register is not initialized by software reset or deep standby transition reset.*
- *This Register is not available for TYPE2-M0+.*

8.3 Sub Clock Control Register (RCK_CTL)

The Sub Clock Control Register (RCK_CTL) controls the clock to RTC, HDMI-CEC/remote control reception.

Power consumption can be reduced by stopping the clock supply to unused resource.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						CECCKE	RTCCKE
Attribute	-						R/W	R/W
Initial value	000000						0	1

Register Functions

[bit7:2] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit1] CECCKE: CEC clock control bit

This bit controls sub clock for HDMI-CEC/remote control reception macro.

Bit	Description
0	Sub clock is not supplied to HDMI-CEC/remote control reception macro. [Initial value]
1	Sub clock is supplied to HDMI-CEC/remote control reception macro.

[bit0] RTCCKE: RTC clock control bit

This bit controls sub clock for RTC macro.

Bit	Description
0	Sub clock is not supplied to RTC macro.
1	Sub clock is supplied to RTC macro. [Initial value]

8.4 RTC Mode Control Register (PMD_CTL)

The RTC Mode Control Register (PMD_CTL) controls either RTC mode or STOP mode and either deep standby RTC mode or deep standby STOP mode.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							RTCE
Attribute				-				R/W
Initial value				0000000				0

Register Functions

[bit7:1] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit0] RTCE: RTC mode control bit

This bit selects transiting to either RTC mode or STOP mode and either deep standby RTC mode or deep standby stop mode.

Bit	Description
0	STOP mode and deep standby stop mode [initial value]
1	RTC mode and deep standby RTC mode

Standby mode is selected when DSTM bit is "0" and deep standby mode is selected when DSTM bit is "1".

Notes:

- *This register is not initialized by software reset and deep standby transition reset.*
- *Writing "1" to RTCE bit is valid only when the SORDY bit of the System Clock Mode Status Register (SCM_STR) is "1".*
- *Sub oscillation is enabled when RTCE bit is "1" regardless of the SOSCE bit value of System Clock Mode Control Register (SCM_CTL) and the SORDY bit value of the System Clock Mode Status Register (SCM_STR).*

8.5 Deep Standby Return Factor Register 1 (WRFSR)

The Deep Standby Return Factor Register 1 (WRFSR) indicates return factors by low-voltage detection reset and the INITX pin input reset that occur in deep standby mode.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						WLVDH	WINITX
Attribute	-						R	R
Initial value	000000						0	0

Register Functions

[bit7:2] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit1] WLVHD: Low-voltage detection reset return bit

This bit indicates returning from deep standby mode by low-voltage detection reset.

Bit	Description
0	Not returned by low-voltage detection reset [initial value]
1	Returned by low-voltage detection reset

[bit0] WINITX: INITX pin input reset return bit

This bit indicates returning from deep standby mode by INITX pin input reset.

Bit	Description
0	Not returned by INITX pin input reset [initial value]
1	Returned by INITX pin input reset

Notes:

- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by other reset factors. In addition, all bits are cleared by reading.*
- *Before transiting to the deep standby mode, ensure that the return factor from the deep standby mode is not set. If the factor is set, clear it.*
- *This register can be set only in the deep standby mode.*

8.6 Deep Standby Return Factor Register 2 (WIFSR)

The Deep Standby Return Factor Register 2 (WIFSR) indicates return factors by WKUPx pin input, low-voltage detection interrupt, and RTC interrupt, and HDMI-CEC/Remote Control Reception interrupt that occur in deep standby mode.

Register Configuration

bit	15	14	13	12	11	10	9	8
Field	WUI11	WUI10	WUI9	WUI8	WUI7	WUI6	WCEC1I	WCEC0I
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	WUI5	WUI4	WUI3	WUI2	WUI1	WUI0	WLVDI	WRTCI
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Register Functions

[bit15:10] WUI11 to WUI6: WKUPx pin input return bits

These bits indicate returning from deep standby mode by WKUPx pin input.

Bit	Description
0	Not returned by WKUPx pin input [initial value]
1	Returned by WKUPx pin input

[bit9] WCEC1I: CEC ch.1 interrupt return bit

This bit indicates returning from deep standby mode by HDMI-CEC/ Remote Control Reception ch.1 interrupt.

Bit	Description
0	Not returned by HDMI-CEC/ Remote Control Reception ch.1 interrupt [initial value]
1	Returned by HDMI-CEC/ Remote Control Reception ch.1 interrupt

[bit8] WCEC0I: CEC ch.0 interrupt return bit

This bit indicates returning from deep standby mode by HDMI-CEC/ Remote Control Reception ch.0 interrupt.

Bit	Description
0	Not returned by HDMI-CEC/ Remote Control Reception ch.0 interrupt [initial value]
1	Returned by HDMI-CEC/ Remote Control Reception ch.0 interrupt

[bit7:2] WUI5 to WUI0: WKUPx pin input return bits

These bits indicate returning from deep standby mode by WKUPx pin input.

Bit	Description
0	Not returned by WKUPx pin input [initial value]
1	Returned by WKUPx pin input

[bit1] WLVDI: LVD interrupt return bit

This bit indicates returning from deep standby mode by LVD interrupt.

Bit	Description
0	Not returned by LVD interrupt [initial value]
1	Returned by LVD interrupt

[bit0] WRTCI: RTC interrupt return bit

This bit indicates returning from deep standby mode by RTC interrupt.

Bit	Description
0	Not returned by RTC interrupt [initial value]
1	Returned by RTC interrupt

Notes:

- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by other reset factor. In addition, all bits are cleared by reading.*
- *Before transiting to the deep standby mode, ensure that the return factor from deep standby mode is not set. If the factor is set, clear it.*
- *This register can be set only in the deep standby mode.*

8.7 Deep Standby Return Enable Register (WIER)

The Deep Standby Return Enable Register (WIER) enables a return by WKUPx pin input, low-voltage detection interrupt, RTC interrupt, and HDMI-CEC/Remote Control Reception interrupt that occur in deep standby mode.

Register Configuration

bit	15	14	13	12	11	10	9	8
Field	WUI11E	WUI10E	WUI9E	WUI8E	WUI7E	WUI6E	WCEC1E	WCEC0E
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	WUI5E	WUI4E	WUI3E	WUI2E	WUI1E	Reserved	WLVDE	WRTCE
Attribute	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Register Functions

[bit15:10] WUI11E to WUI6E: WKUPx pin input return enable bits

A return from deep standby mode by WKUPx pin input is disabled or enabled.

Bit	Description
0	Disable a return by WKUPx pin input [initial value]
1	Enable a return by WKUPx pin input

[bit9] WCEC1E: HDMI-CEC/ Remote Control Reception ch.1 interrupt return enable bit

A return from deep standby mode by HDMI-CEC/ Remote Control Reception ch.1 interrupt is disabled or enabled.

Bit	Description
0	Disable a return by HDMI-CEC/ Remote Control Reception ch.1 interrupt [initial value]
1	Enable a return by HDMI-CEC/ Remote Control Reception ch.1 interrupt

[bit8] WCEC0E: HDMI-CEC/ Remote Control Reception ch.0 interrupt return enable bit

A return from deep standby mode by HDMI-CEC/ Remote Control Reception ch.0 interrupt is disabled or enabled.

Bit	Description
0	Disable a return by HDMI-CEC/ Remote Control Reception ch.0 interrupt [initial value]
1	Enable a return by HDMI-CEC/ Remote Control Reception ch.0 interrupt

[bit7:3] WUI5E to WUI1E: WKUPx pin input return enable bits

A return from deep standby mode by WKUPx pin input is disabled or enabled.

Bit	Description
0	Disable a return by WKUPx pin input [initial value]
1	Enable a return by WKUPx pin input

[bit2] Reserved: Reserved bit

The read value is always "0". Writing has no effect on the operation.

[bit1] WLVDE: LVD interrupt return enable bit

A return from deep standby mode by LVD interrupt is disabled or enabled.

Bit	Description
0	Disable a return by LVD interrupt [initial value]
1	Enable a return by LVD interrupt

[bit0] WRTCE: RTC interrupt return enable bit

A return from deep standby mode by RTC interrupt is disabled or enabled.

Bit	Description
0	Disable a return by RTC interrupt [initial value]
1	Enable a return by RTC interrupt

Notes:

- *A return from deep standby mode by WKUP0 pin input is always enabled.*
- *This register is not initialized by deep standby transition reset.*

8.8 WKUP Pin Input Level Register (WILVR)

The WKUP Pin Input Level Register (WILVR) selects a valid level of WKUP1 to WKUP5 pin inputs that occur in deep standby mode.

Register Configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved					WUI11LV	WUI10LV	WUI9LV
Attribute			-			R/W	R/W	R/W
Initial value			00000			0	0	0

bit	7	6	5	4	3	2	1	0
Field	WUI8LV	WUI7LV	WUI6LV	WUI5LV	WUI4LV	WUI3LV	WUI2LV	WUI1LV
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Register Functions

[bit15:11] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit10:0] WUI11LV to WUI1LV: WKUPx pin input level select bits

A valid level of WKUPx pin input is selected.

Bit	Description
0	Request a return when WKUPx pin input is Low level [initial value]
1	Request a return when WKUPx pin input is High level

Notes:

- *WKUP0 pin input always requests a return in Low level*
  For example, it returns as soon as it transits to deep standby mode when WKUP1 inputs in Low level (WUI1LV = 0).
- *This register is not initialized by deep standby transition reset.*

8.9 Deep Standby RAM Retention Register (DSRAMR)

The Deep Standby RAM Retention Register (DSRAMR) controls the retention of the on-chip SRAM contents in deep standby modes.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						SRAMR	
Attribute	-						R/W	
Initial value	000000						00	

Register Functions

[bit7:2] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit1:0] SRAMR: On-chip SRAM retention control bits

These bits control the retention of the on-chip SRAM contents in deep standby modes.

bit1	bit0	Description
0	0	Not retain the on-chip SRAM contents in the deep standby mode. [initial value]
0	1	Setting is prohibited
1	0	Setting is prohibited
1	1	Retain the on-chip SRAM contents in the deep standby mode.

Note:

- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by other reset factor.*

8.10 Backup Registers 01 to 16 (BUR01 to 16)

The Backup Registers 01 to 16 (BUR01 to 16) are general registers that retain values in deep standby mode.

Register Configuration

bit	31	24	23	16	15	8	7	0
Field	BUR04		BUR03		BUR02		BUR01	
Attribute	R/W		R/W		R/W		R/W	
Initial value	0x00		0x00		0x00		0x00	

bit	31	24	23	16	15	8	7	0
Field	BUR08		BUR07		BUR06		BUR05	
Attribute	R/W		R/W		R/W		R/W	
Initial value	0x00		0x00		0x00		0x00	

bit	31	24	23	16	15	8	7	0
Field	BUR12		BUR11		BUR10		BUR09	
Attribute	R/W		R/W		R/W		R/W	
Initial value	0x00		0x00		0x00		0x00	

bit	31	24	23	16	15	8	7	0
Field	BUR16		BUR15		BUR14		BUR13	
Attribute	R/W		R/W		R/W		R/W	
Initial value	0x00		0x00		0x00		0x00	

Notes:

- *This register is initialized by power-on reset and low-voltage detection reset. It is not initialized by other reset factor.*

8.11 Main OSCillation Crystal Type Selection ConTroL Register (MOSC_CTL)

Main OSCillation crystal type selection ConTroL Register (MOSC_CTL) is used to select the type of external crystal. To fit the proper type of crystal can reduce power consumption.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					IMAINSEL		Reserved
Attribute	-					R/W		-
Initial value	00000					10		0

Register Functions

[bit7:3] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit2:1] IMAINSEL: Main oscillation circuit current setting bits

These bits set the current to Main oscillation circuit.

TYPE2-M0+:

bit1	bit0	Description
0	0	support 4MHz crystal only. Lowest power consumption.
0	1	support 4MHz/8MHz crystal, middle power consumption.
1	0	support 4MHz/8MHz/16MHz crystal, higher power consumption.[Initial setting]
1	1	support 48MHz crystal only, highest power consumption.

TYPE3-M0+:

bit1	bit0	Description
0	0	Setting is prohibited.
0	1	support up to 8MHz crystal, middle power consumption.
1	0	support up to 16MHz crystal, higher power consumption.[Initial setting]
1	1	support up to 48MHz crystal, highest power consumption.

[bit0] Reserved: Reserved bit

The read value is always "0". Writing has no effect on the operation.

Notes:

- *This register is not initialized by software reset or deep standby transition reset.*

8.12 IO State hold control Register (WIOLC_CTL)

IO State hold control Register (WIOLC_CTL) sets the behavior of IO state hold function in deep standby mode, and this register releases IO state held.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							Reserved
Attribute	-							-
Initial value	0000000							0

bit	23	22	21	20	19	18	17	16
Field	Reserved							LHX_ST
Attribute	-							R
Initial value	0000000							1

bit	15	14	13	12	11	10	9	8
Field	Reserved							CONTX
Attribute	-							R/W
Initial value	0000000							0

bit	7	6	5	4	3	2	1	0
Field	Reserved							LH_CL
Attribute	-							R/W
Initial value	0000000							0

Register Function

[bit31:25] Reserved: Reserved bits

"0x00" is read from these bits.
When writing these bits, set them to "0x00".

[bit24] Reserved: Reserved bits

"0x0" is read from this bit.
When writing this bit, set it to "0x0".

[bit23:17] Reserved: Reserved bits

"0x00" is read from these bits.
When writing these bits, set them to "0x00".

[bit16] LHX_ST : IO state bit

Indicates IO state is held or released..

Bit16		Description
Reading	0	IO state is held
	1	IO state is released (through)

[bit15:9] Reserved: Reserved bits

"0x00" is read from these bits.
When writing these bits, set them to "0x00".

[bit8] CONTX : IO State hold function enable bit

Enables IO state hold function.

Bit8		Description
Reading		Reads out the register value.
Writing	0	IO hold function is enable
	1	IO hold function is not enable

[bit7:1] Reserved: Reserved bits

"0x00" is read from these bits.
When writing these bits, set them to "0x00".

[bit0] LH_CL : IO State hold release bit

Release IO state hold. When writing to this bit, High Speed CR clock has to be running.

Bit8		Description
Reading		Read value is always ‘0’
Writing	0	IO state is not affected.
	1	IO state will be released (When Run mode).

8.13 Sub Oscillator IO Control Register (SUBOSC_CTL)

The SUBOSC_CTL register sets a pin as a sub clock (oscillation) pin..

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						SUBXC	
Attribute	-						R/W	
Initial value	-						01	

Register Function

[bit7:2] Reserved: Reserved bits

"0" is read from these bits.

When writing these bits, set them to "0".

[bit1:0] SUBXC: Sub Clock (Oscillation) Pin Setting Register

This bit sets a pin as a sub clock (oscillation) pin.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Does not use two pins of X0A and X1A as sub clock (oscillation) pins but as digital input/output pins.
	01	Uses two pins of X0A and X1A as sub clock (oscillation) pins. [Initial value] (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)
	10	Setting is prohibited.
	11	In Deep standby mode: Setting is prohibited. In other modes: Uses X0A pin as an external clock input pin. Uses X1A pin as a digital input/output.

Notes:

- *Only writing "01" to the SUBXC bit does not make a sub clock start oscillation.*
  To start oscillation, enable oscillation by the SOSCE bit of the System Clock Mode Control Register (SCM_CTL), which is described in the chapter "Clock", after writing "01" to the SUBXC bit.
- *To use external clock, refer to "Using an external clock" of "HANDLING PRECAUTIONS " in Datasheet of each product.*
- *This register is not initialized by deep standby transition reset.*

8.14 CEC Input/Output Control Register (CEC_CTL)

The CEC Input/Output Control Register(CEC_CTL) selects I/O for HDMI-CEC/Remote Control Reception. This register is only for TYPE3-M0+.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved				WS_CECR1B		WS_CECR0B	
Attribute					R/W		R/W	
Initial value	0000				00		00	

Register Functions

[bit7:4] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit3:2] WS_CECR1B: CEC1 input/output selection bits

This selects IO for ch.1 of HDMI-CEC/Remote Control Reception.

Bit		Description
Reading		Reads out the register value
Writing	00	HDMI-CEC/Remote Control Reception ch.1 is not input/output. [Initial value]
	01	CEC1_0 is used, HDMI-EC/Remote Control Reception ch.1 is input/output
	10	Setting is prohibited
	11	Setting is prohibited

[bit1:0] WS_CECR0B: CEC0 input/output selection bits

This selects IO for ch.0 of HDMI-CEC/Remote Control Reception.

Bit		Description
Reading		Reads out the register value
Writing	00	HDMI-CEC/Remote Control Reception ch.0 is not input/output. [Initial value]
	01	CEC0_0 is used, HDMI-EC/Remote Control Reception ch.0 is input/output
	10	Setting is prohibited
	11	Setting is prohibited

Notes:

- *In TYPE3-M0+ product, setting of this register becomes effective, not CECR1B/CECR0B of EPFR18 register. When writing to CECR1B/CECR0B of EPFR18 register, always write 0 to the bit. Setting of this register takes a priority.*
- *If WS_CECR1B/WS_CECR0B is set to “01”, Pull-up resistance of CEC becomes shut off regardless of PCR settings.*
- *This register is not initialized by deep standby transition reset.*

8.15 Serial Wire Debug Control Register (DEBUG_SW_CTL)

Serial Wire Debug Control Register (DEBUG_SW_CTL) sets terminals for serial wire debugging.

Register Configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved							DBG_EN
Attribute				R/W				R/W
Initial value				0000000				1

Register Functions

[bit7:1] Reserved: Reserved bits

The read value is always "0". Writing has no effect on the operation.

[bit0] DBG_EN: SWD enable bits

This bit enables Serial Wire Debug.

Bit8		Description
Reading		Reads out the register value.
Writing	0	Use GPIO port
	1	Use serial Wire debug port

Notes:

- *To enable SWD, SWDEN bit of EPFR00 register has to be set to 1, in addition to enabling DBG_EN.*
- *When SWD port (Serial wire debug port) is used as GPIO port, set this bit to 0.*
- *This register is not initialized by deep standby transition reset.*

9. Usage Precautions

Pay attention to the following points when using low power consumption mode.

For the pin shared for analog input and WKUP, WKUPx pin input is blocked when ADE bit of corresponding analog input setting register (ADE) is set to "1" even if the recovery by WKUPx pin input is allowed. To use the recovery by WKUPx pin input, set ADE bit of corresponding analog input setting register (ADE) to "0" before shifting to deep standby mode.

CHAPTER 6-2: VBAT Domain

This chapter explains the functions and operations of the VBAT power domain.

1. Overview of VBAT Domain
2. Configuration of VBAT Domain
3. Chip Power Supply Control
4. Hibernation Control
5. Procedure for Setting 32 kHz Clock
6.Procedure for Setting VBAT I/O Port
7. Registers
8. Usage Precautions

CODE: 9BFVBATPD-FM0-E03.0

1. Overview of VBAT Domain

The power consumed while the RTC is in operation can be reduced by using the VBAT power supply pin, which provides independent power supply for the RTC (calendar circuit) and the 32 kHz oscillator. It's applied to TYPE2-M0+ products.

Configuration of Power Supply Domain

This family consists of the following three power supply domains.

- CPU Domain

This domain consists of the following circuits.

  - CPU
  - On-chip Flash memory
  - On-chip SRAM
  - Peripheral functions

This domain receives power supply from the VCC power supply pin. The power supply is cut off in deep standby RTC mode and deep standby stop mode.

- Always-ON Domain

This domain consists of the following circuits.

  - On-chip regulator
  - Power management circuit
  - Port circuit
  - Main oscillation circuit and I/O Port
  - RTC(control function, Timer)

This domain receives power supply from the VCC power supply pin.

The VCC power supply pin receives power from the system power supply (on-board regulator).

- VBAT Domain

This domain consists of the following circuits.

  - RTC (Calendar function)
  - 32 kHz oscillation circuit
  - Power-on circuit
  - Backup registers
  - Port circuit

This domain always receives power supply from the VBAT power supply pin.

The VBAT power supply pin receives power from a backup power supply (such as a cell) and the system power supply.

On-Chip Power Gating

In deep standby RTC mode and deep standby stop mode, this family cuts off the power supply for the CPU Domain by using the power switch function built in the chip.

The Always-ON Domain keeps the power supply on even in deep standby RTC mode and deep standby stop mode.

For details of deep standby RTC mode and deep standby stop mode, see Chapter "Low Power Consumption Mode".

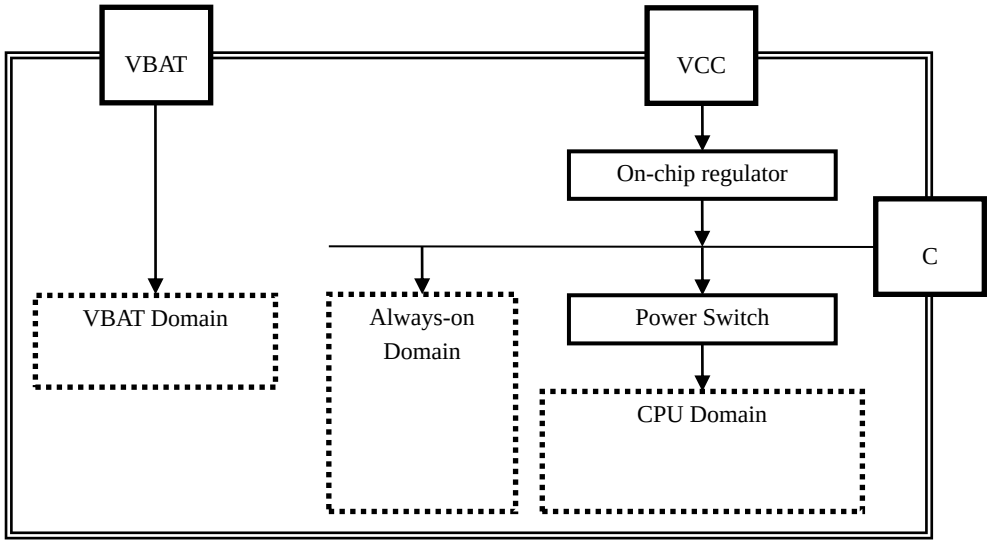
Off-Chip Power Gating

If the system power supply supplying power to the VCC pin of this family is cut off, the power supply for the CPU Domain and that for the Always-on Domain are cut off.

In this situation, the power supply for the VBAT Domain can remain on if a backup power supply supplies power to the VBAT Domain.

The on-board regulator, which controls the system power supply with an alarm time set in the RTC or with a signal input from an external pin, can be turned on or off.

Figure 1-1 Power Supply Configuration of FM0+ Family



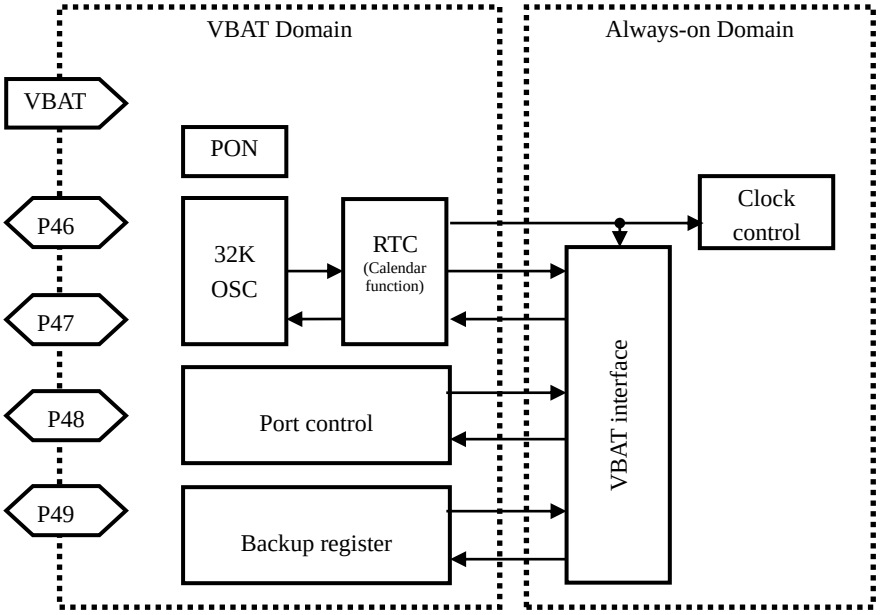
2. Configuration of VBAT Domain

This section explains the internal configuration of the VBAT Domain.

Internal Configuration of VBAT Domain

Figure 2-1 shows the internal configuration of the VBAT Domain and the connection between the VBAT Domain and the Always-on Domain.

Figure 2-1 Internal Configuration of VBAT Domain and Connection Between VBAT Domain and Always-on Domain



- RTC (Calendar function)
  This is a calendar circuit with the frequency compensation function. It is not include Timer function.
- 32 kHz oscillation circuit (32K OSC)
  This is an oscillation circuit that can be connected to a (32768 Hz) crystal oscillator for clocks.
- Power-on circuit (PON)
  This detects the power-on of the VBAT Domain and generates the circuit initialization signal.
- Backup register
  This 32-byte register retains data while power is being supplied to the VBAT power supply pin.
- VBAT I/O ports (P46 to 49, Port Control)
  They are I/O ports driven by power supplied from the VBAT power supply pin.

  The control circuit for the VBAT I/O ports is independent of the control circuit for other than I/O ports except P46 to P49.

2.1 Interfacing with Always-on Domain

This section explains the methods of interfacing the VBAT Domain withe the Always-on Domain.

Overview of Interfacing

The VBAT Domain is driven by the 32 kHz oscillation circuit or a clock divided from PCLK.

Therefore, if an internal bus is directly connected to a register belonging to the VBAT Domain, a bus master such as the CPU is made to wait when accessing such register.

The FM0+ Family has the following two mechanisms to prevent an access from being made to wait.

- A buffer is built in the Always-on Domain. An access from an internal bus is directed to that buffer.
- Data is transferred between the buffer of the Always-on Domain and the register of the VBAT Domain.

In the documents of the FM0+ Family, data transfer operations between the buffer of the Always-on Domain and the register of the VBAT Domain are called as stated below.

- Recall: data transfer from the register of the VBAT Domain to the buffer of the Always-on Domain
- Save: data transfer from the buffer of the Always-on Domain to the register of the VBAT Domain

Since data written to the buffer is erased if the VCC power supply is off, save the data in the register of the VBAT Domain while the VCC power supply is on.

Immediately after the VCC power is turned on or when a reset occurs in the Always ON domain, the buffer value is initialized by an Always ON domain reset signal.

Therefore, before reading data from the buffer, execute a recall operation to restore data retained in the register while the VBAT power supply (backup power supply) was on to the buffer.

The calendar data of the RTC in the buffer is not automatically updated.

Before reading the time data from the buffer, execute the recall operation to transfer the time data saved in the register of the VBAT Domain to the buffer.

Types of Interface Circuit

There are four types of interface circuit as shown in Table 2-1.

Table 2-1 Types of Interface Circuit

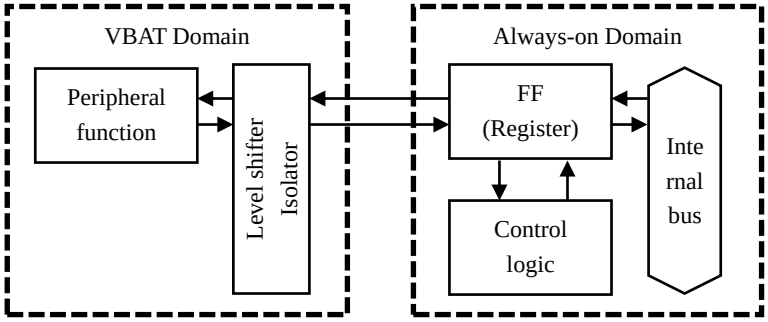
Circuit Type	Always-on Domain	VBAT Domain	Transfer Clock	Figure Number
Type 1	FF available	FF unavailable	-	Figure 2-2
Type 2	FF available	FF available	32 kHz	Figure 2-3
Type 3	FF available	FF available	PCLK1 (divided)	Figure 2-4
Type 4	FF unavailable	FF available	-	Figure 2-5 Configuration of Interface Circuit Type 4

A signal sent from the Always-on Domain to the VBAT Domain is clipped to the VSS by the level shifter and the isolator when the Always-on Domain is powered off.

This function enables the following operations to be executed when the Always-on Domain is powered off: continuing the operation of the calendar function and alarm function of the RTC, holding the pin states of the VBAT I/O ports, retaining data in the backup registers.

- Interface circuit type 1

Figure 2-2 Configuration of Interface Circuit Type 1



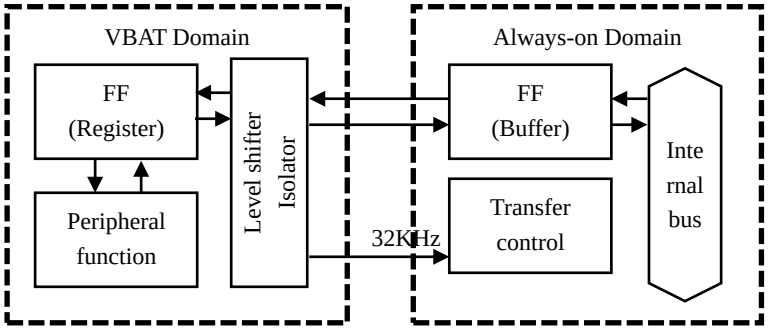
Use this circuit type if the register does not have to retain data when the VCC power supply is off.

Table 2-2 Behavior of Register of Interface Circuit Type 1

	Behavior of Register
Initialization of register	Initialization of register by the reset signal of the Always-on Domain
Bus read	The state of the control circuit (Always-on Domain) and that of the peripheral circuit (VBAT Domain) can be read directly.
Bus write	The register directly affects the operations of the control circuit (Always-on Domain) and those of the peripheral circuit (VBAT Domain).

■ Interface circuit type 2

Figure 2-3 Configuration of Interface Circuit Type 2



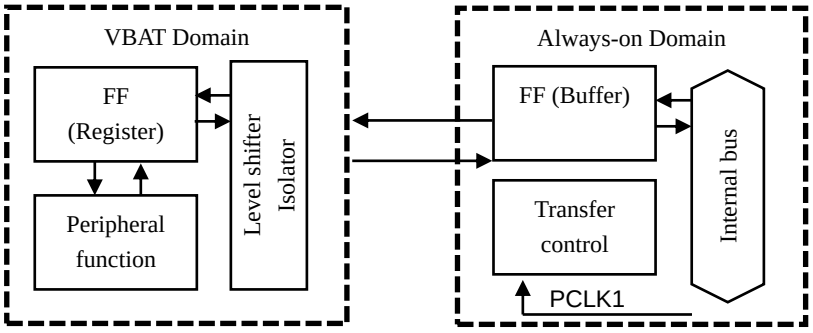
Use this circuit type if the register has to retain data even when the VCC power supply is off.

Table 2-3 Behavior of Register of Interface Circuit Type 2

	Behavior of Register/Buffer
Initialization of register	Initialization of register by the power-on signal of the VBAT Domain
Initialization of buffer	Initialization of register by the reset signal of the Always-on Domain For reset factors, see the chapter “RTC Count Block”.
Bus read	Data in the buffer is read to the bus.
Bus write	Data is written to the buffer.
Recall operation	Data is transferred from the register to the buffer.
Save operation	Data is transferred from the buffer to the register.

■ Interface circuit type 3

Figure 2-4 Configuration of Interface Circuit Type 3



Use this circuit type if the register has to retain data even when the VCC power supply is off.

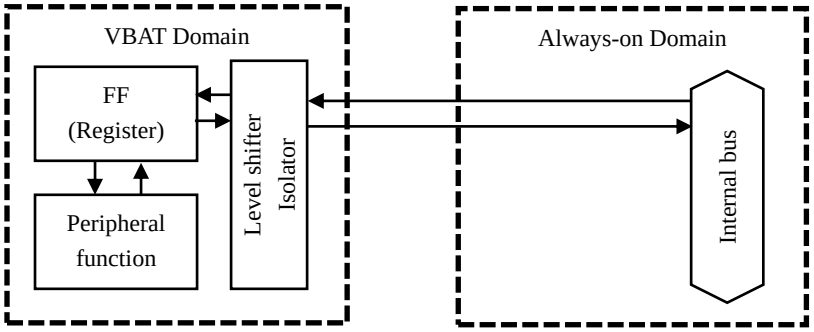
Table 2-4 Behavior of Register of Interface Circuit Type 3

	Behavior of Register/Buffer
Initialization of register	Initialization of register by the power-on signal of the VBAT Domain
Initialization of buffer	Initialization of register by the reset signal of the Always-on Domain Initialization of register by RTC reset.
Bus read	Data in the buffer is read to the bus.
Bus write	Data is written to the buffer.
Recall operation	Data is transferred from the register to the buffer.
Save operation	Data is transferred from the buffer to the register.

The difference between Type 2 Circuit and Type 3 Circuit is the clock for the recall operation and save operation.

■ Interface circuit type 4

Figure 2-5 Configuration of Interface Circuit Type 4



Use this circuit type if the register has to retain data even when the VCC power supply is off.

Table 2-5 Behavior of Register of Interface Circuit Type 4

	Behavior of Register/Buffer
Initialization of register	Initialization of register by the power-on signal of the VBAT Domain
Bus read	Data in the buffer is read to the bus.
Bus write	Directly affects the operation of the VBAT domain.

Circuit Type 4 does not require a recall operation or save operation.

Circuit Connected to Interface Circuit

The major circuits in the VBAT Domain are the RTC(Calendar function), the VBAT port and the buffer register.

The VBAT Domain executes the save operation or the recall operation on the buffer and registers of each circuit together.

(For details of the function of the WTCR20 Register in the following explanation, see "7.5 Control Register (WTCR20)" in chapter "RTC Count Block" in "Timer Part".

- CREAD/CWRITE

Performs a bulk save/recall operation for the registers shown in Table 2-6 List of Registers Transferred by CWRITE/CREAD, which are included in the RTC circuit.

Table 2-6 List of Registers Transferred by CWRITE/CREAD

No.	Register Name	Reference	No.	Register Name	Reference
1	WTSR	[RTCCAL]	2	WTMIR	[RTCCAL]
3	WTHR	[RTCCAL]	4	WTDR	[RTCCAL]
5	WTDW	[RTCCAL]	6	WTMOR	[RTCCAL]
7	WTYR	[RTCCAL]	8	ALMIR	[RTCCAL]
9	ALHR	[RTCCAL]	10	ALDR	[RTCCAL]
11	ALMOR	[RTCCAL]	12	ALYR	[RTCCAL]
13	Reserved	-	14	WTCR11	[RTCCAL]
15	WTCR10 (bit0 only)	[RTCCAL]			

For the function of each register, see [RTCCAL], which stands for Chapter "RTC Count Block" in "Timer Part".

The interface circuit type for registers No.1 to No.14 is type 2.

For WTCR10 register of No.15, this register has different types of bits of interface circuit. Bit:0 ST is type2. Bit:2 RUN is type4. Except bit0,2 of this register bits are normal register bits that are not affected by VBAT domain.

A save operation is started if "1" is written to RTC setting save control bit (CWRITE) in the WTCR20 Register. This save operation is called a CWRITE operation.

A recall operation is started if "1" is written to RTC setting recall control bit (CREAD) in the WTCR20 Register. This recall operation is called a CREAD operation.

The subclock is used as the transfer clock.

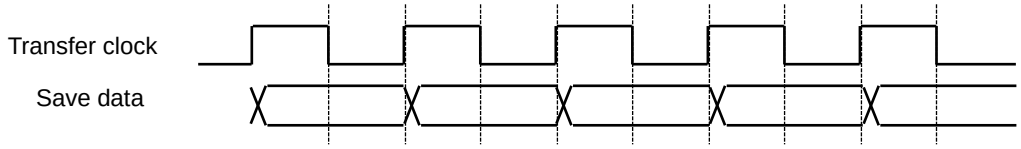
The RTC transfers 1 byte of data for one transfer clock.

In one CREAD/CWRITE operation, the RTC transfers data of registers in sequence from No.1 to No.15 shown in Table 2-6.

Since the registers are 15 bytes in total, the data transfer ends as 15 transfer clocks elapse.

Special notes are provided for the save operation and recall operation. See the notes in “3. Explanation of RTC Count Block Operation and Examples of Setting Procedures” in “RTC Count Block” of “Timer Part”.

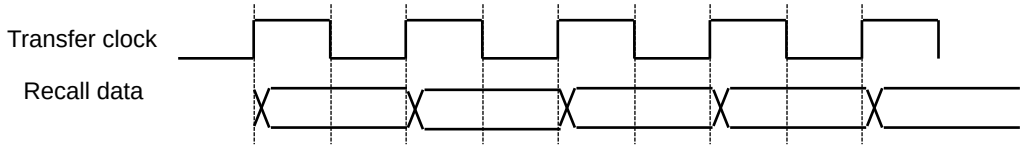
- CWRITE operation waveform



The save data is output from the buffer at a rising edge of the transfer clock and is written to the register at a falling edge of the transfer clock.

Three transfer clocks are required for preprocessing before the start of a transfer and two transfer clocks are also required for postprocessing after the end of a transfer.

- CREAD operation waveform



The recall data is output from the register at a rising edge of the transfer clock and is written to the buffer at the end of all data transfer.

Three transfer clocks are required for preprocessing before the start of a transfer and two transfer clocks are also required for postprocessing after the end of a transfer.

■ PWRITE/PREAD

Performs a bulk save/recall operation for the registers shown in Table 2-7, which are included in the VBAT port circuit.

Table 2-7 List of Registers Transferred by PWRITE/PREAD

No.	Register Name	Reference	No.	Register Name	Reference
1	WTCAL0	[RTCCLK]	2	WTCAL1	[RTCCLK]
3	WTCALPRD	[RTCCLK]	4	WTCALEN	[RTCCLK]
5	WTCOSEL	[RTCCLK]	6	CCS	2.3 32 kHz Oscillation Circuit
7	Reserved	-	8	WTOSCCNT	2.3 32 kHz Oscillation Circuit
9	VBPFR	2.6 VBAT I/O Ports	14	VBPCR	2.6 VBAT I/O Ports
11	VBDDR	2.6 VBAT I/O Ports	12	VBPZR	2.6 VBAT I/O Ports
13	VBDOR	2.6 VBAT I/O Ports			

For the function of each register, see [RTCCLK], which stands for Chapter "RTC Clock Control Block" in "Timer Part", and "2.6 VBAT I/O Ports" and "2.3 32 kHz Oscillation Circuit" in this chapter.

The interface circuit type for registers No.1 to No.13 of the VBAT port circuit is type 3.

A save operation is started if "1" is written to bit5 in the WTCR20 Register. This save operation is called a PWRITE operation.

A recall operation is started if "1" is written to bit4 in the WTCR20 Register. This recall operation is called a PREAD operation.

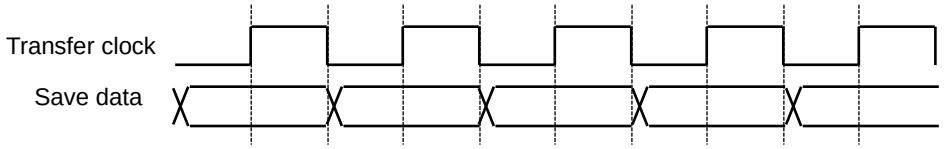
The transfer clock is created by dividing PCLK1 by the value of the VB_CLKDIV Register.

The RTC transfers 1 byte of data for one transfer clock.

In one PREAD/PWRITE operation, the RTC transfers data of registers in sequence from No.1 to No.13 shown in Table 2-7.

Since the registers of the VBAT port circuit are 13 bytes in size, the data transfer ends as 13 transfer clocks elapse.

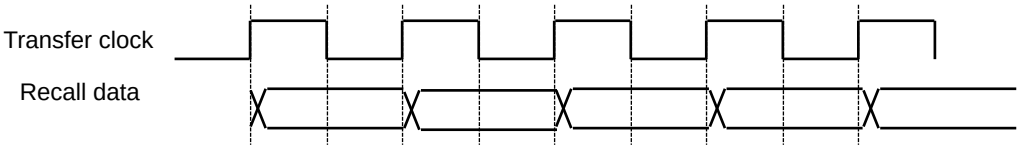
- PWRITE operation waveform



The save data is output from the buffer at a falling edge of the transfer clock and is written to the register at a rising edge of the transfer clock.

One transfer clock is required for preprocessing before the start of a transfer.

- PREAD operation waveform



The recall data is output from the register at a rising edge of the transfer clock and is written to the buffer at a falling edge of the transfer clock.

One transfer clock is required for preprocessing before the start of a transfer and one transfer clock is also required for postprocessing after the end of a transfer.

■ BWRITE/BREAD

The backup registers refer to the 32-byte register area from BREG00 to BREG1F.

For the functions of the backup registers, see "2.5 Backup Registers".

The interface circuit type for the backup registers is type 3.

A save operation is started if "1" is written to Back up register save control bit (BWRITE) in the WTCR20 Register. This save operation is called a BWRITE operation.

A recall operation is started if "1" is written to Back up register recall control bit (BREAD) in the WTCR20 Register. This recall operation is called a BREAD operation.

The transfer clock is created by dividing PCLK1 by the value of the VB_CLKDIV Register.

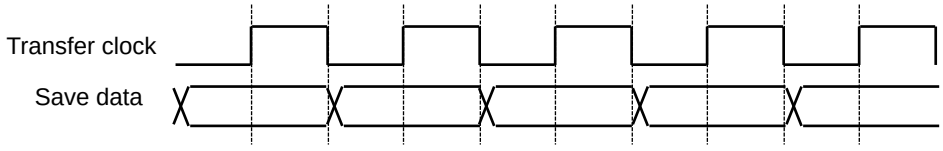
The RTC transfers 1 byte of data for one transfer clock.

In one BREAD/BWRITE operation, the RTC transfers data of registers in sequence from BREG00 to BREG1F.

The RTC starts the transfer from BREG00. The transfer destination or the transfer source is shifted to the next register whenever one transfer clock elapses.

Since the backup register of the FM0+ Family is 32 bytes in size, the data transfer ends as 32 transfer clocks elapse.

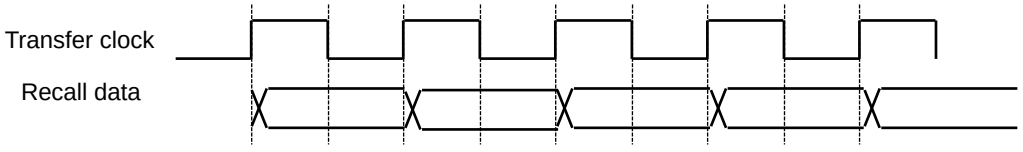
- BWRITE operation waveform



The save data is output from the buffer at a falling edge of the transfer clock and is written to the register at a rising edge of the transfer clock.

One transfer clock is required for preprocessing before the start of a transfer.

- BREAD operation waveform



The recall data is output from the register at a rising edge of the transfer clock and is written to the buffer at a falling edge of the transfer clock.

One transfer clock is required for preprocessing before the start of a transfer and one transfer clock is also required for postprocessing after the end of a transfer.

■ Allowed transfer combination

Though it should be checked that the TRANS bit in the WTCR0 Register is "0" before the start of a recall operation or of a save operation, the transfers in a combination with the "o" mark in the following table can be executed exceptionally.

	CREAD	CWRITE	PREAD	PWRITE	BREAD	BWRITE
CREAD		×	×	×	o	o
CWRITE	×		×	×	o	o
PREAD	×	×		×	o	o
PWRITE	×	×	×		o	o
BREAD	o	o	o	o		×
BWRITE	o	o	o	o	×	

"o" indicates that the transfers in that combination can be executed simultaneously.

"×" indicates that the transfers in that combination cannot be executed simultaneously.

■ Notes on description

In the peripheral manuals of the FM0+ Family, a read access and a write access to a register of interface circuit "type 2" or of interface circuit "type 3" are defined as follows.

Read access: A recall operation is executed and then data in the buffer is read.

Write access: A recall operation is executed to update the entire buffer. Afterward, the part in the buffer corresponding to the data of the write access is replaced with such data, and then a save operation is executed.

■ Usage Precautions

- Do not access the buffer during a save operation or a recall operation. It is not possible to identify whether correct data has been saved while accessing the buffer during a save operation, neither is it possible to identify whether correct data has been read while accessing the buffer during a recall operation.
- Set the VB_CLKDIV Register to a value that makes the transfer clock for PREAD, PWRITE, BREAD and BWRITE generated by dividing PCLK1 become 1 MHz or below.

2.2 RTC

The RTC of the FM0+ Family is a calendar circuit with a 32 kHz frequency compensation function.

Overview of RTC Functions

The RTC has the following functions.

- Clock function
- Alarm function
- Timer function (It exists in Always on domain)
- Frequency compensation function

Configuration of RTC

For details of the functions of the RTC, see Chapter "RTC Count Block" in "Timer Part".

2.3 32 kHz Oscillation Circuit

The 32 kHz oscillation circuit is an oscillation circuit exclusively for the crystal oscillator for the clock, and creates the subclock.

Overview of Functions of 32 kHz Oscillation Circuit

The 32 kHz oscillation circuit has the following functions.

- Oscillation current switch function (It exists in Always on domain)
- Oscillation boost function (It exists in Always on domain)
- Clock generator cooperative operation function

■ Oscillation current switch function

The amplifier circuit of the 32 kHz oscillation circuit is driven by a constant current source.

The current value of the constant current source can be controlled by the value of the CCS Register.

■ Oscillation boost function

In the case of the crystal oscillator for the clock, it takes longer time for the oscillation frequency to stabilize.

The oscillation stabilization time can be shortened by increasing the current supplied to the amplifier circuit for a certain period of time after the start of oscillation.

During the period from the start of oscillation to the end of the oscillation boost time set in the BOOST Register, the current the constant current source supplies is the current value set in the CCB Register. After the above period has elapsed, the current the constant current source supplies switches to the current value set in the CCS Register.

If a current value larger than the one set in the CCS Register is set in the CCB Register, the oscillation boost function can work effectively.

If not using the oscillation boost function, set in the CCB Register a value same as the one set in the CCS Register.

■ Clock generator cooperative operation function

The SOSCNTL bit in the WTOSCCNT Register enables or disables the cooperative operation between the 32 kHz oscillation circuit and the clock generator belonging to the CPU Domain.

With the cooperative operation enabled, the 32 kHz oscillation of this circuit stops when the CPU transits to stop mode or deep standby stop mode.

If the cooperative operation is disabled, the 32 kHz oscillation of this circuit does not stop regardless of the operation mode of the CPU.

Notes:

- *The 32 kHz oscillation circuit does not have the oscillation stabilization wait function or the clock failure detection function. Enable the clock cooperative function and use the clock failure detection function of the CPU Domain.*
- *The appropriate amount of current flowing to the amplifier circuit varies depending on the characteristic (ESR) and load capacitance (CL) of the oscillator connected to the VBAT Domain. Select an appropriate amount of current by performing a matching evaluation between the VBAT Domain and the crystal oscillator.*
- *If using the RTC with a backup power supply for the VBAT Domain instead of the VCC power supply, disable the cooperative operation with the clock generator.*
- *After the 32 kHz oscillation of this circuit has started, do not update the CCB Register or the CCS Register.*

Application of 32 kHz Oscillation Circuit

See "5 Procedure for Setting 32 kHz Clock" for details of application.

Registers Used for 32 kHz Oscillation Circuit

bit	31 - 24	23 - 16	15 - 8	7 - 0	Initial value	Attribute
	Reserved	Reserved	Reserved	VB_CLKDIV	0x00000007	R/W
	Reserved	Reserved	Reserved	WTOSCCNT	0x00000001	R/W
	Reserved	Reserved	Reserved	CCS	0x000000CE	R/W
	Reserved	Reserved	Reserved	CCB	0x000000CE	R/W
	Reserved	Reserved	Reserved	BOOST	0x00000003	R/W
	Reserved	Reserved	Reserved	EWKUP	0x00000000	R/W
	Reserved	Reserved	Reserved	VDET	0x00000080	R/W
	Reserved	Reserved	Reserved	HIBRST	0x00000000	R/W

The interface circuit types for the above registers are type 1 and type 3 and type 4.

The save operation and recall operation of the 32 kHz oscillation circuit are PWRITE and PREAD respectively.

2.4 Power-on Circuit

The FM0+ Family has a power-on circuit independent of the VCC power supply pin detecting the power-on of the VBAT Domain.

Overview of Function of Power-on Circuit

The power-on circuit in the VBAT Domain has the following function.

- VBAT power supply pin rising edge detection function

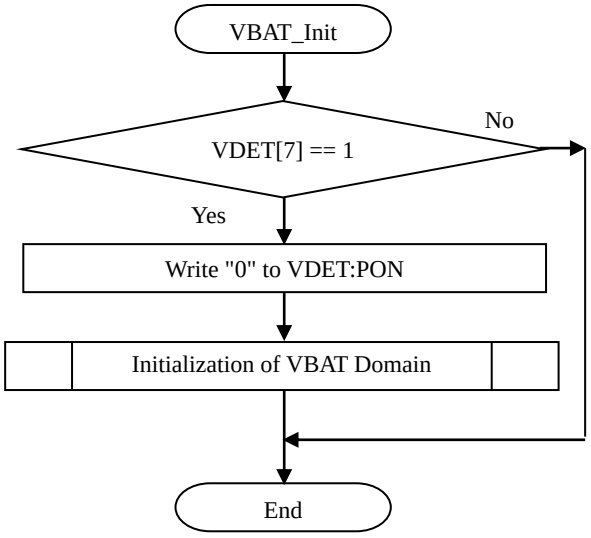
While the power-on circuit is outputting the power-on signal, Power-on bit (PON) in the VDET Register reads “1”.

The power-on signal keeps being asserted until “0” is written to Power-on bit (PON) in the VDET Register.

The power-on signal and the value of Power-on bit (PON) in the VDET Register are not affected by turning on and off of the VCC power supply.

If the power-on circuit in the VBAT Domain is initialized according to the following flow, while a peripheral function is initialized at a VCC rising edge, the initialization of the VBAT Domain that is in operation is skipped and the RTC continues running.

Figure 2-6 Flow of Power-on Judgment and Initialization of VBAT Power Domain



Notes:

- *The power-on circuit in the VBAT Domain does not have the VBAT power supply pin low voltage detection function. (The FM0+ Family does not have the VBAT power supply pin low voltage detection function.)*
- *While the power-on signal is being asserted, the values of the registers of the VBAT Domain (RTC, 32 kHz oscillation circuit, VBAT I/O port control circuit, backup registers) are fixed at their respective initial values. Before setting these circuits, write “0” to Power-on bit (PON) in the VDET Register to clear the power-on signal.*

2.5 Backup Registers

The FM0+ Family has 32-byte backup registers retaining data with the VBAT power supply.

Overview of Function of Backup Registers

A backup register retains values written to it while power is being supplied to the VBAT power supply pin.

The backup register is reset by the power-on circuit immediately after the VBAT power supply has been turned on.

As the application of the VBAT power supply can be checked by reading the value of Power-on bit (PON) in the VDET Register, initialize the backup register with a program whenever necessary.

Configuration of Backup Register and Method of Accessing Backup Register

The interface circuit type for the backup register is type 3.

For details, see “2.1 Interfacing with Always-on Domain”.

The data transfer between a backup register and a data retention register is a batch transfer of data of all areas.

Update data according to the following procedure.

1. Set the VB_CLKDIV Register to a value that makes the transfer clock become 1 MHz or below.
2. Recall (BREAD) data from the data retention register to the buffer register.

   If “1” is written to Back up register recall control bit (BREAD) in the WTCR20 Register, the recall operation starts and Transfer flag bit (TRANS) in the WTCR10 Register becomes “1”.
   If the recall operation ends, Transfer flag bit (TRANS) in the WTCR10 becomes “0”.
3. Modify the content of the buffer register.

   The buffer register allows random read access and random write access.
4. Save data in the buffer register to the data retention register.

   If “1” is written to Back up register save control bit (BWRITE) in the WTCR20 Register, the save operation starts and Transfer flag bit (TRANS) in the WTCR10 Register becomes “1”.
   If the save (BWRITE) operation ends, Transfer flag bit (TRANS) becomes “0”.

* If the power supply of the Always-on Domain is turned off, data of the buffer register is lost. Therefore, always transfer data retained in the buffer register to the data retention register by executing a save (BWRITE) operation.

* While transferring data to the data retention register (Transfer flag bit (TRANS) in the WTCR10 Register is “1”), do not access the buffer register.

* If the Always-on Domain has been reset during the data transfer or the VCC power supply is turned off, the integrity of the data of the data retention register cannot be guaranteed.

Details of Backup Registers

■ List of backup registers

bit	31 – 24	23 – 16	15 – 8	7 – 0	Initial value	Attribute
	BREG03	BREG02	BREG01	BREG00	0x00000000	R/W
	BREG07	BREG06	BREG05	BREG04	0x00000000	R/W
	BREG0B	BREG0A	BREG09	BREG08	0x00000000	R/W
	BREG0F	BREG0E	BREG0D	BREG0C	0x00000000	R/W
	BREG13	BREG12	BREG11	BREG10	0x00000000	R/W
	BREG17	BREG16	BREG15	BREG14	0x00000000	R/W
	BREG1B	BREG1A	BREG19	BREG18	0x00000000	R/W
	BREG1F	BREG1E	BREG1D	BREG1C	0x00000000	R/W

The interface circuit type for the above registers is type 3.

The save operation and recall operation of the backup registers are BWRITE and BREAD respectively.

The backup registers retain data with the VBAT pin power supply.

They can be accessed by byte access, halfword access and word access.

2.6 VBAT I/O Ports

The FM0+ Family has four I/O ports assigned to the VBAT Domain. These I/O ports (VBAT I/O ports) are controlled by the port control circuit (VBAT port control circuit) of the VBAT Domain, and continues operating even when the VCC power supply is turned off. The VBAT port control circuit is independent of the port control circuit explained in “CHAPTER I/O Port” in “FM0+ Family PERIPHERAL MANUAL”. The addresses of registers of the VBAT port control circuit are mapped to an area different the one to which the addresses of registers of the port control circuit are mapped.

Overview of Function of VBAT I/O Ports

The VBAT I/O ports keep operating as long as the VBAT power supply is turned on, even when the VCC power supply is turned off.

The VBAT I/O ports do not have the relocate function.

Configuration of VBAT I/O Ports

The registers of the VBAT port control circuit select the I/O direction, and the function of an I/O port between GPIO port and peripheral function I/O pin.

For the configuration of the VBAT I/O ports, see Figure 2-1 in chapter “I/O Port”. (For registers having the same function, substitute an actual register name for the one used in Figure 2-1.)

Table 2-8 shows a register list and explains the function of each register.

Table 2-8 Register List and Register Function

Register Name	Function
VBPFR[5:4]	This is a register setting whether to use a VBAT I/O port as a special pin (for oscillation) or as a digital I/O pin.
VBPFR[3:0]	This is a register setting whether to use a VBAT I/O port as a GPIO port or as a peripheral function I/O pin.
VBPCR[3:0]	With a VBAT I/O port used as a digital input pin or as a digital bidirectional pin, this is a register setting whether to connect or disconnect the pull-up resistor of a VBAT I/O port.
VBDDR[3:0]	With a VBAT I/O port used as a GPIO port, this is a register setting whether to use that GPIO port as an input pin or as an output pin. Note: If a VBAT I/O port is used as a peripheral function I/O pin, the setting of this register becomes invalid.
VBDIR[3:0]	This is a register reading the level of a VBAT I/O port. − If a VBAT I/O port is used as a digital input pin, this register reads the input level. − If a VBAT I/O port is used as a digital output pin, this register reads the output level. − If a VBAT I/O port is used as a special pin, this register always reads “0”.
VBDOR[3:0]	With a VBAT I/O port used as a GPIO output pin, this is a register setting the output level. − If a bit in VBDOR[3:0] is set to “0”, a GPIO output pin outputs “L” level. − If a bit in VBDOR[3:0] is set to “1”, a GPIO output pin outputs “H” level. Note: If a VBAT I/O port is used as a GPIO input pin or as a peripheral function I/O pin, the setting of this register becomes invalid.
VBPZR[1:0]	This is a register controlling the open drain of a VBAT I/O port. − If a VBAT I/O port outputs “L” level, I/O Port is set to “L” level output. (The pull-up resistor is disconnected regardless of the setting of the PCR Register.) − If a VBAT I/O port outputs “H” level, I/O Port is set to “Hi-Z”, and the open drain is controlled in pseudo manner. (The pull-up resistor is disconnected regardless of the setting of the PCR Register.) − If a VBAT I/O port is used as an input port, I/O Port is set to “Hi-Z”, and their I/O direction changes to the input direction. (The pull-up resistor is disconnected regardless of the setting of the PCR Register.)

The interface circuit type of the VBDIR Register is type 4. The interface circuit type of the other registers are type 3.

The save operation and recall operation of the 32 kHz oscillation circuit are PWRITE and PREAD respectively. (register of interface circuit type 3)

Note:

- *The settings of the I/O Port Control Registers (PFR4[6:9], PCR4[6:9], DDR4[6:9], DIR4[6:9], DOR4[6:9], PZR4[6:9]) have no effect on the operations of the VBAT I/O ports.*

Initial Settings of VBAT I/O Ports

Table 2-9 shows the respective initial states of the VBAT I/O ports.

Table 2-9 Initial States of VBAT I/O Ports.

No.	Pin	Initially Selected Function
1	P46/X0A	This pin can be used as an oscillation pin. (The oscillation has stopped.) The digital input has been cut off and “0” has been input to this pin.
2	P47/X1A	This pin can be used as an oscillation pin. (The oscillation has stopped.) The digital input has been cut off and “0” has been input to this pin.
3	P48/VREGCTL	This is a digital input pin. The output is open drain.
4	P49/VWAKEUP	This is a digital input pin. The output is open drain.

The VBAT I/O ports remain in their respective states described in Table 2-9 while the VBAT power-on circuit is resetting the VBAT Domain.

Procedure for Setting VBAT I/O Ports

- In the case of using 32 kHz oscillation circuit
  See “5 Procedure for Setting 32 kHz Clock” for different setting procedures.
- In the case of controlling hibernation
  See “4 Hibernation Control” for the setting procedure as well as the procedure for setting I/O.
- In the case of using VBAT I/O port as GPIO port
  For the setting method, refer to “CHAPTER I/O Port” in “FM0+ Family PERIPHERAL MANUAL”.

  (For registers having the same function, substitute an actual register name for the one used in that chapter.)

Registers of VBAT I/O Ports

■ List of registers of VBAT I/O ports

bit	31 – 24	23 – 16	15 – 8	7 – 0	Initial value	Attribute
	Reserved	Reserved	Reserved	VBPFR	0x0000001C	R/W
	Reserved	Reserved	Reserved	VBPCR	0x00000000	R/W
	Reserved	Reserved	Reserved	VBDDR	0x00000000	R/W
	Reserved	Reserved	Reserved	VBDIR	0x000000XX	R
	Reserved	Reserved	Reserved	VBDOR	0x0000000F	R/W
	Reserved	Reserved	Reserved	VBPZR	0x00000003	R/W

Configuration of Registers of VBAT I/O Ports and Method of Accessing Those Registers

The interface circuit type for the VBAT I/O port registers is type 3.

For details, see “2.1 Interfacing with Always-on Domain”.

The data transfer between a buffer register and a VBAT I/O port register is a batch transfer of data of all areas.

Update data according to the following procedure.

1. Set the VB_CLKDIV Register to a value that makes the transfer clock become 1 MHz or below.
2. Recall data from the VBAT I/O port retention register to the buffer register.

   If “1” is written to VBAT PORT recall control bit(PREAD) in the control register 20(WTCR20), the recall operation starts and transfer flag bit(TRANS) in the control register 10(WTCR10) becomes “1”.

   If the recall operation ends, the TRANS bit becomes “0”.
3. Modify the content of the buffer register.

   The buffer register allows random read access and random write access.
4. Save data in the buffer register to the VBAT I/O port register.

   If “1” is written to VBAT PORT save control bit(PWRITE) in the control register 20(WTCR20), the save operation starts and Transfer flag bit(TRANS) in the control register(WTCR10) becomes “1”. If the save operation ends, the TRANS bit becomes “0”.

   - Modifying new data in the buffer register alone does not change the state of a VBAT I/O port pin.

     To change a VBAT I/O port register value (pin state), execute a save operation to transfer data in a buffer register corresponding to that VBAT I/O port register to that VBAT I/O port register.
   - While transferring data to the data retention register (TRANS bit in the WTCR10 Register is “1”), do not access the buffer register.
   - If the Always-on Domain has been reset during the data transfer or the VCC power supply is turned off, the integrity of the data of the data retention register cannot be guaranteed.

3. Chip Power Supply Control

This section explains details of applying and cutting off chip power supply.

Table of Combinations of VCC Power Supply and VBAT Power Supply

Table 3-1 shows the respective states of the VCC power supply and the VBAT power supply.

Table 3-1 Combination of VCC Power Supply State and VBAT Power Supply State

	VBAT Power Supply On	VBAT Power Supply Off
VCC power supply on	Normal operation	This combination is prohibited.
VCC power supply off	Only the VBAT Domain continues operating.	Stop of operation

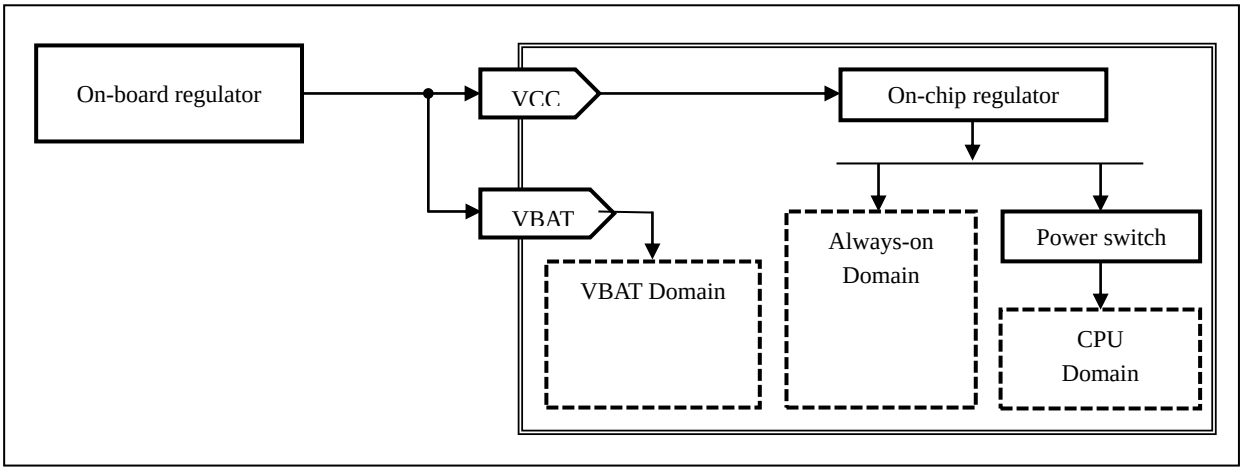
Driving VBAT Power Supply and VCC Power Supply with Same Power Supply

■ Transition of power supply state

If the VBAT power supply and the VCC power supply are driven by the same power supply, the chip power supply state transits between “normal operation” and “stop of operation” shown in .Table 3-1.

When driving the VBAT power supply and the VCC power supply with the same power supply, initialize the VBAT Domain whenever applying the VCC power supply.

Figure 3-1 Example of Driving VBAT Power Supply with VCC Power Supply



If not using the backup power supply for the VBAT power supply, connect the VBAT power supply pin directly to the VCC power supply pin outside the chip.

If the VBAT power supply pin is directly connected to the VCC power supply pin, the prohibited combination of “VCC power supply on and VBAT power supply off” can be avoided.

Driving VBAT Power Supply with battery

■ Transition of power supply state

Figure 3-2 shows how the state of power supply transits when a battery is used as the VBAT power supply. Figure 3-3 shows the respective waveforms of circuits.

Power-on bit (PON) in the VDET Register indicates whether the system power supply has been turned on for the first time.

If the system power supply has been turned on for the first time, do the settings of the circuits in the VBAT Domain.

Figure 3-2 Transition of States with battery Used as VBAT Power Supply

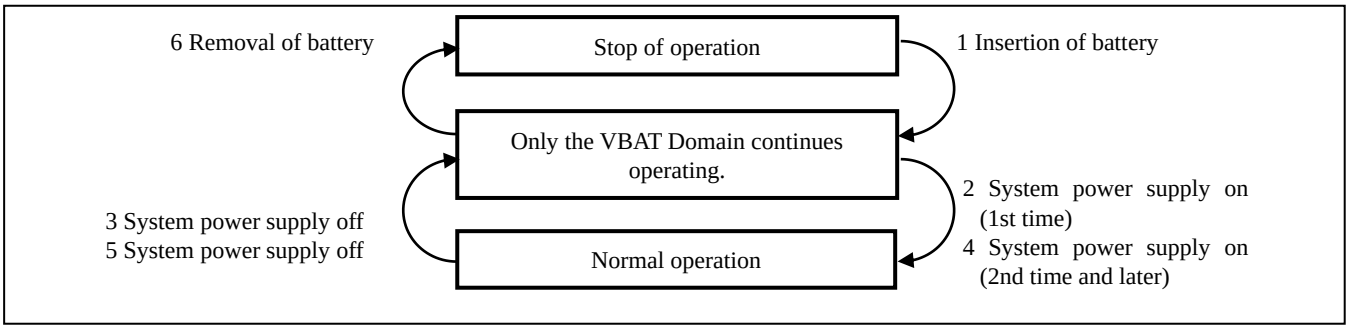
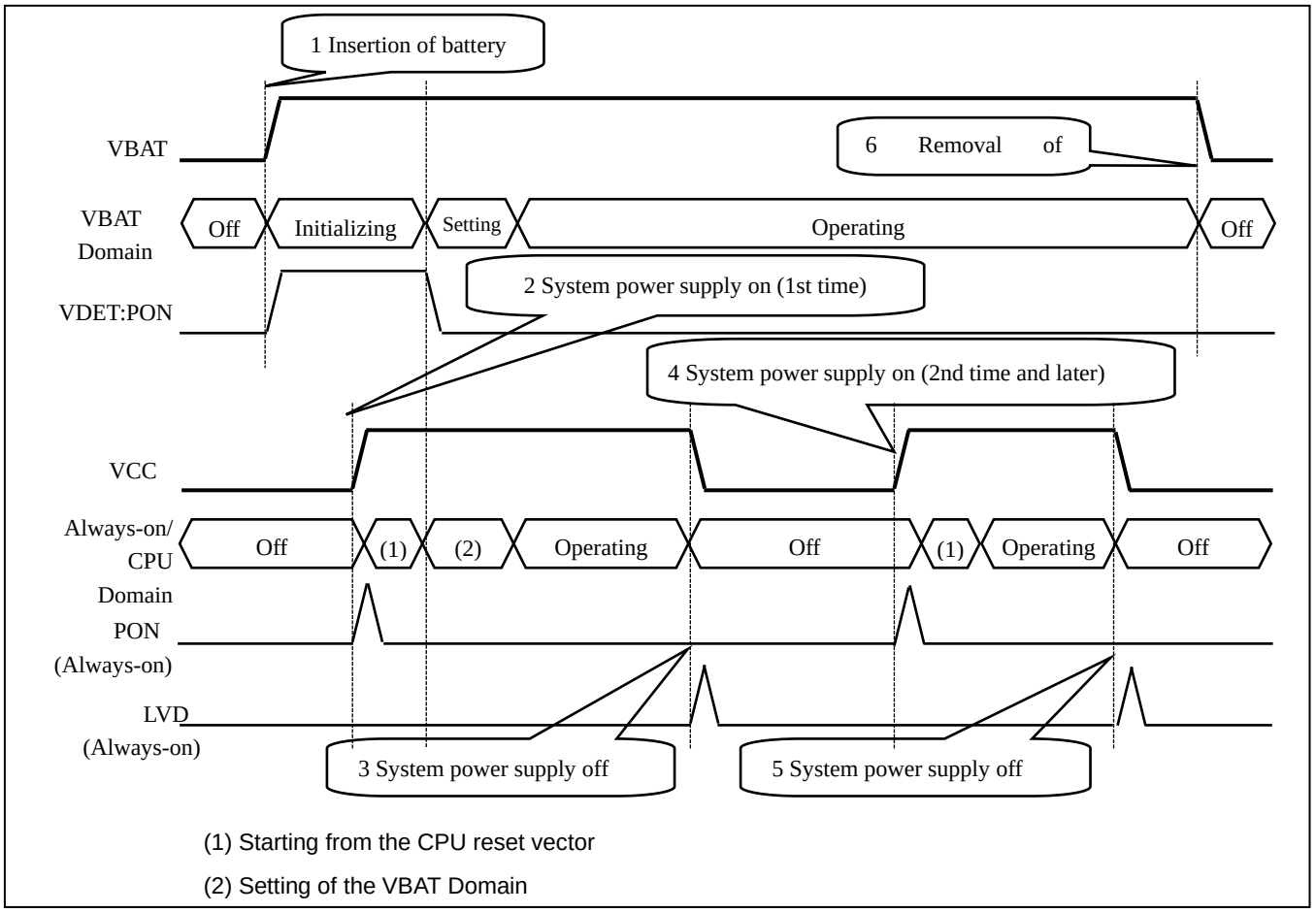


Figure 3-3 Example of Waveforms for Using battery



(1) Starting from the CPU reset vector

(2) Setting of the VBAT Domain

- Examples of power supply configuration

Figure 3-4 Example of Using Primary battery as Backup Power Supply

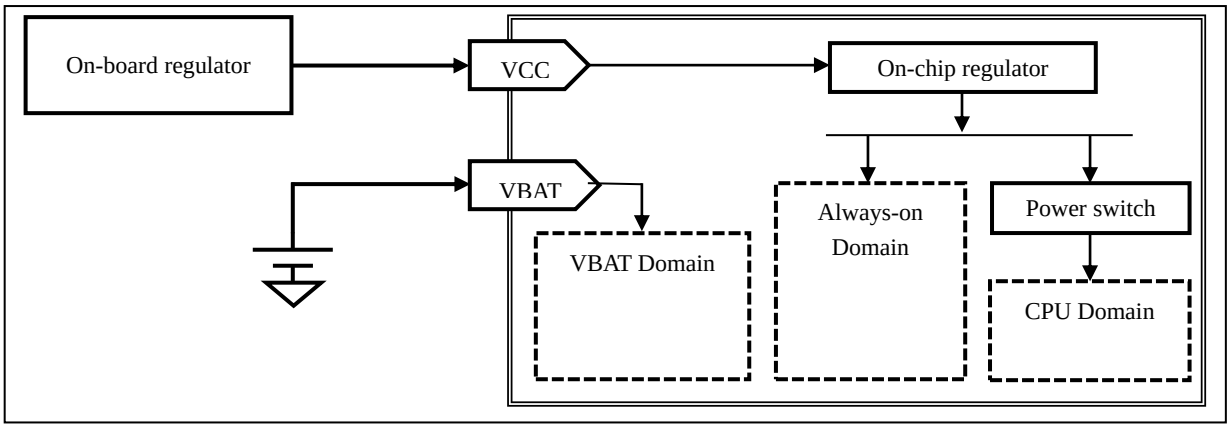
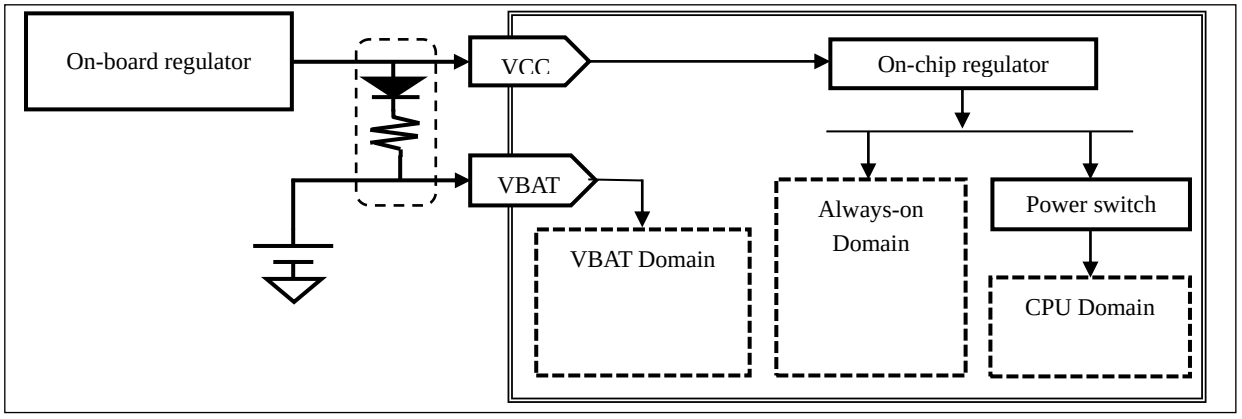


Figure 3-5 Example of Using Secondary battery as Backup Power Supply



The diode and resistor inside the dotted line box trickle charges the secondary battery.

When setting the resistance, check whether the secondary battery used can be trickle charged, and the maximum current of trickle charging.

If the secondary battery used cannot be trickle charged, use it the same as a primary battery.

Notes:

- *Turn off the system power supply before inserting or removing a battery.*
- *If a primary battery is used as the backup power supply, it is not recommended to connect the system power supply to the backup power supply through a diode.*

4. Hibernation Control

This section shows an example of circuit configuration for controlling off-chip power gating through the microcontroller and an example of the sequence of controlling off-chip gating through the microcontroller.

Overview of Hibernation Control

Hibernation control turns on or off the VCC power supply (for both Always-on Domain and CPU Domain) by controlling the standby function of the on-board regulator through the VBAT Domain.

To execute hibernation control, supply the VBAT pin with a backup power supply other than the VCC power supply (system power supply).

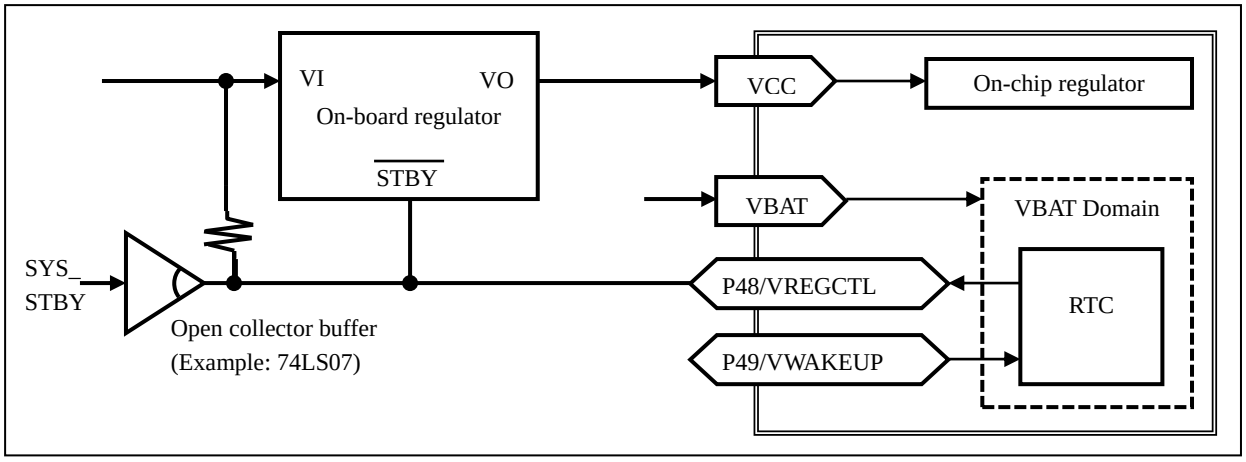
Below are the two sources for making the microcontroller return from the hibernation state.

- Alarm interrupt of the RTC
- Wakeup request to the P49/VWAKEUP pin (A request is made at a rising edge.)

To use the alarm interrupt of the RTC to make the microcontroller return from the hibernation state, keep the 32 kHz oscillation operating even when the VCC power supply is off.

External Connection Examples of FM0+ Family

Figure 4-1 Example of External Connection with Input Voltage (VI) of on-Board Regulator Lower than 5.5 V



Many on-board regulators enter the standby state when their “STBY” input is “L” level.

The P48/VREGCTL pin of the VBAT I/O ports is 5 V tolerant and is a pseudo-open drain pin.

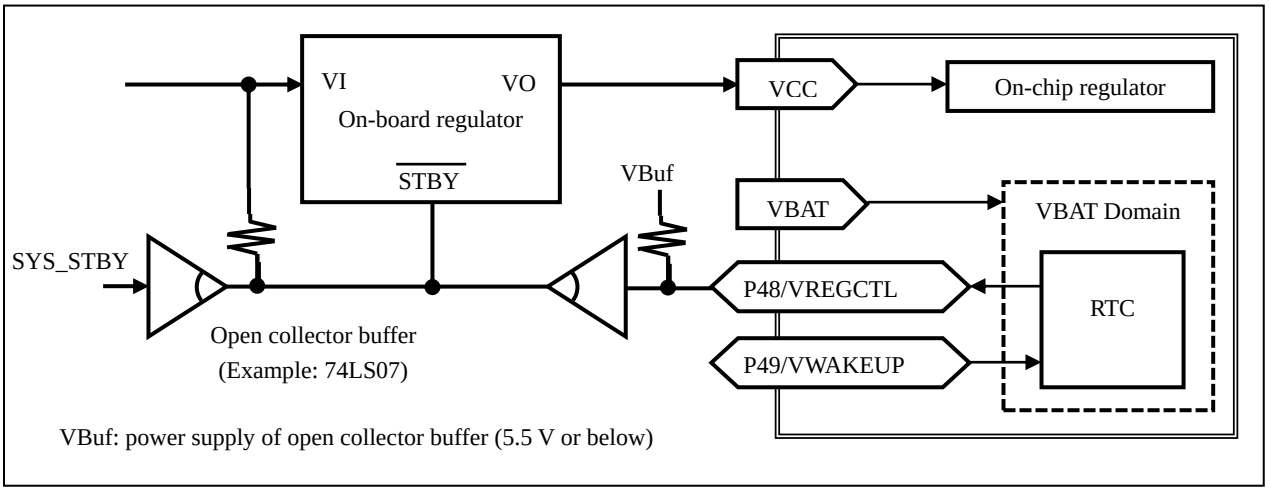
Connect the P48/VREGCTL to the “STBY” input of the on-board regulator, and connect a pull-up resistor to the input voltage (VI) of the on-board regulator

Table 4-1 shows how the on-board regulator operates when the standby signal of the system is buffered by an open collector buffer, and the buffered standby signal and the P48/VREGCTL are connected by a wired OR logic circuit.

Table 4-1 Operation of On-Board Regulator

SYS_STBY	VREGCTL = “L”	VREGCTL = “H”
“L”	Standby mode	Standby mode
“H”	Standby mode	Normal operation mode

Figure 4-2 Example of External Connection with Input Voltage (VI) of On-Board Regulator Higher than 5.5 V



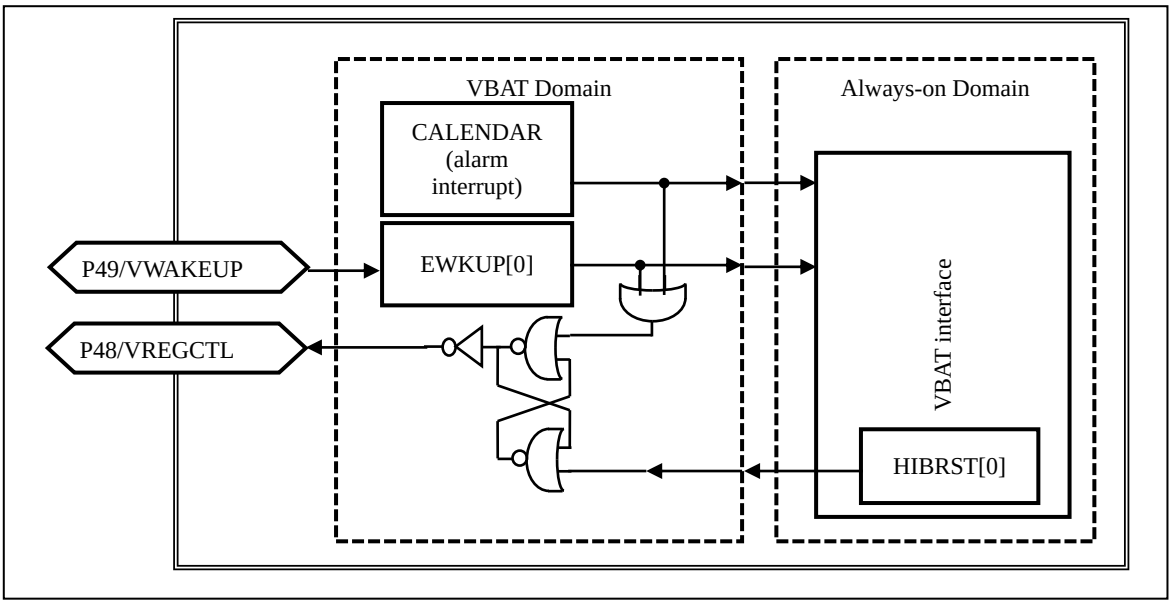
If the input voltage of the on-board regulator is higher than 5.5 V, the standby pin of the on-board regulator cannot be directly controlled by the P48/VREGCTL pin.

Execute buffering with an open collector buffer whose voltage resistance is higher than the input voltage of the on-board regulator.

Block Configuration of Hibernation Controller

The hibernation controller is part of the RTC circuit. Figure 4-3 shows the configuration of the hibernation controller.

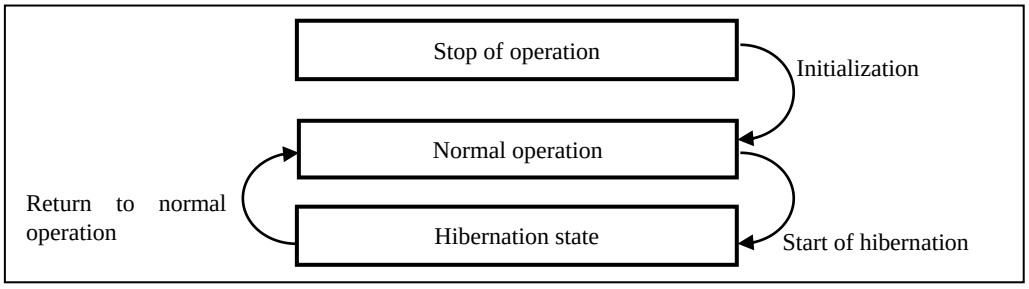
Figure 4-3 Hibernation Controller



Example of Hibernation Operation Flow

Figure 4-4 shows the hibernation operation flow.

Figure 4-4 Example of Hibernation Operation Flow



■ Initial settings of hibernation operation

Below are the initial settings required for the hibernation operation.

- Alarm setting of the RTC

  For the method of setting the alarm, refer to “CHAPTER: RTC Count Block” in “FM0+ Family PERIPHERAL MANUAL Timer Part”.

- Setting of the P49/VWAKEUP pin

  Write “1” to the VPFR1 bit in the VBPFR Register.

- Setting of the P48/VREGCTL pin

  Write “1” to the VPFR0 bit in the VBPFR Register.

The CPU core can transit to the hibernation state even when the alarm setting of the RTC and the setting of the P49/WAKEUP pin are not done.

If the CPU core transits to the hibernation state with both settings not done, it cannot return to the normal operation state.

■ Setting of hibernation start

With both alarm interrupt of the RTC and wakeup (P49/VWAKEUP pin) cleared, if “1” is written to bit0 in the HIBRST Register, the P48/VREGCTL pin becomes “0”, the on-board regulator transits to the standby state and the VCC power supply is turned off.

■ Judging return from hibernation state and operations after return from hibernation state

If an alarm interrupt of the RTC or a wakeup request occurs, the P48/VREGCTL pin becomes “1”, the on-board regulator returns from the standby state and the VCC power supply is turned on.

If the VCC power supply is turned on, the CPU core executes the normal power-on operation.

To judge whether the CPU core has returned from the hibernation state, check whether the following three conditions are met.

- The VBAT Domain has been powered on (Power-on bit (PON) in the VDET Register).
- The alarm interrupt of the RTC has occurred (Alarm coincidence flag bit (INTALI) in the WTCR12 Register).
- A wakeup up request has been made (Wakeup request bit (WUP0) in the EWKUP Register).

Notes:

- *The P48/VREGCTL pin becomes “0” immediately after “1” has been written to Hibernation start bit (HIBRST) in the HIBRST Register.*
- *Complete all operations for turning off the VCC power supply before witting “1” to Hibernation start bit (HIBRST) in the HIBRST Register.*
- *In the hibernation operation, the VCC power supply is assumed to be tuned off with the control of P48/VREGCTL pin.*

  When the on-board regulator is not directly controlled with P48/VREGCTL pin at debugging, turn off the VCC power supply once by manual operation.

5. Procedure for Setting 32 kHz Clock

This section explains recommended sequences of setting the 32 kHz oscillation circuit when using the RTC.

Features of 32 kHz Oscillation Circuit in VBAT Domain

With the 32 kHz oscillation circuit incorporated in the VBAT Domain, even when the CPU Domain and the Always-on Domain are turned off, the 32 kHz oscillation circuit can continue operating and the RTC can continue counting the time.

Linking with Clock Control Circuit

If the 32 kHz oscillation circuit in the VBAT Domain is linked with the clock control circuit is executed, the VBAT Domain becomes compatible with the FM3 Family.

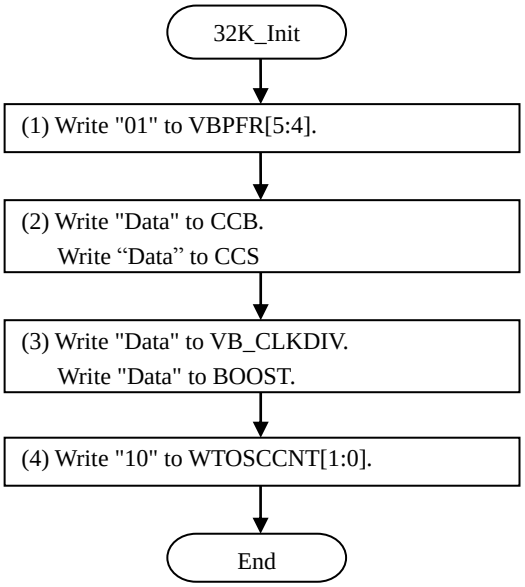
In addition, with the 32 kHz oscillation circuit in the VBAT Domain linked with the clock control circuit and the VCC power supply turned off, if the CPU transits to deep standby mode or deep standby stop mode as the VCC power supply is turned on, the 32 kHz oscillation automatically stops.

If the 32 kHz clock is only used as the clock for subrun mode, power consumption of the backup power supply can be reduced by linking the 32 kHz oscillation circuit with the clock control circuit.

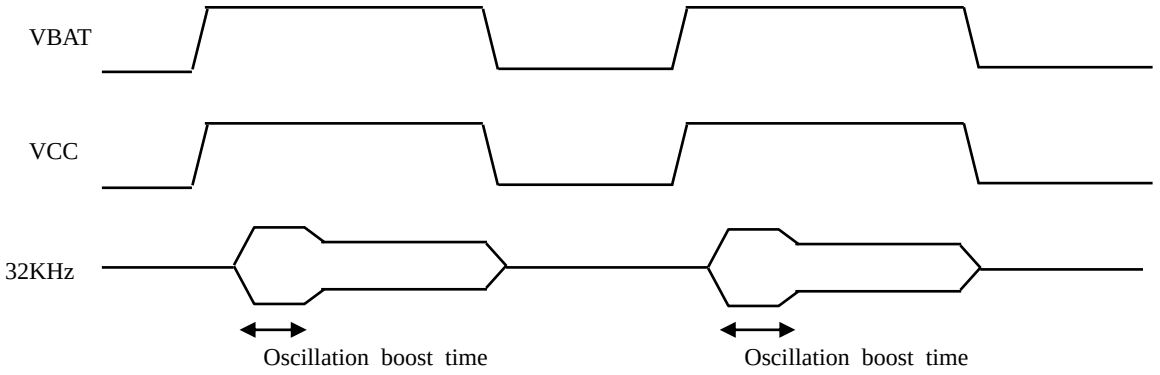
■ Example of setting procedure

(1) Set VBAT I/O Port to use the 32k oscillation circuit.
(2) Set an appropriate current value for the current that is to flow to the oscillation amplifier circuit.
(3) Set the oscillation boost time.
(4) Enable the cooperative operation with the clock control circuit.

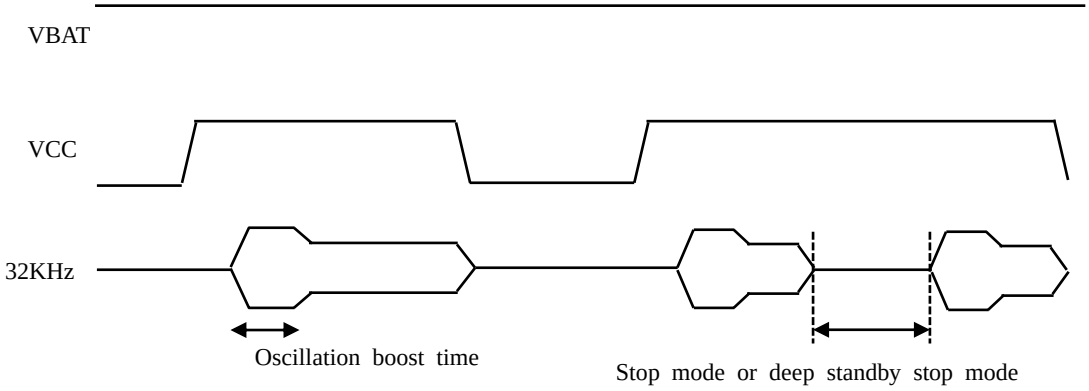
In addition, enable the oscillation.



- Examples of operation
  - No backup power supply is used.



  - The backup power supply is used, and the 32 kHz oscillation circuit is linked with the clock control circuit.



Not Linking with Clock Control Circuit

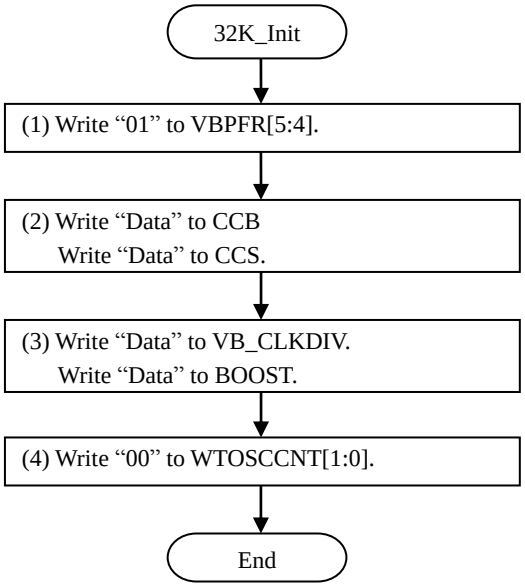
If always using the backup power supply to keep the RTC operating, do not link the 32 kHz oscillation circuit with the clock control circuit.

The average power consumption of the entire system can be reduced by executing the following operations: keep only the VBAT operating with the backup power supply, and use the hibernation control of the VBAT Domain or the external circuit to turn off the VCC power supply while processes by the CPU are not necessary.

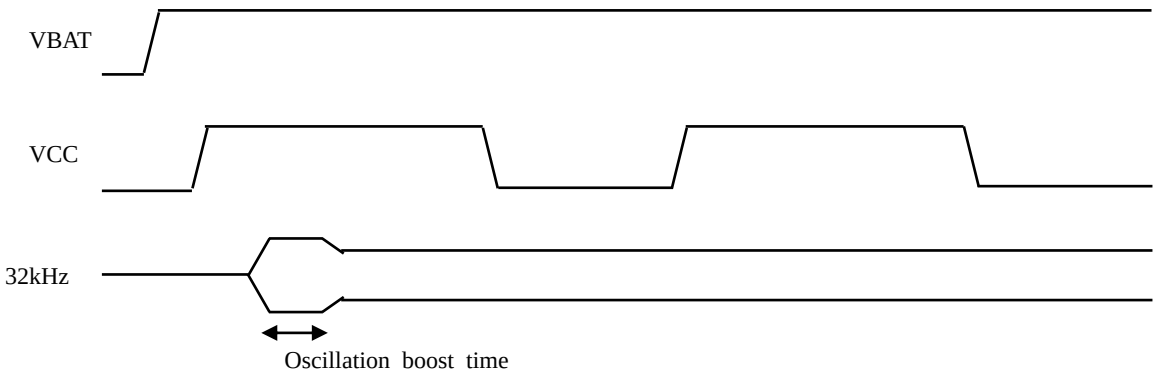
■ Example of setting procedure

(1) Set the Register of VBAT I/O Port to use the 32 kHz oscillation circuit.
(2) Set an appropriate current value for the current that is to flow to the oscillation amplifier circuit.
(3) Set the oscillation boost time.
(4) Disable the cooperative operation with the clock control circuit.

In addition, enable the oscillation.



■ Examples of operation



Not Linking with Clock Control Circuit But Waiting for Oscillation Stabilization

It is necessary to not link the 32 kHz oscillation circuit with the clock control circuit when always using the backup power supply to keep the RTC operating. Nonetheless, the 32 kHz oscillation circuit and RTC in the VBAT Domain do not have the oscillation stabilization wait function.

When the 32 kHz clock is used only for the RTC, a software timer can be used to count the oscillation stabilization wait time. However, if the 32 kHz clock is also used in subrun mode, the clock oscillation stabilization wait function becomes necessary.

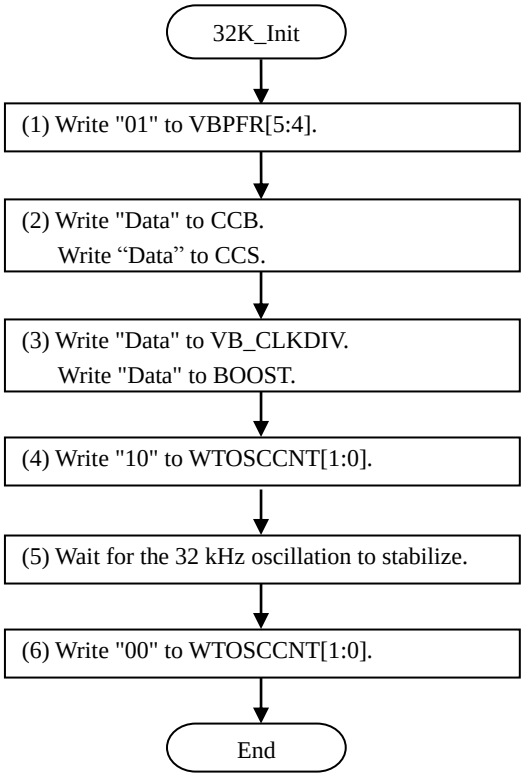
In the above situation, follow the procedure below to enable the oscillation stabilization wait function only at the start of oscillation.

■ Example of setting procedure

Enable the cooperative operation with the clock control circuit and start the oscillation.

After the oscillation stabilization wait time has elapsed, disable the cooperative operation with the clock control circuit.

(1) Set the Register of VBAT I/O Port to use the 32 kHz oscillation circuit.
(2) Set an appropriate current value for the current that is to flow to the oscillation amplifier circuit.
(3) Set the oscillation boost time.
(4) Enable the cooperative operation with the clock control circuit. In addition, enable the oscillation.
(5) Wait for the stabilization of the 32 kHz oscillation.
(6) Disable the cooperative operation with the clock control circuit. (The oscillation keeps being enabled.)



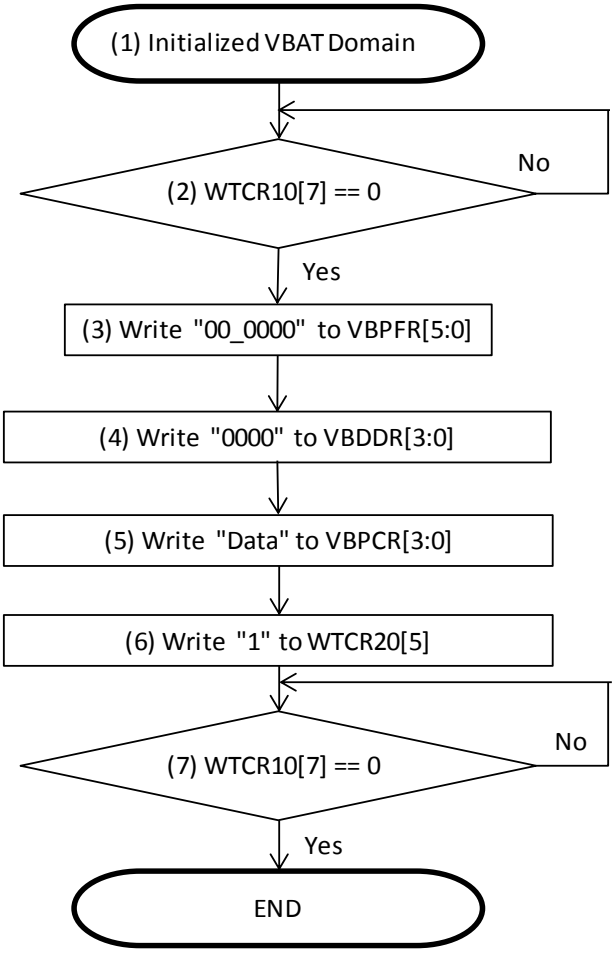
6. Procedure for Setting VBAT I/O Port

■ When using VBAT I/O as a general-purpose I/O input

・Setting procedure example

The following is a setting example of using P46, P47, P48 and P49 all as general-purpose I/O inputs.

(1) Initiate the VBAT domain (see Figure 2-6).
(2) Set the general-purpose IO port to use it as the GPIO pin.
(3) Set the pull-up.
(4) Set the port output direction to the input direction.
(5) Transfer the setting value to the VBAT domain.
(6) Wait until the transfer is completed.

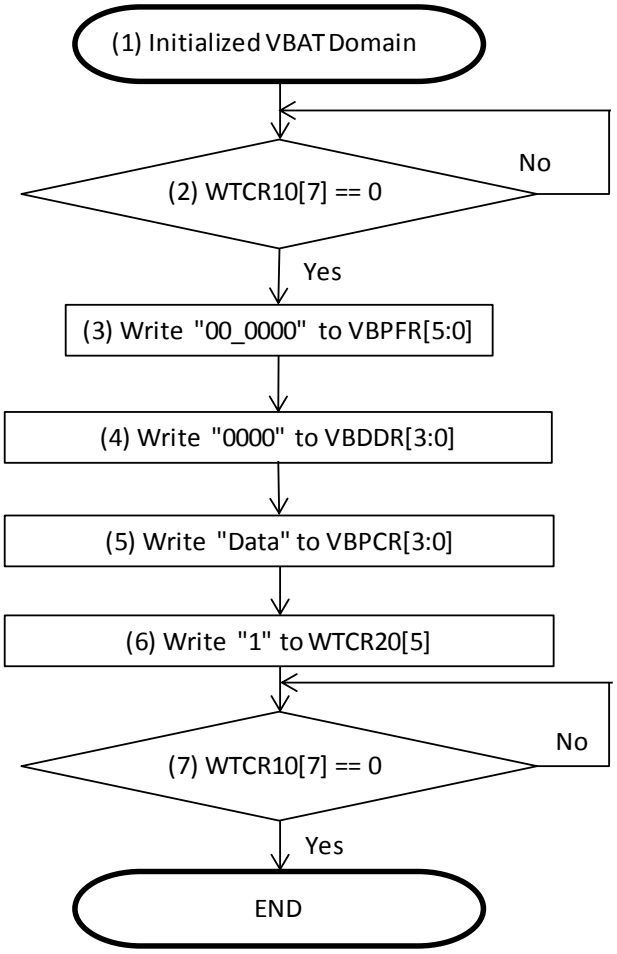


■ When using the VBAT I/O as a general-purpose I/O output:

・Setting procedure example

The following is a setting example of using P46, P47, P48 and P49 all as general-purpose I/O outputs.

(1) Initiate the VBAT domain (see Figure 2-6).
(2) Set the general-purpose IO port to use it as the GPIO pin.
(3) Set the port output direction to the output direction.
(4) Set the port output data register.
(5) Set the port pseudo open drain register (only P48 and P49 can be set).
(6) Transfer the setting value to the VBAT domain.
(7) Wait until the transfer is completed.

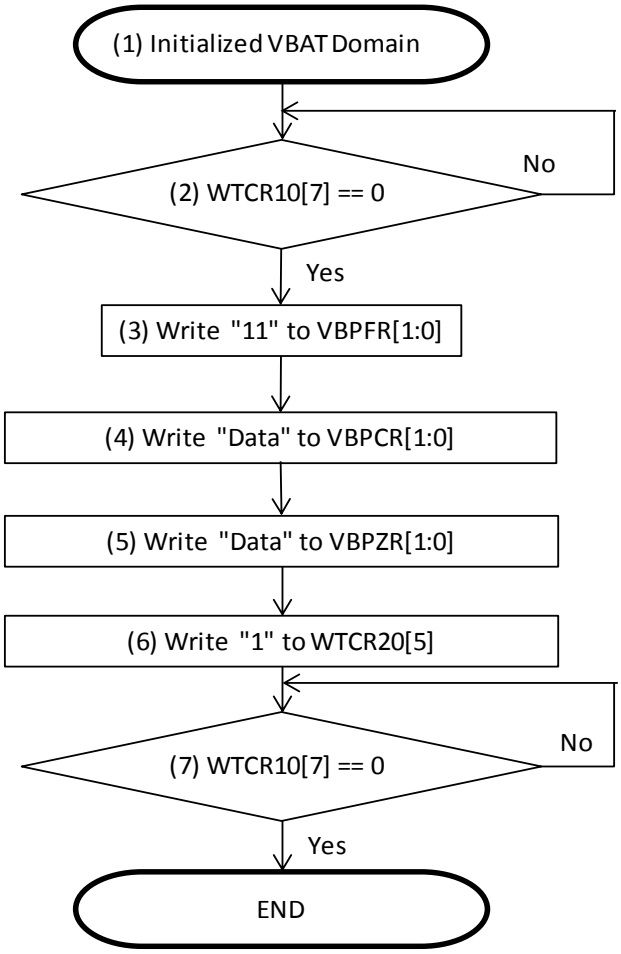


■ When using the VBAT I/O as a peripheral function:

・Setting procedure example

The following is a setting example of using P48 and P49 as peripheral function.

(1) Initiate the VBAT domain (see Figure 2-6).
(2) Set the general-purpose IO port to use it as the peripheral function.
(3) Set the pull-up.
(4) Set the port pseudo open drain.
(5) Transfer the setting value to the VBAT domain.
(6) Wait until the transfer is completed.



7. Registers

This section explains the register list of the VBAT Domain unit.

Table 7-1 shows the registers of the VBAT Domain unit.

Table 7-1 Registers of VBAT Domain Unit.

Abbreviation	Register Name	Reference
VB_CLKDIV	VB_CLKDIV Register	7.1
WTOSCCNT	WTOSCCNT Register	7.2
CCS/CCB	CCS/CCB Register	7.3
BOOST	BOOST Register	7.4
EWKUP	EWKUP Register	7.5
HIBRST	HIBRST Register	7.6
VDET	VEDT Register	7.7
VBPFR	Port Function Set Register	7.8
VBPCR	Pull-up Set Register	7.9
VBDDR	Port I/O Direction Set Register	7.10
VBDIR	Port I/O Data Register	7.11
VBDOR	Port Output Data Register	7.12
VBPZR	Port Pseudo-Open Drain Set Register	7.13

The registers and buffers which exist in always on domain in Table 7-1 Registers of VBAT Domain Unit., except VBDIR, VDET and EWKUP, are cleared by a system reset or RTC reset. Therefore, the save operation must be performed after the value is set again or the recall operation is performed.

7.1 VB_CLKDIV Register

VB_CLKDIV register set the frequency of transfer clock when the buck-up register and port register are transferred simultaneously.

Bit	7	6	5	4	3	2	1	0
Field	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	1	1	1

The interface circuit type for this register is type 1.

[bit7:0] DIV[7:0]: Transfer Clock set bits for PREAD, PWRITE, BREAD, BWRITE

These bits set the transfer clock cycle used in the batch transfer of the backup register and of the port register.

Equation of computing the register value: transfer clock = PCLK / (VB_CLKDIV + 2)

(Set these bits to a value that makes the frequency of the transfer clock used in BREAD/BWRITE and PREAD/PWRITE 1 MHz or below.)

7.2 WTOSCCNT Register

WTOSCCNT Register specifies the operation of 32 kHz Oscillation circuit.

Bit	7	6	5	4	3	2	1	0
Field	Reserved						SOSCNTL	SOSCEX
Attribute	-						R/W	R/W
Initial value	-						0	1

The interface circuit type for this register is type 3.

[bit7:2] Reserved: Reserved bits

These bits read “0b000000”.

In a write access to these bits, write “0b000000” to them.

[bit1] SOSCNTL: Cooperative operation control bit

This bit enables or disables the cooperative operation between the 32 kHz oscillation circuit and the clock control circuit.

Bit		Description
Reading		A read access reads the value of this bit.
Writing	0	The 32 kHz oscillation circuit operates independently as VBAT Dmain. (Initial value)
	1	The 32 kHz oscillation circuit is linked with the clock control circuit.

[bit0] SOSCEX: Oscillation enable bit

This bit enables or disables the operation of the oscillation circuit when the 32 kHz oscillation circuit operates independently as VBAT Domain.

If the 32 kHz oscillation circuit is linked with the clock control circuit, this bit cannot control the operation of the oscillation circuit.

Bit		Description
Reading		A read access reads the value of this bit.
Writing	0	Starts the oscillation.
	1	Stops the oscillation. (Initial value)

7.3 CCS/CCB Register

CCS Register sets the current value when the oscillation sustains.
CCB Register sets the boost current at the oscillation start.

TYPE2-M0+ products

■CCS Register

bit	7	6	5	4	3	2	1	0
Field	CCS							
Attribute	R/W							
Initial value	11001110							

The interface circuit type for this register is type 3.

[bit7:0] CCS: Oscillation sustain current set bits

These bits set the value of current for sustaining oscillation.

■CCB Register

bit	7	6	5	4	3	2	1	0
Field	CCB							
Attribute	R/W							
Initial value	11001110							

The interface circuit type for this register is type 1.

[bit7:0] CCB: Oscillation boost current set bits

These bits set the value of boost current at the start of oscillation.

Table 7-2 Relationship between Settings of CCS/CCB Register and the Modes.

Table 7-2 Relationship between Settings of CCS/CCB Register and the Modes.

CCS/CCB	Mode	Remark
00000100	Low power	Load capacity it will be possible to use a smaller type of crystal oscillator.
11001110	Standard	(Initial value)
Others	Undefined	Setting prohibited

Note:

- *Please be the same setting CCS register and CCB register.*

7.4 BOOST Register

BOOST Register sets the clock value of oscillation boost.

Bit	7	6	5	4	3	2	1	0
Field	Reserved						BOOST1	BOOST0
Attribute	-						R/W	R/W
Initial value	-						1	1

The interface circuit type for this register is type 1.

[bit7:2] Reserved: Reserved bits

These bits read “0b000000”.

In a write access to these bits, write “0b000000” to them.

[bit1:0] BOOST1, BOOST0: Oscillation boost time set bits

These bits set the number of clocks for oscillation boost.

Table 7-3 Settings of Oscillation Boost Time

BOOST1	BOOST0	Oscillation Boost Time
1	1	500 ms [Initial value]
1	0	250 ms
0	1	125 ms
0	0	62.5 ms

7.5 EWKUP Register

EWKUP Register displays and clears the request state of the wakeup.

Bit	7	6	5	4	3	2	1	0
Field	Reserved							WUP0
Attribute	-							R/W
Initial value	-							0

The interface circuit type for this register is type 4.

[bit7:1] Reserved: Reserved bits

These bits read “0b0000000”.

In a write access to these bits, write “0b0000000” to them.

[bit0] WUP0: Wakeup request bit

Bit		Description
Reading	0	The VBAT Domain has accepted no wakeup request.
	1	The VBAT Domain has accepted a wakeup request.
Writing	0	The VBAT Domain clears a wakeup request.
	1	Writing “1” to this bit has no effect on operation.

The wakeup request function is enabled if Port function of P49/VWAKEUP pin set bit (VPFR1) in VBPFR register is set to “1”.

With the wakeup request function enabled, if the VBAT Domains detects a rising edge of the P49/VWAKEUP pin, it accepts a wakeup request and makes an RTC interrupt to the interrupt control circuit.

The wakeup request can be accepted at the 7th PCLK cycle from the wakeup request clear or later. To clear a register in VBAT Domain, 7 PCLK cycles are required. So, if the standby mode or external reset is entered before 7 PCLK cycles have elapsed, the wakeup request is not accepted because PCLK is stopped.

7.6 HIBRST Register

HIBRST Register sets the hibernation start.

Bit	7	6	5	4	3	2	1	0
Field	Reserved							HIBRST
Attribute	-							R/W
Initial value	-							0

The interface circuit type for this register is type 1.

[bit7:1] Reserved: Reserved bits

These bits read “0b0000000”.

In a write access to these bits, write “0b0000000” to them.

[bit0] HIBRST: Hibernation start bit

Bit		Description
Reading		A read access reads the value of this bit.
Writing	0	This bit can write 0. But it has no effect on the hibernation operation.
	1	Starts the hibernation.

The hibernation can be started if VPFR0 bit in Port Function Setup Register (VBPFR) is set to “1”.

7.7 VDET Register

VDET Register indicates the state of power-on circuit and clears the power-on signal.

Bit	7	6	5	4	3	2	1	0
Field	PON	Reserved						
Attribute	R/W	-						
Initial value	1	-						

The interface circuit type for this register is type 4.

[bit7] PON: Power-on bit

This bit indicates the state of the power-on circuit and clears the power-on signal.

Bit		Description
Reading	0	Indicates that the initialization signal of the power-on circuit has been cleared.
	1	Indicates that the power-on circuit has output the initialization signal.
Writing	0	Clears the power-on signal.
	1	Writing “1” to this bit has no effect on operation.

[bit6:0] Reserved: Reserved bits

These bits read “0b0000000”.

In a write access to these bits, write “0b0000000” to them.

7.8 Port Function Set Register (VBPFR)

VBPFR Register selects the usage of pins.

Bit	7	6	5	4	3	2	1	0
Field	Reserved		SPSR1	SPSR0	VPFR3	VPFR2	VPFR1	VPFR0
Attribute	-		R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-		0	1	1	1	0	0

The interface circuit type for this register is type 3.

[bit7:6] Reserved: Reserved bits

These bits read “0b00”.

In a write access to these bits, write “0b00” to them.

[bit5:4] SPSR1, SPSR0: Oscillation pin function set bits

bit5	bit4	Function
0	0	The P46 and P47 pins are used as digital (GPIO) pins.
0	1	The P46 and P47 pins are used as 32 kHz oscillation pins. [Initial value]
1	0	The P46 and P47 pins are used as digital (GPIO) pins.
1	1	The P46 pin is used as an external clock input pin. The P47 pin is used as a digital (GPIO) pin.

[bit3] VPFR3: Port function of P46/X0A pin set bit

[bit2] VPFR2: Port function of P47/X1A pin set bit

Bit		Description
Reading		A read access reads the value of this bit.
Writing	0	The pin corresponding to the VPFR3 bit or to the VPFR2 bit is used as a GPIO port.
	1	The pin corresponding to the VPFR3 bit or to the VPFR2 bit is used as an I/O pin of a peripheral function. (Initial value)

[bit1] VPFR1: Port function of P49/VWAKEUP pin set bit

[bit0] VPFR0: Port function of P48/VREGCTL pin set bit

Bit		Description
Reading		A read access reads the value of this bit.
Writing	0	The pin corresponding to the VPFR1/VPFR0 bit is used as a GPIO port. [Initial value]
	1	The pin corresponding to the VPFR1/VPFR0 bit is used as an I/O pin of a peripheral function.

VBPFR[5:2] setting combinations are as shown in Table 7-4.

Table 7-4 VBPFR[5:2] Setting Combinations.

	VBPFR[5]	VBPFR[4]	VBPFR[3]	VBPFR[2]
GPIO	0	0	0	0
32kHz oscillation	0	1	-	-
GPIO	1	0	0	0
P46 external clock input	1	1	1	0

To use the 32kHz oscillation circuit, set the function setting bit (VBPFR[5:4]) of the oscillation pin to 0b01. This enables the 32kHz oscillation circuit to be used without depending on VBPFR[3:2].

To use P46/X0A as an external clock, set VBPFR[5:2] to 0b1110, and then input an external clock from P46/X0A. In this case, P47 can be used as the GPIO pin.

7.9 Pull-Up Set Register (VBPCR)

VBPCR Register sets the pull-up of pins.

In TYPE2-M0+ products, there is no pull-up function of P46/X0A pin and P47/X1A pin, therefore the settings of the VBPCR[3:2] are invalid.

Bit	7	6	5	4	3	2	1	0
Field	Reserved				VPCR3	VPCR2	VPCR1	VPCR0
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				0	0	0	0

The interface circuit type for this register is type 3.

[bit7:4] Reserved: Reserved bits

These bits read “0b0000”.

In a write access to these bits, write “0b0000” to them.

[bit3] VPCR3: P46/X0A pin pull-up set bit

[bit2] VPCR2: P47/X1A pin pull-up set bit

[bit1] VPCR1: P49/VWAKEUP pin pull-up set bit

[bit0] VPCR0: P48/VREGCTL pin pull-up set bit

Bit		Description
Reading		A read access reads the value of this bit. (Initial value = 0)
Writing	0	The pull-up resistor of the pin corresponding to the VPCR3/VPCR2/VPCR1/VPCR0 bit is disconnected from the pin.
	1	If the pin corresponding to the VPCR3/VPCR2/VPCR1/VPCR0 bit is in the input state (either GPIO function or peripheral function), the pull-up resistor is connected to the pin. If the pin corresponding to the VPCR3/VPCR2/VPCR1/VPCR0 bit is in the output state, the pull-up resistor is disconnected from the pin.

7.10 Port I/O Direction Set Register (VBDDR)

VBDDR Register sets the I/O direction of pins.

In TYPE2-M0+ products, the GPIO function of P46/X0A pin and P47/X1A pin is an input only, therefore they cannot be used as an output port.

Bit	7	6	5	4	3	2	1	0
Field	Reserved				VDDR3	VDDR2	VDDR1	VDDR0
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				0	0	0	0

The interface circuit type for this register is type 3.

[bit7:4] Reserved: Reserved bits

These bits read “0b0000”.

In a write access to these bits, write “0b0000” to them.

[bit3] VDDR3: Port direction of P46/X0A pin set bit

[bit2] VDDR2: Port direction of P47/X1A pin set bit

[bit1] VDDR1: Port direction of P49/VWAKEUP pin set bit

[bit0] VDDR0: Port direction of P48/VREGCTL pin set bit

Bit		Description
Reading		A read access reads the value of this bit. [Initial value = 0]
Writing	0	The GPIO port is used as an input port. If the pin corresponding to the VDDR3/VDDR2/VDDR1/VDDR0 bit is used as an I/O pin of a peripheral function, the setting of the VDDR3/VDDR2/VDDR1/VDDR0 bit is ignored.
	1	The GPIO port is used as an output port. If the pin corresponding to the VDDR3/VDDR2/VDDR1/VDDR0 bit is used as an I/O pin of a peripheral function, the setting of the VDDR3/VDDR2/VDDR1/VDDR0 bit is ignored.

7.11 Port Input Data Register (VBDIR)

VBDIR Register indicates the input data of pins.

Bit	7	6	5	4	3	2	1	0
Field	Reserved				VDIR3	VDIR2	VDIR1	VDIR0
Attribute	-				R	R	R	R
Initial value	-				x	x	x	x

The interface circuit type for this register is type 4.

[bit7:4] Reserved: Reserved bits

These bits read “0b0000”.

In a write access to these bits, write “0b0000” to them.

[bit3] VDIR3: Port input data of P46/X0A pin bit

[bit2] VDIR2: Port input data of P47/X1A pin bit

[bit1] VDIR1: Port input data of P49/VWAKEUP pin bit

[bit0] VDIR0: Port input data of P48/VREGCTL pin bit

Bit		Description
Reading	0	Regardless of the pin function settings (VBPFR[3:0], VBDDR and VBDOR Registers), this bit indicates that the pin is in the “L” level input state or the “L” level output state. If the P46 and P47 pins are used as special function pins according to the settings of the SPSR1 and SPSR0(bit[5:4]) in the VBPFR Register, this bit always reads “0” as the input is blocked.
	1	Regardless of the pin function settings (VBPFR[3:0], VBDDR and VBDOR Registers), this bit indicates that the pin is in the “H” level input state or the “H” level output state.
Writing		Writing a value to this bit has no effect on operation.

7.12 Port Output Data Register (VBDOR)

VBDOR Register sets the data output to pins.

In TYPE2-M0+ products, the GPIO function of P46/X0A pin and P47/X1A pin is an input only, therefore the settings of the VBDOR[3:2] are invalid.

Bit	7	6	5	4	3	2	1	0
Field	Reserved				VDOR3	VDOR2	VDOR1	VDOR0
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				1	1	1	1

The interface circuit type for this register is type 3.

[bit7:4] Reserved: Reserved bits

These bits read “0b0000”.

In a write access to these bits, write “0b0000” to them.

[bit3] VDOR3: Port output data of P46/X0A pin bit

[bit2] VDOR2: Port output data of P47/X1A pin bit

[bit1] VDOR1: Port output data of P49/VWAKEUP pin bit

[bit0] VDOR0: Port output data of P48/VREGCTL pin bit

Bit		Description
Reading		A read access reads the value of this bit. (Initial value = 1)
Writing	0	Outputs “L” level to the GPIO port. If the pin is used as an input pin or as a peripheral function I/O pin, the setting of this bit is ignored.
	1	Outputs “H” level to the GPIO port. If the pin is used as an input pin or as a peripheral function I/O pin, the setting of this bit is ignored.

7.13 Port Pseudo-Open Drain Set Register (VBPZR)

VBPZR Register sets the port pseudo-open drain of a pin.

Bit	7	6	5	4	3	2	1	0
Field	Reserved						VPZR1	VPZR0
Attribute	-						R/W	R/W
Initial value	-						1	1

The interface circuit type for this register is type 3.

[bit7:2] Reserved: Reserved bits

These bits read “0b000000”.

In a write access to these bits, write “0b000000” to them.

[bit1] VPZR1: P49/VWAKEUP pin pseudo-open drain set bit

[bit0] VPZR0: P48/VREGCTL pin pseudo-open drain set bit

Bit		Description
Reading		A read access reads the value of this bit. [Initial value = 1]
Writing	0	If digital “H” level is output from a GPIO port or a peripheral macro, the pin becomes “H” level.
	1	If digital “H” level is output from a GPIO port or a peripheral macro, the pin becomes “Hi-Z”. The pull-up resistor is disconnected regardless of the setting of the PCR Register.

8. Usage Precautions

Note the following when using the backup power supply.

- Charging a primary battery or overcharging a secondary battery may cause battery leakage or fire. Check the features of the battery to be used before deciding the configuration of the circuit around the battery.
- The hibernation control function cannot be used if the on-board regulator has no standby pin. To control the hibernation, select a product that has a standby pin.

CHAPTER 7-1: Interrupts Overview

The interrupts is focusing on system peripheral interrupts control through NVIC (Nested Vector Interrupt Controller) to Cortex-M0+ core.

1. Explanation of Reference Chapter of Interrupts

CODE: 9BFINTTOP_FM0-E03.0

1. Explanation of Reference Chapter of Interrupts

The configuration of interrupts varies according to the product type. See the following chapters corresponding with the product type to be used.

Table 1-1 Correspondence Table for Interrupts

Product Type	Reference
TYPE1-M0+	Chapter "Interrupts (Type-1)" Chapter “Interrupts (Type-1) (A)” Chapter " Interrupts (Type-1) (B)"
TYPE2-M0+	Chapter "Interrupts (Type-2)" Chapter “Interrupts (Type-2) (A)” Chapter " Interrupts (Type-2) (B)"
TYPE3-M0+	Chapter "Interrupts (Type-3)"

CHAPTER 7-2: Interrupts Top (TYPE1)

This chapter explains the interrupt controller (TYPE1) and peripheral interrupt requests (TYPE1).

1. Overview
2. Configuration

CODE: 9AFIRQC_A-E01.0

1. Overview

The interrupt controller (Type-1) determines the priority of interrupt requests and sends the requests to the CPU. The Cortex-M0+ CPU core is equipped with the nested vectored interrupt controller (NVIC) internally within the core. Interrupt signals from several peripherals are aggregated and input to a single interrupt factor vector. The interrupt requests that have occurred can be checked using the interrupt request batch read register.

Features of the Nested Vectored Interrupt Controller (NVIC)

- 32 maskable peripheral interrupt channels (not including the 16 exception interrupts of Cortex-M0+)
- 4 programmable interrupt priority levels (using 2-bit prioritized interrupts)
- Facilitates low-latency exception and interrupt handling
- Implements System Control Registers
- Supports non-maskable interrupt (NMI) input

The NVIC and the processor core interface are closely coupled, providing mechanisms that enable low-latency interrupt processing and efficient processing of late arriving interrupts. The NVIC maintains the nested interrupt information to enable tail chaining of interrupts.

All interrupts are managed by the NVIC, including core exceptions. See "Chapter 5: Exceptions" and "Chapter 8: Nested Vectored Interrupt Controller" in the "Cortex-M0+ Technical Reference Manual" published by Arm for details on exceptions and NVIC.

Note:

- *In the "Cortex-M0+ Technical Reference Manual", all exception type:IRQ are defined as external interrupt inputs. In this manual, exception type:IRQ are expressed as peripheral interrupts. Peripheral interrupts include "External Interrupt and NMI Control Unit" interrupts from external pins and interrupts from peripheral resources within the LSI.*

Interrupt Factor Aggregation Function

The interrupt request signals from each peripheral resource are aggregated into 32 sources and input to the NVIC. Furthermore, the interrupt request signal from the external NMIX pin is logically OR'ed with the hardware watchdog interrupt signal and input to the NVIC.

Peripheral Interrupt Request Batch Read Function

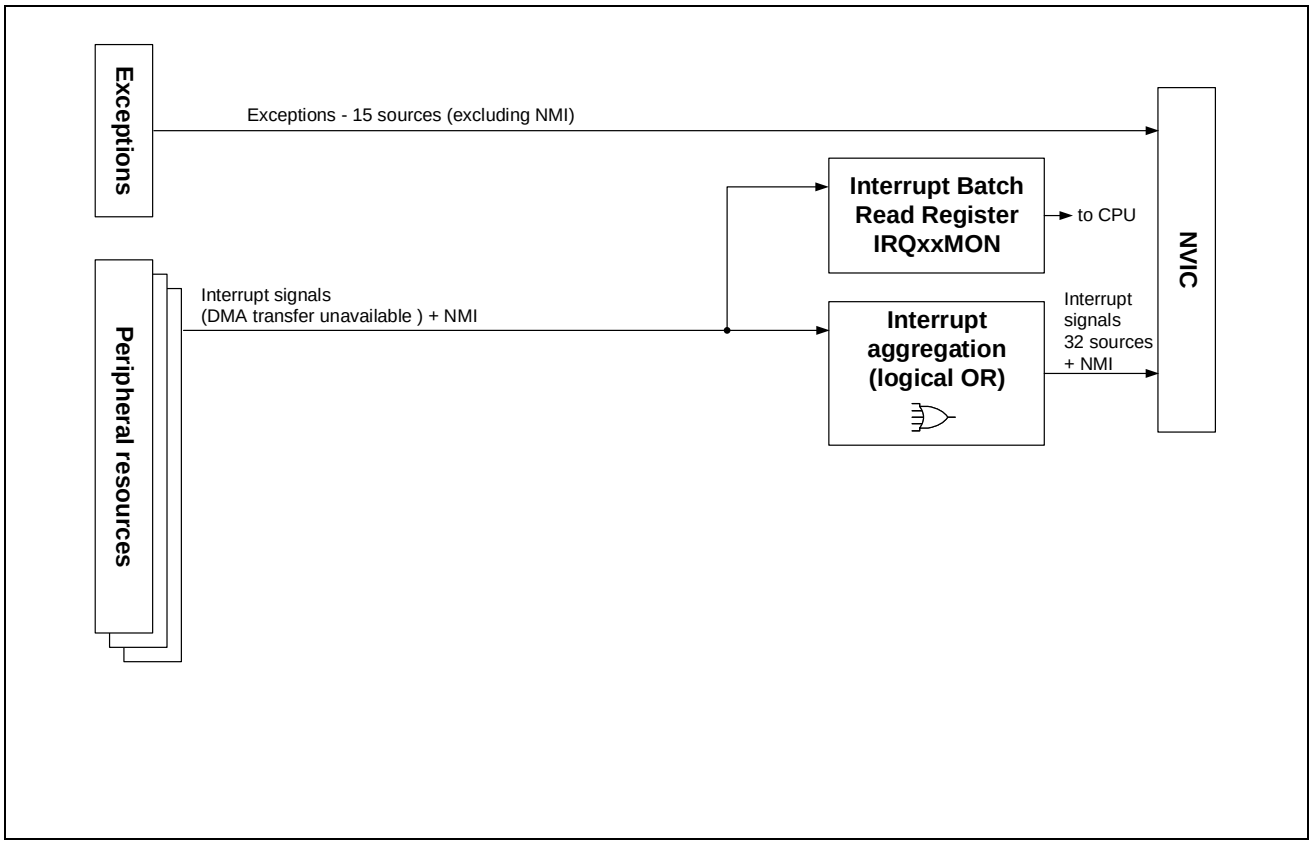
The interrupt request batch read register allows the interrupt request signals from the peripheral resources aggregated into a single interrupt request signal to be read out at once. Reading this register makes it possible to check which interrupt request has occurred. However, the interrupt request flags cannot be cleared by using this function. Clear the interrupt request flags using the registers of each peripheral function.

2. Configuration

This section shows the configuration of the relationship between the interrupt controller (Type-1).

Block Diagram of Interrupt Controller (Type-1)

Figure 2-1 Block Diagram of Interrupt Controller (Type-1)



- Interrupt factor aggregation block

  Aggregate (logical OR) interrupt request signals from each peripheral resource to 32 factors and output them to NVIC.

- Peripheral interrupt request batch read register block

  For interrupt request signals from a peripheral resource aggregated to one interrupt request signal, this register can check what interrupt request of each peripheral resource signal generates such interrupt.

■ Interrupt factor vector relocate function

Two types of the interrupt factor vector shown in 2 can be selected by the IRQCMODE register setting. For IRQCMODE bit, refer to "Interrupts (B)".For the details of each setting, refer to the each chapter as following Table 1-1.

Moreover, the arbitrary interrupt factor can be selected with the RCINTSEL0 and RCINTSEL1 registers. For details on RCINTSEL0 and RCINTSEL1 registers, see "Interrupts (B)".

Table 1-1 Correspondence Table for Interrupt Chapter

IRQCMODE Setting	Reference
IRQCMODE=0 Relocate not selected	Chapter "Interrupts (A)"
IRQCMODE=1 Relocate selected	Chapter "Interrupts (B)"

Table 1-2 Exceptions and Interrupt Factor Vectors List

Vector No.	IRQ No.	Exceptions and Interrupt Factor Vectors	
		IRQCMODE=0	IRQCMODE=1
0	-	Stack pointer initial value	
1	-	Reset	
2	-	Non-Maskable Interrupt (NMI) / Hardware Watchdog Timer	
3	-	Hard Fault	
4	-	Reserved	
5	-	Reserved	
6	-	Reserved	
7 to 10	-	Reserved	
11	-	SVCall (Supervisor Call)	
12	-	Reserved	
13	-	Reserved	
14	-	PendSV	
15	-	SysTick	
16	0	Anomalous Frequency Detection by Clock Supervisor (FCS)	
17	1	Software Watchdog Timer	
18	2	Low Voltage Detector (LVD)	
19	3	MFT unit 0, unit 1, unit 2 Wave Form Generator / DTIF(Motor Emergency Stop)	Selecting the interrupt factor with RCINTSEL0 register
20	4	External Pin Interrupt ch.0 to ch.7	Selecting the interrupt factor with RCINTSEL0 register
21	5	External Pin Interrupt ch.8 to ch.31	Selecting the interrupt factor with RCINTSEL0 register
22	6	Dual Timer / Quad Counter (QPRC) ch.0, ch.1, ch.2	Selecting the interrupt factor with RCINTSEL0 register
23	7	Reception Interrupt of MFS ch.0 / Reception Interrupt of MFS ch.8	Selecting the interrupt factor with RCINTSEL1 register
24	8	Transmission Interrupt and Status Interrupt of MFS ch.0 / Transmission Interrupt and Status Interrupt of MFS ch.8	Selecting the interrupt factor with RCINTSEL1 register

Vector No.	IRQ No.	Exceptions and Interrupt Factor Vectors	
		IRQCMODE=0	IRQCMODE=1
25	9	Reception Interrupt of MFS ch.1 / Reception Interrupt of MFS ch.9	Selecting the interrupt factor with RCINTSEL1 register
26	10	Transmission Interrupt and Status Interrupt of MFS ch.1 / Transmission Interrupt and Status Interrupt of MFS ch.9	Selecting the interrupt factor with RCINTSEL1 register
27	11	Reception Interrupt of MFS ch.2 / Reception Interrupt of MFS ch.10	MFT unit 0 Wave Form Generator / DTIF(Motor Emergency Stop) / Reception Interrupt, Transmission Interrupt, and Status Interrupt of MFS ch.8
28	12	Transmission Interrupt and Status Interrupt of MFS ch.2 / Transmission Interrupt and Status Interrupt of MFS ch.10	External pin interrupt ch.0 to ch.7
29	13	Reception Interrupt of MFS ch.3 / Reception Interrupt of MFS ch.11	External pin interrupt ch.8 to ch.31
30	14	Transmission Interrupt and Status Interrupt of MFS ch.3 / Transmission Interrupt and Status Interrupt of MFS ch.11	Dual Timer / Quad Counter (QPRC) ch.0
31	15	Reception Interrupt of MFS ch.4 / Reception Interrupt of MFS ch.12	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.0
32	16	Transmission Interrupt and Status Interrupt of MFS ch.4 / Transmission Interrupt and Status Interrupt of MFS ch.12	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.1
33	17	Reception Interrupt of MFS ch.5 / Reception Interrupt of MFS ch.13	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.2
34	18	Transmission Interrupt and Status Interrupt of MFS ch.5 / Transmission Interrupt and Status Interrupt of MFS ch.13	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.3
35	19	Reception Interrupt of MFS ch.6 / Reception Interrupt of MFS ch.14	Reception Interrupt of MFS ch.4
36	20	Transmission Interrupt and Status Interrupt of MFS ch.6 / Transmission Interrupt and Status Interrupt of MFS ch.14	Transmission Interrupt and Status Interrupt of MFS ch.4
37	21	Reception Interrupt of MFS ch.7 / Reception Interrupt of MFS ch.15	Reception Interrupt of MFS ch.5
38	22	Transmission Interrupt and Status Interrupt of MFS ch.7 / Transmission Interrupt and Status Interrupt of MFS ch.15	Transmission Interrupt and Status Interrupt of MFS ch.5
39	23	PPG ch.0/2/4/8/10/12/16/18/20	
40	24	External Main OSC / External Sub OSC / Main PLL / Watch Counter/Real Time Counter	
41	25	A/D Converter unit 0	A/D Converter unit 0 /Reception Interrupt, Transmission Interrupt, and Status Interrupt of MFS ch.9
42	26	A/D Converter unit 1	A/D Converter unit 1 /Reception Interrupt, Transmission Interrupt, and Status Interrupt of MFS ch.10

Vector No.	IRQ No.	Exceptions and Interrupt Factor Vectors	
		IRQCMODE=0	IRQCMODE=1
43	27	A/D Converter unit 2 / LCD Controller	A/D Converter unit 2/ LCD Controller / Reception Interrupt, Transmission Interrupt, and Status Interrupt of MFS ch.11
44	28	MFT unit 0, unit 1, unit 2 Free-run Timer	MFT unit 0 Free-run Timer, Input Capture, Output Compare
45	29	MFT unit 0, unit 1, unit 2 Input Capture	MFT unit 1 Free-run Timer, Input Capture, Output Compare
46	30	MFT unit 0, unit 1, unit 2 Output Compare	MFT unit 2 Free-run Timer, Input Capture, Output Compare
47	31	Base Timer ch.0 to ch.7 / Flash RDY interrupt /Flash HANG interrupt	

CHAPTER 7-3: Interrupts (TYPE1-A)

This chapter explains Exception and Interrupt Factor Vectors and Registers (TYPE1) at IRQCMODE=0.

1. Exception and Interrupt Factor Vectors
2. Registers
3. Usage Precautions

CODE: 9AFIRQC_A-E01.0

1. Exception and Interrupt Factor Vectors

This section shows a vector table of the exceptions and interrupts input to the NVIC.

Table 1-1 Exception and Interrupt Factor Vectors

Vector No.	IRQ No.	Exception and Interrupt Factor	Vector Offset
0	-	Stack pointer initial value	0x00
1	-	Reset	0x04
2	-	Non-Maskable Interrupt (NMI) / Hardware Watchdog Timer	0x08
3	-	Hard Fault	0x0C
4	-	Reserved	0x10
5	-	Reserved	0x14
6	-	Reserved	0x18
7-10	-	Reserved	0x1C - 0x2B
11	-	SVCall (Supervisor Call)	0x2C
12	-	Reserved	0x30
13	-	Reserved	0x34
14	-	PendSV	0x38
15	-	SysTick	0x3C
16	0	Anomalous Frequency Detection by Clock Supervisor (FCS)	0x40
17	1	Software Watchdog Timer	0x44
18	2	Low Voltage Detector (LVD)	0x48
19	3	MFT unit 0, unit 1, unit 2 Wave Form Generator / DTIF(Motor Emergency Stop)	0x4C
20	4	External Pin Interrupt Request ch.0 to ch.7	0x50
21	5	External Pin Interrupt Request ch.8 to ch.31	0x54
22	6	Dual Timer / Quad Counter (QPRC) ch.0, ch.1, ch.2	0x58
23	7	Reception Interrupt Request of MFS ch.0 / Reception Interrupt Request of MFS ch.8	0x5C
24	8	Transmission Interrupt Request and Status Interrupt Request of MFS ch.0 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.8	0x60
25	9	Reception Interrupt Request of MFS ch.1 / Reception Interrupt Request of MFS ch.9	0x64
26	10	Transmission Interrupt Request and Status Interrupt Request of MFS ch.1 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.9	0x68
27	11	Reception Interrupt Request of MFS ch.2 / Reception Interrupt Request of MFS ch.10	0x6C
28	12	Transmission Interrupt Request and Status Interrupt Request of MFS ch.2 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.10	0x70
29	13	Reception Interrupt Request of MFS ch.3 / Reception Interrupt Request of MFS ch.11	0x74
30	14	Transmission Interrupt Request and Status Interrupt Request of MFS ch.3 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.11	0x78
31	15	Reception Interrupt Request of MFS ch.4 / Reception Interrupt Request of MFS ch.12	0x7C
32	16	Transmission Interrupt Request and Status Interrupt Request of MFS ch.4 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.12	0x80
33	17	Reception Interrupt Request of MFS ch.5 / Reception Interrupt Request of MFS ch.13	0x84
34	18	Transmission Interrupt Request and Status Interrupt Request of MFS ch.5 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.13	0x88

Vector No.	IRQ No.	Exception and Interrupt Factor	Vector Offset
35	19	Reception Interrupt Request of MFS ch.6 / Reception Interrupt Request of MFS ch.14	0x8C
36	20	Transmission Interrupt Request and Status Interrupt Request of MFS ch.6 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.14	0x90
37	21	Reception Interrupt Request of MFS ch.7 / Reception Interrupt Request of MFS ch.15	0x94
38	22	Transmission Interrupt Request and Status Interrupt Request of MFS ch.7 / Transmission Interrupt Request and Status Interrupt Request of MFS ch.15	0x98
39	23	PPG ch.0/2/4/8/10/12/16/18/20	0x9C
40	24	External Main OSC / External Sub OSC / Main PLL / Watch Counter/ Real Time Counter	0xA0
41	25	A/D Converter unit 0	0xA4
42	26	A/D Converter unit 1	0xA8
43	27	A/D Converter unit 2 / LCD Controller	0xAC
44	28	MFT unit 0, unit 1, unit 2 Free-run Timer	0xB0
45	29	MFT unit 0, unit 1, unit 2 Input Capture	0xB4
46	30	MFT unit 0, unit 1, unit 2 Output Compare	0xB8
47	31	Base Timer ch.0 to ch.7/ Flash RDY interrupt / Flash HANG interrupt	0xBC

The priorities of the exceptions for vectors No. 4 to No. 15 can be configured using the System Handler Priority Registers (address 0xE000ED18, 0xE000ED1C, 0xE000ED20) built into the NVIC. The priorities of the peripheral interrupts for vectors No. 16 and after can be configured using the IRQ Priority Registers (address 0xE000E400 to 0xE000E42C) built into the NVIC.

The sources of the interrupts for vectors No. 2 and No. 16 to No. 47 can be checked using the batch read register. See "Chapter 8: Nested Vectored Interrupt Controller" in the "Cortex-M0+ Technical Reference Manual" for details on the other exceptions and interrupts.

Furthermore, for the interrupts of vectors No. 2 and No. 16 to No. 47, the sources that are batch read may be a signal that multiple interrupt factors are logical OR’ed within each of the peripheral macros. See the descriptions of each of the peripheral resource interrupts for details.

2. Registers

This section describes the DMA transfer request selection registers and the interrupt request batch read registers.

List of DMA Transfer Request Selection Registers and Interrupt Request Batch Read Registers

Table 2-1 List of DMA Transfer Request Selection Registers and Interrupt Request Batch Read Registers

Abbreviation	Register Name	Reference
DRQSEL	DMA Request Selection Register	2.1
EXC02MON	EXC02 Batch Read Register	2.2
IRQ00MON	IRQ00 Batch Read Register	2.3
IRQ01MON	IRQ01 Batch Read Register	2.4
IRQ02MON	IRQ02 Batch Read Register	2.5
IRQ03MON	IRQ03 Batch Read Register	2.6
IRQ04MON	IRQ04 Batch Read Register	2.7
IRQ05MON	IRQ05 Batch Read Register	2.8
IRQ06MON	IRQ06 Batch Read Register	2.9
IRQ07MON	IRQ07 Batch Read Register	2.10
IRQ08MON	IRQ08 Batch Read Register	2.12
IRQ09MON	IRQ09 Batch Read Register	2.10
IRQ10MON	IRQ10 Batch Read Register	2.12
IRQ11MON	IRQ11 Batch Read Register	2.10
IRQ12MON	IRQ12 Batch Read Register	2.12
IRQ13MON	IRQ13 Batch Read Register	2.10
IRQ14MON	IRQ14 Batch Read Register	2.12
IRQ15MON	IRQ15 Batch Read Register	2.10
IRQ16MON	IRQ16 Batch Read Register	2.12
IRQ17MON	IRQ17 Batch Read Register	2.10
IRQ18MON	IRQ18 Batch Read Register	2.12
IRQ19MON	IRQ19 Batch Read Register	2.11
IRQ20MON	IRQ20 Batch Read Register	2.13
IRQ21MON	IRQ21 Batch Read Register	2.11
IRQ22MON	IRQ22 Batch Read Register	2.13
IRQ23MON	IRQ23 Batch Read Register	2.14
IRQ24MON	IRQ24 Batch Read Register	2.15
IRQ25MON	IRQ25 Batch Read Register	2.16
IRQ26MON	IRQ26 Batch Read Register	2.16
IRQ27MON	IRQ27 Batch Read Register	2.17
IRQ28MON	IRQ28 Batch Read Register	2.18
IRQ29MON	IRQ29 Batch Read Register	2.19
IRQ30MON	IRQ30 Batch Read Register	2.20
IRQ31MON	IRQ31 Batch Read Register	2.21

See "Chapter 5: Nested Vectored Interrupt Controller" in the "Cortex-M0+ Technical Reference Manual" for details on the registers in the NVIC.

2.1 DMA Request Selection Register (DRQSEL)

The DMA Request Selection Register (DRQSEL) selects whether interrupt signals that can start DMA transfers are output as interrupt requests to the CPU or output as transfer requests to the DMAC.

Register Configuration

bit	31	5	4	3	2	1	0
Field	Reserved	Reserved					
Attribute	R/W	R					
Initial value	00000000000000000000000000000000	00000					

Register Functions

[bit31:5] Reserved: Reserved bits

Write 0.to these bits.

[bit4:0] Reserved: Reserved bits

A reserved bit reads "0".

2.2 EXC02 Batch Read Register (EXC02MON)

The EXC02 Batch Read Register (EXC02MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 2.

EXC02MON indicates the status of the interrupt requests of the hardware watchdog timer and NMIX external pin.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved														HWINT	NMI
Attribute	R														R	R
Initial value	00000000000000														0	0

Register Functions

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1] HWINT:

Bit	Description
0	No hardware watchdog timer interrupt request
1	Hardware watchdog timer interrupt request

[bit0] NMI:

Bit	Description
0	No NMIX external pin interrupt request
1	NMIX external pin interrupt request

2.3 IRQ00 Batch Read Register (IRQ00MON)

The IRQ00 Batch Read Register (IRQ00MON) can batch-read the interrupt request allocated to interrupt factor vector No. 16.

IRQ00MON indicates the status of the interrupt request of anomalous frequency detection by the CSV.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															FCSINT
Attribute	R															R
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] FCSINT:

Bit	Description
0	No anomalous frequency detection by CSV interrupt request
1	Anomalous frequency detection by CSV interrupt request

2.4 IRQ01 Batch Read Register (IRQ01MON)

The IRQ01 Batch Read Register (IRQ01MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 17.

IRQ01MON indicates the status of the interrupt request of the software watchdog timer.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															SWWDTINT
Attribute	R															R
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] SWWDTINT:

Bit	Description
0	No software watchdog timer interrupt request
1	Software watchdog timer interrupt request

2.5 IRQ02 Batch Read Register (IRQ02MON)

The IRQ02 Batch Read Register (IRQ02MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 18.

IRQ02MON indicates the status of the interrupt request of low voltage detection (LVD).

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															LVDINT
Attribute	R															R
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] LVDINT:

Bit	Description
0	No low voltage detection (LVD) interrupt request
1	Low voltage detection (LVD) interrupt request

2.6 IRQ03 Batch Read Register (IRQ03MON)

The IRQ03 Batch Read Register (IRQ03MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 19.

IRQ02MON indicates the status of the interrupt requests of MFT unit 0, MFT unit 1 and MFT unit 2.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				WAVE2INT				WAVE1INT				WAVE0INT			
Attribute	R				R				R				R			
Initial value	0000				0000				0000				0000			

Register Functions

[bit31:12] Reserved: Reserved bits

A reserved bit reads "0".

[bit11:8] WAVE2INT:

Bit No.	Bit	Description
11	0	No WFG timer 54 interrupt request in MFT unit 2
	1	WFG timer 54 interrupt request in MFT unit 2
10	0	No WFG timer 32 interrupt request in MFT unit 2
	1	WFG timer 32 interrupt request in MFT unit 2
9	0	No WFG timer 10 interrupt request in MFT unit 2
	1	WFG timer 10 interrupt request in MFT unit 2
8	0	No DTIF (motor emergency stop) interrupt request in MFT unit 2
	1	DTIF (motor emergency stop) interrupt request in MFT unit 2

[bit7:4] WAVE1INT:

Bit No.	Bit	Description
7	0	No WFG timer 54 interrupt request in MFT unit 1
	1	WFG timer 54 interrupt request in MFT unit 1
6	0	No WFG timer 32 interrupt request in MFT unit 1
	1	WFG timer 32 interrupt request in MFT unit 1
5	0	No WFG timer 10 interrupt request in MFT unit 1
	1	WFG timer 10 interrupt request in MFT unit 1
4	0	No DTIF (motor emergency stop) interrupt request in MFT unit 1
	1	DTIF (motor emergency stop) interrupt request in MFT unit 1

[bit3:0] WAVE0INT:

Bit No.	Bit	Description
3	0	No WFG timer 54 interrupt request in MFT unit 0
	1	WFG timer 54 interrupt request in MFT unit 0
2	0	No WFG timer 32 interrupt request in MFT unit 0
	1	WFG timer 32 interrupt request in MFT unit 0
1	0	No WFG timer 10 interrupt request in MFT unit 0
	1	WFG timer 10 interrupt request in MFT unit 0
0	0	No DTIF (motor emergency stop) interrupt request in MFT unit 0
	1	DTIF (motor emergency stop) interrupt request in MFT unit 0

2.7 IRQ04 Batch Read Register (IRQ04MON)

The IRQ04 Batch Read Register (IRQ04MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 20.

IRQ04MON indicates the status of the interrupt requests of the external interrupt ch.0 to ch.7.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								EXTINT							
Attribute	R								R							
Initial value	0x00								0x00							

Register Functions

[bit31:8] Reserved: Reserved bits

A reserved bit reads "0".

[bit7:0] EXTINT:

Bit No.	Bit	Description
7	0	No interrupt request of external interrupt ch.7
	1	Interrupt request of external interrupt ch.7
6	0	No interrupt request of external interrupt ch.6
	1	Interrupt request of external interrupt ch.6
5	0	No interrupt request of external interrupt ch.5
	1	Interrupt request of external interrupt ch.5
4	0	No interrupt request of external interrupt ch.4
	1	Interrupt request of external interrupt ch.4
3	0	No interrupt request of external interrupt ch.3
	1	Interrupt request of external interrupt ch.3
2	0	No interrupt request of external interrupt ch.2
	1	Interrupt request of external interrupt ch.2
1	0	No interrupt request of external interrupt ch.1
	1	Interrupt request of external interrupt ch.1
0	0	No interrupt request of external interrupt ch.0
	1	Interrupt request of external interrupt ch.0

2.8 IRQ05 Batch Read Register (IRQ05MON)

The IRQ05 Batch Read Register (IRQ05MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 21.

IRQ05MON indicates the status of the interrupt requests of the external interrupt ch.8 to ch.31.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved								EXTINT							
Attribute	R								R							
Initial value	0x00								0x00							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EXTINT															
Attribute	R															
Initial value	0x0000															

Register Functions

[bit31:24] Reserved: Reserved bits

A reserved bit reads "0".

[bit23:0] EXTINT:

Bit No.	Bit	Description
23	0	No interrupt request of external interrupt ch.31
	1	Interrupt request of external interrupt ch.31
22	0	No interrupt request of external interrupt ch.30
	1	Interrupt request of external interrupt ch.30
21	0	No interrupt request of external interrupt ch.29
	1	Interrupt request of external interrupt ch.29
20	0	No interrupt request of external interrupt ch.28
	1	Interrupt request of external interrupt ch.28
19	0	No interrupt request of external interrupt ch.27
	1	Interrupt request of external interrupt ch.27
18	0	No interrupt request of external interrupt ch.26
	1	Interrupt request of external interrupt ch.26
17	0	No interrupt request of external interrupt ch.25
	1	Interrupt request of external interrupt ch.25
16	0	No interrupt request of external interrupt ch.24
	1	Interrupt request of external interrupt ch.24
15	0	No interrupt request of external interrupt ch.23
	1	Interrupt request of external interrupt ch.23
14	0	No interrupt request of external interrupt ch.22
	1	Interrupt request of external interrupt ch.22

Bit No.	Bit	Description
13	0	No interrupt request of external interrupt ch.21
	1	Interrupt request of external interrupt ch.21
12	0	No interrupt request of external interrupt ch.20
	1	Interrupt request of external interrupt ch.20
11	0	No interrupt request of external interrupt ch.19
	1	Interrupt request of external interrupt ch.19
10	0	No interrupt request of external interrupt ch.18
	1	Interrupt request of external interrupt ch.18
9	0	No interrupt request of external interrupt ch.17
	1	Interrupt request of external interrupt ch.17
8	0	No interrupt request of external interrupt ch.16
	1	Interrupt request of external interrupt ch.16
7	0	No interrupt request of external interrupt ch.15
	1	Interrupt request of external interrupt ch.15
6	0	No interrupt request of external interrupt ch.14
	1	Interrupt request of external interrupt ch.14
5	0	No interrupt request of external interrupt ch.13
	1	Interrupt request of external interrupt ch.13
4	0	No interrupt request of external interrupt ch.12
	1	Interrupt request of external interrupt ch.12
3	0	No interrupt request of external interrupt ch.11
	1	Interrupt request of external interrupt ch.11
2	0	No interrupt request of external interrupt ch.10
	1	Interrupt request of external interrupt ch.10
1	0	No interrupt request of external interrupt ch.9
	1	Interrupt request of external interrupt ch.9
0	0	No interrupt request of external interrupt ch.8
	1	Interrupt request of external interrupt ch.8

2.9 IRQ06 Batch Read Register (IRQ06MON)

The IRQ06 Batch Read Register (IRQ06MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 22.

IRQ06MON indicates the status of the interrupt requests of the QPRC and dual timer.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved												QUD2INT			
Attribute	R												R			
Initial value	0x000												0000			

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	QUD2INT		QUD1INT						QUD0INT						TIMINT	
Attribute	R		R						R						R	
Initial value	0x0000		000000						000000						00	

Register Functions

[bit31:20] Reserved: Reserved bits

A reserved bit reads "0".

[bit19:14] QUD2INT:

Bit No.	Bit	Description
19	0	No PC match & RC match interrupt request of QPRC ch.2
	1	PC match & RC match interrupt request of QPRC ch.2
18	0	No interrupt request detected RC out of range on QPRC ch.2
	1	Interrupt request detected RC out of range on QPRC ch.2
17	0	No PC count invert interrupt request of QPRC ch.2
	1	PC count invert interrupt request of QPRC ch.2
16	0	No overflow/underflow/zero index interrupt request of QPRC ch.2
	1	Overflow/underflow/zero index interrupt request of QPRC ch.2
15	0	No PC&RC match interrupt request of QPRC ch.2
	1	PC&RC match interrupt request of QPRC ch.2
14	0	No PC match interrupt request of QPRC ch.2
	1	PC match interrupt request of QPRC ch.2

[bit13:8] QUD1INT:

Bit No.	Bit	Description
13	0	No PC match & RC match interrupt request of QPRC ch.1
	1	PC match & RC match interrupt request of QPRC ch.1
12	0	No interrupt request detected RC out of range on QPRC ch.1
	1	Interrupt request detected RC out of range on QPRC ch.1
11	0	No PC count invert interrupt request of QPRC ch.1
	1	PC count invert interrupt request of QPRC ch.1
10	0	No overflow/underflow/zero index interrupt request of QPRC ch.1
	1	Overflow/underflow/zero index interrupt request of QPRC ch.1
9	0	No PC&RC match interrupt request of QPRC ch.1
	1	PC&RC match interrupt request of QPRC ch.1
8	0	No PC match interrupt request of QPRC ch.1
	1	PC match interrupt request of QPRC ch.1

[bit7:2] QUD0INT:

Bit No.	Bit	Description
7	0	No PC match & RC match interrupt request of QPRC ch.0
	1	PC match & RC match interrupt request of QPRC ch.0
6	0	No interrupt request detected RC out of range on QPRC ch.0
	1	Interrupt request detected RC out of range on QPRC ch.0
5	0	No PC count invert interrupt request of QPRC ch.0
	1	PC count invert interrupt request of QPRC ch.0
4	0	No overflow/underflow/zero index interrupt request of QPRC ch.0
	1	Overflow/underflow/zero index interrupt request of QPRC ch.0
3	0	No PC&RC match interrupt request of QPRC ch.0
	1	PC&RC match interrupt request of QPRC ch.0
2	0	No PC match interrupt request of QPRC ch.0
	1	PC match interrupt request of QPRC ch.0

[bit1:0] TIMINT:

Bit No.	Bit	Description
1	0	No dual timer TIMINT2 interrupt request
	1	Dual timer TIMINT2 interrupt request
0	0	No dual timer TIMINT1 interrupt request
	1	Dual timer TIMINT1 interrupt request

2.10 IRQ07/09/11/13/15/17 Batch Read Register (IRQxxMON)

The IRQ07 Batch Read Register (IRQ07MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 23.
The IRQ09 Batch Read Register (IRQ09MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 25.
The IRQ11 Batch Read Register (IRQ11MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 27.
The IRQ13 Batch Read Register (IRQ13MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 29.
The IRQ15 Batch Read Register (IRQ15MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 31.
The IRQ17 Batch Read Register (IRQ17MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 33.

IRQ07MON indicates the status of the reception interrupt request of MFS ch.0 / ch.8.
IRQ09MON indicates the status of the reception interrupt request of MFS ch.1 / ch.9.
IRQ11MON indicates the status of the reception interrupt request of MFS ch.2 / ch.10.
IRQ13MON indicates the status of the reception interrupt request of MFS ch.3 / ch.11.
IRQ15MON indicates the status of the reception interrupt request of MFS ch.4 / ch.12.
IRQ17MON indicates the status of the reception interrupt request of MFS ch.5 / ch.13.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved														MFSINT	
Attribute	R														R	
Initial value	00000000000000														00	

Register Functions

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1:0] MFSINT:

Bit No.	Bit	Description
1	0	No reception interrupt request of the corresponding MFS channel (ch.8 to ch.13)
	1	Reception interrupt request of the corresponding MFS channel (ch.8 to ch.13)
0	0	No reception interrupt request of the corresponding MFS channel (ch.0 to ch.5)
	1	Reception interrupt request of the corresponding MFS channel (ch.0 to ch.5)

2.11 IRQ19/21 Batch Read Register (IRQxxMON)

The IRQ19 Batch Read Register (IRQ19MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 35.
The IRQ21 Batch Read Register (IRQ21MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 37.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved											Reserved	Reserved		MFSINT	
Attribute	R											R	R		R	
Initial value	00000000000											0	00		00	

Register Functions

[bit31:5] Reserved: Reserved bits
A reserved bit reads "0".

[bit4] Reserved: Reserved bit
A reserved bit reads "0".

[bit3:2] Reserved: Reserved bits
A reserved bit reads "0".

[bit1:0] MFSINT:

Bit No.	Bit	Description
1	0	No reception interrupt request of the corresponding MFS channel (ch.14, ch.15)
	1	Reception interrupt request of the corresponding MFS channel (ch.14, ch.15)
0	0	No reception interrupt request of the corresponding MFS channel (ch.6, ch.7)
	1	Reception interrupt request of the corresponding MFS channel (ch.6, ch.7)

2.12 IRQ08/10/12/14/16/18 Batch Read Register (IRQxxMON)

The IRQ08 Batch Read Register (IRQ08MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 24.
The IRQ10 Batch Read Register (IRQ10MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 26.
The IRQ12 Batch Read Register (IRQ12MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 28.
The IRQ14 Batch Read Register (IRQ14MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 30.
The IRQ16 Batch Read Register (IRQ16MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 32.
The IRQ18 Batch Read Register (IRQ18MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 34.

IRQ08MON indicates the status of the transmission interrupt request and the status interrupt request of MFS ch.0 and ch.8.
IRQ10MON indicates the status of the transmission interrupt request and the status interrupt request of MFS ch.1 and ch.9.
IRQ12MON indicates the status of the transmission interrupt request and the status interrupt request of MFS ch.2 and ch.10.
IRQ14MON indicates the status of the transmission interrupt request and the status interrupt request of MFS ch.3 and ch.11.
IRQ16MON indicates the status of the transmission interrupt request and the status interrupt request of MFS ch.4 and ch.12.
IRQ18MON indicates the status of the transmission interrupt request and the status interrupt request of MFS ch.5 and ch.13.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved												MFSINT			
Attribute	R												R			
Initial value	00000000000000												0000			

Register Functions

[bit31:4] Reserved: Reserved bits
A reserved bit reads "0".

[bit3:0] MFSINT:

Bit No.	Bit	Description
3	0	No status interrupt request of the corresponding MFS channel (ch.8 to ch.13)
	1	Status interrupt request of the corresponding MFS channel (ch.8 to ch.13)
2	0	No transmission interrupt request of the corresponding MFS channel (ch.8 to ch.13)
	1	Transmission interrupt request of the corresponding MFS channel (ch.8 to ch.13)
1	0	No status interrupt request of the corresponding MFS channel (ch.0 to ch.5)
	1	Status interrupt request of the corresponding MFS channel (ch.0 to ch.5)
0	0	No transmission interrupt request of the corresponding MFS channel (ch.0 to ch.5)
	1	Transmission interrupt request of the corresponding MFS channel (ch.0 to ch.5)

2.13 IRQ20/22 Batch Read Register (IRQxxMON)

The IRQ20 Batch Read Register (IRQ20MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 36.
The IRQ22 Batch Read Register (IRQ22MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 38.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved											Reserved	MFSINT			
Attribute	R											R	R			
Initial value	00000000000											0	0000			

Register Functions

[bit31:5] Reserved: Reserved bits

A reserved bit reads "0".

[bit4] Reserved: Reserved bit

A reserved bit reads "0".

[bit3:0] MFSINT:

Bit No.	Bit	Description
3	0	No status interrupt request of the corresponding MFS channel (ch.14, ch.15)
	1	Status interrupt request of the corresponding MFS channel (ch.14, ch.15)
2	0	No transmission interrupt request of the corresponding MFS channel (ch.14, ch.15)
	1	Transmission interrupt request of the corresponding MFS channel (ch.14, ch.15)
1	0	No status interrupt request of the corresponding MFS channel (ch.6, ch.7)
	1	Status interrupt request of the corresponding MFS channel (ch.6, ch.7)
0	0	No transmission interrupt request of the corresponding MFS channel (ch.6, ch.7)
	1	Transmission interrupt request of the corresponding MFS channel (ch.6, ch.7)

2.14 IRQ23 Batch Read Register (IRQ23MON)

The IRQ23 Batch Read Register (IRQ23MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 39.

IRQ23MON indicates the status of the interrupt request of the PPG.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved							PPGINT								
Attribute	R							R								
Initial value	0000000							000000000								

Register Functions

[bit31:9] Reserved: Reserved bits

A reserved bit reads "0".

[bit8:0] PPGINT:

Bit No.	Bit	Description
8	0	No interrupt request of PPG ch.20
	1	Interrupt request of PPG ch.20
7	0	No interrupt request of PPG ch.18
	1	Interrupt request of PPG ch.18
6	0	No interrupt request of PPG ch.16
	1	Interrupt request of PPG ch.16
5	0	No interrupt request of PPG ch.12
	1	Interrupt request of PPG ch.12
4	0	No interrupt request of PPG ch.10
	1	Interrupt request of PPG ch.10
3	0	No interrupt request of PPG ch.8
	1	Interrupt request of PPG ch.8
2	0	No interrupt request of PPG ch.4
	1	Interrupt request of PPG ch.4
1	0	No interrupt request of PPG ch.2
	1	Interrupt request of PPG ch.2
0	0	No interrupt request of PPG ch.0
	1	Interrupt request of PPG ch.0

2.15 IRQ24 Batch Read Register (IRQ24MON)

The IRQ24 Batch Read Register (IRQ24MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 40.

IRQ24MON indicates the status of the interrupt requests of the RTC, watch counter, main PLL oscillation, sub oscillation and main clock oscillation.

Register Configuration

bit	31							8
Field	Reserved							
Attribute	R							
Initial value	0x000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved		RTCINT	WCINT	Reserved	MPLLINT	SOSCINT	MOSCINT
Attribute	R		R	R	R	R	R	R
Initial value	00		0	0	0	0	0	0

Register Functions

[bit31:6] Reserved: Reserved bits

A reserved bit reads "0".

[bit5] RTCINT:

Bit	Description
0	No RTC interrupt request
1	RTC interrupt request

[bit4] WCINT:

Bit	Description
0	No watch counter interrupt request
1	Watch counter interrupt request

[bit3] Reserved: Reserved bit

A reserved bit reads "0".

[bit2] MPLLINT:

Bit	Description
0	No stabilization wait completion interrupt request for main PLL oscillation
1	Stabilization wait completion interrupt request for main PLL oscillation

[bit1] SOSCINT:

Bit	Description
0	No stabilization wait completion interrupt request for sub-clock oscillation
1	Stabilization wait completion interrupt request for sub-clock oscillation

[bit0] MOSCINT:

Bit	Description
0	No stabilization wait completion interrupt request for main clock oscillation
1	Stabilization wait completion interrupt request for main clock oscillation

2.16 IRQ25/26 Batch Read Register (IRQxxMON)

The IRQ25 Batch Read Register (IRQ25MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 41.
The IRQ26 Batch Read Register (IRQ26MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 42.

IRQ25MON indicates the status of the interrupt request of A/D converter unit 0.
IRQ26MON indicates the status of the interrupt request of A/D converter unit 1.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved											ADCINT				
Attribute	R											R				
Initial value	00000000000											00000				

Register Functions

[bit31:5] Reserved: Reserved bits

A reserved bit reads "0".

[bit4:0] ADCINT:

Bit No.	Bit	Description
4	0	No range comparison result interrupt request in the corresponding A/D converter unit
	1	Range comparison result interrupt request in the corresponding A/D converter unit
3	0	No conversion result comparison interrupt request in the corresponding A/D converter unit
	1	Conversion result comparison interrupt request in the corresponding A/D converter unit
2	0	No FIFO overrun interrupt request in the corresponding A/D converter unit
	1	FIFO overrun interrupt request in the corresponding A/D converter unit
1	0	No scan conversion interrupt request in the corresponding A/D converter unit
	1	Scan conversion interrupt request in the corresponding A/D converter unit
0	0	No priority conversion interrupt request in the corresponding A/D converter unit
	1	Priority conversion interrupt request in the corresponding A/D converter unit

2.17 IRQ27 Batch Read Register (IRQ27MON)

The IRQ27 Batch Read Register (IRQ27MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 43.

IRQ27MON indicates the status of the interrupt requests of A/D converter unit 2 and LCD controller.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved										LCDCINT	ADCINT				
Attribute	R										R	R				
Initial value	0000000000										0	00000				

Register Functions

[bit31:6] Reserved: Reserved bits

A reserved bit reads "0".

[bit5] LCDCINT:

Bit	Description
0	No interrupt request for LCD controller
1	Interrupt request for LCD controller

[bit4:0] ADCINT:

Bit No.	Bit	Description
4	0	No range comparison result interrupt request in the corresponding A/D converter unit 2
	1	Range comparison result interrupt request in the corresponding A/D converter unit 2
3	0	No conversion result comparison interrupt request in the A/D converter unit 2
	1	Conversion result comparison interrupt request in the A/D converter unit 2
2	0	No FIFO overrun interrupt request in the A/D converter unit 2
	1	FIFO overrun interrupt request in the A/D converter unit 2
1	0	No scan conversion interrupt request in the A/D converter unit 2
	1	Scan conversion interrupt request in the A/D converter unit 2
0	0	No priority conversion interrupt request in the A/D converter unit 2
	1	Priority conversion interrupt request in the A/D converter unit 2

2.18 IRQ28 Batch Read Register (IRQ28MON)

The IRQ28 Batch Read Register (IRQ28MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 44.

IRQ28MON indicates the status of the interrupt requests of MFT unit 0, MFT unit 1 and MFT unit 2.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved														FRT2INT	
Attribute	R														R	
Initial value	00000000000000														00	

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	FRT2INT				FRT1INT						FRT0INT					
Attribute	R				R						R					
Initial value	0000				000000						000000					

Register Functions

[bit31:18] Reserved: Reserved bits

A reserved bit reads "0".

[bit17:12] FRT2INT:

Bit No.	Bit	Description
17	0	No zero detection interrupt request of the free run timer ch.2 in the MFT unit 2
	1	Zero detection interrupt request of the free run timer ch.2 in the MFT unit 2
16	0	No zero detection interrupt request of the free run timer ch.1 in the MFT unit 2
	1	Zero detection interrupt request of the free run timer ch.1 in the MFT unit 2
15	0	No zero detection interrupt request of the free run timer ch.0 in the MFT unit 2
	1	Zero detection interrupt request of the free run timer ch.0 in the MFT unit 2
14	0	No peak value detection interrupt request of the free run timer ch.2 in the MFT unit 2
	1	Peak value detection interrupt request of the free run timer ch.2 in the MFT unit 2
13	0	No peak value detection interrupt request of the free run timer ch.1 in the MFT unit 2
	1	Peak value detection interrupt request of the free run timer ch.1 in the MFT unit 2
12	0	No peak value detection interrupt request of the free run timer ch.0 in the MFT unit 2
	1	Peak value detection interrupt request of the free run timer ch.0 in the MFT unit 2

[bit11:6] FRT1INT:

Bit No.	Bit	Description
11	0	No zero detection interrupt request of the free run timer ch.2 in the MFT unit 1
	1	Zero detection interrupt request of the free run timer ch.2 in the MFT unit 1
10	0	No zero detection interrupt request of the free run timer ch.1 in the MFT unit 1
	1	Zero detection interrupt request of the free run timer ch.1 in the MFT unit 1
9	0	No zero detection interrupt request of the free run timer ch.0 in the MFT unit 1
	1	Zero detection interrupt request of the free run timer ch.0 in the MFT unit 1
8	0	No peak value detection interrupt request of the free run timer ch.2 in the MFT unit 1
	1	Peak value detection interrupt request of the free run timer ch.2 in the MFT unit 1
7	0	No peak value detection interrupt request of the free run timer ch.1 in the MFT unit 1
	1	Peak value detection interrupt request of the free run timer ch.1 in the MFT unit 1
6	0	No peak value detection interrupt request of the free run timer ch.0 in the MFT unit 1
	1	Peak value detection interrupt request of the free run timer ch.0 in the MFT unit 1

[bit5:0] FRT0INT:

Bit No.	Bit	Description
5	0	No zero detection interrupt request of the free run timer ch.2 in the MFT unit 0
	1	Zero detection interrupt request of the free run timer ch.2 in the MFT unit 0
4	0	No zero detection interrupt request of the free run timer ch.1 in the MFT unit 0
	1	Zero detection interrupt request of the free run timer ch.1 in the MFT unit 0
3	0	No zero detection interrupt request of the free run timer ch.0 in the MFT unit 0
	1	Zero detection interrupt request of the free run timer ch.0 in the MFT unit 0
2	0	No peak value detection interrupt request of the free run timer ch.2 in the MFT unit 0
	1	Peak value detection interrupt request of the free run timer ch.2 in the MFT unit 0
1	0	No peak value detection interrupt request of the free run timer ch.1 in the MFT unit 0
	1	Peak value detection interrupt request of the free run timer ch.1 in the MFT unit 0
0	0	No peak value detection interrupt request of the free run timer ch.0 in the MFT unit 0
	1	Peak value detection interrupt request of the free run timer ch.0 in the MFT unit 0

2.19 IRQ29 Batch Read Register (IRQ29MON)

The IRQ29 Batch Read Register (IRQ29MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 45.

IRQ29MON indicates the status of the interrupt requests of MFT unit 0, MFT unit 1 and MFT unit 2.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				ICU2INT				ICU1INT				ICU0INT			
Attribute	R				R				R				R			
Initial value	0000				0000				0000				0000			

Register Functions

[bit31:12] Reserved: Reserved bits

A reserved bit reads "0".

[bit11:8] ICU2INT:

Bit No.	Bit	Description
11	0	No interrupt request of the input capture ch.3 in the MFT unit 2
	1	Interrupt request of the input capture ch.3 in the MFT unit 2
10	0	No interrupt request of the input capture ch.2 in the MFT unit 2
	1	Interrupt request of the input capture ch.2 in the MFT unit 2
9	0	No interrupt request of the input capture ch.1 in the MFT unit 2
	1	Interrupt request of the input capture ch.1 in the MFT unit 2
8	0	No interrupt request of the input capture ch.0 in the MFT unit 2
	1	Interrupt request of the input capture ch.0 in the MFT unit 2

[bit7:4] ICU1INT:

Bit No.	Bit	Description
7	0	No interrupt request of the input capture ch.3 in the MFT unit 1
	1	Interrupt request of the input capture ch.3 in the MFT unit 1
6	0	No interrupt request of the input capture ch.2 in the MFT unit 1
	1	Interrupt request of the input capture ch.2 in the MFT unit 1
5	0	No interrupt request of the input capture ch.1 in the MFT unit 1
	1	Interrupt request of the input capture ch.1 in the MFT unit 1
4	0	No interrupt request of the input capture ch.0 in the MFT unit 1
	1	Interrupt request of the input capture ch.0 in the MFT unit 1

[bit3:0] ICU0INT:

Bit No.	Bit	Description
3	0	No interrupt request of the input capture ch.3 in the MFT unit 0
	1	Interrupt request of the input capture ch.3 in the MFT unit 0
2	0	No interrupt request of the input capture ch.2 in the MFT unit 0
	1	Interrupt request of the input capture ch.2 in the MFT unit 0
1	0	No interrupt request of the input capture ch.1 in the MFT unit 0
	1	Interrupt request of the input capture ch.1 in the MFT unit 0
0	0	No interrupt request of the input capture ch.0 in the MFT unit 0
	1	Interrupt request of the input capture ch.0 in the MFT unit 0

2.20 IRQ30 Batch Read Register (IRQ30MON)

The IRQ30 Batch Read Register (IRQ30MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 46.

IRQ30MON indicates the status of the interrupt requests of MFT unit 0, MFT unit 1 and MFT unit 2.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved														OCU2INT	
Attribute	R														R	
Initial value	00000000000000														00	

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OCU2INT				OCU1INT						OCU0INT					
Attribute	R				R						R					
Initial value	0000				000000						000000					

Register Functions

[bit31:18] Reserved: Reserved bits

A reserved bit reads "0".

[bit17:12] OCU2INT:

Bit No.	Bit	Description
17	0	No interrupt request of the output compare ch.5 in the MFT unit 2
	1	Interrupt request of the output compare ch.5 in the MFT unit 2
16	0	No interrupt request of the output compare ch.4 in the MFT unit 2
	1	Interrupt request of the output compare ch.4 in the MFT unit 2
15	0	No interrupt request of the output compare ch.3 in the MFT unit 2
	1	Interrupt request of the output compare ch.3 in the MFT unit 2
14	0	No interrupt request of the output compare ch.2 in the MFT unit 2
	1	Interrupt request of the output compare ch.2 in the MFT unit 2
13	0	No interrupt request of the output compare ch.1 in the MFT unit 2
	1	Interrupt request of the output compare ch.1 in the MFT unit 2
12	0	No interrupt request of the output compare ch.0 in the MFT unit 2
	1	Interrupt request of the output compare ch.0 in the MFT unit 2

[bit11:6] OCU1INT:

Bit No.	Bit	Description
11	0	No interrupt request of the output compare ch.5 in the MFT unit 1
	1	Interrupt request of the output compare ch.5 in the MFT unit 1
10	0	No interrupt request of the output compare ch.4 in the MFT unit 1
	1	Interrupt request of the output compare ch.4 in the MFT unit 1
9	0	No interrupt request of the output compare ch.3 in the MFT unit 1
	1	Interrupt request of the output compare ch.3 in the MFT unit 1
8	0	No interrupt request of the output compare ch.2 in the MFT unit 1
	1	Interrupt request of the output compare ch.2 in the MFT unit 1
7	0	No interrupt request of the output compare ch.1 in the MFT unit 1
	1	Interrupt request of the output compare ch.1 in the MFT unit 1
6	0	No interrupt request of the output compare ch.0 in the MFT unit 1
	1	Interrupt request of the output compare ch.0 in the MFT unit 1

[bit5:0] OCU0INT:

Bit No.	Bit	Description
5	0	No interrupt request of the output compare ch.5 in the MFT unit 0
	1	Interrupt request of the output compare ch.5 in the MFT unit 0
4	0	No interrupt request of the output compare ch.4 in the MFT unit 0
	1	Interrupt request of the output compare ch.4 in the MFT unit 0
3	0	No interrupt request of the output compare ch.3 in the MFT unit 0
	1	Interrupt request of the output compare ch.3 in the MFT unit 0
2	0	No interrupt request of the output compare ch.2 in the MFT unit 0
	1	Interrupt request of the output compare ch.2 in the MFT unit 0
1	0	No interrupt request of the output compare ch.1 in the MFT unit 0
	1	Interrupt request of the output compare ch.1 in the MFT unit 0
0	0	No interrupt request of the output compare ch.0 in the MFT unit 0
	1	Interrupt request of the output compare ch.0 in the MFT unit 0

2.21 IRQ31 Batch Read Register (IRQ31MON)

The IRQ31 Batch Read Register (IRQ31MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 47.

IRQ31MON indicates the status of the interrupt requests of the Flash memory and base timer.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved				FLASHINT	Reserved										
Attribute	R				R	R										
Initial value	0000				0	00000000000										

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	BTINT															
Attribute	R															
Initial value	0x0000															

Register Functions

[bit31:28] Reserved: Reserved bits

A reserved bit reads "0".

[bit27] FLASHINT:

Bit	Description
0	No RDY, HANG interrupt request for flash memory
1	RDY, HANG interrupt request for flash memory

[bit26:16] Reserved: Reserved bits

A reserved bit reads "0".

[bit15:0] BTINT:

Bit No.	Bit	Description
15	0	No IRQ1 interrupt request on the base timer ch.7
	1	IRQ1 interrupt request on the base timer ch.7
14	0	No IRQ0 interrupt request on the base timer ch.7
	1	IRQ0 interrupt request on the base timer ch.7
13	0	No IRQ1 interrupt request on the base timer ch.6
	1	IRQ1 interrupt request on the base timer ch.6
12	0	No IRQ0 interrupt request on the base timer ch.6
	1	IRQ0 interrupt request on the base timer ch.6
11	0	No IRQ1 interrupt request on the base timer ch.5
	1	IRQ1 interrupt request on the base timer ch.5

Bit No.	Bit	Description
10	0	No IRQ0 interrupt request on the base timer ch.5
	1	IRQ0 interrupt request on the base timer ch.5
9	0	No IRQ1 interrupt request on the base timer ch.4
	1	IRQ1 interrupt request on the base timer ch.4
8	0	No IRQ0 interrupt request on the base timer ch.4
	1	IRQ0 interrupt request on the base timer ch.4
7	0	No IRQ1 interrupt request on the base timer ch.3
	1	IRQ1 interrupt request on the base timer ch.3
6	0	No IRQ0 interrupt request on the base timer ch.3
	1	IRQ0 interrupt request on the base timer ch.3
5	0	No IRQ1 interrupt request on the base timer ch.2
	1	IRQ1 interrupt request on the base timer ch.2
4	0	No IRQ0 interrupt request on the base timer ch.2
	1	IRQ0 interrupt request on the base timer ch.2
3	0	No IRQ1 interrupt request on the base timer ch.1
	1	IRQ1 interrupt request on the base timer ch.1
2	0	No IRQ0 interrupt request on the base timer ch.1
	1	IRQ0 interrupt request on the base timer ch.1
1	0	No IRQ1 interrupt request on the base timer ch.0
	1	IRQ1 interrupt request on the base timer ch.0
0	0	No IRQ0 interrupt request on the base timer ch.0
	1	IRQ0 interrupt request on the base timer ch.0

As shown in the Table 2-1, base timer interrupt factors IRQ0 and IRQ1 differ depending on the base timer function to be used.

Table 2-1 Interrupt Factors for Each Function of the Base Timer

Function	Interrupt Factor IRQ0	Interrupt Factor IRQ1
16-bit PWM timer	Underflow detection/ duty match detection	Timer start trigger detection
16-bit PPG timer	Underflow detection	Timer start trigger detection
16/32-bit reload timer	Underflow detection	Timer start trigger detection
16/32-bit PWC timer	Overflow detection	Measurement finished detection

3. Usage Precautions

Be careful with the following points when using the interrupt controller.

- The interrupt request signals from each of the peripheral resources are notified by level. When exiting from interrupt processing, always clear the interrupt request.
- The NMIX pin is allocated shared with a general-purpose port. The initial value after reset is released is configured as a general-purpose port, and the NMI input is masked. In order to use the NMI, enable NMI by configuring the port setting. See the chapter of "External Interrupt and NMI Control Unit" for details.
- See the chapters for each of the macros for the correlation between the specific event detection registers and interrupt enable registers in each peripheral resource.

CHAPTER 7-4: Interrupts (TYPE1-B)

This chapter explains Exception and Interrupt Factor Vectors and Registers (TYPE1) at IRQCMODE=1.

1. Exception and Interrupt Factor Vectors
2. Registers
3. Usage Precautions

CODE: 9AFIRQC_B-E01.0

1. Exception and Interrupt Factor Vectors

This section shows a vector table of the exceptions and interrupts input to the NVIC.

Table 1-1-1 Exception and Interrupt Factor Vectors

Vector No.	IRQ No.	Exception and Interrupt Factor	Vector Offset
0	-	Stack pointer initial value	0x00
1	-	Reset	0x04
2	-	Non-Maskable Interrupt (NMI) / Hardware Watchdog Timer	0x08
3	-	Hard Fault	0x0C
4	-	Reserved	0x10
5	-	Reserved	0x14
6	-	Reserved	0x18
7-10	-	Reserved	0x1C - 0x2B
11	-	SVCall (Supervisor Call)	0x2C
12	-	Reserved	0x30
13	-	Reserved	0x34
14	-	PendSV	0x38
15	-	SysTick	0x3C
16	0	Anomalous Frequency Detection by Clock Supervisor (FCS)	0x40
17	1	Software Watchdog Timer	0x44
18	2	Low Voltage Detector (LVD)	0x48
19	3	Interrupt source selected in the INTSEL0 bits in the RCINTSEL0 Register	0x4C
20	4	Interrupt source selected in the INTSEL1 bits in the RCINTSEL0 Register	0x50
21	5	Interrupt source selected in the INTSEL2 bits in the RCINTSEL0 Register	0x54
22	6	Interrupt source selected in the INTSEL3 bits in the RCINTSEL0 Register	0x58
23	7	Interrupt source selected in the INTSEL0 bits in the RCINTSEL1 Register	0x5C
24	8	Interrupt source selected in the INTSEL1 bits in the RCINTSEL1 Register	0x60
25	9	Interrupt source selected in the INTSEL2 bits in the RCINTSEL1 Register	0x64
26	10	Interrupt source selected in the INTSEL3 bits in the RCINTSEL1 Register	0x68
27	11	MFT unit 0 Wave Form Generator / DTIF(Motor Emergency Stop) / Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.8	0x6C
28	12	External Pin Interrupt Request ch.0 to ch.7	0x70
29	13	External Pin Interrupt Request ch.8 to ch.31	0x74
30	14	Dual Timer / Quad Counter (QPRC) ch.0	0x78
31	15	Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.0	0x7C
32	16	Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.1	0x80
33	17	Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.2	0x84
34	18	Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.3	0x88
35	19	Reception Interrupt Request of MFS ch.4	0x8C

Vector No.	IRQ No.	Exception and Interrupt Factor	Vector Offset
36	20	Transmission Interrupt Request and Status Interrupt Request of MFS ch.4	0x90
37	21	Reception Interrupt Request of MFS ch.5	0x94
38	22	Transmission Interrupt Request and Status Interrupt Request of MFS ch.5	0x98
39	23	PPG ch.0/2/4/8/10/12/16/18/20	0x9C
40	24	External Main OSC / External Sub OSC / Main PLL / Watch Counter/Real Time Counter	0xA0
41	25	A/D Converter unit 0 / Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.9	0xA4
42	26	A/D Converter unit 1 / Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.10	0xA8
43	27	A/D Converter unit 2 / LCD Controller / Reception Interrupt Request, Transmission Interrupt Request and Status Interrupt Request of MFS ch.11	0xAC
44	28	MFT unit 0 Free-run Timer, Input Capture, Output Compare	0xB0
45	29	MFT unit 1 Free-run Timer, Input Capture, Output Compare	0xB4
46	30	MFT unit 2 Free-run Timer, Input Capture, Output Compare	0xB8
47	31	Base Timer ch.0 to ch.7 / Flash RDY interrupt / Flash HANG interrupt	0xBC

The priorities of the exceptions for vectors No. 4 to No. 15 can be configured using the System Handler Priority Registers (address 0xE000ED18, 0xE000ED1C, 0xE000ED20) built into the NVIC. The priorities of the peripheral interrupts for vectors No. 16 and after can be configured using the IRQ Priority Registers (address 0xE000E400 to 0xE000E42C) built into the NVIC.

The sources of the interrupts for vectors No. 2 and No. 16 to No. 47 can be checked using the batch read register. See "Chapter 5: Nested Vectored Interrupt Controller" in the "Cortex-M0+ Technical Reference Manual" for details on the other exceptions and interrupts.

Furthermore, for the interrupts of vectors No. 2 and No. 16 to No. 47, the sources that are batch read may be a signal that multiple interrupt factors are logical OR’ed within each of the peripheral macros. See the descriptions of each of the peripheral resource interrupts for details.

2. Registers

This section describes the DMA transfer request selection register and the interrupt request batch read registers.

List of DMA Transfer Request Selection Register and Interrupt Request Batch Read Registers

Table 2-1 List of DMA Transfer Request Selection Register and Interrupt Request Batch Read Registers

Abbreviation	Register Name	Reference
DRQSEL	DMA Request Selection Register	2.1
EXC02MON	EXC02 Batch Read Register	0
IRQ00MON	IRQ00 Batch Read Register	2.3
IRQ01MON	IRQ01 Batch Read Register	2.4
IRQ02MON	IRQ02 Batch Read Register	2.5
IRQ03MON	IRQ03 Batch Read Register	2.6
IRQ04MON	IRQ04 Batch Read Register	
IRQ05MON	IRQ05 Batch Read Register	
IRQ06MON	IRQ06 Batch Read Register	
IRQ07MON	IRQ07 Batch Read Register	
IRQ08MON	IRQ08 Batch Read Register	
IRQ09MON	IRQ09 Batch Read Register	
IRQ10MON	IRQ10 Batch Read Register	
IRQ11MON	IRQ11 Batch Read Register	2.7
IRQ12MON	IRQ12 Batch Read Register	2.8
IRQ13MON	IRQ13 Batch Read Register	2.9
IRQ14MON	IRQ14 Batch Read Register	2.10
IRQ15MON	IRQ15 Batch Read Register	2.11
IRQ16MON	IRQ16 Batch Read Register	
IRQ17MON	IRQ17 Batch Read Register	
IRQ18MON	IRQ18 Batch Read Register	
IRQ19MON	IRQ19 Batch Read Register	2.12
IRQ20MON	IRQ20 Batch Read Register	2.13
IRQ21MON	IRQ21 Batch Read Register	2.12
IRQ22MON	IRQ22 Batch Read Register	2.13
IRQ23MON	IRQ23 Batch Read Register	2.14
IRQ24MON	IRQ24 Batch Read Register	2.15
IRQ25MON	IRQ25 Batch Read Register	2.16
IRQ26MON	IRQ26 Batch Read Register	
IRQ27MON	IRQ27 Batch Read Register	2.17
IRQ28MON	IRQ28 Batch Read Register	2.18
IRQ29MON	IRQ29 Batch Read Register	
IRQ30MON	IRQ30 Batch Read Register	2.19
IRQ31MON	IRQ31 Batch Read Register	2.20
IRQCMODE	Interrupt Factor Vector Relocate Setting Register	2.21
RCINTSEL0	Interrupt Factor Selection Register 0	2.22
RCINTSEL1	Interrupt Factor Selection Register 1	2.23

See "Chapter 5: Nested Vectored Interrupt Controller" in the "Cortex-M0+ Technical Reference Manual" for details on the registers in the NVIC.

2.1 DMA Request Selection Register (DRQSEL)

The DMA Request Selection Register (DRQSEL) selects whether interrupt signals that can start DMA transfers are output as interrupt requests to the CPU or output as transfer requests to the DMAC.

Register Configuration

bit	31	5	4	3	2	1	0
Field	Reserved				Reserved		
Attribute	R/W				R		
Initial value	00000000000000000000000000000000				00000		

Register Functions

[bit31:0] Reserved: Reserved bits

Write 0.to these bits.

[bit4:0] Reserved: Reserved bits

A reserved bit reads "0".

2.2 EXC02 Batch Read Register (EXC02MON)

The EXC02 Batch Read Register (EXC02MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 2.

EXC02MON indicates the status of the interrupt requests of the hardware watchdog timer and NMIX external pin.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved														HWINT	NMI
Attribute	R														R	R
Initial value	00000000000000														0	0

Register Functions

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1] HWINT:

Bit	Description
0	No hardware watchdog timer interrupt request
1	Hardware watchdog timer interrupt request

[bit0] NMI:

Bit	Description
0	No NMIX external pin interrupt request
1	NMIX external pin interrupt request

2.3 IRQ00 Batch Read Register (IRQ00MON)

The IRQ00 Batch Read Register (IRQ00MON) can batch-read the interrupt request allocated to interrupt factor vector No. 16.

IRQ00MON indicates the status of the interrupt request of anomalous frequency detection by the CSV.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															FCSINT
Attribute	R															R
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] FCSINT:

Bit	Description
0	No anomalous frequency detection by CSV interrupt request
1	Anomalous frequency detection by CSV interrupt request

2.4 IRQ01 Batch Read Register (IRQ01MON)

The IRQ01 Batch Read Register (IRQ01MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 17.

IRQ01MON indicates the status of the interrupt request of the software watchdog timer.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															SWWDTINT
Attribute	R															R
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] SWWDTINT:

Bit	Description
0	No software watchdog timer interrupt request
1	Software watchdog timer interrupt request

2.5 IRQ02 Batch Read Register (IRQ02MON)

The IRQ02 Batch Read Register (IRQ02MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 18.

IRQ02MON indicates the status of the interrupt request of low voltage detection (LVD).

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															LVDINT
Attribute	R															R
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] LVDINT:

Bit	Description
0	No low voltage detection (LVD) interrupt request
1	Low voltage detection (LVD) interrupt request

2.6 IRQ03 to IRQ10 Batch Read Register (IRQ03MON to IRQ10MON)

The IRQ03 Batch Read Register (IRQ03MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 19.
The IRQ04 Batch Read Register (IRQ04MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 20.
The IRQ05 Batch Read Register (IRQ05MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 21.
The IRQ06 Batch Read Register (IRQ06MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 22.
The IRQ07 Batch Read Register (IRQ07MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 23.
The IRQ08 Batch Read Register (IRQ08MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 24.
The IRQ09 Batch Read Register (IRQ09MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 25.
The IRQ10 Batch Read Register (IRQ10MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 26.

IRQ03MON indicates the status of the interrupt request selected in the INTSEL0 bits in the RCINTSEL0 Register.
IRQ04MON indicates the status of the interrupt request selected in the INTSEL1 bits in the RCINTSEL0 Register.
IRQ05MON indicates the status of the interrupt request selected in the INTSEL2 bits in the RCINTSEL0 Register.
IRQ06MON indicates the status of the interrupt request selected in the INTSEL3 bits in the RCINTSEL0 Register.
IRQ07MON indicates the status of the interrupt request selected in the INTSEL0 bits in the RCINTSEL1 Register.
IRQ08MON indicates the status of the interrupt request selected in the INTSEL1 bits in the RCINTSEL1 Register.
IRQ09MON indicates the status of the interrupt request selected in the INTSEL2 bits in the RCINTSEL1 Register.
IRQ10MON indicates the status of the interrupt request selected in the INTSEL3 bits in the RCINTSEL1 Register.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															RCINT
Attribute	R															R
Initial value	00000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits
A reserved bit reads "0".

[bit0] RCINT:

Bit	Description
0	No interrupt request selected with relevant RCINTSEL0:INTSELx/RCINTSEL1:INTSELx
1	Interrupt request selected with relevant RCINTSEL0:NTSELx/RCINTSEL1:INTSELx*

*: If the base timer is selected as the interrupt factor, this bit is set to "1" by either one of the interrupt factors IRQ0 and IRQ1.

2.7 IRQ11 Batch Read Register (IRQxxMON)

The IRQ11 Batch Read Register (IRQ11MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 27.

IRQ11MON indicates the status of the interrupt requests of MFT unit 0 and MFS ch.8.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved									MFSINT			WAVEINT			
Attribute	R									R			R			
Initial value	000000000									000			0000			

Register Functions

[bit31:7] Reserved: Reserved bits

A reserved bit reads "0".

[bit6:4] MFSINT:

Bit No.	Bit	Description
6	0	No status interrupt request of the corresponding MFS channel
	1	Status interrupt request of the corresponding MFS channel
5	0	No transmission interrupt request of the corresponding MFS channel
	1	Transmission interrupt request of the corresponding MFS channel
4	0	No reception interrupt request of the corresponding MFS channel
	1	Reception interrupt request of the corresponding MFS channel

[bit3:0] WAVEINT:

Bit No.	Bit	Description
3	0	No interrupt request of WFG timer 54 in the corresponding MFT unit
	1	Interrupt request of WFG timer 54 in the corresponding MFT unit
2	0	No interrupt request of WFG timer 32 in the corresponding MFT unit
	1	Interrupt request of WFG timer 32 in the corresponding MFT unit
1	0	No interrupt request of WFG timer 10 in the corresponding MFT unit
	1	Interrupt request of WFG timer 10 in the corresponding MFT unit
0	0	No interrupt request of DTIF (Motor emergency stop) in the corresponding MFT unit
	1	Interrupt request of DTIF (Motor emergency stop) in the corresponding MFT unit

2.8 IRQ12 Batch Read Register (IRQ12MON)

The IRQ12 Batch Read Register (IRQ12MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 28.

IRQ12MON indicates the status of the interrupt requests of external interrupt ch.0 to ch.7.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								EXTINT							
Attribute	R								R							
Initial value	0x00								0x00							

Register Functions

[bit31:8] Reserved: Reserved bits

A reserved bit reads "0".

[bit7:0] EXTINT:

Bit No.	Bit	Description
7	0	No interrupt request of external interrupt ch.7
	1	Interrupt request of external interrupt ch.7
6	0	No interrupt request of external interrupt ch.6
	1	Interrupt request of external interrupt ch.6
5	0	No interrupt request of external interrupt ch.5
	1	Interrupt request of external interrupt ch.5
4	0	No interrupt request of external interrupt ch.4
	1	Interrupt request of external interrupt ch.4
3	0	No interrupt request of external interrupt ch.3
	1	Interrupt request of external interrupt ch.3
2	0	No interrupt request of external interrupt ch.2
	1	Interrupt request of external interrupt ch.2
1	0	No interrupt request of external interrupt ch.1
	1	Interrupt request of external interrupt ch.1
0	0	No interrupt request of external interrupt ch.0
	1	Interrupt request of external interrupt ch.0

2.9 IRQ13 Batch Read Register (IRQ13MON)

The IRQ13 Batch Read Register (IRQ13MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 29.

IRQ13MON indicates the status of the interrupt requests of external interrupt ch.8 to ch.31.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved								EXTINT							
Attribute	R								R							
Initial value	0x00								0x00							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EXTINT															
Attribute	R															
Initial value	0x0000															

Register Functions

[bit31:24] Reserved: Reserved bits

A reserved bit reads "0".

[bit23:0] EXTINT:

Bit No.	Bit	Description
23	0	No interrupt request of external interrupt ch.31
	1	Interrupt request of external interrupt ch.31
22	0	No interrupt request of external interrupt ch.30
	1	Interrupt request of external interrupt ch.30
21	0	No interrupt request of external interrupt ch.29
	1	Interrupt request of external interrupt ch.29
20	0	No interrupt request of external interrupt ch.28
	1	Interrupt request of external interrupt ch.28
19	0	No interrupt request of external interrupt ch.27
	1	Interrupt request of external interrupt ch.27
18	0	No interrupt request of external interrupt ch.26
	1	Interrupt request of external interrupt ch.26
17	0	No interrupt request of external interrupt ch.25
	1	Interrupt request of external interrupt ch.25
16	0	No interrupt request of external interrupt ch.24
	1	Interrupt request of external interrupt ch.24
15	0	No interrupt request of external interrupt ch.23
	1	Interrupt request of external interrupt ch.23
14	0	No interrupt request of external interrupt ch.22
	1	Interrupt request of external interrupt ch.22
13	0	No interrupt request of external interrupt ch.21
	1	Interrupt request of external interrupt ch.21

Bit No.	Bit	Description
12	0	No interrupt request of external interrupt ch.20
	1	Interrupt request of external interrupt ch.20
11	0	No interrupt request of external interrupt ch.19
	1	Interrupt request of external interrupt ch.19
10	0	No interrupt request of external interrupt ch.18
	1	Interrupt request of external interrupt ch.18
9	0	No interrupt request of external interrupt ch.17
	1	Interrupt request of external interrupt ch.17
8	0	No interrupt request of external interrupt ch.16
	1	Interrupt request of external interrupt ch.16
7	0	No interrupt request of external interrupt ch.15
	1	Interrupt request of external interrupt ch.15
6	0	No interrupt request of external interrupt ch.14
	1	Interrupt request of external interrupt ch.14
5	0	No interrupt request of external interrupt ch.13
	1	Interrupt request of external interrupt ch.13
4	0	No interrupt request of external interrupt ch.12
	1	Interrupt request of external interrupt ch.12
3	0	No interrupt request of external interrupt ch.11
	1	Interrupt request of external interrupt ch.11
2	0	No interrupt request of external interrupt ch.10
	1	Interrupt request of external interrupt ch.10
1	0	No interrupt request of external interrupt ch.9
	1	Interrupt request of external interrupt ch.9
0	0	No interrupt request of external interrupt ch.8
	1	Interrupt request of external interrupt ch.8

2.10 IRQ14 Batch Read Register (IRQ14MON)

The IRQ14 Batch Read Register (IRQ14MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 30.

IRQ14MON indicates the status of the interrupt requests of dual timer and QPRC.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								QUDINT						TIMINT	
Attribute	R								R						R	
Initial value	0x00								000000						00	

Register Functions

[bit31:8] Reserved: Reserved bits

A reserved bit reads "0".

[bit7:2] QUDINT:

Bit No.	Bit	Description
7	0	No PC match & RC match interrupt request of QPRC ch.0
	1	PC match & RC match interrupt request of QPRC ch.0
6	0	No interrupt request detected RC out of range on QPRC ch.0
	1	Interrupt request detected RC out of range on QPRC ch.0
5	0	No PC count invert interrupt request of QPRC ch.0
	1	PC count invert interrupt request of QPRC ch.0
4	0	No overflow/underflow/zero index interrupt request of QPRC ch.0
	1	Overflow/underflow/zero index interrupt request of QPRC ch.0
3	0	No PC&RC match interrupt request of QPRC ch.0
	1	PC&RC match interrupt request of QPRC ch.0
2	0	No PC match interrupt request of QPRC ch.0
	1	PC match interrupt request of QPRC ch.0

[bit1:0] TIMINT:

Bit No.	Bit	Description
1	0	No dual timer TIMINT2 interrupt request
	1	Dual timer TIMINT2 interrupt request
0	0	No dual timer TIMINT1 interrupt request
	1	Dual timer TIMINT1 interrupt request

2.11 IRQ15 to IRQ18 Batch Read Register (IRQ15MON to IRQ18MON)

The IRQ15 Batch Read Register (IRQ15MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 31
The IRQ16 Batch Read Register (IRQ16MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 32.
The IRQ17 Batch Read Register (IRQ17MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 33.
The IRQ18 Batch Read Register (IRQ18MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 34.

IRQ15MON indicates the status of the reception interrupt request, transmission interrupt request and status request of MFS ch.0.
IRQ16MON indicates the status of the reception interrupt request, transmission interrupt request and status request of MFS ch.1.
IRQ17MON indicates the status of the reception interrupt request, transmission interrupt request and status request of MFS ch.2.
IRQ18MON indicates the status of the reception interrupt request, transmission interrupt request and status request of MFS ch.3.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved													MFSINT		
Attribute	R													R		
Initial value	0000000000000													000		

Register Functions

[bit31:3] Reserved: Reserved bits

A reserved bit reads "0".

[bit2:0] MFSINT:

Bit No.	Bit	Description
2	0	No status interrupt request of the corresponding MFS channel
	1	Status interrupt request of the corresponding MFS channel
1	0	No transmission interrupt request of the corresponding MFS channel
	1	Transmission interrupt request of the corresponding MFS channel
0	0	No reception interrupt request of the corresponding MFS channel
	1	Reception interrupt request of the corresponding MFS channel

2.12 IRQ19/21 Batch Read Register (IRQxxMON)

The IRQ19 Batch Read Register (IRQ19MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 35.
The IRQ21 Batch Read Register (IRQ21MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 37.

IRQ19MON indicates the status of the reception interrupt request of MFS ch.4.
IRQ21MON indicates the status of the reception interrupt request of MFS ch.5.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															MFSINT
Attribute	R															R
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] MFSINT:

Bit No.	Bit	Description
0	0	No reception interrupt request of the corresponding MFS channel
	1	Reception interrupt request of the corresponding MFS channel

2.13 IRQ20/22 Batch Read Register (IRQxxMON)

The IRQ20 Batch Read Register (IRQ20MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 36.
The IRQ22 Batch Read Register (IRQ22MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 38.

IRQ20MON indicates the status of the transmission interrupt request and the status of the status interrupt request of MFS ch.4.
IRQ22MON indicates the status of the transmission interrupt request and the status of the status interrupt request of MFS ch.5.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved														MFSINT	
Attribute	R														R	
Initial value	00000000000000														00	

Register Functions

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1:0] MFSINT:

Bit No.	Bit	Description
1	0	No status interrupt request of the corresponding MFS channel
	1	Status interrupt request of the corresponding MFS channel
0	0	No transmission interrupt request of the corresponding MFS channel
	1	Transmission interrupt request of the corresponding MFS channel

2.14 IRQ23 Batch Read Register (IRQ23MON)

The IRQ23 Batch Read Register (IRQ23MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 39.

IRQ23MON indicates the status of the interrupt request of the PPG.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved							PPGINT								
Attribute	R							R								
Initial value	0000000							000000000								

Register Functions

[bit31:9] Reserved: Reserved bits

A reserved bit reads "0".

[bit8:0] PPGINT:

Bit No.	Bit	Description
8	0	No interrupt request of PPG ch.20
	1	Interrupt request of PPG ch.20
7	0	No interrupt request of PPG ch.18
	1	Interrupt request of PPG ch.18
6	0	No interrupt request of PPG ch.16
	1	Interrupt request of PPG ch.16
5	0	No interrupt request of PPG ch.12
	1	Interrupt request of PPG ch.12
4	0	No interrupt request of PPG ch.10
	1	Interrupt request of PPG ch.10
3	0	No interrupt request of PPG ch.8
	1	Interrupt request of PPG ch.8
2	0	No interrupt request of PPG ch.4
	1	Interrupt request of PPG ch.4
1	0	No interrupt request of PPG ch.2
	1	Interrupt request of PPG ch.2
0	0	No interrupt request of PPG ch.0
	1	Interrupt request of PPG ch.0

2.15 IRQ24 Batch Read Register (IRQ24MON)

The IRQ24 Batch Read Register (IRQ24MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 40.

1. IRQ24MON indicates the status of the interrupt requests of the RTC, watch counter, main PLL oscillation, sub oscillation and main clock oscillation.

Register Configuration

bit	31							8
Field	Reserved							
Attribute	R							
Initial value	0x000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved		RTCINT	WCINT	Reserved	MPLLINT	SOSCINT	MOSCINT
Attribute	R		R	R	R	R	R	R
Initial value	00		0	0	0	0	0	0

Register Functions

[bit31:6] Reserved: Reserved bits

A reserved bit reads "0".

[bit5] RTCINT:

Bit	Description
0	No RTC interrupt request
1	RTC interrupt request

[bit4] WCINT:

Bit	Description
0	No watch counter interrupt request
1	Watch counter interrupt request

[bit3] Reserved: Reserved bit

A reserved bit reads "0".

[bit2] MPLLINT:

Bit	Description
0	No stabilization wait completion interrupt request for main PLL oscillation
1	Stabilization wait completion interrupt request for main PLL oscillation

[bit1] SOSCINT:

Bit	Description
0	No stabilization wait completion interrupt request for sub-clock oscillation
1	Stabilization wait completion interrupt request for sub-clock oscillation

[bit0] MOSCINT:

Bit	Description
0	No stabilization wait completion interrupt request for main clock oscillation
1	Stabilization wait completion interrupt request for main clock oscillation

2.16 IRQ25/26 Batch Read Register (IRQxxMON)

The IRQ25 Batch Read Register (IRQ25MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 41.
The IRQ26 Batch Read Register (IRQ26MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 42.

IRQ25MON indicates the status of the interrupt requests of A/D converter unit 0 and MFS ch.9.
IRQ26MON indicates the status of the interrupt request of A/D converter unit 1 and MFS ch.10.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								MFSINT			ADCINT				
Attribute	R								R			R				
Initial value	0x00								000			00000				

Register Functions

[bit31:8] Reserved: Reserved bits

A reserved bit reads "0".

[bit7:5] MFSINT:

Bit No.	Bit	Description
7	0	No status interrupt request of the corresponding MFS channel
	1	Status interrupt request of the corresponding MFS channel
6	0	No transmission interrupt request of the corresponding MFS channel
	1	Transmission interrupt request of the corresponding MFS channel
5	0	No reception interrupt request of the corresponding MFS channel
	1	Reception interrupt request of the corresponding MFS channel

[bit4:0] ADCINT:

Bit No.	Bit	Description
4	0	No range comparison result interrupt request in the corresponding A/D converter unit
	1	Range comparison result interrupt request in the corresponding A/D converter unit
3	0	No conversion result comparison interrupt request in the corresponding A/D converter unit
	1	Conversion result comparison interrupt request in the corresponding A/D converter unit
2	0	No FIFO overrun interrupt request in the corresponding A/D converter unit
	1	FIFO overrun interrupt request in the corresponding A/D converter unit
1	0	No scan conversion interrupt request in the corresponding A/D converter unit
	1	Scan conversion interrupt request in the corresponding A/D converter unit
0	0	No priority conversion interrupt request in the corresponding A/D converter unit
	1	Priority conversion interrupt request in the corresponding A/D converter unit

2.17 IRQ27 Batch Read Register (IRQ27MON)

The IRQ27 Batch Read Register (IRQ27MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 43.

IRQ27MON indicates the status of the interrupt requests of A/D converter unit 2, LCD controller and MFS ch.11.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved							MFSINT			LCDCINT	ADCINT				
Attribute	R							R			R	R				
Initial value	0000000							000			0	00000				

Register Functions

[bit31:9] Reserved: Reserved bits

A reserved bit reads "0".

[bit8:6] MFSINT:

Bit No.	Bit	Description
8	0	No status interrupt request of MFS ch.11
	1	Status interrupt request of the corresponding MFS ch.11
7	0	No transmission interrupt request of the corresponding MFS ch.11
	1	Transmission interrupt request of the corresponding MFS ch.11
6	0	No reception interrupt request of the corresponding MFS ch.11
	1	Reception interrupt request of the corresponding MFS ch.11

[bit5] LCDCINT:

Bit	Description
0	No interrupt request for LCD controller
1	Interrupt request for LCD controller

[bit4:0] ADCINT:

Bit No.	Bit	Description
4	0	No range comparison result interrupt request in the corresponding A/D converter unit 2
	1	Range comparison result interrupt request in the corresponding A/D converter unit 2
3	0	No conversion result comparison interrupt request in the A/D converter unit 2
	1	Conversion result comparison interrupt request in the A/D converter unit 2
2	0	No FIFO overrun interrupt request in the A/D converter unit 2
	1	FIFO overrun interrupt request in the A/D converter unit 2
1	0	No scan conversion interrupt request in the A/D converter unit 2
	1	Scan conversion interrupt request in the A/D converter unit 2
0	0	No priority conversion interrupt request in the A/D converter unit 2
	1	Priority conversion interrupt request in the A/D converter unit 2

2.18 IRQ28/29 Batch Read Register (IRQxxMON)

The IRQ28 Batch Read Register (IRQ28MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 44.
The IRQ29 Batch Read Register (IRQ29MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 45.

IRQ28MON indicates the status of the interrupt request of MFT unit 0.
IRQ29MON indicates the status of the interrupt request of MFT unit 1.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OCUINT						ICUINT				FRTINT					
Attribute	R						R				R					
Initial value	000000						0000				000000					

Register Functions

[bit31:16] Reserved: Reserved bits
A reserved bit reads "0".

[bit15:10] OCUINT:

Bit No.	Bit	Description
15	0	No interrupt request of the output compare ch.5 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.5 in the corresponding MFT unit
14	0	No interrupt request of the output compare ch.4 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.4 in the corresponding MFT unit
13	0	No interrupt request of the output compare ch.3 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.3 in the corresponding MFT unit
12	0	No interrupt request of the output compare ch.2 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.2 in the corresponding MFT unit
11	0	No interrupt request of the output compare ch.1 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.1 in the corresponding MFT unit
10	0	No interrupt request of the output compare ch.0 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.0 in the corresponding MFT unit

[bit9:6] ICUINT:

Bit No.	Bit	Description
9	0	No interrupt request of the input capture ch.3 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.3 in the corresponding MFT unit
8	0	No interrupt request of the input capture ch.2 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.2 in the corresponding MFT unit
7	0	No interrupt request of the input capture ch.1 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.1 in the corresponding MFT unit
6	0	No interrupt request of the input capture ch.0 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.0 in the corresponding MFT unit

[bit5:0] FRTINT:

Bit No.	Bit	Description
5	0	No zero detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
	1	Zero detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
4	0	No zero detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
	1	Zero detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
3	0	No zero detection interrupt request of the free run timer ch.0 in the corresponding MFT unit
	1	Zero detection interrupt request of the free run timer ch.0 in the corresponding MFT unit
2	0	No peak value detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
	1	Peak value detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
1	0	No peak value detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
	1	Peak value detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
0	0	No peak value detection interrupt request of the free run timer ch.0 in the corresponding MFT unit
	1	Peak value detection interrupt request of the free run timer ch.0 in the corresponding MFT unit

2.19 IRQ30 Batch Read Register (IRQ30MON)

The IRQ30 Batch Read Register (IRQ30MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 46.

IRQ30MON indicates the status of the interrupt requests of MFT unit 2.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved								Reserved							
Attribute	R								R							
Initial value	0x00								0x00							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	OCUINT						ICUINT				FRTINT					
Attribute	R						R				R					
Initial value	000000						0000				000000					

Register Functions

[bit31:24] Reserved: Reserved bits

A reserved bit reads "0".

[bit23:16] Reserved: Reserved bits

A reserved bit reads "0".

[bit15:10] OCUINT:

Bit No.	Bit	Description
15	0	No interrupt request of the output compare ch.5 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.5 in the corresponding MFT unit
14	0	No interrupt request of the output compare ch.4 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.4 in the corresponding MFT unit
13	0	No interrupt request of the output compare ch.3 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.3 in the corresponding MFT unit
12	0	No interrupt request of the output compare ch.2 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.2 in the corresponding MFT unit
11	0	No interrupt request of the output compare ch.1 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.1 in the corresponding MFT unit
10	0	No interrupt request of the output compare ch.0 in the corresponding MFT unit
	1	Interrupt request of the output compare ch.0 in the corresponding MFT unit

[bit9:6] ICUINT:

Bit No.	Bit	Description
9	0	No interrupt request of the input capture ch.3 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.3 in the corresponding MFT unit
8	0	No interrupt request of the input capture ch.2 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.2 in the corresponding MFT unit
7	0	No interrupt request of the input capture ch.1 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.1 in the corresponding MFT unit
6	0	No interrupt request of the input capture ch.0 in the corresponding MFT unit
	1	Interrupt request of the input capture ch.0 in the corresponding MFT unit

[bit5:0] FRTINT:

Bit No.	Bit	Description
5	0	No zero detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
	1	Zero detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
4	0	No zero detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
	1	Zero detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
3	0	No zero detection interrupt request of the free run timer ch.0 in the corresponding MFT unit
	1	Zero detection interrupt request of the free run timer ch.0 in the corresponding MFT unit
2	0	No peak value detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
	1	Peak value detection interrupt request of the free run timer ch.2 in the corresponding MFT unit
1	0	No peak value detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
	1	Peak value detection interrupt request of the free run timer ch.1 in the corresponding MFT unit
0	0	No peak value detection interrupt request of the free run timer ch.0 in the corresponding MFT unit
	1	Peak value detection interrupt request of the free run timer ch.0 in the corresponding MFT unit

2.20 IRQ31 Batch Read Register (IRQ31MON)

The IRQ31 Batch Read Register (IRQ31MON) can batch-read the interrupt requests allocated to interrupt factor vector No. 47.

IRQ31MON indicates the status of the interrupt requests of the Flash memory and base timer.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved				FLASHINT	Reserved										
Attribute	R				R	R										
Initial value	0000				0	00000000000										

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	BTINT															
Attribute	R															
Initial value	0x0000															

Register Functions

[bit31:28] Reserved: Reserved bits

A reserved bit reads "0".

[bit27] FLASHINT:

Bit	Description
0	No RDY, HANG interrupt request for flash memory
1	RDY, HANG interrupt request for flash memory

[bit26:16] Reserved: Reserved bits

A reserved bit reads "0".

[bit15:0] BTINT:

Bit No.	Bit	Description
15	0	No IRQ1 interrupt request on the base timer ch.7
	1	IRQ1 interrupt request on the base timer ch.7
14	0	No IRQ0 interrupt request on the base timer ch.7
	1	IRQ0 interrupt request on the base timer ch.7
13	0	No IRQ1 interrupt request on the base timer ch.6
	1	IRQ1 interrupt request on the base timer ch.6
12	0	No IRQ0 interrupt request on the base timer ch.6
	1	IRQ0 interrupt request on the base timer ch.6
11	0	No IRQ1 interrupt request on the base timer ch.5
	1	IRQ1 interrupt request on the base timer ch.5

Bit No.	Bit	Description
10	0	No IRQ0 interrupt request on the base timer ch.5
	1	IRQ0 interrupt request on the base timer ch.5
9	0	No IRQ1 interrupt request on the base timer ch.4
	1	IRQ1 interrupt request on the base timer ch.4
8	0	No IRQ0 interrupt request on the base timer ch.4
	1	IRQ0 interrupt request on the base timer ch.4
7	0	No IRQ1 interrupt request on the base timer ch.3
	1	IRQ1 interrupt request on the base timer ch.3
6	0	No IRQ0 interrupt request on the base timer ch.3
	1	IRQ0 interrupt request on the base timer ch.3
5	0	No IRQ1 interrupt request on the base timer ch.2
	1	IRQ1 interrupt request on the base timer ch.2
4	0	No IRQ0 interrupt request on the base timer ch.2
	1	IRQ0 interrupt request on the base timer ch.2
3	0	No IRQ1 interrupt request on the base timer ch.1
	1	IRQ1 interrupt request on the base timer ch.1
2	0	No IRQ0 interrupt request on the base timer ch.1
	1	IRQ0 interrupt request on the base timer ch.1
1	0	No IRQ1 interrupt request on the base timer ch.0
	1	IRQ1 interrupt request on the base timer ch.0
0	0	No IRQ0 interrupt request on the base timer ch.0
	1	IRQ0 interrupt request on the base timer ch.0

As shown in the Table 2-2, base timer interrupt factors IRQ0 and IRQ1 differ depending on the base timer function to be used.

Table 2-2 Interrupt Factors for Each Function of the Base Timer

Function	Interrupt Factor IRQ0	Interrupt Factor IRQ1
16-bit PWM timer	Underflow detection/ duty match detection	Timer start trigger detection
16-bit PPG timer	Underflow detection	Timer start trigger detection
16/32-bit reload timer	Underflow detection	Timer start trigger detection
16/32-bit PWC timer	Overflow detection	Measurement finished detection

2.21 Interrupt Factor Vector Relocate Setting Register (IRQCMODE)

The Interrupt Factor Vector Relocate Setting Register (IRQCMODE) selects whether the interrupt factor vectors are assigned according to Table 1-1 in chapter "Interrupts (A)" or to Table 1-1-1 in this chapter.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															IRQCMODE
Attribute	R/W															R/W
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] IRQCMODE:

Bit	Description
0	Assigns the interrupt factor vector according to Table 1-1-1 in chapter "Interrupts (A)".
1	Assigns the interrupt factor vector according to Table 1-1-1 in this chapter.

2.22 Interrupt Factor Selection Register 0 (RCINTSEL0)

The Interrupt Factor Selection Register 0 (RCINTSEL0) selects the interrupt factors for the interrupt vectors No. 19 to No. 22. This register is valid when the IRQQCMODE:IRQCMODE bit is "1".

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	INTSEL3								INTSEL2							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	INTSEL1								INTSEL0							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

Register Functions

[bit31:24] INTSEL3:
These bits select* the interrupt factor for the interrupt vector No. 22.

[bit23:16] INTSEL2:
These bits select* the interrupt factor for the interrupt vector No. 21.

[bit15:8] INTSEL1:
These bits select* the interrupt factor for the interrupt vector No. 20.

[bit7:0] INTSEL0:
These bits select* the interrupt factor for the interrupt vector No. 19.

*: See Table 2-3 for details of selection interrupt factors

Notes:

- *The interrupt factors selected by RCINTSEL0 are masked with IRQ11 to IRQ31. (The usable bits in IRQ11MON to IRQ31MON Registers are also masked.)*
- *Ensure that all interrupt factors selected in the INTSEL0 to INTSEL7 bits are different when selecting interrupt factors.*

2.23 Interrupt Factor Selection Register 1 (RCINTSEL1)

The Interrupt Factor Selection Register 1 (RCINTSEL1) selects the interrupt factors for the interrupt vectors No. 23 to No. 26. This register is valid when the IRQQCMODE:IRQCMODE bit is "1".

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	INTSEL7								INTSEL6							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	INTSEL5								INTSEL4							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

Register Functions

[bit31:24] INTSEL7:
These bits select* the interrupt factor for the interrupt vector No. 26.

[bit23:16] INTSEL6:
These bits select* the interrupt factor for the interrupt vector No. 25.

[bit15:8] INTSEL5:
These bits select* the interrupt factor for the interrupt vector No. 24.

[bit7:0] INTSEL4:
These bits select* the interrupt factor for the interrupt vector No. 23.

*: See Table 2-3 for details of selection interrupt factors

Notes:

- *The interrupt factors selected by RCINTSEL0 are masked with IRQ11 to IRQ31. (The usable bits in IRQ11MON to IRQ31MON Registers are also masked.)*
- *Ensure that all interrupt factors selected in the INTSEL0 to INTSEL7 bits are different when selecting interrupt factors.*

Table 2-3 Interrupt Factor Selection

Setting of RCINTSELx:INTSELx	Interrupt Factor
0x00	No interrupt factor is selected.
0x01	External interrupt ch.0
0x02	External interrupt ch.1
0x03	External interrupt ch.2
0x04	External interrupt ch.3
0x05	External interrupt ch.4
0x06	External interrupt ch.5
0x07	External interrupt ch.6
0x08	External interrupt ch.7
0x09	External interrupt ch.8
0x0A	External interrupt ch.9
0x0B	External interrupt ch.10
0x0C	External interrupt ch.11
0x0D	IRQ0/IRQ1 of base timer ch.0
0x0E	IRQ0/IRQ1 of base timer ch.1
0x0F	IRQ0/IRQ1 of base timer ch.2
0x10	IRQ0/IRQ1 of base timer ch.3
0x11	IRQ0/IRQ1 of base timer ch.4
0x12	IRQ0/IRQ1 of base timer ch.5
0x13	IRQ0/IRQ1 of base timer ch.6
0x14	IRQ0/IRQ1 of base timer ch.7
0x15	Reception interrupt of MFS ch.0
0x16	Reception interrupt of MFS ch.1
0x17	Reception interrupt of MFS ch.2
0x18	Reception interrupt of MFS ch.3
0x19	Zero detection interrupt of MFT unit 0 free-run timer ch.0
0x1A	Zero detection interrupt of MFT unit 1 free-run timer ch.0
0x1B	Zero detection interrupt of MFT unit 2 free-run timer ch.0
0x1C	Reserved
0x1D	Reserved
0x1E	Reserved
0x1F	Reserved
0x20	Reception interrupt of MFS ch.8
0x21	Reception interrupt of MFS ch.9
0x22	Reception interrupt of MFS ch.10
0x23	Reception interrupt of MFS ch.11
0x24	Reception interrupt of MFS ch.12
0x25	Reception interrupt of MFS ch.13
0x26	Reception interrupt of MFS ch.14
0x27	Reception interrupt of MFS ch.15
0x28	Transmission/Status Interrupt of MFS ch.8
0x29	Transmission/Status Interrupt of MFS ch.9
0x2A	Transmission/Status Interrupt of MFS ch.10
0x2B	Transmission/Status Interrupt of MFS ch.11

Setting of RCINTSELx:INTSELx	Interrupt Factor
0x2C	Transmission/Status Interrupt of MFS ch.12
0x2D	Transmission/Status Interrupt of MFS ch.13
0x2E	Transmission/Status Interrupt of MFS ch.14
0x2F	Transmission/Status Interrupt of MFS ch.15
0x30 to 0xFF	Reserved

3. Usage Precautions

Be careful with the following points when using the interrupt controller.

- The interrupt request signals from each of the peripheral resources are notified by level. When exiting from interrupt processing, always clear the interrupt request.
- The NMIX pin is allocated shared with a general-purpose port. The initial value after reset is released is configured as a general-purpose port, and the NMI input is masked. In order to use the NMI, enable NMI by configuring the port setting. See the chapter of "External Interrupt and NMI Control Unit" for details.
- See the chapters for each of the macros for the correlation between the specific event detection registers and interrupt enable registers in each peripheral resource.

CHAPTER 7-5: Interrupts Top (TYPE2)



This chapter explains interrupt controller (TYPE2) and peripheral interrupt requests (TYPE2).

1. Overview
2. Configuration

CODE: 9BFIRQC_B_FM0+-E01.0 ???

1. Overview

The Cortex-M0+ CPU core is equipped with the Nested Vectored Interrupt Controller (NVIC) inside the core. The NVIC supports reserved system exceptions and 32 peripheral interrupts, and can set the priority order of 4 interrupt priority levels (with a built-in 2-bit register). This section explains interrupt signals from peripheral functions installed in the microcontroller (Type-2) and the connection between the NVIC and the interrupt signals (Type-2).

2. Configuration

Block Diagram

Figure 1-1 Connection between Interrupt Signals (Type-2) and NVIC

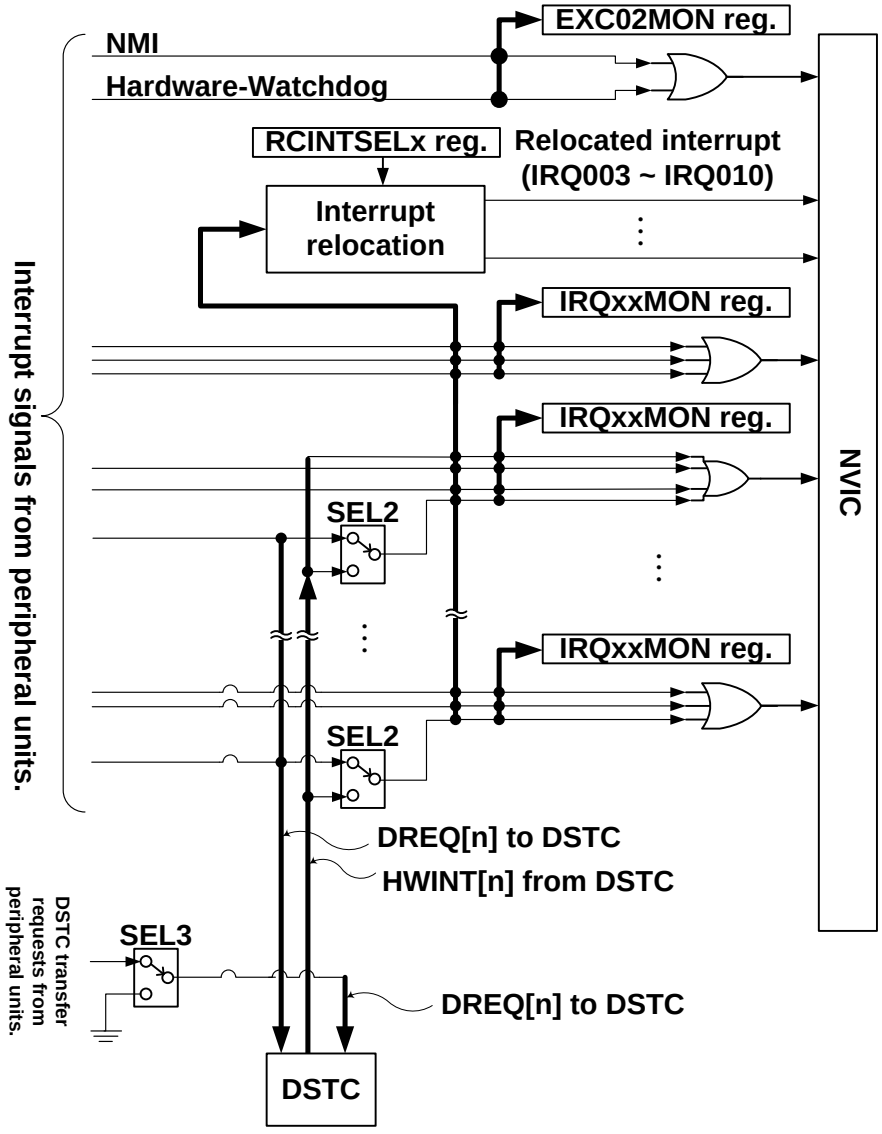


Figure 1-1 illustrates how the NVIC is connected to the interrupt signals input from peripheral functions, and the DSTC. Details of the connection are explained below.

NVIC

The NVIC supports reserved system exceptions and 32 peripheral interrupts. For details of the NVIC, refer to "Cortex-M0+ Technical Reference Manual". In "Cortex-M0+ Technical Reference Manual", an exception other than the reserved system exceptions is defined as "external interrupt (IRQ)". In this document, the external interrupt (IRQ) is called a peripheral interrupt to differentiate the external interrupt (IRQ) from the external interrupt from a microcontroller external input pin.

The interrupt priority register of the NVIC has a 2-bit configuration and can set 4 interrupt priority levels.

The respective priorities of reserved system exception no. 4 to no. 15 can be set by using the System Handler Priority Registers (addresses: 0xE000ED18, 0xE000ED1C, 0xE000ED20) installed in the NVIC.

The respective priorities of peripheral interrupts of exception no. 16 to no. 47 can be set by using the IRQ Priority Registers (addresses: 0xE000E400 to 0xE000E41C) installed in the NVIC.

The NVIC supports non-maskable interrupt (NMI) input.

Interrupt Aggregation and Batch Read Registers

Interrupt signals to be input from all peripheral functions ("Interrupt signals from peripheral units in Figure 1-1) are aggregated by the logic OR circuit in the figure. The aggregated interrupt signals are then connected to one of the 32 peripheral interrupts of the NVIC. See Table4-1 to check to which peripheral function an interrupt output of the NVIC is assigned.

Since interrupt signals are aggregated by the logical OR circuit, one interrupt of the NVIC is generated by multiple sources. When an interrupt is generated, the source that causes that interrupt can be identified by reading Interrupt Batch Read Registers (IRQxxxMON Register in Figure 1-1). The Interrupt Batch Read Registers (IRQ000MON to IRQ31MON) cover all interrupt inputs of the NVIC.

Each bit of IRQxxxMON register in the case of non-equipped in each product, is a reserved bit.

The non-maskable interrupt signal (NMI) which is from the external interrupt and NMI controllers, and the interrupt signal (HW-Watchdog) from the hardware watchdog timer are aggregated by a logical OR circuit and then connected to the input of exception no. 2 of the NVIC. When an interrupt of exception no. 2 is generated, the source of the interrupt, which is either external interrupt and NMI controllers or hardware watchdog timer, can be identified by reading the EXC02MON Register.

The NMI pin of the microcontroller is shared with a general-purpose port. After a reset has been released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the I/O port setting. For details, see Chapter "I/O Port". The NMI input signals are input to the NVIC via the external interrupt and NMI controllers.

Interrupt Relocate Function

Two types of the interrupt factor vector shown in 2can be selected by the IRQCMODE register setting. For IRQCMODE bit, refer to "Interrupts (B)".For the details of each setting, refer to the each chapter as following Table 1-1.
Moreover, the arbitrary interrupt factor can be selected with the RCINTSEL0 and RCINTSEL1 registers. For details on RCINTSEL0 and RCINTSEL1 registers, see "Interrupts (B)".

Table 1-1 Correspondence Table for Interrupt Chapter

IRQCMODE Setting	Reference
IRQCMODE=0 Relocate not selected	Chapter "Interrupts (A)"
IRQCMODE=1 Relocate selected	Chapter "Interrupts (B)"

DSTC Transfer Request and DSTC Transfer End Notification Selection

There are two types of peripheral functions using DMA transfer of DSTC; one using interrupts as transfer requests to DSTC and another one handling interrupts and transfer requests to DSTC separately.

The peripheral function using interrupts as transfer requests to DSTC can use the interrupt signal as the DMA transfer request signal to DSTC. With the DREQENB[n] register settings of DSTC, the interrupt signals from the peripheral functions are recognized as the DMA transfer requests.

The peripheral function handling interrupts and transfer requests to DSTC separately is programmable CRC. This peripheral function hold the DSTC transfer requests separately from interrupts.

Table 1-1 shows the differences of functions.

Table 1-2 Differences Among the Peripheral Functions Using DSTC

1. Peripheral Function Type	2. When DREQENB[n]=0			3. When DREQENB[n]=1		
	Circuit State	IRQxxxMON	Notification to NVIC	Circuit State	IRQxxxMON	Notification to NVIC
Peripheral functions handling interrupts and transfer requests to DSTC separately (programmable CRC)	SEL3 selects GND.	Interrupts from the peripheral functions are displayed. (The transfer completion interrupt from DSTC does not occurs.)	Interrupts from the peripheral functions (The transfer completion interrupt from DSTC does not occurs.)	SEL3 selects the DMA transfer requests from the peripheral functions.	Interrupts from the peripheral functions and transfer completion interrupts from DSTC are displayed separately by different registers.	Both interrupts from the peripheral functions and interrupts from DSTC
Peripheral functions using interrupts as the transfer requests to DSTC (peripheral functions using DSTC transfer other than programmable CRC)	SEL2 selects interrupts from peripheral functions.	Interrupts from the peripheral functions are displayed.	Interrupts from the peripheral functions	SEL2 selects interrupts from DSTC.	Interrupts from DSTC are displayed.	Interrupts from DSTC

In case of the peripheral functions using interrupts as transfer requests to DSTC, either of the DMA transfer completion notification signal output from DSTC (HWINT[n] from DSTC in the figure) or the interrupt signal from peripheral functions is selected by the selector circuit (SEL2 in the figure) as the interrupt signal input to NVIC. Switching of SEL2 is performed according to the DREQENB[n] register setting.

When the DMA transfer with DSTC is selected, the transfer completion interrupts from DSTC occurs instead of interrupts from peripheral. Because of the circuit configuration as shown in the figure, the transfer completion interrupts from DSTC can be read from the corresponding bit of the IRQxxxMON register. In addition, interrupt relocate function can be applied.

In case of the peripheral functions handling interrupts and transfer requests to DSTC separately, the DREQENB register setting determines whether the DSTC transfer requests from peripheral functions is connected to DSTC or not (SEL3 in the figure). In this case, in addition, interrupts from peripheral functions and transfer completion interrupts from DSTC are input to NVIC respectively, without the selector circuit (SEL2 in the figure). The transfer completion interrupts from DSTC and interrupts of peripheral functions can be read from the corresponding bit of the different IRQxxxMON registers respectively. In addition, interrupt relocate function can be applied.

For details of DSTC transfer requests from each peripheral function, refer to the chapter for each peripheral function.

DMA Transfer Acceptance Signal Connection

There are peripheral function blocks for which transfer request signals (interrupt signals) have to be cleared after the DMA transfer to those peripherals has ended. The transfer request signals for such peripheral functions are to be cleared by the DSTC. If the DMA transfer by the DSTC is selected in SEL2, the DMA transfer acknowledge signal (NOT ILLUSTRATED IN FIGURE 1-1) from the DSTC is connected to a peripheral function.

DMA Transfer Stop Signal Connection

The DMA transfer stop request signal is output from the multi-function serial unit (to be called MFS later in this document). According to the selection made in SEL2, the MFS (NOT ILLUSTRATED IN FIGURE 1-1) is connected to the DSTC as explained below.

If the connection between the DSTC and the MFS is selected in SEL2, the DSTC stops a transfer operation according to the transfer stop request signal. The DSTC cannot execute the transfer operation until the transfer stop request signal from the MFS is negated. The transfer stop request signal from the MFS is aggregated with the transfer end interrupt (HWINT[n] signal) of the DSTC by logical OR, and is notified to the NVIC as an interrupt signal.

CHAPTER 7-6: Interrupts (TYPE2-A)

This chapter explains Exception and Interrupt Factor Vectors and Registers at IRQCMODE=0.

1. Lists of Interrupts
2. Registers
3. Usage Precautions

CODE: 9BFIRQC_B_FM0+-E01.0 ???

1. Lists of Interrupts

This section shows a list of sources of exceptions and interrupt sources input to the NVIC, and a list of interrupts that can be transferred by the DMA transfer by the DSTC.

1.1 List of Exceptions and Interrupts

Table 1-1 shows a list of sources of exceptions and interrupt to be input to the NVIC. Below are details of columns in the table.

Exc no.
NVIC exception number

IRQ no.
Peripheral interrupt number (number = Exc no. - 16)

Vector offset
Storage offset address of the vector that an interrupt refers to

Bit
This indicates the number of a bit in a Batch Read Register (IRQxxxMON or EXC02MON) from which an interrupt source is read out. In the case of a single IRQ having multiple bit numbers. multiple sources are aggregated by logical OR, and a source can be read output from its corresponding bit. In the case of a single IRQ having only bit number "0", no multiple sources are aggregated by logical OR. "-" in this column indicates that there is no Batch Read Register for that exception or interrupt.

DSTC
A number, as the described value, indicates that it is the peripheral function using interrupts as transfer requests to DTSC and it is interrupt signal compatible with DMA transfer by DSTC, and the number shows the bit number of the DREQENB[n] register of DSTC. In this case, the DREQENB[n] register setting of DSTC determines the connection of SEL2 in the figure 1 1.

"*", as the described value, indicates that it is the peripheral function handling interrupts and transfer requests to DSTC separately, and only DMA transfer completion interrupt by DSTC is input. Transfer request signals from peripheral functions are not input to NVIC, and the transfer completion interrupt (HWINT[n]) generated from DSTC when the transfer is completed is input to NVIC without the selector circuit (SEL2) in the Figure 1 1.

"-", as the described value, indicates that it is not compatible with DMA transfer by DSTC.

Exception source and interrupt source
This column contains exception sources and interrupt sources. Some interrupts have multiple sources. Such interrupt sources of a peripheral function are aggregated by logical OR. Even if only one interrupt source from a peripheral source is shown, such peripheral function may have multiple interrupt sources aggregated by logical OR. For details, refer to the respective details of peripheral functions.

Table 1-1 List of Exception Sources and Interrupt Sources (IRQCMODE=0)

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
0	-	0x000	-	-	(Stack pointer initial value)
1	-	0x004	-	-	Reset
2	-	0x008	1	-	Hardware watchdog timer interrupt
			0	-	Non-maskable interrupt (NMI)
3	-	0x00C	-	-	Hard fault
4	-	0x010	-	-	Reserved
5	-	0x014	-	-	Reserved
6	-	0x018	-	-	Reserved
7 ~ 10	-	0x01C | 0x02B	-	-	Reserved
11	-	0x02C	-	-	SVCall (supervisor call)
12	-	0x030	-	-	Reserved
13	-	0x034	-	-	Reserved
14	-	0x038	-	-	PendSV
15	-	0x03C	-	-	SysTick
16	0	0x040	0	-	Anomalous frequency detection interrupt by Clock supervisor (FCS)
17	1	0x044	0	-	Software watchdog timer interrupt
18	2	0x048	0	-	Low-voltage detection (LVD) interrupt
19	3	0x04C	3	25	MFT unit 0 WFG timer 54 interrupt
			2	24	MFT unit 0 WFG timer 32 interrupt
			1	23	MFT unit 0 WFG timer 10 interrupt
			0	-	MFT unit 0 DTIF (motor emergency stop) interrupt
20	4	0x050	7	7	External Pin Interrupt ch.7
			6	6	External Pin Interrupt ch.6
			5	5	External Pin Interrupt ch.5
			4	4	External Pin Interrupt ch.4
			3	3	External Pin Interrupt ch.3
			2	2	External Pin Interrupt ch.2
			1	1	External Pin Interrupt ch.1
			0	0	External Pin Interrupt ch.0

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
21	5	0x054	15	-	External Pin Interrupt ch.23
			14	-	External Pin Interrupt ch.22
			13	-	External Pin Interrupt ch.21
			12	-	External Pin Interrupt ch.20
			11	-	External Pin Interrupt ch.19
			10	-	External Pin Interrupt ch.18
			9	-	External Pin Interrupt ch.17
			8	-	External Pin Interrupt ch.16
			7	-	External Pin Interrupt ch.15
			6	-	External Pin Interrupt ch.14
			5	-	External Pin Interrupt ch.13
			4	12	External Pin Interrupt ch.12
			3	11	External Pin Interrupt ch.11
			2	10	External Pin Interrupt ch.10
			1	9	External Pin Interrupt ch.9
			0	8	External Pin Interrupt ch.8
22	6	0x058	1	-	Dual timer interrupt ch.2
			0	-	Dual timer interrupt ch.1
23	7	0x05C	0	46	Reception Interrupt of MFS ch.0
24	8	0x060	1	-	Status Interrupt of MFS ch.0
			0	47	Transmit Interrupt of MFS ch.0
25	9	0x064	0	48	Reception Interrupt of MFS ch.1
26	10	0x068	1	-	Status Interrupt of MFS ch.1
			0	49	Transmit Interrupt of MFS ch.1
27	11	0x06C	0	50	Reception Interrupt of MFS ch.2
28	12	0x070	1	-	Status Interrupt of MFS ch.2
			0	51	Transmit Interrupt of MFS ch.2
29	13	0x074	0	52	Reception Interrupt of MFS ch.3
30	14	0x078	1	-	Status Interrupt of MFS ch.3
			0	53	Transmit Interrupt of MFS ch.3
31	15	0x07C	0	54	Reception Interrupt of MFS ch.4
32	16	0x080	1	-	Status Interrupt of MFS ch.4
			0	55	Transmit Interrupt of MFS ch.4
33	17	0x084	0	56	Reception Interrupt of MFS ch.5
34	18	0x088	1	-	Status Interrupt of MFS ch.5
			0	57	Transmit Interrupt of MFS ch.5
35	19	0x08C	0	58	Reception Interrupt of MFS ch.6
36	20	0x090	1	-	Status Interrupt of MFS ch.6
			0	59	Transmit Interrupt of MFS ch.6
37	21	0x094	0	60	Reception Interrupt of MFS ch.7
38	22	0x098	1	-	Status Interrupt of MFS ch.7
			0	61	Transmit Interrupt of MFS ch.7

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
39	23	0x09C	12	-	DSTC ERINT interrupt
			11	-	DSTC SWINT interrupt
			10	-	Smart card (IC card) interrupt ch.0
			9	-	HDMI-CEC remote control reception interrupt ch.0
			8	-	Reserved
			7	-	Reserved
			6	-	Reserved
			5	-	Reserved
			4	-	Reserved
			3	-	Reserved
			2	44	PPG interrupt ch.4
			1	43	PPG interrupt ch.2
			0	42	PPG interrupt ch.0
40	24	0x0A0	10	-	Smart card (IC card) interrupt ch.1
			9	-	HDMI-CEC remote control reception interrupt ch.1
			8~6	-	reserved
			5	-	Real timer counter interrupt
			4	45	Watch counter interrupt
			3	-	PLL of USB oscillation stabilization wait completion interrupt
			2	-	Main PLL oscillation stabilization wait completion interrupt
			1	-	Sub clock oscillation stabilization wait completion interrupt
			0	-	Main clock oscillation stabilization wait completion interrupt
41	25	0x0A4	4	-	ADC unit0 Range comparison result interrupt
			3	-	ADC unit0 result comparison interrupt
			2	-	ADC unit0 FIFO overrun interrupt
			1	62	ADC unit0 Scan conversion interrupt
			0	63	ADC unit0 Priority conversion interrupt
42	26	0x0A8	5	-	USB ch.0 host SOFIRQ interrupt USB ch.0 host CMPIRQ interrupt
			4	-	USB ch.0 host DIRQ interrupt USB ch.0 host URIRQ interrupt USB ch.0 host RWKIRQ interrupt USB ch.0 host CNNIRQ interrupt
			3	-	USB ch.0 device SPK interrupt
			2	-	USB ch.0 device SUSP interrupt USB ch.0 device SOF interrupt USB ch.0 device BRST interrupt USB ch.0 device CONF interrupt USB ch.0 device WKUP interrupt
			1	-	USB ch.0 device endpoint 0 DRQO interrupt
			0	-	USB ch.0 device endpoint 0 DRQI interrupt
43	27	0x0AC	5	-	LCD controller interrupt
			4~0	-	Reserved

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
44	28	0x0B0	5	31	MFT unit 0 Free-run Timer zero detection interrupt ch.2
			4	30	MFT unit 0 Free-run Timer zero detection interrupt ch.1
			3	29	MFT unit 0 Free-run Timer zero detection interrupt ch.0
			2	28	MFT unit 0 Free-run Timer peak detection interrupt ch.2
			1	27	MFT unit 0 Free-run Timer peak detection interrupt ch.1
			0	26	MFT unit 0 Free-run Timer peak detection interrupt ch.0
45	29	0x0B4	8	-	USB ch.0 device endpoint 5 DRQ interrupt
			7	-	USB ch.0 device endpoint 4 DRQ interrupt
			6	-	USB ch.0 device endpoint 3 DRQ interrupt
			5	-	USB ch.0 device endpoint 2 DRQ interrupt
			4	-	USB ch.0 device endpoint 1 DRQ interrupt
			3	35	MFT unit 0 ICU input edge detection interrupt ch.3
			2	34	MFT unit 0 ICU input edge detection interrupt ch.2
			1	33	MFT unit 0 ICU input edge detection interrupt ch.1
			0	32	MFT unit 0 ICU input edge detection interrupt ch.0
46	30	0x0B8	5	41	MFT unit 0 OCU match detection interrupt ch.5
			4	40	MFT unit 0 OCU match detection interrupt ch.4
			3	39	MFT unit 0 OCU match detection interrupt ch.3
			2	38	MFT unit 0 OCU match detection interrupt ch.2
			1	37	MFT unit 0 OCU match detection interrupt ch.1
			0	36	MFT unit 0 OCU match detection interrupt ch.0
47	31	0x0BC	27	-	Flash memory RDY, HANG interrupt
			26~16	-	reserved
			15	-	Base timer ch.7 source 1 (IRQ1)
			14	-	Base timer ch.7 source 0 (IRQ0)
			13	-	Base timer ch.6 source 1 (IRQ1)
			12	22	Base timer ch.6 source 0 (IRQ0)
			11	-	Base timer ch.5 source 1 (IRQ1)
			10	-	Base timer ch.5 source 0 (IRQ0)
			9	-	Base timer ch.4 source 1 (IRQ1)
			8	21	Base timer ch.4 source 0 (IRQ0)
			7	20	Base timer ch.3 source 1 (IRQ1)
			6	19	Base timer ch.3 source 0 (IRQ0)
			5	18	Base timer ch.2 source 1 (IRQ1)
			4	17	Base timer ch.2 source 0 (IRQ0)
			3	16	Base timer ch.1 source 1 (IRQ1)
			2	15	Base timer ch.1 source 0 (IRQ0)
			1	14	Base timer ch.0 source 1 (IRQ1)
			0	13	Base timer ch.0 source 0 (IRQ0)

1.2 Interrupt Signals Input to DSTC

Table 1-2 shows interrupt signals input as transfer request signals to the DSTC. Numbers in the table correspond to the numbers of the DREQENB[n] Registers of the DSTC.

Table 1-2 List of Interrupt Signals Input to DSTC

Number	Interrupt Signal Name
0	External pin interrupt ch.0
1	External pin interrupt ch.1
2	External pin interrupt ch.2
3	External pin interrupt ch.3
4	External pin interrupt ch.4
5	External pin interrupt ch.5
6	External pin interrupt ch.6
7	External pin interrupt ch.7
8	External pin interrupt ch.8
9	External pin interrupt ch.9
10	External pin interrupt ch.10
11	External pin interrupt ch.11
12	External pin interrupt ch.12
13	Base timer ch.0 source 0 (IRQ0) interrupt
14	Base timer ch.0 source 1 (IRQ1) interrupt
15	Base timer ch.1 source 0 (IRQ0) interrupt
16	Base timer ch.1 source 1 (IRQ1) interrupt
17	Base timer ch.2 source 0 (IRQ0) interrupt
18	Base timer ch.2 source 1 (IRQ1) interrupt
19	Base timer ch.3 source 0 (IRQ0) interrupt
20	Base timer ch.3 source 1 (IRQ1) interrupt
21	Base timer ch.4 source 0 (IRQ0) interrupt
22	Base timer ch.6 source 0 (IRQ0) interrupt
23	MFT unit 0 WFG timer 10 interrupt
24	MFT unit 0 WFG timer 32 interrupt
25	MFT unit 0 WFG timer 54 interrupt
26	MFT unit 0 FRT ch.0 peak value detection interrupt
27	MFT unit 0 FRT ch.1 peak value detection interrupt
28	MFT unit 0 FRT ch.2 peak value detection interrupt
29	MFT unit 0 FRT ch.0 zero detection interrupt
30	MFT unit 0 FRT ch.1 zero detection interrupt
31	MFT unit 0 FRT ch.2 zero detection interrupt
32	MFT unit 0 ICU ch.0 input edge detection interrupt
33	MFT unit 0 ICU ch.1 input edge detection interrupt
34	MFT unit 0 ICU ch.2 input edge detection interrupt
35	MFT unit 0 ICU ch.3 input edge detection interrupt
36	MFT unit 0 OCU ch.0 match detection interrupt
37	MFT unit 0 OCU ch.1 match detection interrupt
38	MFT unit 0 OCU ch.2 match detection interrupt

Number	Interrupt Signal Name
39	MFT unit 0 OCU ch.3 match detection interrupt
40	MFT unit 0 OCU ch.4 match detection interrupt
41	MFT unit 0 OCU ch.5 match detection interrupt
42	PPG ch.0 interrupt
43	PPG ch.2 interrupt
44	PPG ch.4 interrupt
45	Watch counter interrupt
46	MFS ch.0 reception interrupt
47	MFS ch.0 transmission interrupt
48	MFS ch.1 reception interrupt
49	MFS ch.1 transmission interrupt
50	MFS ch.2 reception interrupt
51	MFS ch.2 transmission interrupt
52	MFS ch.3 reception interrupt
53	MFS ch.3 transmission interrupt
54	MFS ch.4 reception interrupt
55	MFS ch.4 transmission interrupt
56	MFS ch.5 reception interrupt
57	MFS ch.5 transmission interrupt
58	MFS ch.6 reception interrupt
59	MFS ch.6 transmission interrupt
60	MFS ch.7 reception interrupt
61	MFS ch.7 transmission interrupt
62	A/D converter unit 0 scan conversion interrupt
63	A/D converter unit 0 priority conversion interrupt

2. Registers

This section explains the respective details of registers.

Register List

Abbreviation	Register Name	Reference
EXC02MON	EXC02 Batch Read Register	2.1
IRQ00MON	IRQ000 Batch Read Register	2.2
IRQ01MON	IRQ001 Batch Read Register	
IRQ02MON	IRQ002 Batch Read Register	
IRQ03MON	IRQ003 Batch Read Register	
IRQ04MON	IRQ004 Batch Read Register	
IRQ05MON	IRQ005 Batch Read Register	
IRQ06MON	IRQ006 Batch Read Register	
IRQ07MON	IRQ007 Batch Read Register	
IRQ08MON	IRQ008 Batch Read Register	
IRQ09MON	IRQ009 Batch Read Register	
IRQ10MON	IRQ010 Batch Read Register	
IRQ11MON	IRQ011 Batch Read Register	
IRQ12MON	IRQ012 Batch Read Register	
IRQ13MON	IRQ013 Batch Read Register	
IRQ14MON	IRQ014 Batch Read Register	
IRQ15MON	IRQ015 Batch Read Register	
IRQ16MON	IRQ016 Batch Read Register	
IRQ17MON	IRQ017 Batch Read Register	
IRQ18MON	IRQ018 Batch Read Register	
IRQ19MON	IRQ019 Batch Read Register	
IRQ20MON	IRQ020 Batch Read Register	
IRQ21MON	IRQ021 Batch Read Register	
IRQ22MON	IRQ022 Batch Read Register	
IRQ23MON	IRQ023 Batch Read Register	
IRQ24MON	IRQ024 Batch Read Register	
IRQ25MON	IRQ025 Batch Read Register	
IRQ26MON	IRQ026 Batch Read Register	
IRQ27MON	IRQ027 Batch Read Register	
IRQ28MON	IRQ028 Batch Read Register	
IRQ29MON	IRQ029 Batch Read Register	
IRQ30MON	IRQ030 Batch Read Register	
IRQ31MON	IRQ031 Batch Read Register	

2.1 EXC02 Batch Read Register (EXC02MON)

The EXC02 Batch Read Register (EXC02MON) can read out at once all interrupts (NMI and hardware watchdog interrupt) assigned to exception no. 2.

Register Configuration

bit	31							8
Field	Reserved							
Attribute	R							
Initial value	0x000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved						HWINT	NMI
Attribute	R						R	R
Initial value	000000						0	0

Register Function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1] HWINT

Value	Description
0	There is no interrupt request of the hardware watchdog timer.
1	An interrupt request of the hardware watchdog timer has been made.

[bit0] NMI

Value	Description
0	There is no interrupt request of the NMIX external pin.
1	An interrupt request of the NMIX external pin has been made.

2.2 IRQxx Batch Read Register (IRQxxMON)

The IRQxx Batch Read Register indicates 32 registers that are IRQ00MON – IRQ31MON. These are corresponding with IRQ00 – IRQ31 interrupt inputs of the NVIC respectively. The status of interrupt signals that are aggregated by logic OR circuit can be read out by these registers.

See Table 1-1 to check which bit in this register is assigned to which interrupt from peripheral function.

Register Configuration

Bit	31	30	29	28	27	26	25	24
Field	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Field	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Register Function

[bit31:0] INT31 – INT00

Operation	Description
Write	Write access is ignored.
Read 0	The interrupt assigned to this bit does not occur.
Read 1	The interrupt assigned to this bit occurs.

3. Usage Precautions

Note the following when using the interrupt controller.

- The interrupt controller is notified of the interrupt request signals from peripheral functions in terms of level. When exiting the processing of an interrupt, always clear the interrupt request for that interrupt.
- When other interruption with a high priority is processing, the interrupt request from peripheral functions notified to NVIC is pended inside NVIC. When canceling the interrupt request which was pended inside NVIC, clear the interrupt request from peripheral functions, and clear the interrupt request which was pended inside NVIC by the Interrupt Clear-Pending Registers (addresses: 0xE000E280 to 0xE000E29C) installed in the NVIC.
- The NMIX pin is shared with a general-purpose port. After a reset is released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the port setting. For details, see Chapter "External Interrupt and NMI Controller".
- If the DMA transfer by the DSTC is used, the transfer end interrupt (HWINT[n]) from the DSTC is generated instead of the interrupt from a peripheral function. Due to the above configuration, the NVIC makes an interrupt from a peripheral function, and a transfer end interrupt from the DSTC jump to the same interrupt vector. Use the DREQENB[n] Register to select the interrupt to be processed.
  However, for some peripheral functions (MFSI2S, HS-SPICNT and programmable CRC) handling interrupts and transfer requests to DSTC separately, the DREQENB register setting of DSTC determines whether the DMA transfer is performed or not. In this case, interrupts from peripheral functions and transfer completion interrupts from DSTC are input to NVIC respectively.
- For the relationship between specific event detection registers and interrupt enable registers in a peripheral function, see the chapter on that peripheral function.

CHAPTER 7-7: Interrupts (TYPE2-B)

This chapter explains Exception and Interrupt Factor Vectors and Registers at IRQCMODE=1.

1. Lists of Interrupts
2. Registers
3. Usage Precautions

CODE: 9BFIRQC_B_FM0+-E01.0 ???

1. Lists of Interrupts

This section shows a list of sources of exceptions and interrupt sources input to the NVIC, and a list of interrupts that can be transferred by the DMA transfer by the DSTC.

1.1 List of Exceptions and Interrupts

Table 1-1 shows a list of sources of exceptions and interrupt to be input to the NVIC. Below are details of columns in the table.

Exc no.
NVIC exception number

IRQ no
Peripheral interrupt number (number = Exc no. - 16)

Vector offset
Storage offset address of the vector that an interrupt refers to

Bit
This indicates the number of a bit in a Batch Read Register (IRQxxxMON or EXC02MON) from which an interrupt source is read out. In the case of a single IRQ having multiple bit numbers. multiple sources are aggregated by logical OR, and a source can be read output from its corresponding bit. In the case of a single IRQ having only bit number "0", no multiple sources are aggregated by logical OR. "-" in this column indicates that there is no Batch Read Register for that exception or interrupt.

DSTC
A number, as the described value, indicates that it is the peripheral function using interrupts as transfer requests to DTSC and it is interrupt signal compatible with DMA transfer by DSTC, and the number shows the bit number of the DREQENB[n] register of DSTC. In this case, the DREQENB[n] register setting of DSTC determines the connection of SEL2 in the figure 1 1.

"*", as the described value, indicates that it is the peripheral function handling interrupts and transfer requests to DSTC separately, and only DMA transfer completion interrupt by DSTC is input. Transfer request signals from peripheral functions are not input to NVIC, and the transfer completion interrupt (HWINT[n]) generated from DSTC when the transfer is completed is input to NVIC without the selector circuit (SEL2) in the Figure 1 1.

"-", as the described value, indicates that it is not compatible with DMA transfer by DSTC.

Exception source and interrupt source
This column contains exception sources and interrupt sources. Some interrupts have multiple sources. Such interrupt sources of a peripheral function are aggregated by logical OR. Even if only one interrupt source from a peripheral source is shown, such peripheral function may have multiple interrupt sources aggregated by logical OR. For details, refer to the respective details of peripheral functions.

Table 1-1 List of Exception Sources and Interrupt Sources (IRQCMODE=1)

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
0	-	0x000	-	-	(Stack pointer initial value)
1	-	0x004	-	-	Reset
2	-	0x008	1	-	Hardware watchdog timer interrupt
			0	-	Non-maskable interrupt (NMI)
3	-	0x00C	-	-	Hard fault
4	-	0x010	-	-	Reserved
5	-	0x014	-	-	Reserved
6	-	0x018	-	-	Reserved
7 ~ 10	-	0x01C | 0x02B	-	-	Reserved
11	-	0x02C	-	-	SVCall (supervisor call)
12	-	0x030	-	-	Reserved
13	-	0x034	-	-	Reserved
14	-	0x038	-	-	PendSV
15	-	0x03C	-	-	SysTick
16	0	0x040	0	-	Anomalous frequency detection interrupt by Clock supervisor (FCS)
17	1	0x044	0	-	Software watchdog timer interrupt
18	2	0x048	0	-	Low-voltage detection (LVD) interrupt
19	3	0x04C	0	-	Relocatable Interrupt (This selected by the INTSEL0:RCINTSEL0 Register.)
20	4	0x050	0	-	Relocatable Interrupt (This selected by the INTSEL1:RCINTSEL0 Register.)
21	5	0x054	0	-	Relocatable Interrupt (This selected by the INTSEL2:RCINTSEL0 Register.)
22	6	0x058	0	-	Relocatable Interrupt (This selected by the INTSEL3:RCINTSEL0 Register.)
23	7	0x05C	0	-	Relocatable Interrupt (This selected by the INTSEL4:RCINTSEL1 Register.)
24	8	0x060	0	-	Relocatable Interrupt (This selected by the INTSEL5:RCINTSEL1 Register.)
25	9	0x064	0	-	Relocatable Interrupt (This selected by the INTSEL6:RCINTSEL1 Register.)
26	10	0x068	0	-	Relocatable Interrupt (This selected by the INTSEL7:RCINTSEL1 Register.)
27	11	0x06C	3	25	MFT unit0 Wave Form Generator 54
			2	24	MFT unit0 Wave Form Generator 32
			1	23	MFT unit0 Wave Form Generator 10
			0	-	MFT unit 0 DTIF (motor emergency stop) interrupt

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
28	12	0x070	7	7	External Pin Interrupt ch.7
			6	6	External Pin Interrupt ch.6
			5	5	External Pin Interrupt ch.5
			4	4	External Pin Interrupt ch.4
			3	3	External Pin Interrupt ch.3
			2	2	External Pin Interrupt ch.2
			1	1	External Pin Interrupt ch.1
			0	0	External Pin Interrupt ch.0
29	13	0x074	15	-	External Pin Interrupt ch.23
			14	-	External Pin Interrupt ch.22
			13	-	External Pin Interrupt ch.21
			12	-	External Pin Interrupt ch.20
			11	-	External Pin Interrupt ch.19
			10	-	External Pin Interrupt ch.18
			9	-	External Pin Interrupt ch.17
			8	-	External Pin Interrupt ch.16
			7	-	External Pin Interrupt ch.15
			6	-	External Pin Interrupt ch.14
			5	-	External Pin Interrupt ch.13
			4	12	External Pin Interrupt ch.12
			3	11	External Pin Interrupt ch.11
			2	10	External Pin Interrupt ch.10
			1	9	External Pin Interrupt ch.9
			0	8	External Pin Interrupt ch.8
30	14	0x078	1	-	Dual timer interrupt ch.2
			0	-	Dual timer interrupt ch.1
31	15	0x07C	2	-	Status Interrupt of MFS ch.0
			1	47	Transmit Interrupt of MFS ch.0
			0	46	Reception Interrupt of MFS ch.0
32	16	0x080	2	-	Status Interrupt of MFS ch.1
			1	49	Transmit Interrupt of MFS ch.1
			0	48	Reception Interrupt of MFS ch.1
33	17	0x084	2	-	Status Interrupt of MFS ch.2
			1	51	Transmit Interrupt of MFS ch.2
			0	50	Reception Interrupt of MFS ch.2
34	18	0x088	2	-	Status Interrupt of MFS ch.3
			1	53	Transmit Interrupt of MFS ch.3
			0	52	Reception Interrupt of MFS ch.3
35	19	0x08C	0	54	Reception Interrupt of MFS ch.4
36	20	0x090	1	-	Status Interrupt of MFS ch.4
			0	55	Transmit Interrupt of MFS ch.4
37	21	0x094	0	56	Reception Interrupt of MFS ch.5

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
38	22	0x098	1	-	Status Interrupt of MFS ch.5
			0	57	Transmit Interrupt of MFS ch.5
39	23	0x09C	12	-	DSTC ERINT interrupt
			11	-	DSTC SWINT interrupt
			10	-	Smart card (IC card) interrupt ch.0
			9	-	HDMI-CEC remote control reception interrupt ch.0
			8~3	-	reserved
			2	44	PPG interrupt ch.4
			1	43	PPG interrupt ch.2
			0	42	PPG interrupt ch.0
40	24	0x0A0	10	-	Smart card (IC card) interrupt ch.1
			9	-	HDMI-CEC remote control reception interrupt ch.1
			8~6	-	reserved
			5	-	Real timer counter interrupt
			4	45	Watch counter interrupt
			3	-	PLL of USB oscillation stabilization wait completion interrupt
			2	-	Main PLL oscillation stabilization wait completion interrupt
			1	-	Sub clock oscillation stabilization wait completion interrupt
			0	-	Main clock oscillation stabilization wait completion interrupt
41	25	0x0A4	4	-	ADC unit0 Range comparison result interrupt
			3	-	ADC unit0 result comparison interrupt
			2	-	ADC unit0 FIFO overrun interrupt
			1	62	ADC unit0 Scan conversion interrupt
			0	63	ADC unit0 Priority conversion interrupt
42	26	0x0A8	5	-	USB ch.0 host SOFIRQ interrupt USB ch.0 host CMPIRQ interrupt
			4	-	USB ch.0 host DIRQ interrupt USB ch.0 host URIRQ interrupt USB ch.0 host RWKIRQ interrupt USB ch.0 host CNNIRQ interrupt
			3	-	USB ch.0 device SPK interrupt
			2	-	USB ch.0 device SUSP interrupt USB ch.0 device SOF interrupt USB ch.0 device BRST interrupt USB ch.0 device CONF interrupt USB ch.0 device WKUP interrupt
			1	-	USB ch.0 device endpoint 0 DRQO interrupt
			0	-	USB ch.0 device endpoint 0 DRQI interrupt
43	27	0x0AC	5	-	LCD controller interrupt
			4~0	-	reserved

Exc. No.	IRQ No.	Vector Offset	Bit	DSTC	Exception and Interrupt Source
44	28	0x0B0	15	41	MFT unit 0 OCU match detection interrupt ch.5
			14	40	MFT unit 0 OCU match detection interrupt ch.4
			13	39	MFT unit 0 OCU match detection interrupt ch.3
			12	38	MFT unit 0 OCU match detection interrupt ch.2
			11	37	MFT unit 0 OCU match detection interrupt ch.1
			10		MFT unit 0 OCU match detection interrupt ch.0
			9		MFT unit 0 ICU input edge detection interrupt ch.3
			8		MFT unit 0 ICU input edge detection interrupt ch.2
			7		MFT unit 0 ICU input edge detection interrupt ch.1
			6		MFT unit 0 ICU input edge detection interrupt ch.0
			5	31	MFT unit 0 Free-run Timer zero detection interrupt ch.2
			4	30	MFT unit 0 Free-run Timer zero detection interrupt ch.1
			3	29	MFT unit 0 Free-run Timer zero detection interrupt ch.0
			2	28	MFT unit 0 Free-run Timer peak detection interrupt ch.2
			1	27	MFT unit 0 Free-run Timer peak detection interrupt ch.1
			0	26	MFT unit 0 Free-run Timer peak detection interrupt ch.0
45	29	0x0B4	8	36	USB ch.0 device endpoint 5 DRQ interrupt
			7	35	USB ch.0 device endpoint 4 DRQ interrupt
			6	34	USB ch.0 device endpoint 3 DRQ interrupt
			5	33	USB ch.0 device endpoint 2 DRQ interrupt
			4	32	USB ch.0 device endpoint 1 DRQ interrupt
			3~0	-	reserved
46	30	0x0B8	31~0	-	reserved
47	31	0x0BC	27	-	Flash memory RDY, HANG interrupt
			26~16	-	reserved
			15	-	Base timer ch.7 source 1 (IRQ1)
			14	-	Base timer ch.7 source 0 (IRQ0)
			13	-	Base timer ch.6 source 1 (IRQ1)
			12	22	Base timer ch.6 source 0 (IRQ0)
			11	-	Base timer ch.5 source 1 (IRQ1)
			10	-	Base timer ch.5 source 0 (IRQ0)
			9	-	Base timer ch.4 source 1 (IRQ1)
			8	21	Base timer ch.4 source 0 (IRQ0)
			7	20	Base timer ch.3 source 1 (IRQ1)
			6	19	Base timer ch.3 source 0 (IRQ0)
			5	18	Base timer ch.2 source 1 (IRQ1)
			4	17	Base timer ch.2 source 0 (IRQ0)
			3	16	Base timer ch.1 source 1 (IRQ1)
			2	15	Base timer ch.1 source 0 (IRQ0)
			1	14	Base timer ch.0 source 1 (IRQ1)
			0	13	Base timer ch.0 source 0 (IRQ0)

1.2 Interrupt Signals Input to DSTC

Table 1-2 shows interrupt signals input as transfer request signals to the DSTC. Numbers in the table correspond to the numbers of the DREQENB[n] Registers of the DSTC.

Table 1-2 List of Interrupt Signals Input to DSTC

Number	Interrupt Signal Name
0	External pin interrupt ch.0
1	External pin interrupt ch.1
2	External pin interrupt ch.2
3	External pin interrupt ch.3
4	External pin interrupt ch.4
5	External pin interrupt ch.5
6	External pin interrupt ch.6
7	External pin interrupt ch.7
8	External pin interrupt ch.8
9	External pin interrupt ch.9
10	External pin interrupt ch.10
11	External pin interrupt ch.11
12	External pin interrupt ch.12
13	Base timer ch.0 source 0 (IRQ0) interrupt
14	Base timer ch.0 source 1 (IRQ1) interrupt
15	Base timer ch.1 source 0 (IRQ0) interrupt
16	Base timer ch.1 source 1 (IRQ1) interrupt
17	Base timer ch.2 source 0 (IRQ0) interrupt
18	Base timer ch.2 source 1 (IRQ1) interrupt
19	Base timer ch.3 source 0 (IRQ0) interrupt
20	Base timer ch.3 source 1 (IRQ1) interrupt
21	Base timer ch.4 source 0 (IRQ0) interrupt
22	Base timer ch.6 source 0 (IRQ0) interrupt
23	MFT unit 0 WFG timer 10 interrupt
24	MFT unit 0 WFG timer 32 interrupt
25	MFT unit 0 WFG timer 54 interrupt
26	MFT unit 0 FRT ch.0 peak value detection interrupt
27	MFT unit 0 FRT ch.1 peak value detection interrupt
28	MFT unit 0 FRT ch.2 peak value detection interrupt
29	MFT unit 0 FRT ch.0 zero detection interrupt
30	MFT unit 0 FRT ch.1 zero detection interrupt
31	MFT unit 0 FRT ch.2 zero detection interrupt
32	USB ch.0 device DRQ interrupt of endpoint 1
33	USB ch.0 device DRQ interrupt of endpoint 2
34	USB ch.0 device DRQ interrupt of endpoint 3
35	USB ch.0 device DRQ interrupt of endpoint 4
36	USB ch.0 device DRQ interrupt of endpoint 5
37	MFT unit 0 OCU ch.1 match detection interrupt
38	MFT unit 0 OCU ch.2 match detection interrupt

Number	Interrupt Signal Name
39	MFT unit 0 OCU ch.3 match detection interrupt
40	MFT unit 0 OCU ch.4 match detection interrupt
41	MFT unit 0 OCU ch.5 match detection interrupt
42	PPG ch.0 interrupt
43	PPG ch.2 interrupt
44	PPG ch.4 interrupt
45	Watch counter interrupt
46	MFS ch.0 reception interrupt
47	MFS ch.0 transmission interrupt
48	MFS ch.1 reception interrupt
49	MFS ch.1 transmission interrupt
50	MFS ch.2 reception interrupt
51	MFS ch.2 transmission interrupt
52	MFS ch.3 reception interrupt
53	MFS ch.3 transmission interrupt
54	MFS ch.4 reception interrupt
55	MFS ch.4 transmission interrupt
56	MFS ch.5 reception interrupt
57	MFS ch.5 transmission interrupt
58	-
59	-
60	-
61	-
62	A/D converter unit 0 scan conversion interrupt
63	A/D converter unit 0 priority conversion interrupt

2. Registers

This section explains the respective details of registers.

Register List

Abbreviation	Register Name	Reference
EXC02MON	EXC02 Batch Read Register	2.1
IRQ00MON	IRQ000 Batch Read Register	2.2
IRQ01MON	IRQ001 Batch Read Register	
IRQ02MON	IRQ002 Batch Read Register	
IRQ03MON	IRQ003 Batch Read Register	
IRQ04MON	IRQ004 Batch Read Register	
IRQ05MON	IRQ005 Batch Read Register	
IRQ06MON	IRQ006 Batch Read Register	
IRQ07MON	IRQ007 Batch Read Register	
IRQ08MON	IRQ008 Batch Read Register	
IRQ09MON	IRQ009 Batch Read Register	
IRQ10MON	IRQ010 Batch Read Register	
IRQ11MON	IRQ011 Batch Read Register	
IRQ12MON	IRQ012 Batch Read Register	
IRQ13MON	IRQ013 Batch Read Register	
IRQ14MON	IRQ014 Batch Read Register	
IRQ15MON	IRQ015 Batch Read Register	
IRQ16MON	IRQ016 Batch Read Register	
IRQ17MON	IRQ017 Batch Read Register	
IRQ18MON	IRQ018 Batch Read Register	
IRQ19MON	IRQ019 Batch Read Register	
IRQ20MON	IRQ020 Batch Read Register	
IRQ21MON	IRQ021 Batch Read Register	
IRQ22MON	IRQ022 Batch Read Register	
IRQ23MON	IRQ023 Batch Read Register	
IRQ24MON	IRQ024 Batch Read Register	
IRQ25MON	IRQ025 Batch Read Register	
IRQ26MON	IRQ026 Batch Read Register	
IRQ27MON	IRQ027 Batch Read Register	
IRQ28MON	IRQ028 Batch Read Register	
IRQ29MON	IRQ029 Batch Read Register	
IRQ30MON	IRQ030 Batch Read Register	
IRQ31MON	IRQ031 Batch Read Register	
IRQCMODE	Interrupt Factor Vector Relocate Setting Register	2.3
RCINTSEL0	Interrupt Factor Selection Register 0	2.4
RCINTSEL1	Interrupt Factor Selection Register 1	2.5

2.1 EXC02 Batch Read Register (EXC02MON)

The EXC02 Batch Read Register (EXC02MON) can read out at once all interrupts (NMI and hardware watchdog interrupt) assigned to exception no. 2.

Register Configuration

bit	31							8
Field	Reserved							
Attribute	R							
Initial value	0x000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved						HWINT	NMI
Attribute	R						R	R
Initial value	000000						0	0

Register Function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1] HWINT

Value	Description
0	There is no interrupt request of the hardware watchdog timer.
1	An interrupt request of the hardware watchdog timer has been made.

[bit0] NMI

Value	Description
0	There is no interrupt request of the NMIX external pin.
1	An interrupt request of the NMIX external pin has been made.

2.2 IRQxx Batch Read Register (IRQxxMON)

The IRQxx Batch Read Register indicates 32 registers that are IRQ00MON – IRQ31MON. These are corresponding with IRQ00 – IRQ31 interrupt inputs of the NVIC respectively. The status of interrupt signals that are aggregated by logic OR circuit can be read out by these registers.

See Table 1-1 to check which bit in this register is assigned to which interrupt from peripheral function.

Register Configuration

Bit	31	30	29	28	27	26	25	24
Field	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Field	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Field	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Register Function

[bit31:0] INT31 – INT00

Operation	Description
Write	Write access is ignored.
Read 0	The interrupt assigned to this bit does not occur.
Read 1	The interrupt assigned to this bit occurs.

2.3 Interrupt Factor Vector Relocate Setting Register (IRQCMODE)

The Interrupt Factor Vector Relocate Setting Register (IRQCMODE) selects whether the interrupt factor vectors are assigned according to Table 1-1 in chapter "Interrupts (A)" or to Table 1-1 in this chapter.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															IRQCMODE
Attribute	R/W															R/W
Initial value	000000000000000															0

Register Functions

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] IRQCMODE:

Bit	Description
0	Assigns the interrupt factor vector according to Table 1-1 in chapter "Interrupts (A)".
1	Assigns the interrupt factor vector according to Table 1-1 in this chapter.

2.4 Interrupt Factor Selection Register 0 (RCINTSEL0)

The Interrupt Factor Selection Register 0 (RCINTSEL0) selects the interrupt factors for the exception No. 19 to No. 22 (IRQ03 to IRQ06). This register is valid when the IRQQCMODE:IRQCMODE bit is "1".

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	INTSEL3								INTSEL2							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	INTSEL1								INTSEL0							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

Register Functions

[bit31:24] INTSEL3:
These bits select* the interrupt factor for the exception No. 22 (IRQ06).

[bit23:16] INTSEL2:
These bits select* the interrupt factor for the exception No. 21 (IRQ05).

[bit15:8] INTSEL1:
These bits select* the interrupt factor for the exception No. 20 (IRQ04).

[bit7:0] INTSEL0:
These bits select* the interrupt factor for the exception No. 19 (IRQ03).

*: See Table 2-1 for details of selection interrupt factors

Notes:

- *The interrupt factors selected by RCINTSEL0 are masked with IRQ11 to IRQ31. (The usable bits in IRQ11MON to IRQ31MON Registers are also masked.)*
- *Ensure that all interrupt factors selected in the INTSEL0 to INTSEL7 bits are different when selecting interrupt factors.*

2.5 Interrupt Factor Selection Register 1 (RCINTSEL1)

The Interrupt Factor Selection Register 1 (RCINTSEL1) selects the interrupt factors for the exception No. 23 to No. 26 (IRQ07 to IRQ10). This register is valid when the IRQQCMODE:IRQCMODE bit is "1".

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	INTSEL7								INTSEL6							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	INTSEL5								INTSEL4							
Attribute	R/W								R/W							
Initial value	0x00								0x00							

Register Functions

[bit31:24] INTSEL7:
These bits select* the interrupt factor for the exception No. 26 (IRQ10).

[bit23:16] INTSEL6:
These bits select* the interrupt factor for the exception No. 25 (IRQ09).

[bit15:8] INTSEL5:
These bits select* the interrupt factor for the exception No. 24 (IRQ08).

[bit7:0] INTSEL4:
These bits select* the interrupt factor for the exception No. 23 (IRQ07).

*: See Table 2-1 for details of selection interrupt factors

Notes:

- *The interrupt factors selected by RCINTSEL0 are masked with IRQ11 to IRQ31. (The usable bits in IRQ11MON to IRQ31MON Registers are also masked.)*
- *Ensure that all interrupt factors selected in the INTSEL0 to INTSEL7 bits are different when selecting interrupt factors.*

Table 2-1 Interrupt Factor Selection

Setting of RCINTSELx:INTSELx	Interrupt Factor
0x00	No interrupt factor is selected.
0x01	External interrupt ch.0
0x02	External interrupt ch.1
0x03	External interrupt ch.2
0x04	External interrupt ch.3
0x05	External interrupt ch.4
0x06	External interrupt ch.5
0x07	External interrupt ch.6
0x08	External interrupt ch.7
0x09	External interrupt ch.8
0x0A	External interrupt ch.9
0x0B	External interrupt ch.10
0x0C	External interrupt ch.11
0x0D	IRQ0/IRQ1 of base timer ch.0
0x0E	IRQ0/IRQ1 of base timer ch.1
0x0F	IRQ0/IRQ1 of base timer ch.2
0x10	IRQ0/IRQ1 of base timer ch.3
0x11	IRQ0/IRQ1 of base timer ch.4
0x12	IRQ0/IRQ1 of base timer ch.5
0x13	IRQ0/IRQ1 of base timer ch.6
0x14	IRQ0/IRQ1 of base timer ch.7
0x15	Reception interrupt of MFS ch.0
0x16	Reception interrupt of MFS ch.1
0x17	Reception interrupt of MFS ch.2
0x18	Reception interrupt of MFS ch.3
0x19	Zero detection interrupt of MFT unit 0 free-run timer ch.0
0x1A to 0xFF	Reserved

3. Usage Precautions

Note the following when using the interrupt controller.

- The interrupt controller is notified of the interrupt request signals from peripheral functions in terms of level. When exiting the processing of an interrupt, always clear the interrupt request for that interrupt.
- When other interruption with a high priority is processing, the interrupt request from peripheral functions notified to NVIC is pended inside NVIC. When canceling the interrupt request which was pended inside NVIC, clear the interrupt request from peripheral functions, and clear the interrupt request which was pended inside NVIC by the Interrupt Clear-Pending Registers (addresses: 0xE000E280 to 0xE000E29C) installed in the NVIC.
- The NMIX pin is shared with a general-purpose port. After a reset is released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the port setting. For details, see Chapter "External Interrupt and NMI Controller".
- If the DMA transfer by the DSTC is used, the transfer end interrupt (HWINT[n]) from the DSTC is generated instead of the interrupt from a peripheral function. Due to the above configuration, the NVIC makes an interrupt from a peripheral function, and a transfer end interrupt from the DSTC jump to the same interrupt vector. Use the DREQENB[n] Register to select the interrupt to be processed.
  However, for some peripheral functions (MFSI2S, HS-SPICNT and programmable CRC) handling interrupts and transfer requests to DSTC separately, the DREQENB register setting of DSTC determines whether the DMA transfer is performed or not. In this case, interrupts from peripheral functions and transfer completion interrupts from DSTC are input to NVIC respectively.
- For the relationship between specific event detection registers and interrupt enable registers in a peripheral function, see the chapter on that peripheral function.

CHAPTER 7-8: Interrupts (TYPE3)

This chapter explains the interrupt controller (TYPE3) and peripheral interrupt requests (TYPE3).

1. Overview
2. Configuration and Function
3. Lists of Interrupts
4. Registers
5. Usage Precautions

CODE: 9BFIRQC_B_FM0-T3-E03.0

1. Overview

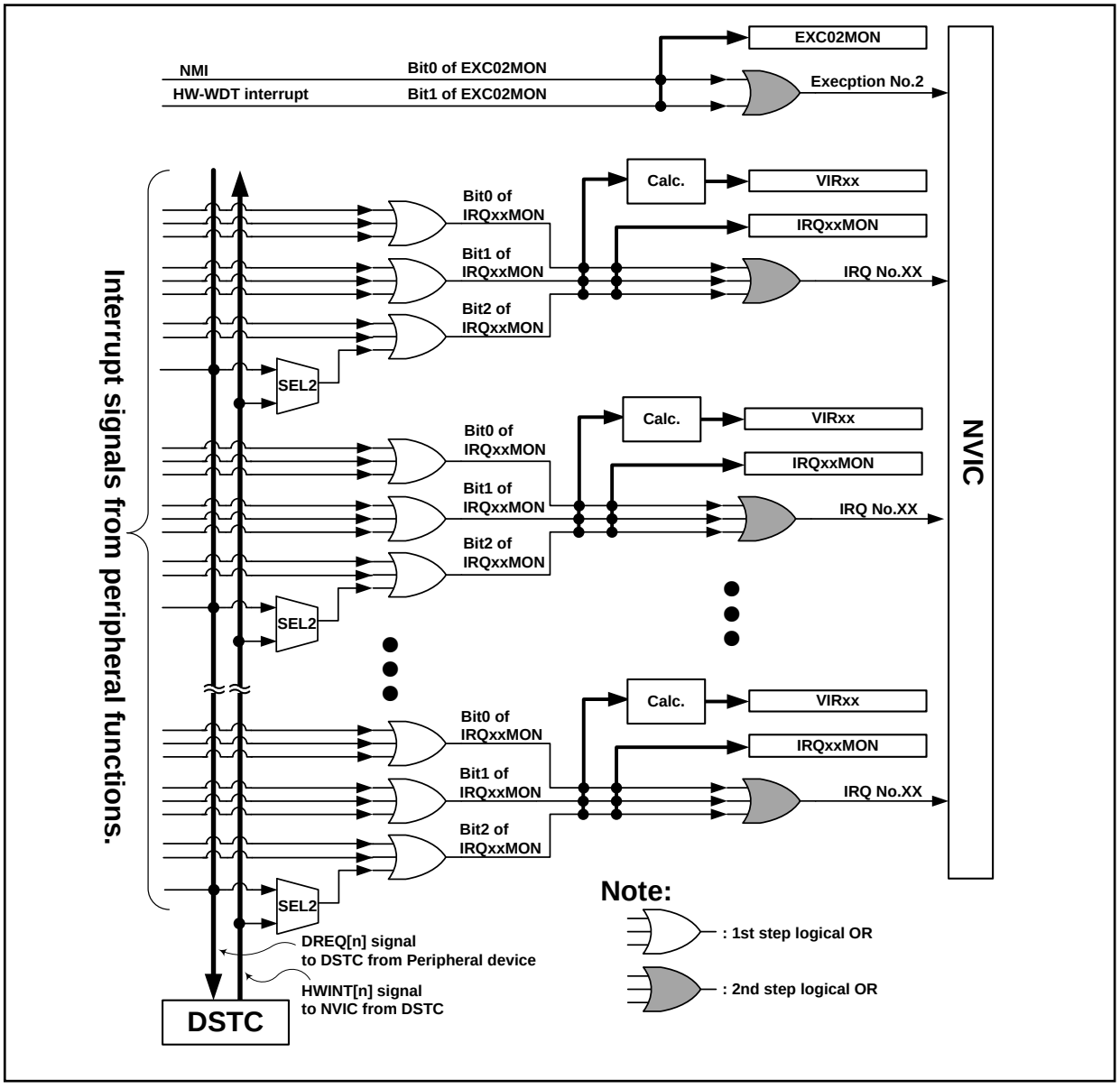
In this product, CPU core is equipped with the Nested Vectored Interrupt Controller (NVIC). The NVIC supports reserved system exceptions and 32 peripheral interrupts, and can set the priority order of 4 interrupt priority levels (with a built-in 2-bit register). This section explains the connection between the interrupt signals from peripheral functions installed in the microcontroller and the NVIC/DSTC.

2. Configuration and Function

2.1 Connection diagram

Figure 2-1 illustrates how the NVIC/DSTC is connected to the interrupt signals input from peripheral functions. Details are explained below.

Figure 2-1 Connection between Interrupt Signals and NVIC/DSTC



2.2 NVIC (Nested Vectored Interrupt Controller)

The NVIC (in Figure 2-1) supports reserved system exceptions and 32 peripheral interrupts. For details of the NVIC, refer to Cortex-M0+ Technical Reference Manual. In Cortex-M0 Technical Reference Manual, an exception other than the reserved system exceptions is defined as external interrupt (IRQ). In this document, the external interrupt (IRQ) is called a peripheral interrupt because of the distinction between the external interrupt (IRQ) and the external interrupt with an external input pin of the microcontroller.

The interrupt priority register of the NVIC has a 2-bit configuration and can set 4 interrupt priority levels.

The respective priorities of reserved system exception no.11, no.14, no.15 can be set by using the System Handler Priority Registers (addresses: 0xE000ED1C, 0xE000ED20) installed in the NVIC.

The respective priorities of peripheral interrupts no.16 to no.47 can be set by using the IRQ Priority Registers (addresses: 0xE000E400 to 0xE000E41C) installed in the NVIC.

The NVIC supports non-maskable interrupt (NMI) input.

2.3 Interrupt Aggregation and Batch Read Registers

Interrupt signals from peripheral functions (in Figure 2-1) are aggregated by the 1st step logical OR circuit (in Figure 2-1) and the 2nd step logical OR circuit (in Figure 2-1). The aggregated interrupt signals are then connected to one of the 32 peripheral interrupts of the NVIC. See Table 3-1,Table 3-2 and Table 3-3 to check which peripheral function interrupt signal is assigned to which IRQ input of the NVIC.

Since interrupt signals are aggregated by the logical OR circuit, one interrupt of the NVIC is generated by multiple sources. When an interrupt is generated, by reading Interrupt Batch Read Registers (IRQxxMON in Figure 2-1), CPU is able to identify which source, that is aggregated by 2nd step logical OR, generated that interrupt. The Interrupt Batch Read Registers cover all interrupt inputs of the NVIC. The 32 registers (IRQ00MON to IRQ31MON) are supported.

By reading Interrupt Batch Read Registers, CPU is not able to identify which source, that is aggregated by 1st step logical OR, generated interrupt. It is need to read an interrupt register in each peripheral function for identifying interrupt source.

The non-maskable interrupt signals (NMI) from the external interrupt and NMI controllers, and HW Watchdog interrupt from the hardware watchdog timer are aggregated by a 2nd step logical OR then connected to exception no.2 that is the input of the NVIC. When an exception no.2 is generated, the source of the interrupt, which is either NMI or HW-Watchdog interrupt, can be identified by reading the EXC02 Batch Read Register (EXC02MON in Figure 2-1).

The NMIX pin is shared with a general-purpose port. After a reset is released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the port setting. For details, see Chapter “I/O port” and “External Interrupt and NMI Controller”.

2.4 Vector Indicate Register

When interrupt occurs, CPU can use Vector Indicate Register (VIRxx in Figure 2-1) to quickly branch the interrupt operation. VIRxx consists of 32 registers (VIR00 – VIR32) that are corresponding with IRQ00–IRQ31 input of the NVIC. When interrupt occurs, CPU is able to read out a routine address value from VIRxx. This address value is decided depending on the status of interrupt signals that is aggregated by the 2nd step logic OR circuit. By placing top address of the interrupt handler in this address area, the CPU is able to quickly branch the interrupt operation. The usage of VIR00 in the event of interrupt (IRQ00) is explained below.

As shown in Table 2-1, place the top address value of IRQ interrupt handler in address area: 0x0000 0040 - 0x0000 00BC. Also, in address area: 0x0000 00C0 - 0x0000 01FC, place the top address value of the interrupt handler corresponding with the interrupt of each Bit0, Bit1 and Bit2 that are aggregated by the 2nd step logic OR circuit.

Table 2-1 Table of Interrupt vector

Address	Data
0x0000 0000	Stack pointer initial value
0x0000 0004	Exception 1: Reset vector
0x0000 0008	Exception 2: NMI/HW-WDT Handler top address
0x0000 000C	Exception 3: Hard fault Handler top address
0x0000 0010 - 0x0000 0028	Reserved
0x0000 002C	Exception 11: SVCall Handler top address
0x0000 0030 - 0x0000 0034	Reserved
0x0000 0038	Exception 14:PendSV Handler top address
0x0000 003C	Exception 15:SysTick Handler top address
0x0000 0040	IRQ00 Handler top address
0x0000 0044	IRQ01 Handler top address
....	....
0x0000 00B8	IRQ30 Handler top address
0x0000 00BC	IRQ31 Handler top address
0x0000 00C0	IRQ00 - bit0 Handler top address
0x0000 00C4	IRQ01 - bit0 Handler top address
....	....
0x0000 0138	IRQ30 - bit0 Handler top address
0x0000 013C	IRQ31 - bit0 Handler top address
0x0000 0140	IRQ00 - bit1 Handler top address
0x0000 0144	IRQ01 - bit1 Handler top address
....	....
0x0000 01B8	IRQ30 - bit1 Handler top address
0x0000 01BC	IRQ31 - bit1 Handler top address
0x0000 01C0	IRQ00 - bit2 Handler top address
0x0000 01C4	IRQ01 - bit2 Handler top address
....	....
0x0000 01F8	IRQ14 - bit2 Handler top address
0x0000 01FC	IRQ15 - bit2 Handler top address

Table 2-2 Status of IRQ00 and read value of VIR00

IRQ00 input signal status			VIR00 read value
Bit0 interrupt status	Bit1 interrupt status	Bit2 interrupt status	
1	Ignored	Ignored	0x0000 00C0 + VIR_OFFSET
0	1	Ignored	0x0000 0140 + VIR_OFFSET
0	0	1	0x0000 01C0 + VIR_OFFSET
0	0	0	Undefined value

When IRQ00 occurs, IRQ00 interrupt handler starts. Then CPU read out the address value from VIR00 in this IRQ00 interrupt handler. The address value to be read from VIR00 is shown in Table 2-2 . In case of Bit0 interrupt occurrence, 0x0000 00C0 is read. In case of Bit1 interrupt occurrence, 0x0000 0140 is read. Similarly, in case of Bit2 interrupt occurrence, 0x0000 01C0 is read, respectively. In the IRQ00 interrupt routine, user has to program so as to read the address from VIR00 register and branch to its address. In this way, branch operation of interrupt handler can perform quickly.

The read value of VIRxx is described in “VIR value” of Table 3-1,Table 3-2 and Table 3-3. The behavior of VIRxx other than VIR00 is also similar to Table 2-2. The notice of usage of VIRxx is explained below.

CPU must access to VIRxx by word (32bit-width) size. In case of using VIRxx, the priority order of interrupt judge operation is fixed, that is Bit0 > Bit1 > Bit2.

If any interrupt does not occurs (Bit0=Bit1=Bit2=0) , the read value of VIRxx is undefined value. Therefore, after occurring interrupt, CPU should read VIRxx in interrupt handler operation before clearing interrupt signal. VIRxx value read out does not indicate whether the interrupt occurred. After clearing interrupt signal, CPU cannot use VIRxx again in that interrupt handler..

By using VIR_OFFSET register, common offset value can be set to VIRxx. VIRxx exists in Single Cycle I/O area in CM0+, so CPU can access VIRxx by 1cycle.

By reading VIRxx, CPU is not able to identify the interrupt factor that is aggregated by 1st step logical OR. CPU needs to read an interrupt register in each peripheral function for identifying interrupt factor.

2.5 Connection of DSTC transfer request and transfer completion

Certain interrupt signals from peripheral functions can be used as DMA transfer request signals to the DSTC. With the DREQENB[n] register settings of DSTC, the interrupt signals from the peripheral functions are recognized as the DMA transfer requests (DREQ[n] signal to DSTC from peripheral in Figure 2-1).

The interrupt signal input to NVIC is selected by the selector circuit (SEL2 in Figure 2-1). Either of the interrupt signal from peripheral functions or the DMA transfer completion notification signal output from DSTC (HWINT[n] from DSTC to NVIC in Figure 2-1) is selected. Switching of SEL2 is performed according to the DREQENB[n] register setting.

Since the circuit configuration as shown in the figure, when the DMA transfer with DSTC is selected, the transfer completion interrupts from DSTC generates instead of interrupts from peripheral. The transfer completion interrupts from DSTC can be read from the corresponding bit of the IRQxxMON register. In addition, VIR function can be applied.For details of DSTC transfer requests from each peripheral function, refer to the chapter for each peripheral function.

2.6 Connection of DMA Transfer Acknowledge Signal

There are peripheral function blocks for which transfer request signals (interrupt signals) have to be cleared after the DMA transfer to those peripherals has ended. The transfer request signals for such peripheral functions are to be cleared by the DSTC. If the DMA transfer by the DSTC is selected in SEL2, the DMA transfer acknowledge signal (NOT ILLUSTRATED in Figure 2-1) from the DSTC is connected to a peripheral function.

2.7 Connection of DMA Transfer Stop Signal

The DMA transfer stop request signal is output from the multi-function serial unit (to be called MFS later in this document). According to the selection made in SEL2, the MFS (NOT ILLUSTRATED in Figure 2-1) is connected to the DSTC as explained below.

If the connection between the DSTC and the MFS is selected in SEL2, the DSTC stops a transfer operation according to the transfer stop request signal. The DSTC does not execute the transfer operation until the transfer stop request signal from the MFS is negated. The transfer stop request signal from the MFS is aggregated with the transfer end interrupt (HWINT[n] signal) of the DSTC by logical OR, and is notified to the NVIC as an interrupt signal.

3. Lists of Interrupts

This section shows a list of exceptions and interrupt sources input to the NVIC, and a list of interrupts that can be transferred by the DMA transfer by the DSTC.

3.1 Lists of exceptions and interrupts

Table 3-1,Table 3-2 and Table 3-3 show a list of exceptions and interrupt to be input to the NVIC. Below are details of columns in the table.

Exception No.

This shows exception number of the NVIC

IRQ No.

This shows peripheral interrupt number of the NVIC (IRQ No. = Exception No. - 16)

Vector offset

This shows the storage address of the interrupt vector, which is referred to when interrupt occurs. The described value + VTOR(Vector table offset register） in the NVIC is referred to.

Batch read register – Name

This shows the name of Batch Read Register. "-" in this column indicates that there is no Batch Read Register for that exception or interrupt.

Batch read register – bit

This is shown that which bit of Batch Read Register is assign to each exception and interrupt from peripheral function. "-" in this column indicates that there is no Batch Read Register for that exception or interrupt.

VIR value

This shows the read value from VIRxx when interrupt occurs. The described value + VIR_OFFSET is read. "-" in this column indicates that there is no VIRxx

Exception or Interrupt name

This shows the name of exception or interrupt from peripheral functions.

If multiple interrupt source name are described in one bit of Batch Read Register, it shows that those interrupts are aggregated by the 1st step logic OR circuit. Even if only one interrupt source name is described, it may have multiple interrupt sources aggregated by logical OR in peripheral function. For details, refer to the respective peripheral functions.

IRQxx interrupt signal to NVIC is generated from each bit of Batch Read Register by the 2nd step logic OR circuit.

In each product, if peripheral function is not equipped, the interrupt signal of that peripheral function never occur. It has no effect to IRQxxMON, VIRxx and NVIC.

DSTC

This shows the support of DMA transfer by DSTC.

“A number”, as the described value, indicates that DMA transfer by DSTC is supported. Also, the number shows the bit number of the DREQENB[n] register of DSTC. The DREQENB[n] register setting of DSTC determines the connection of SEL2 in the Figure 2-1.

In the case of DREQENB[n] = 0, the interrupt signal from peripheral function is connected to NVIC, and DSTC ignored that interrupt signal.

In the case of DREQENB[n] = 1, the interrupt signal from peripheral function is connected to DSTC for DMA request, and the transfer completion interrupt (HWINT[n]) is connected to NVIC.

"-", as the described value, indicates that it is not supported DMA transfer by DSTC.

Table 3-1 List of exception and interrupt sources (1of 3)

Exception No.	IRQ No.	Vector Offset	Batch read register		VIR value	Exception or Interrupt name	DSTC
			Name	bit			
0	-	0x000	-	-	-	(Stack pointer initial value)	-
1	-	0x004	-	-	-	Reset	-
2	-	0x008	EXC02MON	0	-	Non-maskable interrupt (NMI)	-
				1	-	Hardware watchdog timer interrupt	-
3	-	0x00C	-	-	-	Hard fault	-
4	-	0x010	-	-	-	Reserved	-
5	-	0x014	-	-	-	Reserved	-
6	-	0x018	-	-	-	Reserved	-
7	-	0x01C	-	-	-	Reserved	-
8	-	0x020	-	-	-	Reserved	-
9	-	0x024	-	-	-	Reserved	-
10	-	0x028	-	-	-	Reserved	-
11	-	0x02C	-	-	-	SVCall(supervisor call)	-
12	-	0x030	-	-	-	Reserved	-
13	-	0x034	-	-	-	Reserved	-
14	-	0x038	-	-	-	PendSV	-
15	-	0x03C	-	-	-	SysTick	-
16	0	0x040	IRQ00MON	0	0x0C0	Anomalous frequency detection interrupt by CSV	-
				1	0x140	Software watchdog timer interrupt	-
				2	0x1C0	Low-voltage detection (LVD) interrupt	-
17	1	0x044	IRQ01MON	0	0x0C4	MFS ch.0 reception interrupt	0
				1	0x144	MFS ch.0 transmission interrupt	1
				2	0x1C4	MFS ch.0 status interrupt	-
18	2	0x048	IRQ02MON	0	0x0C8	MFS ch.1 reception interrupt	2
				1	0x148	MFS ch.1 transmission interrupt	3
				2	0x1C8	MFS ch.1 status interrupt	-
19	3	0x04C	IRQ03MON	0	0x0CC	Reserved	-
				1	0x14C	Reserved	-
				2	0x1CC	Reserved	-
20	4	0x050	IRQ04MON	0	0x0D0	MFS ch.3 reception interrupt	6
				1	0x150	MFS ch.3 transmission interrupt	7
				2	0x1D0	MFS ch.3 status interrupt	-
21	5	0x054	IRQ05MON	0	0x0D4	MFS ch.4 reception interrupt	8
				1	0x154	MFS ch.4 transmission interrupt	9
				2	0x1D4	MFS ch.4 status interrupt	-
22	6	0x058	IRQ06MON	0	0x0D8	Reserved	-
				1	0x158	Reserved	-
				2	0x1D8	Reserved	-
23	7	0x05C	IRQ07MON	0	0x0DC	MFS ch.6 reception interrupt	12
						I2CSLAVE reception interrupt	48
				1	0x15C	MFS ch.6 transmission interrupt	13
						I2CSLAVE transmission interrupt	49
				2	0x1DC	MFS ch.6 status interrupt	-
						I2CSLAVE status interrupt	-

Table 3-2 List of exception and interrupt sources (2of 3)

Exception No.	IRQ No.	Vector Offset	Batch read register		VIR value	Exception or Interrupt name	DSTC
			Name	bit			
24	8	0x060	IRQ08MON	0	0x0E0	MFS ch.7 reception interrupt	14
				1	0x160	MFS ch.7 transmission interrupt	15
				2	0x1E0	MFS ch.7 status interrupt	-
25	9	0x064	IRQ09MON	0	0x0E4	A/D converter unit0 priority conversion interrupt	50
				1	0x164	A/D converter unit0 scan conversion interrupt	51
				2	0x1E4	A/D converter unit0 FIFO overrun interrupt	-
						A/D converter unit0 conversion result comparison int.	-
						A/D converter unit0 range comparison result int.	-
26	10	0x068	IRQ10MON	0	0x0E8	USB ch.0 device endpoint1 DRQ interrupt	52
				1	0x168	USB ch.0 device endpoint2 DRQ interrupt	53
				2	0x1E8	USB ch.0 device endpoint3 DRQ interrupt	54
27	11	0x06C	IRQ11MON	0	0x0EC	USB ch.0 device endpoint4 DRQ interrupt	55
				1	0x16C	USB ch.0 device endpoint5 DRQ interrupt	56
				2	0x1EC	USB ch.0 device endpoint0 DRQI interrupt	-
28	12	0x070	IRQ12MON	0	0x0F0	USB ch.0 device endpoint0 DRQO interrupt	-
				1	0x170	USB ch.0 device SUSP interrupt	-
						USB ch.0 device SOF interrupt	-
						USB ch.0 device BRST interrupt	-
						USB ch.0 device CONF interrupt	-
						USB ch.0 device WKUP interrupt	-
				2	0x1F0	USB ch.0 device SPK interrupt	-
29	13	0x074	IRQ13MON	0	0x0F4	USB ch.0 host DIRQ interrupt	-
						USB ch.0 host URIRQ interrupt	-
						USB ch.0 host RWKIRQ interrupt	-
						USB ch.0 host CNNIRQ interrupt	-
				1	0x174	USB ch.0 host SOFIRQ interrupt	-
						USB ch.0 host CMPIRQ interrupt	-
				2	0x1F4	Reserved	-
30	14	0x078	IRQ14MON	0	0x0F8	Main PLL oscillation stabilization wait completion int.	-
						Main clock oscillation stabilization wait completion int.	-
						Sub clock oscillation stabilization wait completion int.	-
				1	0x178	Reserved	-
				2	0x1F8	Reserved	-
31	15	0x07C	IRQ15MON	0	0x0FC	Watch counter interrupt	57
				1	0x17C	Real timer counter (RTC) interrupt	-
						Dual timer ch.1 interrupt	-
						Dual timer ch.2 interrupt	-
				2	0x1FC	Reserved	-

Table 3-3 List of exception and interrupt sources (3of 3)

Exception No.	IRQ No.	Vector Offset	Batch read register		VIR value	Exception or Interrupt name	DSTC
			Name	bit			
32	16	0x080	IRQ16MON	0	0x100	External pin interrupt ch.0	16
				1	0x180	External pin interrupt ch.1	17
33	17	0x084	IRQ17MON	0	0x104	External pin interrupt ch.2	18
				1	0x184	External pin interrupt ch.3	19
34	18	0x088	IRQ18MON	0	0x108	External pin interrupt ch.4	20
				1	0x188	External pin interrupt ch.5	21
35	19	0x08C	IRQ19MON	0	0x10C	External pin interrupt ch.6	22
				1	0x18C	External pin interrupt ch.7	23
36	20	0x090	IRQ20MON	0	0x110	External pin interrupt ch.8	24
				1	0x190	Reserved	-
37	21	0x094	IRQ21MON	0	0x114	Reserved	-
				1	0x194	Reserved	-
38	22	0x098	IRQ22MON	0	0x118	External pin interrupt ch.12	28
				1	0x198	External pin interrupt ch.13	29
39	23	0x09C	IRQ23MON	0	0x11C	Reserved	-
				1	0x19C	External pin interrupt ch.15	31
40	24	0x0A0	IRQ24MON	0	0x120	Base timer ch.0 source0 (IRQ0) interrupt	32
						Base timer ch.0 source1 (IRQ1) interrupt	33
				1	0x1A0	Base timer ch.4 source0 (IRQ0) interrupt	34
						Base timer ch.4 source1 (IRQ1) interrupt	35
41	25	0x0A4	IRQ25MON	0	0x124	Base timer ch.1 source0 (IRQ0) interrupt	36
						Base timer ch.1 source1 (IRQ1) interrupt	37
				1	0x1A4	Base timer ch.5 source0 (IRQ0) interrupt	38
						Base timer ch.5 source1 (IRQ1) interrupt	39
42	26	0x0A8	IRQ26MON	0	0x128	Base timer ch.2 source0 (IRQ0) interrupt	40
						Base timer ch.2 source1 (IRQ1) interrupt	41
				1	0x1A8	Base timer ch.6 source0 (IRQ0) interrupt	42
						Base timer ch.6 source1 (IRQ1) interrupt	43
43	27	0x0AC	IRQ27MON	0	0x12C	Base timer ch.3 source0 (IRQ0) interrupt	44
						Base timer ch.3 source1 (IRQ1) interrupt	45
				1	0x1AC	Base timer ch.7 source0 (IRQ0) interrupt	46
						Base timer ch.7 source1 (IRQ1) interrupt	47
44	28	0x0B0	IRQ28MON	0	0x130	CEC Reception/Remote reception ch.0 interrupt	-
						CEC Transmission ch.0 interrupt	-
				1	0x1B0	CEC Reception/Remote reception ch.1 interrupt	-
						CEC Transmission ch.1 interrupt	-
45	29	0x0B4	IRQ29MON	0	0x134	Smart Card ch.1 interrupt	-
				1	0x1B4	FLASH memory RDY/HANG interrupt	-
46	30	0x0B8	IRQ30MON	0	0x138	DSTC SW transfer complete interrupt	-
				1	0x1B8	DSTC error interrupt	-
47	31	0x0BC	IRQ31MON	0	0x13C	Reserved	-
				1	0x1BC	Reserved	-

3.2 Lists of transfer request signals Input to DSTC

Table 3-4 show interrupt signals input as transfer request signals to the DSTC. Numbers in the table correspond to the numbers of the DREQENB[n] Registers of the DSTC.

Table 3-4 List of interrupt signals Input to the DSTC

No.	Source of DSTC transfer request
0	MFS ch.0 reception interrupt
1	MFS ch.0 transmission interrupt
2	MFS ch.1 reception interrupt
3	MFS ch.1 transmission interrupt
4	Reserved
5	Reserved
6	MFS ch.3 reception interrupt
7	MFS ch.3 transmission interrupt
8	MFS ch.4 reception interrupt
9	MFS ch.4 transmission interrupt
10	Reserved
11	Reserved
12	MFS ch.6 reception interrupt
13	MFS ch.6 transmission interrupt
14	MFS ch.7 reception interrupt
15	MFS ch.7 transmission interrupt
16	External pin interrupt ch.0
17	External pin interrupt ch.1
18	External pin interrupt ch.2
19	External pin interrupt ch.3
20	External pin interrupt ch.4
21	External pin interrupt ch.5
22	External pin interrupt ch.6
23	External pin interrupt ch.7
24	External pin interrupt ch.8
25	Reserved
26	Reserved
27	Reserved
28	External pin interrupt ch.12
29	External pin interrupt ch.13
30	Reserved
31	External pin interrupt ch.15

No.	Source of DSTC transfer request
32	Base timer ch.0 source0 (IRQ0) interrupt
33	Base timer ch.0 source1 (IRQ1) interrupt
34	Base timer ch.4 source0 (IRQ0) interrupt
35	Base timer ch.4 source1 (IRQ1) interrupt
36	Base timer ch.1 source0 (IRQ0) interrupt
37	Base timer ch.1 source1 (IRQ1) interrupt
38	Base timer ch.5 source0 (IRQ0) interrupt
39	Base timer ch.5 source1 (IRQ1) interrupt
40	Base timer ch.2 source0 (IRQ0) interrupt
41	Base timer ch.2 source1 (IRQ1) interrupt
42	Base timer ch.6 source0 (IRQ0) interrupt
43	Base timer ch.6 source1 (IRQ1) interrupt
44	Base timer ch.3 source0 (IRQ0) interrupt
45	Base timer ch.3 source1 (IRQ1) interrupt
46	Base timer ch.7 source0 (IRQ0) interrupt
47	Base timer ch.7 source1 (IRQ1) interrupt
48	I2CSLAVE reception interrupt
49	I2CSLAVE transmission interrupt
50	A/D converter unit0 priority conversion interrupt
51	A/D converter unit0 scan conversion interrupt
52	USB ch.0 device endpoint1 DRQ interrupt
53	USB ch.0 device endpoint2 DRQ interrupt
54	USB ch.0 device endpoint3 DRQ interrupt
55	USB ch.0 device endpoint4 DRQ interrupt
56	USB ch.0 device endpoint5 DRQ interrupt
57	Watch counter interrupt
58	Reserved
59	Reserved
60	Reserved
61	Reserved
62	Reserved
63	Reserved

4. Registers

This section explains the respective details of registers in this block.

4.1 Lists of registers

Table 4-1 shows list of registers in this block.

Table 4-1 Register list

Abbreviation	Register name	Reference
EXC02MON	EXC02 Batch Read Register	4.2
IRQxxMON (xx is 00 to 31)	IRQxx Batch Read Register (xx is 00 to 31)	4.3
VIR_OFFSET	VIR OFFSET Register	4.4
VIRxx (xx is 00 to 31)	Vector Indicate Register xx (xx is 00 to 31)	4.5
ODDPKS	USB ch.0 Odd Packet Size DMA Enable Register	4.6

4.2 EXC02 Batch Read Register (EXC02MON)

The EXC02 Batch Read Register (EXC02MON) can read out at once all interrupts (NMI and hardware watchdog interrupt) assigned to exception no. 2.

Register configuration

bit	31							8
Field	Reserved							
Attribute	R							
Initial value	0x000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved						HWINT	NMI
Attribute	R						R	R
Initial value	000000						0	0

Register function

[bit31:2] Reserved: Reserved bits

A reserved bit reads 0.

[bit1] HWINT

Operation	Description
Write	Write access is Ignored.
Read 0	The hardware watchdog timer interrupt request occurs
Read 1	The hardware watchdog timer interrupt request does not occur

[bit0] NMI

Operation	Description
Write	Write access is Ignored.
Read 0	The interrupt request of the NMIX external pin occurs.
Read 1	The interrupt request of the NMIX external pin does not occur

4.3 IRQxx Batch Read Register (IRQxxMON)

The IRQxx Batch Read Register indicates 32 registers that are IRQ00MON – IRQ31MON. These are corresponding with IRQ00 – IRQ31 interrupt inputs of the NVIC respectively. The status of interrupt signals that are aggregated by the 2nd step logic OR circuit can be read out by these registers.

See Table 3-1,Table 3-2 and Table 3-3 to check which bit in this register is assigned to which interrupt from peripheral function

Register configuration

bit	31	3	2	1	0
Field	Reserved		INT2	INT1	INT0
Attribute	R		R	R	R
Initial value	00000000000000000000000000000		0	0	0

Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads 0.

[bit2] INT2

Operation	Description
Write	Write access is Ignored.
Read 0	The interrupt assigned to Bit2 of this register does not occur.
Read 1	The interrupt assigned to Bit2 of this register occurs.

[bit1] INT1

Operation	Description
Write	Write access is Ignored.
Read 0	The interrupt assigned to Bit1 of this register does not occur.
Read 1	The interrupt assigned to Bit1 of this register occurs.

[bit0] INT0

Operation	Description
Write	Write access is Ignored.
Read 0	The interrupt assigned to Bit0 of this register does not occur.
Read 1	The interrupt assigned to Bit0 of this register occurs.

By only reading Interrupt Batch Read Registers, the interrupt source aggregated by 1st step logical OR cannot be identified. CPU needs to read an interrupt register in each peripheral function for identifying interrupt source.

4.4 VIR Offset Register (VIR_OFFSET)

VIR_OFFSET register is used to define a common offset value for VRxx. When interrupt occurs, the read value of VIRxx is a summation of this register value and VIR value in Table 3-1,Table 3-2 and Table 3-3.

Register configuration

bit	31	0
Field	VIR_OFFSET	
Attribute	R/W	
Initial value	0x000000	

Register function

[bit31:0] VIR_OFFSET

Operation	Description
Write	Define the offset value of VIRxx.
Read	Read out register value.

Usually, this register should be set same value as VTOR(Vector table offset register） in the NVIC.

Bit[7:0] of this register must always be set 0x00. If you need to change this register value, it should be done in system initialization. This register must not change in interrupt operation.

4.5 Vector Indicate Register xx (VIRxx)

When interrupt occurs, CPU can use Vector Indicate Register to quickly branch the interrupt operation. VIRxx consists of 32 registers (VIR00 – VIR32) that are corresponding with IRQ00 –IRQ31 input of the NVIC. When interrupt occurs, CPU is able to read out a routine address value from VIRxx. This address value is decided depending on the status of interrupt signals that is aggregated by the 2nd step logic OR circuit. By placing top address of the interrupt handler in this address area, the CPU is able to branch the interrupt operation quickly.

Register configuration

bit	31	0
Field	VIRxx	
Attribute	R	
Initial value	xxxx xxxx xxxx xxxx	

Register function

[bit31:0] VIRxx

Operation	Description
Write	Write access is ignored
Read	Read out address value that is corresponding with status of interrupt signal that is aggregated by the 2nd step logic OR circuit

See “2.4 Vector Indicate Register” for usage of this register. The read value of VIRxx is described in “VIR value” of Table 3-1,Table 3-2 and Table 3-3. Behavior example of VIR00 is described in Table 2-2.

CPU must access to VIRxx by word (32bit-width) size. In case of using VIRxx, the priority order of interrupt judge operation is fixed, that is Bit0 > Bit1 > Bit2.

If any interrupt does not occurs (Bit0=Bit1=Bit2=0), the read value of VIRxx is undefined value. Therefore, after occurring interrupt, CPU should read VIRxx in interrupt handler operation before clearing interrupt signal. VIRxx value read out does not indicate whether the interrupt occurs or not. After clearing interrupt signal, CPU cannot use VIRxx again in that interrupt handler..

By using VIR_OFFSET register, common offset value can be set to VIRxx. VIRxx exists in Single Cycle I/O area in CM0+, so CPU can access VIRxx by 1cycle.

By reading VIRxx , CPU is not able to identify the interrupt source that is aggregated by 1st step logical OR. CPU needs to read an interrupt register in each peripheral function for identifying interrupt factor.

Notes:

- *Pay attention to the following points when using VIRx registers.*
  After occurring the interrupts, there are some interrupts, such as MFS TX or RX interrupt, that these interrupt signals will be automatically cleared by status changes of FIFOs, without CPU clearing the signals. In the case of these interrupt signals, read value of the IRQMON register in the interrupt handler, check that the interrupt is occurred and then read the VIRx registers to branch the process.

4.6 USB ch.0 Odd Packet Size DMA Enable Register (ODDPKS)

If data is transferred in the IN direction in USB ch.0 automatic transfer in which the DSTC is used, only in the last data in the last packet, the effective bit width is compulsorily converted into 1 byte (8 bits) before the data is written to a USB endpoint.

Register configuration

bit	31	5	4	3	2	1	0
Field	Reserved		ODDPKS				
Attribute	R		R/W				
Initial value	All 0		00000				

Register function

[bit31:5] Reserved: Reserved bits

Write 0 to a reserved bit. A reserved bit reads 0.

[bit4] ODDPKS4

Operation	Description
Write 0	There is no conversion of the size for DMA transfer by the DSTC.
Write 1	When transfer destination address in the DSTC is USB.EP5DT, the last transfer size is converted into one byte.
Read	Read out the register value.

[bit3] ODDPKS3

Operation	Description
Write 0	There is no conversion of the size for DMA transfer by the DSTC.
Write 1	When transfer destination address in the DSTC is USB.EP4DT, the last transfer size is converted into one byte.
Read	Read out the register value.

[bit2] ODDPKS2

Operation	Description
Write 0	There is no conversion of the size for DMA transfer by the DSTC.
Write 1	When transfer destination address in the DSTC is USB.EP3DT, the last transfer size is converted into one byte.
Read	Read out the register value.

[bit1] ODDPKS1

Operation	Description
Write 0	There is no conversion of the size for DMA transfer by the DSTC.
Write 1	When transfer destination address in the DSTC is USB.EP2DT, the last transfer size is converted into one byte.
Read	Read out the register value.

[bit0] ODDPKS0

Operation	Description
Write 0	There is no conversion of the size for DMA transfer by the DSTC.
Write 1	When transfer destination address in the DSTC is USB.EP1DT, the last transfer size is converted into one byte.
Read	Read out the register value.

Notes:

- *This register is valid only when on USB ch.0 data is transferred in the IN direction in USB data size automatic transfer mode*

- *When transferring a packet whose number of bytes is an even number, do not write 1 to any of the ODDPKS4, ODDPKS3, ODDPKS2, ODDPKS1 and ODDPKS0 bits.*

5. Usage Precautions

Note the following when using the interrupt controller.

- The interrupt request signals from peripheral functions are notified to the interrupt controller by level sense method. When exiting the processing of an interrupt, always clear the interrupt request signals from peripheral functions.

- When other interruption with a high priority is processing, the interrupt request from peripheral functions notified to NVIC is pended inside NVIC. When canceling the interrupt request which was pended inside NVIC, clear the interrupt request from peripheral functions, and clear the interrupt request which was pended inside NVIC by the Interrupt Clear-Pending Registers (addresses: 0xE000E280) installed in the NVIC.

- The NMIX pin is shared with a general-purpose port. After a reset is released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the port setting. For details, see Chapter “I/O port” and “External Interrupt and NMI Controller”.

- If the DMA transfer by the DSTC is used, the transfer end interrupt (HWINT[n]) from the DSTC is generated instead of the interrupt from a peripheral function. Due to the above configuration, the NVIC makes an interrupt from a peripheral function, and a transfer end interrupt from the DSTC jump to the same interrupt vector. Use the DREQENB[n] Register to select the interrupt to be processed.

- For the relationship between specific event detection registers and interrupt enable registers in a peripheral function, see the chapter on that peripheral function.

CHAPTER 8: External Interrupt and NMI Control Sections



This chapter explains the functions and operations of the external interrupt and NMI control sections.

1. Overview
2. Block Diagram
3. Operations and Setting Procedure Examples
4. Registers

CODE: 9BFEXTINT-FM0-E03.0_FW12-E1.04

1. Overview

The external interrupt and NMI control sections have the following features.

- Has up to 32 external interrupt input pins and one NMI input pin mounted.
- Possible to select the detection condition of an external interrupt from the following five types.
  - High level
  - Low level
  - Rising edge
  - Falling edge
  - Both rising edge and falling edges (Only in TYPE3-M0+ products).
- Possible to use an external interrupt input or NMI input to return from standby mode.
- Possible to enable/disable NMI interrupt in TYPE3-M0+ products.

2. Block Diagram

The following shows the block diagram of the external interrupt and NMI control sections.

Figure 2-1 Block Diagram of External Interrupt and NMI Control Sections (For TYPE1/2-M0+)

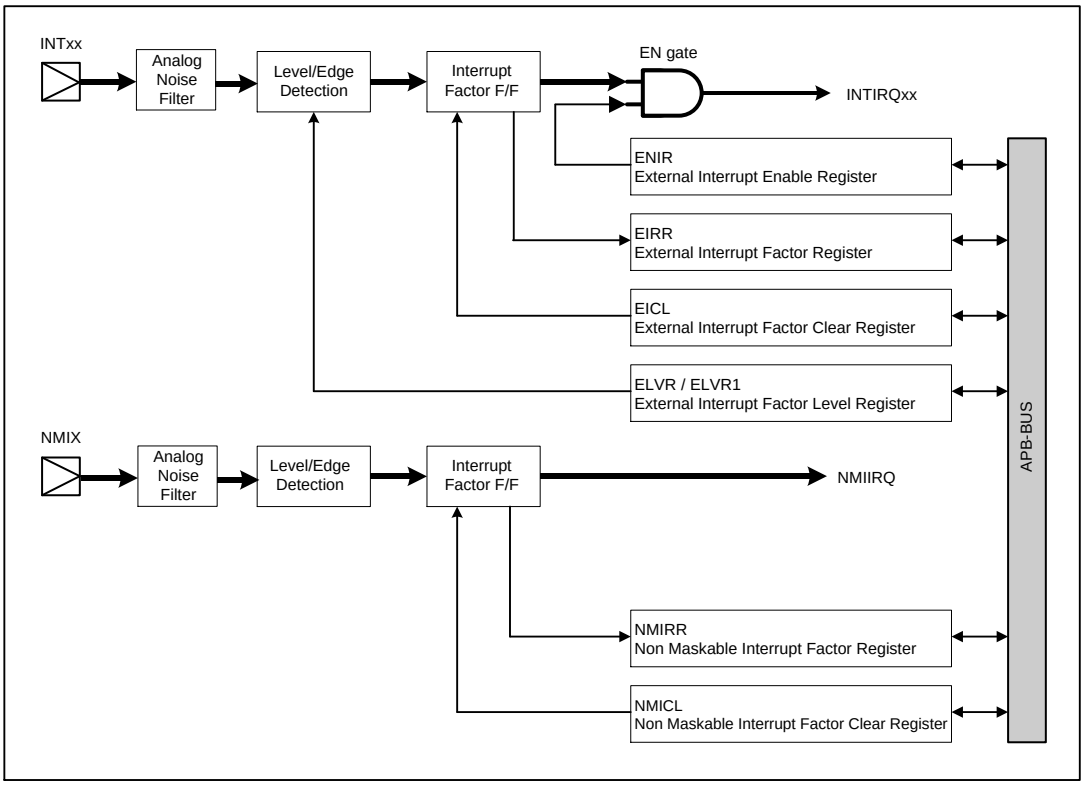
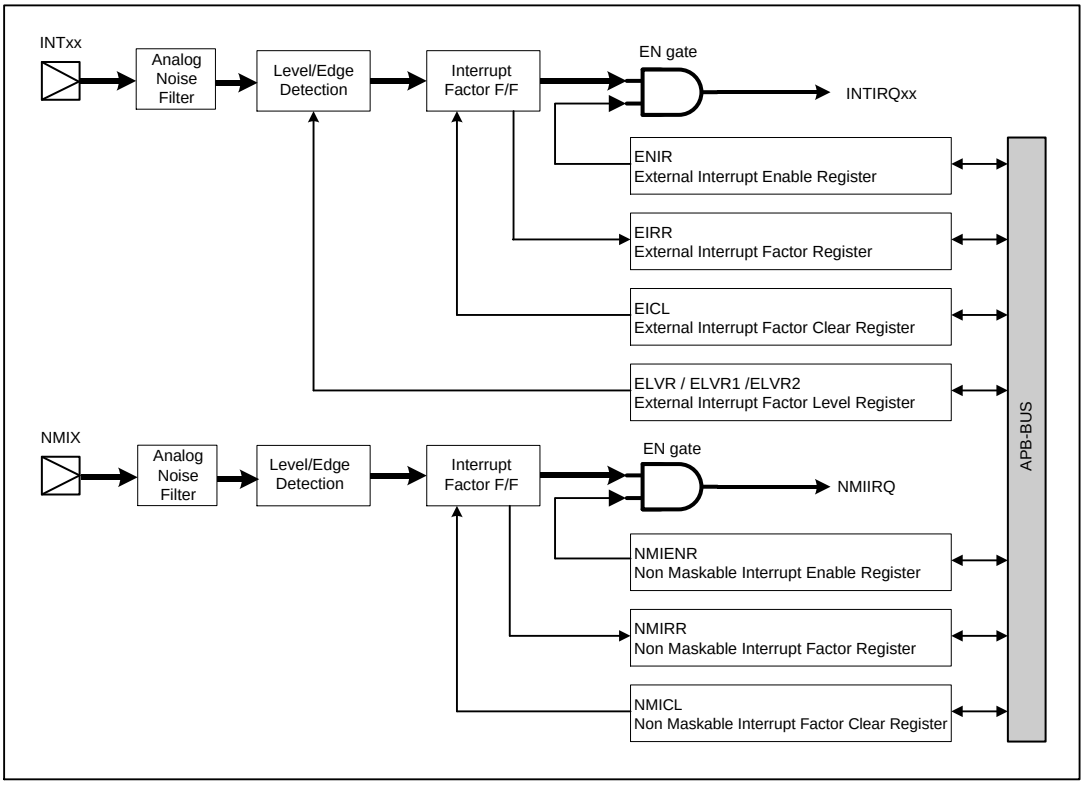


Figure 2-2 Block Diagram of External Interrupt and NMI Control Sections (For TYPE3-M0+)



3. Operations and Setting Procedure Examples

This section explains operations and setting procedure examples.

3.1. Operations of External Interrupt Control Section
3.2. Operations of NMI Control Section
3.3. Returning from the Timer, Stop, RTC Mode

3.1 Operations of External Interrupt Control Section

This section shows the operations of the external interrupt control section.

Overview of Operations in External Interrupt Control Section

The external interrupt control section outputs an external interrupt request to the interrupt controller in the following procedure.

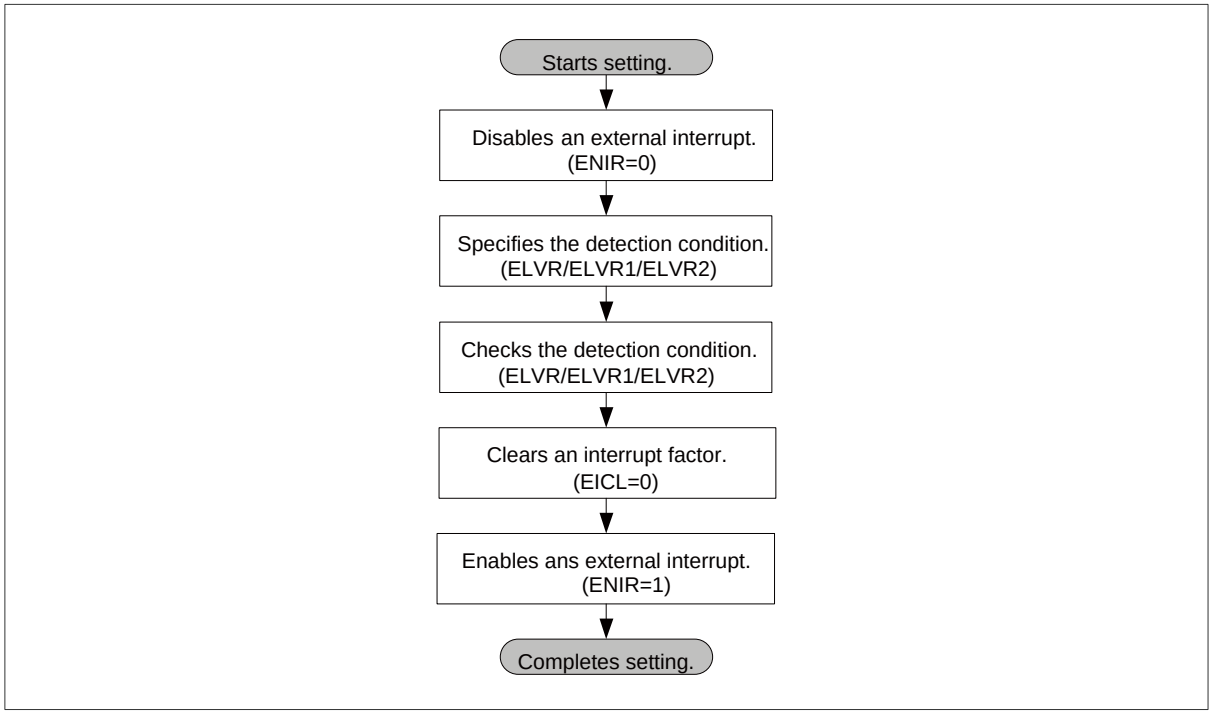
1. The signal input to pin INTxx detects the edge or level specified in the External Interrupt Level Register (ELVR/ELVR1/ELVR2). The edge or level to be detected can be selected from the following five types: High level, Low level, Rising edge, Falling edge, Both rising and falling edge
   (Both rising and falling edge is supported in the TYPE3-M0+ product only.)
2. The detected interrupt input is held in the interrupt factor F/F.
   It is read with the External Interrupt Factor Register (EIRR).
   3. If an external interrupt is enabled with the External Interrupt Enable Register (ENIR), an external interrupt request (INTIRQxx) to the interrupt controller is asserted.
4. The held interrupt factor is cleared with the External Interrupt Factor Clear Register (EICL), an external interrupt request (INTIRQxx) to the interrupt controller is negated. .

Initial Setting Procedure

Execute the following steps to configure external interrupt setting.

1. Disable an external interrupt with the External Interrupt Enable Register (ENIR=0).
2. Specify the detection condition (effective edge or level) with the External Interrupt Factor Level Register (ELVR/ELVR1/ELVR2).
3. Read the External Interrupt Factor Level Register (Any one of ELVR/ELVR1/ELVR2).
4. Clear the external interrupt factor with the External Interrupt Factor Clear Register (EICL=0).
5. Enable the external interrupt with the External Interrupt Enable Register (ENIR=1).

Figure 3-1 External Interrupt Setting Procedure

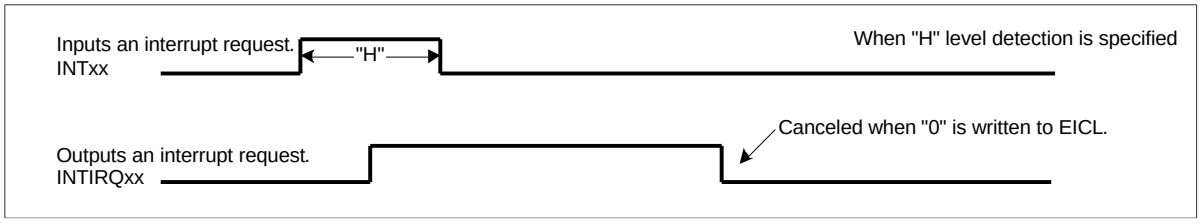


Set and Clear an External Interrupt Request

When the external interrupt is detected, an interrupt factor is held in the External Interrupt Factor Register (EIRR). Figure 3-2 shows waveform in case of High level detection condition. When the external interrupt detection condition is set to the H or L level, an interrupt factor is held in the External Interrupt Factor Register (EIRR) even if an external interrupt request input (INTxx) is negated. Therefore, an external interrupt request (INTIRQxx) to the interrupt controller remains to be asserted. After receiving the interrupt, the CPU should clear the External Interrupt Factor Register (EIRR) with the External Interrupt Factor Clear Register (EICL=0)

When level detection condition is selected and during period of INTxx pin input is asserted, even I f the CPU clears the External Interrupt Factor Register (EIRR) with the External Interrupt Factor Clear register (EICL), this bit will set 1 again.

Figure 3-2 Clearing an Interrupt Factor

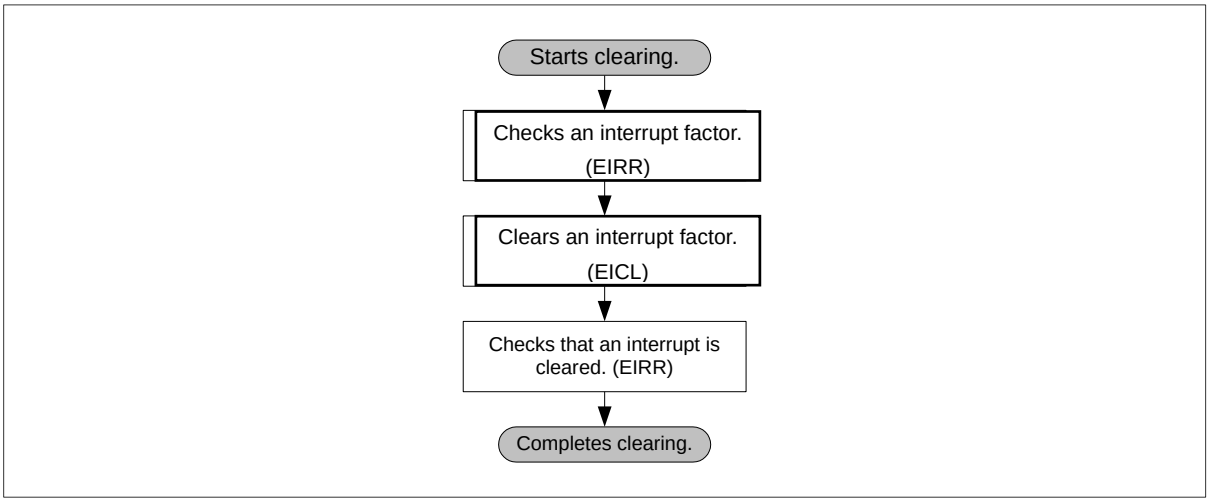


Procedure of Clearing External Interrupt factor

Execute the following steps to cancel an external interrupt request.

1. Read the External Interrupt Factor Register (EIRR), and check the interrupt factor.
2. Write "0" to the corresponding bit in the External Interrupt Factor Clear Register (EICL) to clear it.
3. Read the External Interrupt Factor Register (EIRR), and check that the interrupt factor is cleared.

Figure 3-3 Clearing an External Interrupt Request



Note:

- *When other interruption with a high priority is processing, the interrupt request from peripheral functions notified to NVIC is pended inside NVIC. When canceling the interrupt request which was pended inside NVIC, clear the interrupt request from peripheral functions, and clear the*

interrupt request which was pended inside NVIC by the Interrupt Clear-Pending Registers (addresses: 0xE000E280) installed in the NVIC.

3.2 Operations of NMI Control Section

This section shows the operations of the NMI control section.

Overview of NMI Control Section

The NMI control section outputs an NMI interrupt request (NMIIRQ) to the CPU if the edge or level is detected from the signal input to the NMI input pin (NMIX).
The following edge or level is detected.（This condition is not able to be changed.）

- Run mode: Falling edge
- Sleep mode: Falling edge
- Timer mode: L level
- RTC mode: L level
- Stop mode: L level
- Deep standby mode: NMI request is not available in this mode.

Note:

- *NMI request is not available for to return from Deep standby mode. However, NMIX input pin and WKUP input pin is shared to same input pin. Therefore, it is available for to return by WKUP input pin. For details, see 5.Operations in Deep Standby Modes in CHAPTER 6: Low Power Consumption Mode.*

NMIENR（NMI Enable Register）

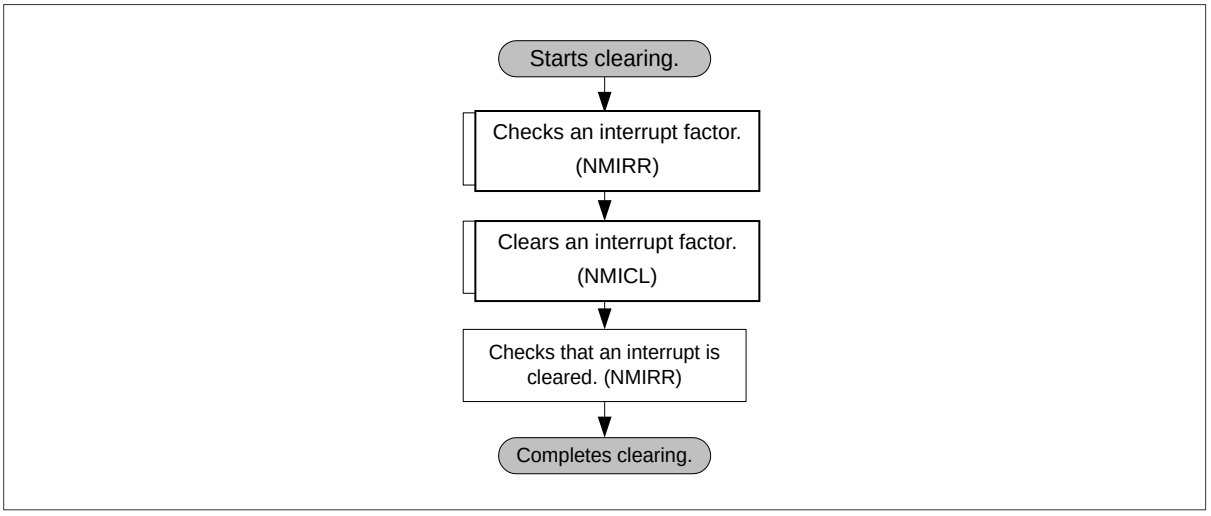
In case of the TYPE3-M0+ product, NMI request can be enabled or disabled with setting of the NMIENR (NMI Enable Register).

Canceling an NMI Request

To cancel an NMI request, clear the request register in the same way as for an external interrupt request. Execute the following steps to cancel an NMI interrupt request.

1. Read the NMI Factor Register (NMIRR), and check the interrupt factor.
2. Write 0 to the corresponding bit in the NMI Factor Clear Register (NMICL) to clear it.
3. Read the NMI Factor Register (NMIRR), and check that the interrupt factor is cleared.

Figure 3-4 Canceling an NMI Request



3.3 Returning from the Timer, Stop, RTC Mode

This section shows a return from the Timer, Stop, RTC mode.

Overview

An external interrupt and NMI requests can be used to return from the Timer, Stop, RTC mode.
In these modes, the signal first input to pin INTxx or NMIX is input asynchronously, and the device can return from these modes to RUN mode.

Setting before Changing to the Timer, Stop, RTC Mode

To use an external interrupt request, specify the pin used to return from these modes and also specify the effective detection level before changing to these modes.

- Pin used to return from these modes.: Interrupt request output enable (ENIR = 1)
- Pin not used to return from these modes. Interrupt request output disable (ENIR = 0)

To use an NMI request, only the L level is detected, and cannot select the detection condition. NMIENR register should be enabled in the TYPE3-M0+ product.

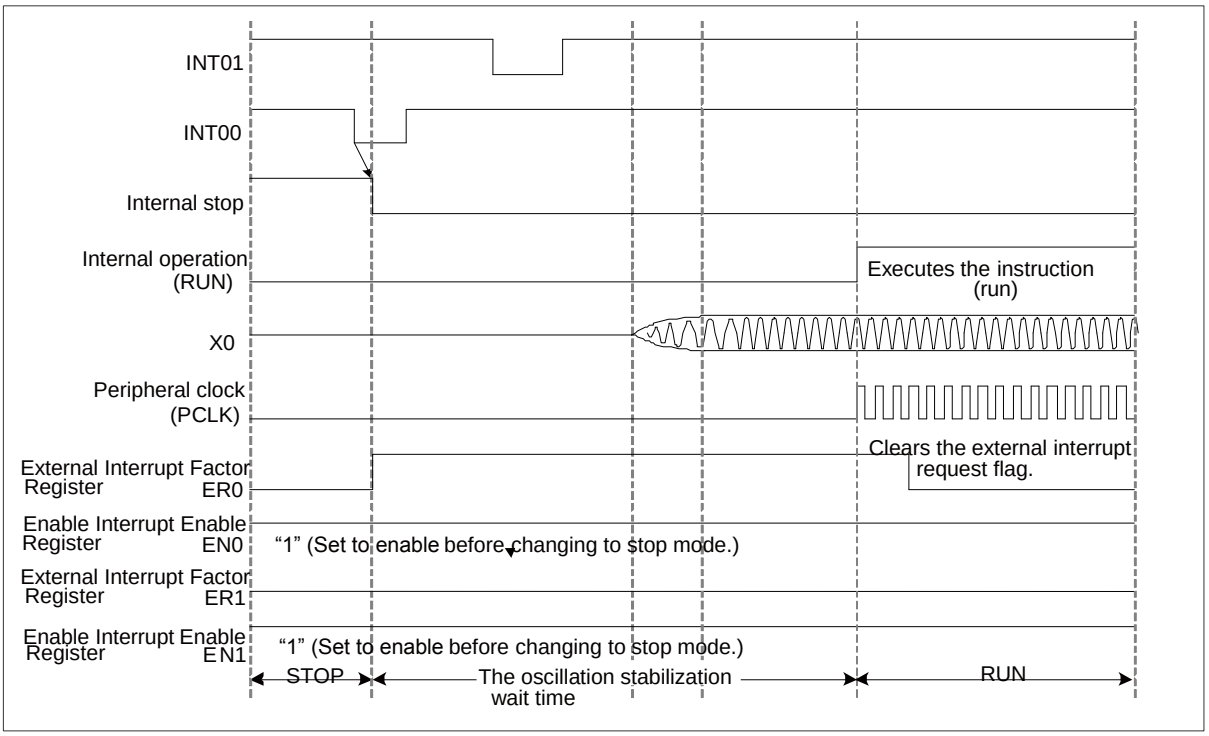
Returning from the Timer, Stop, RTC Mode

For external interrupt request, if the pre-specified effective level is detected in the pin used to return from these modes, the device returns from these modes.
For NMI request, if the L level is detected in stop mode, the device returns from these modes.

Notes on Returning from the Timer, Stop, RTC Mode

Any other external interrupt requests cannot be recognized until the oscillation stabilization wait time lapses after these modes were released.
(For INT01 in Figure 3-5, any external interrupt requests cannot be recognized.)
Therefore, to input an external interrupt after these modes were released, input an external interrupt signal after the oscillation stabilization wait time lapsed.

Figure 3-5 Returning from the Timer, Stop, RTC Mode



4. Registers

This section provides a list of registers.

Register List

The following shows a list of registers in the external interrupt and NMI control sections.

Table 4-1 Registers in External Interrupt and NMI Control Sections

Abbreviation	Register name	Reference
ENIR	External Interrupt Enable Register	4.1
EIRR	External Interrupt Factor Register	4.2
EICL	External Interrupt Factor Clear Register	4.3
ELVR	External Interrupt Factor Level Register	4.4
ELVR1	External Interrupt Factor Level Register 1	4.5
NMIRR	Non Maskable Interrupt Factor Register	4.6
NMICL	Non Maskable Interrupt Factor Clear Register	4.7
ELVR2	External Interrupt Factor Level Register 2	4.8
NMIENR	Non Maskable Interrupt Enable Register	4.9

4.1 External Interrupt Enable Register (ENIR)

The ENIR register is used to control masking an external interrupt request output.

Register configuration

bit	31															16
Field	EN[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15															0
Field	EN[15:0]															
Attribute	R/W															
Initial value	0x0000															

Register functions

[bit31:0] EN31 to EN0: External interrupt enable bits

EN31 to EN0 bits correspond to pins INT31 to INT00.

It should be set 0 to the bit corresponding to a pin that is not defined in the product specifications.

bit	Description
Writing 0	Disables the output of an external interrupt request of INTx pin corresponding to the relevant bit.
Writing 1	Enables the output of an external interrupt request of INTx pin corresponding to the relevant bit.
Reading	Read out register value.

This function enables the interrupt request output corresponding to the bit that is set to 1 in this register, and outputs a request to the interrupt controller. The pin corresponding to the bit that is set to 0 holds an interrupt factor, but outputs no request to the interrupt controller.

4.2 External Interrupt Factor Register (EIRR)

The EIRR register indicates that an external interrupt request is detected.

Register configuration

bit	31															16
Field	ER[31:16]															
Attribute	R															
Initial value	0xXXXX															

bit	15															0
Field	ER[15:0]															
Attribute	R															
Initial value	0xXXXX															

Register functions

[bit31:0] ER31 to ER0: External interrupt request detection bits

ER31 to ER0 bits correspond to pins INT31 to INT00.

The bit corresponding to a pin that is not defined in the product specifications is indefinite.

bit	Description
Reading 0	Detects no external interrupt request of INTx pin corresponding to the relevant bit.
Reading 1	Detects an external interrupt request of INTx pin corresponding to the relevant bit.
Writing	No effect on operation

Notes:

- *When level detection condition is selected and during period of INTxx pin input is asserted, even I f the CPU clears the External Interrupt Factor Register (EIRR) with the External Interrupt Factor Clear register (EICL), this bit will set 1 again.*
- *When Initialization, the bit in the External Interrupt Factor Register (EIRR) may be set to 1. After set the GPIO to external interrupt pin and set ELVR/ELVR1/ELVR2, clear the External Interrupt Factor Register (EIRR) .*

4.3 External Interrupt Factor Clear Register (EICL)

The EICL register is used to clear the held interrupt factor.

Register configuration

bit	31															16
Field	ECL[31:16]															
Attribute	W															
Initial value	0xFFFF															

bit	15															0
Field	ECL[15:0]															
Attribute	W															
Initial value	0xFFFF															

Register functions

[bit31:0] ECL31 to ECL0: External interrupt factor clear bits

ECL31 to ECL0 bits correspond to pins INT31 to INT00.

It is not possible to write 0 to the bit corresponding to a pin that is not defined in the product specifications.

bit	Description
Writing 0	Clears an external interrupt factor of INTx pin corresponding to the relevant bit.
Writing 1	No effect on operation
Reading	Always reads 1.

4.4 External Interrupt Factor Level Register (ELVR)

The ELVR is used to select the level or edge of the signal detected as an external interrupt request.

Register configuration

bit	31															16
Field	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8
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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15															0
Field	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register functions

[bit31:0] LA15 to LA0 or LB15 to LB0: External interrupt request detection level selection bits

LA15 to LA0 or LB15 to LB0 bits correspond to pins INT15 to INT00 on a 2-bit (LA and LB) basis.
It should be set 00 to the bit corresponding to a pin that is not defined in the product specifications.
If the edge or level selected with this bit is detected, it is recognized as an external interrupt request.

	LBx	LAx	Description
Writing	0	0	Detects the Low level.
	0	1	Detects the High level.
	1	0	Detects the rising edge.
	1	1	Detects the falling edge.
Reading	-	-	Reads out register values

Notes:

- *If ELVR/ELVR1/ELVR2 is rewritten to change the detection condition, an invalid interrupt factor may occur. To avoid an invalid interrupt factor from occurring, keep the procedure shown in Figure 3-1 when changing the detection condition.*
- *To detect the edge or level specified in ELVR/ELVR1/ELVR2, at least 3T (T: PCLK cycle) is required as the pulse width. If a signal that does not satisfy the pulse width is input, it is not guaranteed that correct operations will be carried out.*

4.5 External Interrupt Factor Level Register 1 (ELVR1)

The ELVR1 is used to select the level or edge of the signal detected as an external interrupt request.

Register configuration

bit	31															16
Field	LB31	LA31	LB30	LA30	LB29	LA29	LB28	LA28	LB27	LA27	LB26	LA26	LB25	LA25	LB24	LA24
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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15															0
Field	LB23	LA23	LB22	LA22	LB21	LA21	LB20	LA20	LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16
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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register functions

[bit31:0] LA31 to LA16 or LB31 to LB16: External interrupt request detection level selection bits

LA31 to LA16 or LB31 to LB16 bits correspond to pins INT31 to INT16 on a 2-bit (LA and LB) basis.
It should be set 00 to the bit corresponding to a pin that is not defined in the product specifications.
If the edge or level selected with this bit is detected, it is recognized as an external interrupt request.

	LBx	LAx	Description
Writing	0	0	Detects the Low level.
	0	1	Detects the High level.
	1	0	Detects the rising edge.
	1	1	Detects the falling edge.
Reading	-	-	Reads out register values

Notes:

- *If ELVR/ELVR1/ELVR2 is rewritten to change the detection condition, an invalid interrupt factor may occur. To avoid an invalid interrupt factor from occurring, keep the procedure shown in Figure 3-1 when changing the detection condition.*
- *To detect the edge or level specified in ELVR/ELVR1/ELVR2, at least 3T (T: PCLK cycle) is required as the pulse width. If a signal that does not satisfy the pulse width is input, it is not guaranteed that correct operations will be carried out.*

4.6 Non Maskable Interrupt Factor Register (NMIRR)

The NMIRR Register indicates that a non maskable interrupt (NMI) request is detected.

Register configuration

bit	15	1	0
Field	Reserved		NR
Attribute	-		R
Initial value	-		0

Register functions

[bit15:1] Reserved: Reserved bits

The read value is undefined.

They have no effect in write mode.

[bit0] NR: NMI interrupt request detection bit

The NR bit corresponds to NMIX pin.

bit	Description
Reading 0	Detects no NMI interrupt request.
Reading 1	Detects an NMI interrupt request.
Writing	No effect on operation

Note:

In case of the TYPE1-M0+, TYPE2-M0+ products.

- *When the I/O port which is mapped to NMI input pin is changed to GPIO or other peripheral function from NMI (write EPFR00.NMIS = 1), input level of the I/O port should be held high level, and change the I/O port. Internal NMI signal is tied to high level in case of the I/O port is selected to GPIO or other peripheral function. Therefore, when input level of the I/O port is low, to change the I/O port from GPIO or other peripheral function to NMI, is caused to change of internal NMI signal high to low. So, falling edge will be detected, NMI request occurred.*

In case of the TYPE3-M0+ product.

- *When the I/O port which is mapped to NMI input pin is changed to GPIO or other peripheral function from NMI (write EPFR00.NMIS = 1), It should be changed in the state that is disabled the NMI with the Non Maskable Interrupt Enable Register (NMIENR=0). After changing GPIO, clear NMIRR with NMICL. And enable the NMI with (NMIENR=1).*

4.7 Non Maskable Interrupt Factor Clear Register (NMICL)

The NMICL register is used to clear the held interrupt factor.

Register configuration

bit	15	1	0
Field	Reserved		NCL
Attribute	-		R/W
Initial value	-		1

Register functions

[bit15:1] Reserved: Reserved bits

The read value is undefined.

They have no effect in write mode.

[bit0] NCL: NMI interrupt factor clear bit

The NCL bit corresponds to NMIX pin.

bit	Description
Writing 0	Clears an NMI interrupt factor.
Writing 1	No effect on operation
Reading	Always reads 1.

4.8 External Interrupt Factor Level Register 2 (ELVR2)

The ELVR2 is used to select the both rising and falling edges of the signal detected as an external interrupt request.This register is equipped in TYPE3-M0+ products only. In TYPE1-M0+, TYPE2-M0+ products, this detection condition is not supported.

Register configuration

bit	31															16
Field	LC31	LC30	LC29	LC28	LC27	LC26	LC25	LC24	LC23	LC22	LC21	LC20	LC19	LC18	LC17	LC16
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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15															0
Field	LC15	LC14	LC13	LC12	LC11	LC10	LC9	LC8	LC7	LC6	LC5	LC4	LC3	LC2	LC1	LC0
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
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register functions

[bit31:0] LC31 to LC0: External interrupt request detection level selection bits

LC31 to LC0 bits correspond to pins INT31 to INT0.

It should be set 0 the bit corresponding to a pin that is not defined in the product specifications.

If either rising edge or falling edges are detected, it is recognized as an external interrupt request.

	LCx	Description
Writing	0	Detects the edge or level selected with ELVR and ELVR1.
	1	Detects the both rising and falling edges. Setting of ELVR/ELVR1 is ignored.
Reading	-	Reads out register value.

Notes:

- *If ELVR/ELVR1/ELVR2 is rewritten to change the detection condition, an invalid interrupt factor may occur. To avoid an invalid interrupt factor from occurring, keep the procedure shown in Figure 3-1 when changing the detection condition.*
- *To detect the edge or level specified in ELVR/ELVR1/ELVR2, at least 3T (T: PCLK cycle) is required as the pulse width. If a signal that does not satisfy the pulse width is input, it is not guaranteed that correct operations will be carried out.*

4.9 Non Maskable Interrupt Enable Register (NMIENR)

The NMIENR register is used to enable or disable NMI interrupt. This register is equipped in TYPE3-M0+ products only. In TYPE1-M0+, TYPE2-M0+ products, this function is not supported.

Register configuration

bit	7	1	0
Field	Reserved		NE0
Attribute	-		R/W
Initial value	-		0

Register functions

[bit7:1] Reserved: Reserved bits

The read value is undefined.
They have no effect in write mode.

[bit0] NE0: NMI enable bit

This bit enables or disables NMI interrupt.

bit	Function
Writing 0	Disables an NMI interrupt [Initial Value]
Writing 1	Enables an NMI interrupt
Reading	Reads current setting value

Notes:

- *Before enabling NMI interrupt, make sure to check interrupt factor with NMIRR register and clear it with NMICL register. Without clearing interrupt factor, there is a possibility to occur NMI interrupt when enabling NMI interrupt.*
- *The initial value of this register is "0", so NMI is disabled.*

CHAPTER 9: DMAC

This chapter explains DMAC.

1. Overview of DMAC
2. Configuration of DMAC
3. Functions and Operations of DMAC
4. DMAC Control
5. Registers of DMAC
6. Usage Precautions

CODE: 9BFDMAC_FM0-E03.0_MHDMAC-E01.0

1. Overview of DMAC

DMAC (Direct Memory Access Controller) is a function block that transfers data at high speed without CPU. Using DMAC improves the system performance.

Overview of DMAC

- DMAC has its own bus which is independent from the CPU bus; therefore, it allows for transfer operation even when the CPU bus is accessed.
- It consists of maximum 8 channels enabled to execute 8 types of different DMA transfers independently from one another.
- It can set the address of the transfer destination, the address of the transfer source, the size of transfer data, the source of transfer request and the transfer mode, and control the start of transfer operation, the forced termination of transfer and the pause of transfer for each channel.
- It can control the batch start of transfers, the forced batch termination of transfers and the batch pause of transfers for all of the channels.
- When multiple channels are operating simultaneously, it can select the priority of such channel operations from the fixed method or the rotated method.
- It supports hardware DMA transfer using an interrupt signal from Peripherals.
- It complies with the system bus (AHB), supporting 32-bit address space (4 Gbytes).

Overview of Functions of Each Channel

- The addresses of the transfer source and transfer destination can be incremented or fixed.
- Reload function for the addresses of the transfer source and transfer destination (i.e. function to return the values to the original settings upon completion of the transfer) is available.
- The size of data to be transferred can be selected from the following three specifications:

  Transfer data width: (Select from byte/half-word/word)
  Setting the number of blocks: (Select from 1 to 16)
  Setting the number of transfers: (Select from 1 to 65536)
  (For information about the difference between the number of blocks and the number of transfers, see 3 Functions and Operations of DMAC.)

- Whether or not to give notification of the successful completion of transfer and unsuccessful completion of transfer can be specified.
- Transfer mode can be selected from the following five types:

  Software-Block transfer
  Software-Burst transfer
  Hardware-Demand transfer
  Hardware-Block transfer
  Hardware-Burst transfer

Transfer Modes

Software transfer is a method used to start DMAC by direct instruction from CPU.

Hardware transfer is a method using an interrupt signal from a Peripheral as the DMAC transfer request signal to start DMAC directly when the Peripheral issues a transfer request.

Multifunction serial unit and ADC unit directly instruct DMAC to start data transfer, when sending/receiving data or A/D conversion data needs to be transferred. External interrupt unit and Base timer unit directly instruct DMAC to start data transfer at a transfer timing. In either of the cases, data can be transferred without CPU by making such setting beforehand.

Abbreviations

This chapter contains the following terms: DE, DS, DH, PR, EB, PB, ST, IS, BC, TC, MS, TW, FS, FD, RC, RS, RD, EI, CI, SS, EM. All of these terms refer to each bit of DMAC control registers (DMACR, DMACSA, DMACDA, DMACA, DMACB). See "5 Registers of DMAC".

2. Configuration of DMAC

This section explains the system configuration of DMAC and the I/O signals of DMAC.

2.1. DMAC and System Configuration

2.2. I/O Signals of DMAC

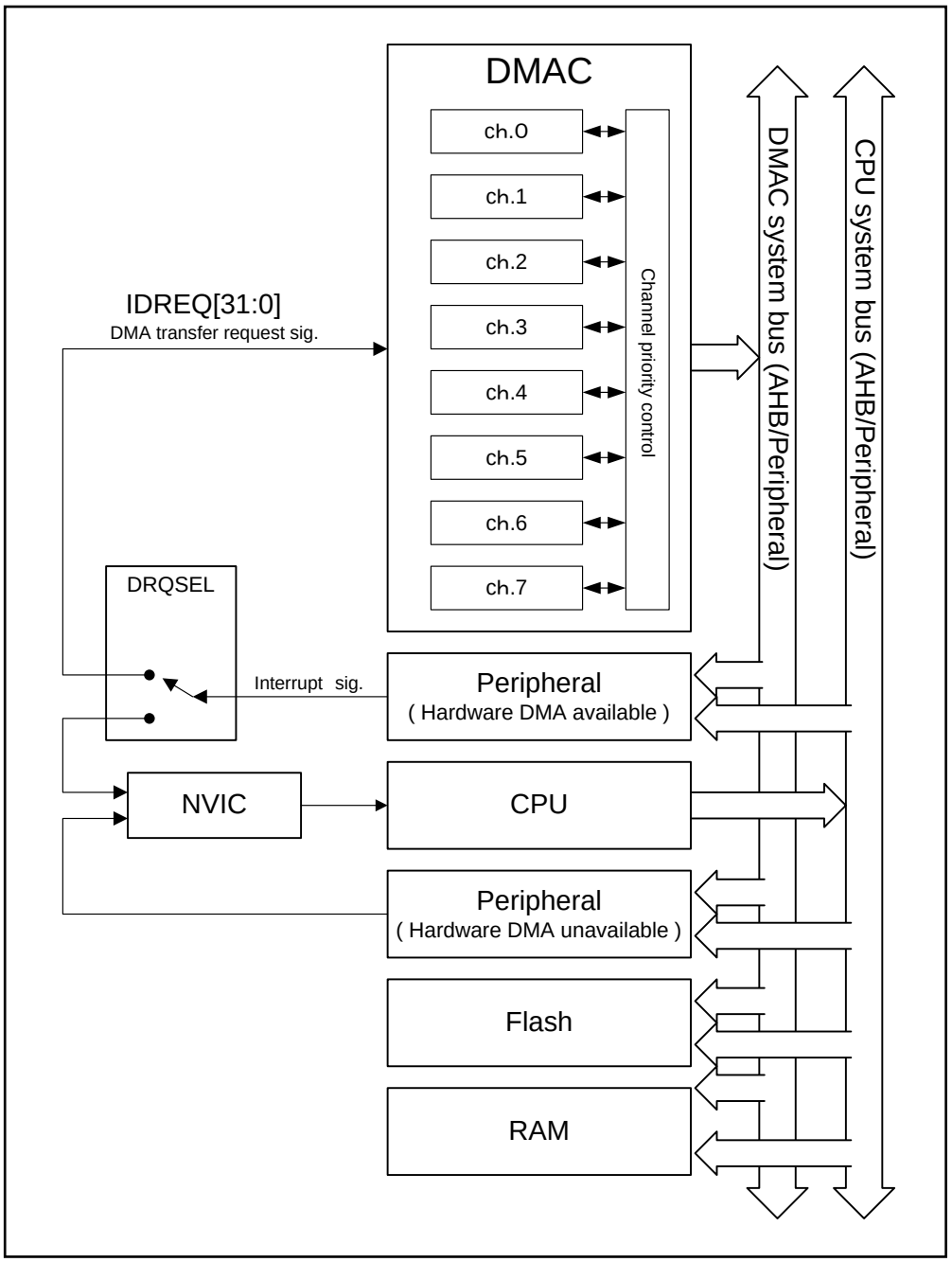
2.1 DMAC and System Configuration

This section explains DMAC and its system configuration.

Block Diagram

Figure 2-1 shows a diagram of DMAC and its system configuration.

Figure 2-1 Block Diagram of DMAC and System Configuration



Explanation of Block Diagram

- DMAC

DMAC is in maximum 8-ch configuration. Each channel performs independent transfer. The priority controller controls the transfer operations of these channels, when there is a conflict among them.

- Connection to the system

The diagram of the system configuration in the figure has been simplified for explanation purposes. For more details, see the chapter "System Overview". DMAC is connected to CPU, Flash, RAM and Peripherals via the system bus. It has its own bus that is independent from the CPU bus, allowing for transfer operation at CPU bus access. It accesses any address area in the system by specifying the address of transfer destination and transfer source for each channel in order to transfer data between the memory and Peripheral. Since some areas cannot be accessed from DMAC, check the memory map.

- Connection of the hardware transfer request signal

The interrupt signal from the Peripheral supporting hardware transfer is selected in the interrupt controller block (indicated as DRQSEL in Figure 2-1) either to be used as the interrupt signal to CPU or the DMA transfer request signal to DMAC.

When performing DMA transfer by hardware request, connect the interrupt signal from each Peripheral as the transfer request signal to DMAC in advance by setting DRQSEL. The interrupt signal from the Peripheral that does not support hardware transfer cannot be used as the DMA transfer request signal. When the interrupt signal is used as the transfer request signal to DMAC, it cannot be used as the interrupt signal to CPU. See the chapter "Interrupts".

There are 32 DMA transfer request signals to be input to DMAC. For the correspondence between each signal and Peripheral, see Table 2-1 in the next section.

Interrupt signals from the peripheral that is not integrated cannot be selected. It should be noted that for a Peripheral with multiple channels and multiple interrupt factors, some interrupts support DMA transfer, while others don’t.

In the case of hardware transfer, each channel of DMAC selects one transfer request signal out of the above 32 transfer request signals in its operation. The IS register is used for the selection.

- Connection of the hardware transfer request clear signal

Some of the Peripherals that support hardware transfer are required to clear the transfer request signal (interrupt signal) after the completion of the transfer. Although it is not illustrated in Figure 2-1, the transfer request signal is cleared for such Peripherals via DMAC by selecting it by DRQSEL.

- Connection of the hardware transfer stop request signal

The multifunction serial unit (hereinafter abbreviated as "MFS") outputs the DMA transfer stop request signal. Although it is not illustrated in Figure 2-1, MFS’s transfer stop request signal is connected to DMAC, when MFS is selected by DRQSEL. When the transfer stop request signal is asserted, DMAC stops the transfer operation. It is configured to mask the succeeding transfer request signals.

Conditions that are asserted by MFS's transfer stop request signal show below.

  - If received interrupts are enabled (SCR:RIE=1), a received interrupt occurs (SSR:PE bit, FRE bit, or ORE bit is set to 1).
  - If chip select error interrupt are enabled (SACSR:CSEIE=1), a chip select error interrupt occurs (SACSR:CSE bit is set to 1).

- Interrupt signal from DMAC

Although it is not illustrated in Figure 2-1, an interrupt signal used to give notification of transfer completion is connected to NVIC. Each channel has 8 interrupt outputs.

2.2 I/O Signals of DMAC

This section explains the I/O signals of DMAC.

Transfer Request Signals to be Input to DMAC

Table 2-1 shows a list of the transfer request signals to be input to DMAC and the interrupt signals from the corresponding Peripherals.

Table 2-1 List of Transfer Request Signals and Interrupt Signals from Corresponding Peripherals

IDREQ No.	Interrupt Signal of Corresponding Peripheral
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Scan conversion interrupt signal from A/D converter unit0
6	Scan conversion interrupt signal from A/D converter unit1
7	Scan conversion interrupt signal from A/D converter unit2
8	Interrupt signal from IRQ0 of base timer ch.0
9	Interrupt signal from IRQ0 of base timer ch.2
10	Interrupt signal from IRQ0 of base timer ch.4
11	Interrupt signal from IRQ0 of base timer ch.6
12	Receiving interrupt signal from MFS ch.0
13	Sending interrupt signal from MFS ch.0
14	Receiving interrupt signal from MFS ch.1
15	Sending interrupt signal from MFS ch.1
16	Receiving interrupt signal from MFS ch.2
17	Sending interrupt signal from MFS ch.2
18	Receiving interrupt signal from MFS ch.3
19	Sending interrupt signal from MFS ch.3
20	Receiving interrupt signal from MFS ch.4
21	Sending interrupt signal from MFS ch.4
22	Receiving interrupt signal from MFS ch.5
23	Sending interrupt signal from MFS ch.5
24	Receiving interrupt signal from MFS ch.6
25	Sending interrupt signal from MFS ch.6
26	Receiving interrupt signal from MFS ch.7
27	Sending interrupt signal from MFS ch.7
28	Interrupt signal from external interrupt unit ch.0
29	Interrupt signal from external interrupt unit ch.1
30	Interrupt signal from external interrupt unit ch.2
31	Interrupt signal from external interrupt unit ch.3

Interrupt Signals Output from DMAC

Table 2-2 shows a list of the interrupt signals output from DMAC.

Table 2-2 List of Interrupt Signals from DMAC

Name of Interrupt Signal	Interrupt Factor Register	Interrupt Enable Register	Interrupt Type
DIRQ0	DMACB0:SS[2:0]	DMACB0.CI	ch.0 successful transfer completion interrupt
		DMACB0.EI	ch.0 unsuccessful transfer completion interrupt
DIRQ1	DMACB1:SS[2:0]	DMACB1.CI	ch.1 successful transfer completion interrupt
		DMACB1.EI	ch.1 unsuccessful transfer completion interrupt
DIRQ2	DMACB2:SS[2:0]	DMACB2.CI	ch.2 successful transfer completion interrupt
		DMACB2.EI	ch.2 unsuccessful transfer completion interrupt
DIRQ3	DMACB3:SS[2:0]	DMACB3.CI	ch.3 successful transfer completion interrupt
		DMACB3.EI	ch.3 unsuccessful transfer completion interrupt
DIRQ4	DMACB4:SS[2:0]	DMACB4.CI	ch.4 successful transfer completion interrupt
		DMACB4.EI	ch.4 unsuccessful transfer completion interrupt
DIRQ5	DMACB5:SS[2:0]	DMACB5.CI	ch.5 successful transfer completion interrupt
		DMACB5.EI	ch.5 unsuccessful transfer completion interrupt
DIRQ6	DMACB6:SS[2:0]	DMACB6.CI	ch.6 successful transfer completion interrupt
		DMACB6.EI	ch.6 unsuccessful transfer completion interrupt
DIRQ7	DMACB7:SS[2:0]	DMACB7.CI	ch.7 successful transfer completion interrupt
		DMACB7.EI	ch.7 unsuccessful transfer completion interrupt

Reference: Interrupt Generation Factors and Clearing (For details, see "4 DMAC Control".)

Interrupt from each channel is generated by the following factors:

- Upon the successful completion of channel transfer, "101" is set to SS[2:0] of the channel. If the above value is set to SS[2:0] with CI=1 (successful transfer completion interrupt enabled), a successful transfer completion interrupt occurs.
- Upon the unsuccessful completion of channel transfer, "001", "010", "011" and "100" are set to SS[2:0] of the channel. If the above value is set to SS[2:0] with EI=1 (unsuccessful transfer completion interrupt enabled), an unsuccessful transfer completion interrupt occurs.
- The successful transfer completion interrupt and the unsuccessful transfer completion interrupt undergo logic OR; therefore, if either of the interrupts occurs, an interrupt occurs from the channel.

Interrupt from each channel can be cleared by writing "000" to SS[2:0].

3. Functions and Operations of DMAC

This section explains the operations of DMAC in each transfer mode.

3.1. Software-Block Transfer
3.2. Software-Burst Transfer
3.3. Hardware-Demand Transfer
3.4. Hardware-Block Transfer & Burst Transfer
3.5. Channel Priority Control

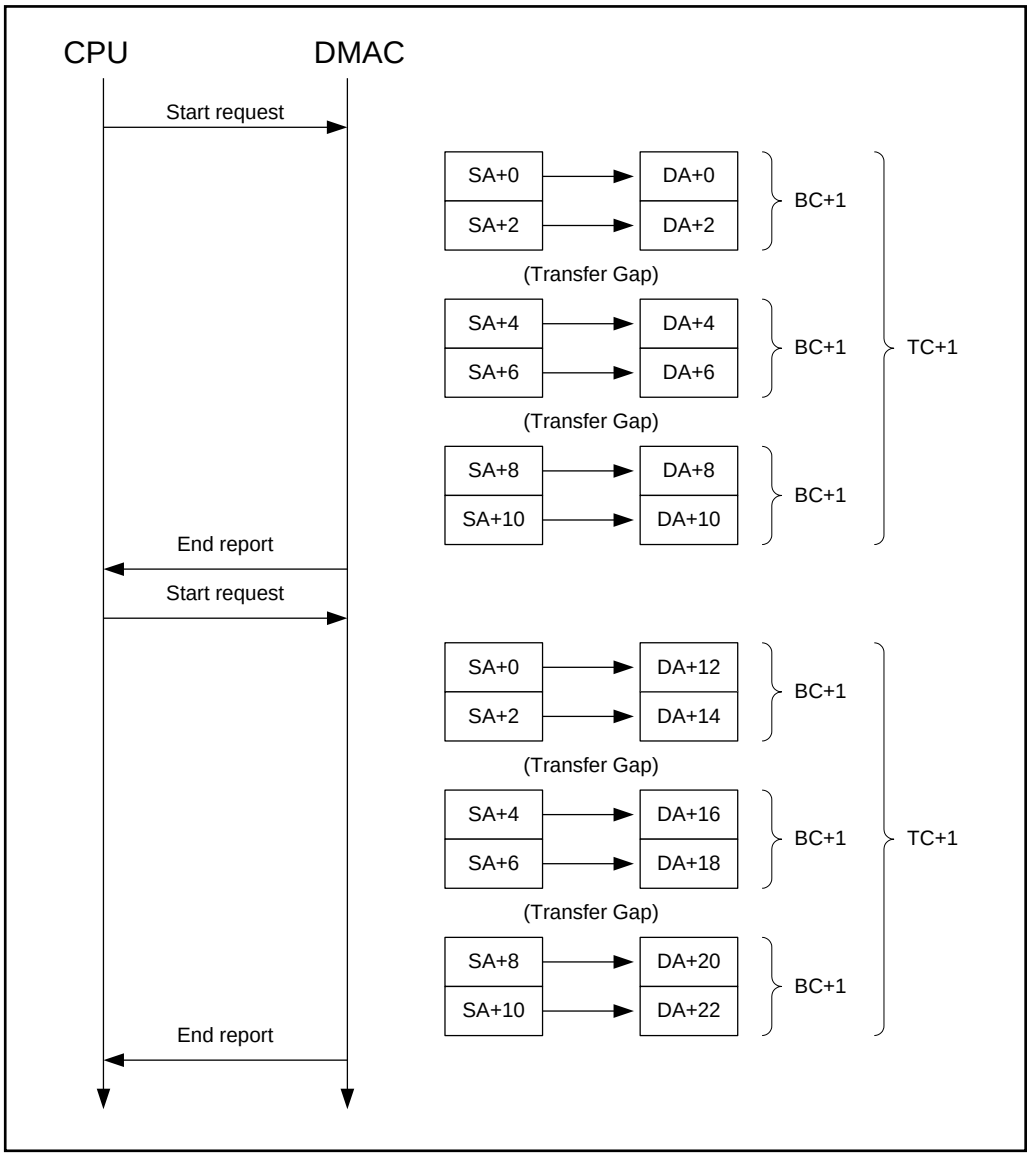
3.1 Software-Block Transfer

This section explains Software-Block transfer.

Figure 3-1 shows an example of the operation of Software-Block transfer. In this example, the following settings apply.

- Transfer mode: Software request Block transfer (ST=1, IS[5:0]=000000, MS=00)
- Transfer source start address: SA(DMACSA=SA)
- Transfer source address control: Increment and reload available (FS=0, RS=1)
- Transfer destination start address: DA(DMACDA=DA)
- Transfer destination address control: Increment and reload not available (FD=0, RD=0)
- Transfer data size: Half-word (16 bits), the number of blocks = 2, the number of transfers = 3 (TW=01, BC=1, TC=2)
- BC/TC reload: Reload available (RC=1)

Figure 3-1 Example of Operation of Software-Block Transfer



DMAC performs the following operation, when the transfer content is set from CPU and then the start of the transfer is instructed.

- Due to the specification of the transfer data width, each transfer is performed by half-word (16bits).
- According to the start addresses of the transfer source and transfer destination, the data width and the incremented/fixed specification, the transfer is performed in the area from the address SA to address DA, for the number of blocks (=BC+1).
- In the case of Block transfer, a Transfer Gap occurs every time transfer of one block is completed.
- DMAC performs data transfer for the number of blocks (=BC+1) by the number of transfers (=TC+1). The size of data to be transferred by each transfer request from CPU is "Data width (TW) × Number of blocks (BC+1) × Number of transfers (TC+1)".
- Once the transfer is completed, DMAC notifies CPU of the completion.
- If the start of transfer is instructed again after the completion of the transfer, the transfer is restarted from the previous transfer start address (SA+0), because the transfer source address has been set to be reloaded (RS=1). As the transfer destination address has not been specified to be reloaded (RD=0), the transfer is started from the next address (DA+12) after the previous transfer end address. Also, as the reload of BC/TC has been specified, the same values as for the previous transfer are reloaded for the number of blocks and the number of transfers for the next transfer.

Transfer Gap is a time period during which no transfer is performed, and it is inserted to prevent one of the DMAC channels from taking the possession of the system bus access right. If multiple channels have transfer requests, DMAC switches the channels that will perform the transfer operation at the timing of the Transfer Gap. The frequency of Transfer Gap generation can be controlled by adjusting the settings of BC and TC.

Moreover, the bus access right is also passed on to CPU at the Transfer Gap timing. System buses in this product are in Multi-layered configuration with a special system bus dedicated to DMA. For this reason, if there is no conflict between CPU and the destination of access, transfer can be performed at the same time as the CPU operation. Even if there is a conflict between CPU and the destination of access, the CPU operation is little affected, as long as the DMAC transfer is in a different address area group (RAM and Peripheral, or Flash memory and RAM, etc.). However, if the transfer is in the same address area group (RAM and RAM, etc.), the CPU operation and/or system performance may be affected, depending on the number of blocks used; therefore, attention must be paid.

("Address area group" mentioned above refers to a group of address areas that are connected on the AHB system bus with the same bus bridge.)

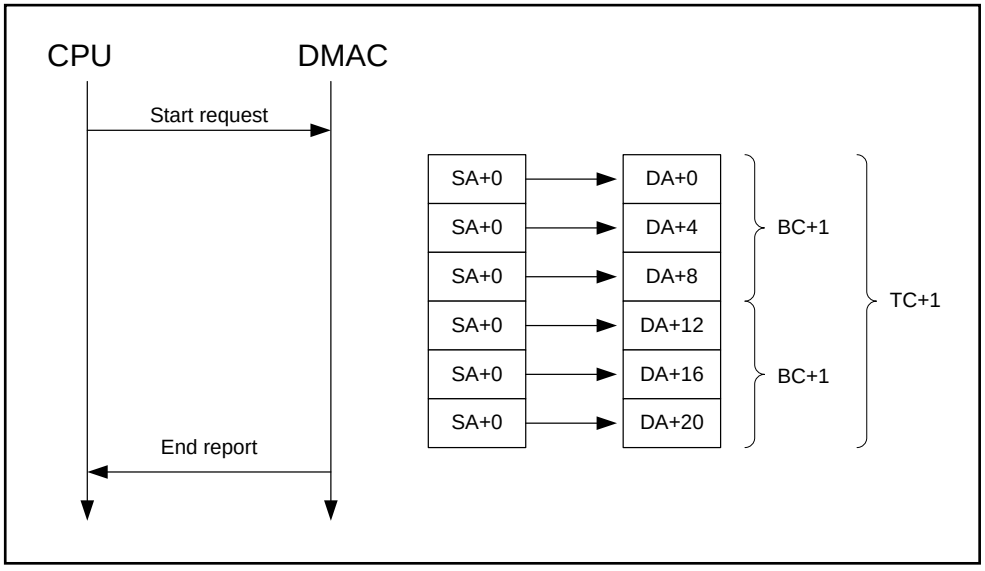
3.2 Software-Burst Transfer

This section explains Software-Burst transfer.

Figure 3-2 shows an example of the operation of Software-Burst transfer. In this example, the following settings apply.

- Transfer mode: Software request Burst transfer (ST=1, IS[5:0]=000000, MS=01)
- Transfer source start address: SA(DMACSA=SA)
- Transfer source address: Fixed, reload available (FS=1, RS=1)
- Transfer destination start address: DA(DMACDA=DA)
- Transfer destination address: Increment and reload not available (FD=0, RD=0)
- Transfer data size: Word (32 bits), the number of blocks =3, the number of transfers =2 (TW=10, BC=2, TC=1)
- Reload of the number of transfers: Number of transfers to be reloaded (RC=1)

Figure 3-2 Example of Operation of Software-Burst Transfer



DMAC performs the following operation, when the transfer content is set from CPU and then the start of the transfer is instructed.

- Due to the specification of the transfer data width, each transfer is performed by word (32bits).
- According to the start addresses of the transfer source and transfer destination, the data width and the incremented/fixed specification, the transfer is performed in the area from the address SA to address DA, for the number of blocks (=BC+1). As the transfer source address is specified to be fixed, it is the same as the transfer source start address (SA+0).
- In the case of Burst transfer, the transfer is executed continuously without generating Transfer Gaps.
- DMAC performs data transfer for the number of blocks (=BC+1) by the number of transfers (=TC+1). The size of data to be transferred by each transfer request from CPU is "Data width (TW) □ Number of blocks (BC+1) □ Number of transfers (TC+1)".

- When the transfer is completed, DMAC notifies CPU of the completion.

In the case of Burst transfer, no Transfer Gap is generated, unlike the Block transfer. As the channel to be controlled takes the possession of the system bus access right, it can be used to put the priority on that particular channel.

3.3 Hardware-Demand Transfer

This section explains Hardware-Demand transfer.

Hardware-Demand transfer is used when performing DMA transfer by the transfer request signal from the Peripherals of MFS and ADC.

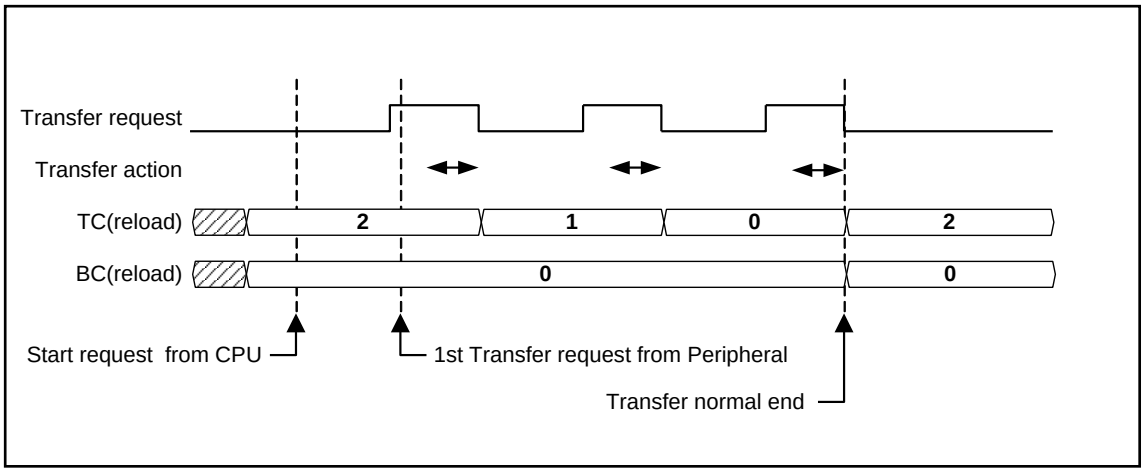
Hardware-Demand transfer is a method used to receive the transfer request signal from Peripherals on a signal level. If the transfer request signal is on High level, transfer is executed. If the transfer request signal is on Low level, no transfer is executed. Transfer is executed by setting the output of the interrupt signal from each Peripheral to High level (with interrupt request) when transfer data exists, or to Low level (without transfer request) when no transfer data exists.

In the case of Hardware-Demand transfer, always specify 1 (BC=0) as the number of blocks.

Figure 3-3 shows an example of the operation of Hardware-Demand transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode: Hardware-Demand transfer (ST=0, IS= Peripheral at the transfer request source, MS=10)
- Transfer data size: Number of blocks = 1, Number of transfers = 3 (BC=0, TC=2)

Figure 3-3 Example of Operation of Hardware-Demand Transfer



The operation of Hardware-Demand transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the transfer request, it performs one transfer and then waits for the next transfer request. During the wait period, a Transfer Gap is generated. Every time a transfer request is received, it performs the same operation for the number of transfers (TC+1). The total number of transfers to be performed is (TC+1). Match the number of transfer requests from the Peripheral and the number of DMAC transfers (TC+1). Once all of the transfers are completed, DMAC notifies CPU of the completion.

3.4 Hardware-Block Transfer & Burst Transfer

This section explains Hardware-Block transfer and Burst transfer.

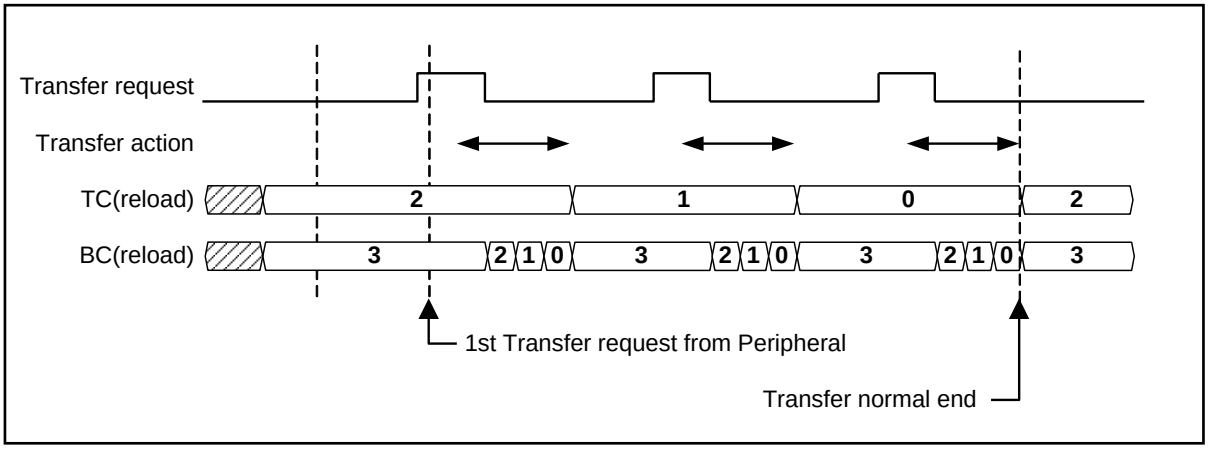
Hardware-Block transfer or Hardware-Burst transfer is used when performing DMA transfer by the transfer request signal from the Peripheral of the base timer or external interrupt.

Hardware-Block transfer and Hardware-Burst transfer are methods used to receive the transfer request signal at the rising edge of the signal. Transfer is executed, when the rising edge of the transfer request signal is detected. DMAC's transfer start timing can be specified by the output of the interrupt signal from each Peripheral.

Figure 3-4 shows an example of the operation of Hardware-Block transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode: Hardware-Block transfer
  (ST=0, IS= Peripheral at the transfer request source, MS=00)
- Transfer data size: Number of blocks = 4, Number of transfers = 3 (BC=3, TC=2)

Figure 3-4 Example of Operation of Hardware-Block Transfer



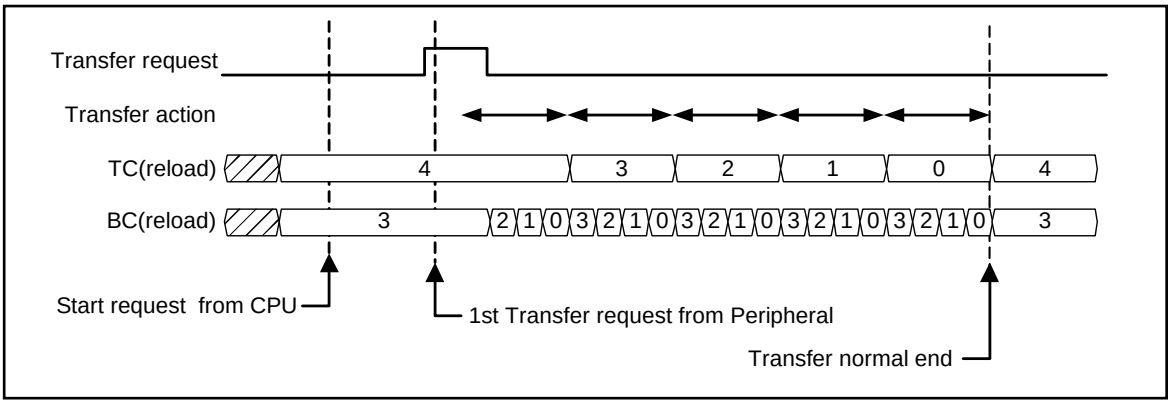
The operation of Hardware-Block transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the transfer request, it performs transfers for the number of blocks (=BC+1) and then waits for the next transfer request. During the wait period, a Transfer Gap is generated. Every time a transfer request is received, it performs the same operation for the number of transfers (TC+1). The total number of transfers to be performed is (BC+1) □ (TC+1). Match the number of transfer requests from the Peripheral and the number of DMAC transfers (TC+1). Once all of the transfers are completed, DMAC notifies CPU of the completion.

Figure 3-5 shows an example of the operation of Hardware-Burst transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode: Hardware-Burst transfer
  (ST=0, IS= Peripheral at the transfer request source, MS=01)
- Transfer data size: Number of blocks =4, Number of transfers = 5 (BC=3, TC=4)

Figure 3-5 Example of Operation of Hardware-Burst Transfer



The operation of Hardware-Burst transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the first transfer request, it performs all of the transfers for the number of times calculated by (BC+1) × (TC+1). During the Hardware-Burst transfer, no Transfer Gap is generated. Once all of the transfers are completed, DMAC notifies CPU of the completion.

3.5 Channel Priority Control

This section explains the channel priority control.

Channel Priority Control

If multiple channels have transfer requests, DMAC switches the channel subject to the transfer among them at the timing of the Transfer Gap of each channel. At this point, the next channel to which the transfer will be performed is determined according to the priority control. The priority control can be selected from either fixed priority or rotated priority by the PR. Figure 3-6 shows an explanatory diagram. In this figure, the X axis indicates the time axis. The arrows indicate transfer timings of each channel to perform its transfer operation when all of the channels issue transfer requests simultaneously.

Operation in Fixed Priority Mode (PR=0)

In fixed priority mode, the channel with the smallest channel number among all the channels with a transfer request has the priority to perform transfer operation.

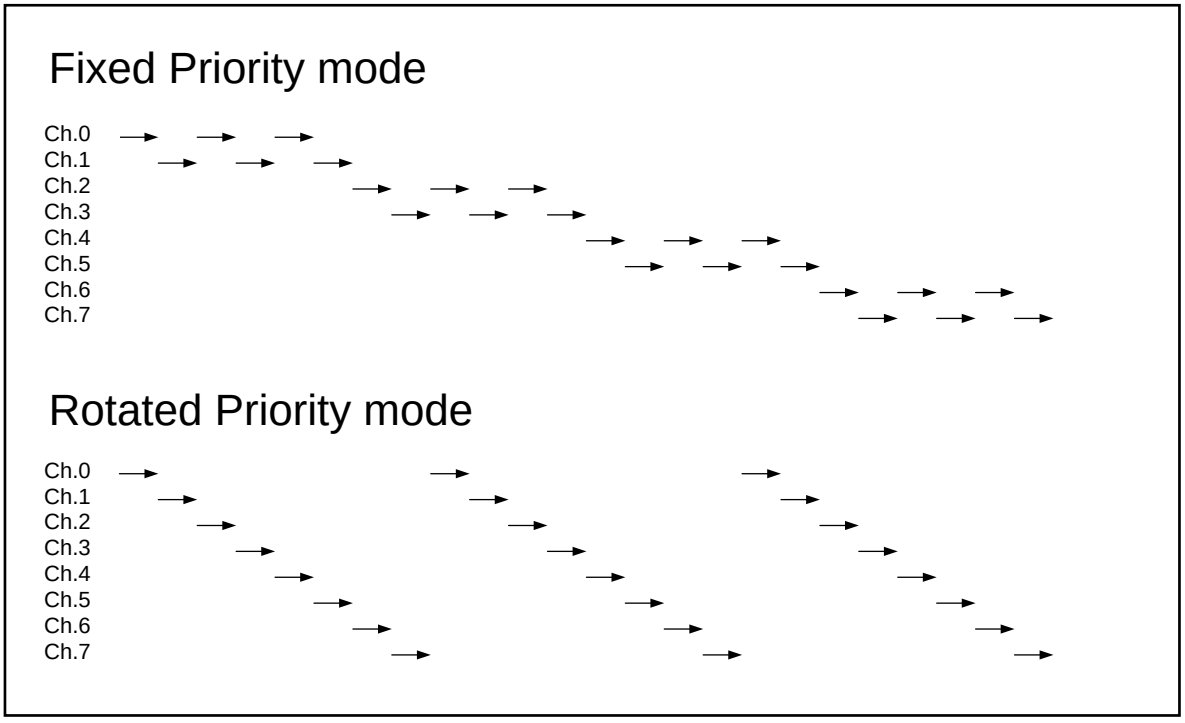
(Priority order: ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7)

First, the channel with the highest priority performs its transfer (ch.0 in Figure 3-6). As the channel with the highest priority halts the transfer operation at the timing of a Transfer Gap, then, the channel with the second highest priority performs its transfer operation (ch.1 in Figure 3-6). For this reason, the channels with the highest and the second highest priority perform the transfer operations alternately. After that, when the channel with higher priority completes its transfer, the channel with lower priority starts its transfer operation (ch.3 in Figure 3-6).

Operation in Rotated Priority Mode (PR=1)

In rotate priority mode, all channels perform their transfer operations equally.

Figure 3-6 Explanatory Diagram of Channel Priority Control



4. DMAC Control

This section explains DMAC control methods in details.

4.1. Overview of DMAC Control
4.2. DMAC Operation and Control Procedure for Software Transfer
4.3. DMAC Operation and Control Procedure for Hardware (EM=0) Transfer
4.4. DMAC Operation and Control Procedure for Hardware (EM=1) Transfer

4.1 Overview of DMAC Control

This section provides an overview of DMAC control.

The control register of each channel of DMAC has EB (individual-channel operation enable bit) and PB (individual-channel pause bit). By manipulating these bits, the start of DMA transfer operation (operation enabled), the forced termination of transfer operation (operation disabled) and the pause of transfer operation can be controlled by channel. The control register also has DE (all-channel operation enable bit) and DH (all-channel pause bit), which allow the transfer operations of all channels to be controlled at once.

Each channel is originally in the operation-prohibited state (Disable state) in which the transfer content (the address of the transfer source, the address of the transfer destination, the transfer data width, the number of transfers, the transfer mode, etc.) are specified for each channel to its configuration register. Then, the transfer operations are controlled by writing to EB, PB, DE and DH to instruct the transfer operations to be started or paused.

Once each channel completes its transfer, it sets the end code to SS (Stop Status) to give the notification of its stop state. An interrupt can be generated upon the completion of transfer. After the transfer ends, each channel clears EB and PB and returns to the operation-prohibited state (Disable state).

The following sections describe the operations of and control procedures for DMA transfer by software request and hardware DMA transfer by transfer request from Peripherals.
The following terms are used in the explanations as instructions from CPU, which refer to writing the following values to the EB, PB, DE and DH bits.

- Instruction to enable individual-channel operation (write EB=1, PB=0)
- Instruction to disable individual-channel operation (write EB=0)
- Instruction to pause individual-channel operation (write EB=1, PB=1)
- Instruction to enable all-channel operation (write DE=1, DH=0000)
- Instruction to disable all-channel operation (write DE=0)
- Instruction to pause all-channel operation (write DE=1, DH!=0000)

4.2 DMAC Operation and Control Procedure for Software Transfer

This section explains DMAC operation and control procedure for software transfer.

Figure 4-1 Transitional Diagram of Software DMA Transfer State

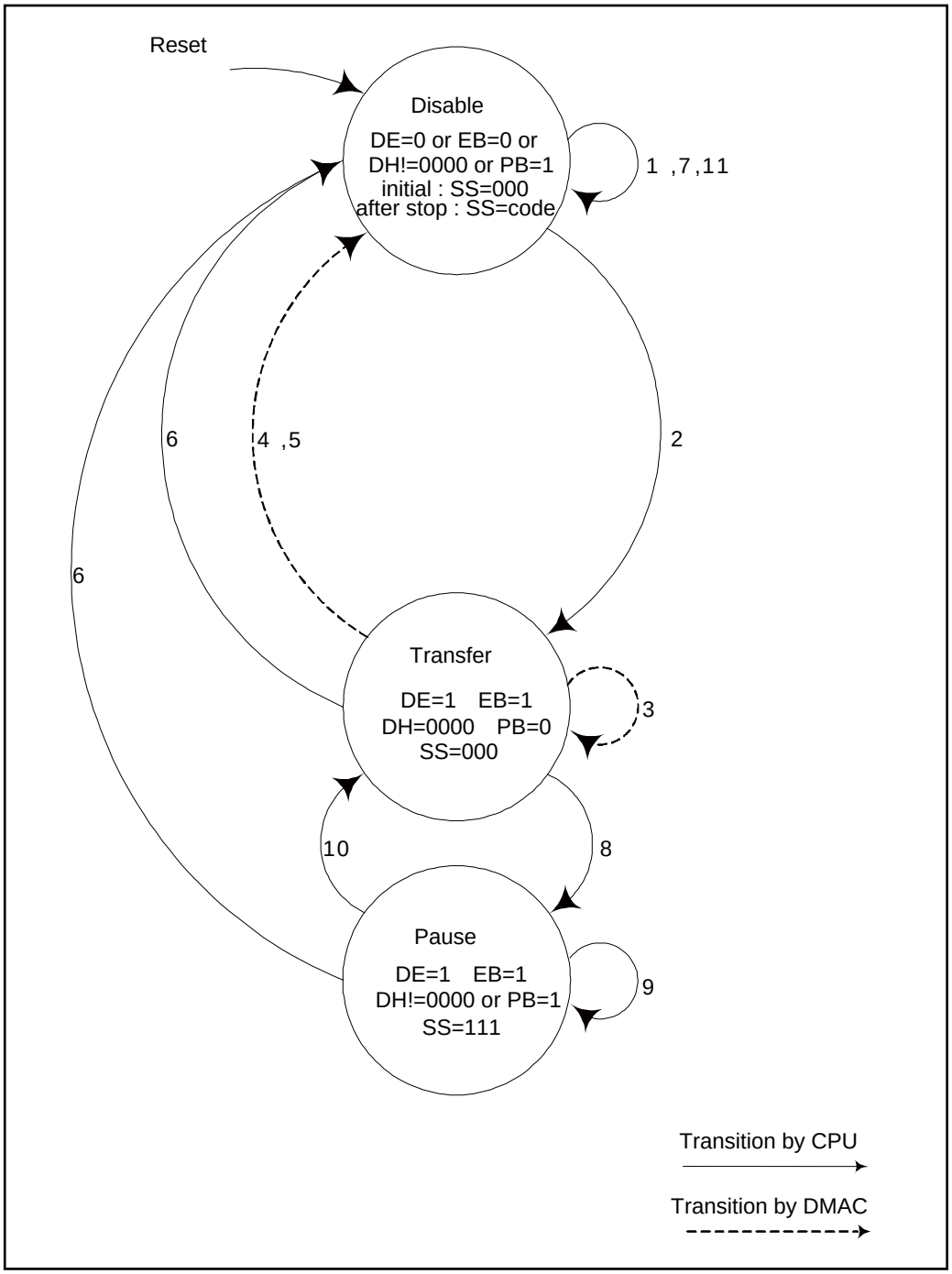


Figure 4-1 shows a transitional diagram of the states of the channel to be controlled for software transfer. The numbers next to the transitional lines in the figure correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC operation.

Description of Each State

- Disable state

  In this state, the transfer of the channel to be controlled is prohibited. Channels in this state do nothing and wait for instruction from CPU. At the system reset, DE=0, EB=0, DH=0000 and PB=0 apply to this Disable state.

- Transfer state

  In this state, the transfer of the channel to be controlled is enabled. Channels in this state perform transfer operation as specified. Once all of the transfer operations are completed, they return to the Disable state. The state is also changed as instructed by CPU.

- Pause state

  In this state, the channel to be controlled has its transfer operation on pause due to an instruction to pause, issued by CPU, and is waiting for another instruction from CPU.

Explanation of Control Procedure

1. Disable state / Preparation for transfer

   Specify via CPU the transfer content for the channel to be controlled (writing to DMACSA, DMACDA, DMACA and DMACB). For details of transfer content to be specified, see “5. Registers of DMAC”. When generating an interrupt from DMAC upon the completion of transfer, set EI and CI.

   The following restrictions apply to software transfer. Specify ST=1 and IS[5:0]=000000. Demand transfer mode cannot be specified to MS. Always set "0" to EM.

   Give an instruction to enable all-channel operation and set PR. Data can also be written to DMACA at the same time in Step 2.

2. Disable state => Transfer state / Start of transfer

   Give an instruction to enable individual-channel operation from CPU. When DE=1, EB=1, DH=0000 and PB=0 are set, the channel to be controlled moves to Transfer state.

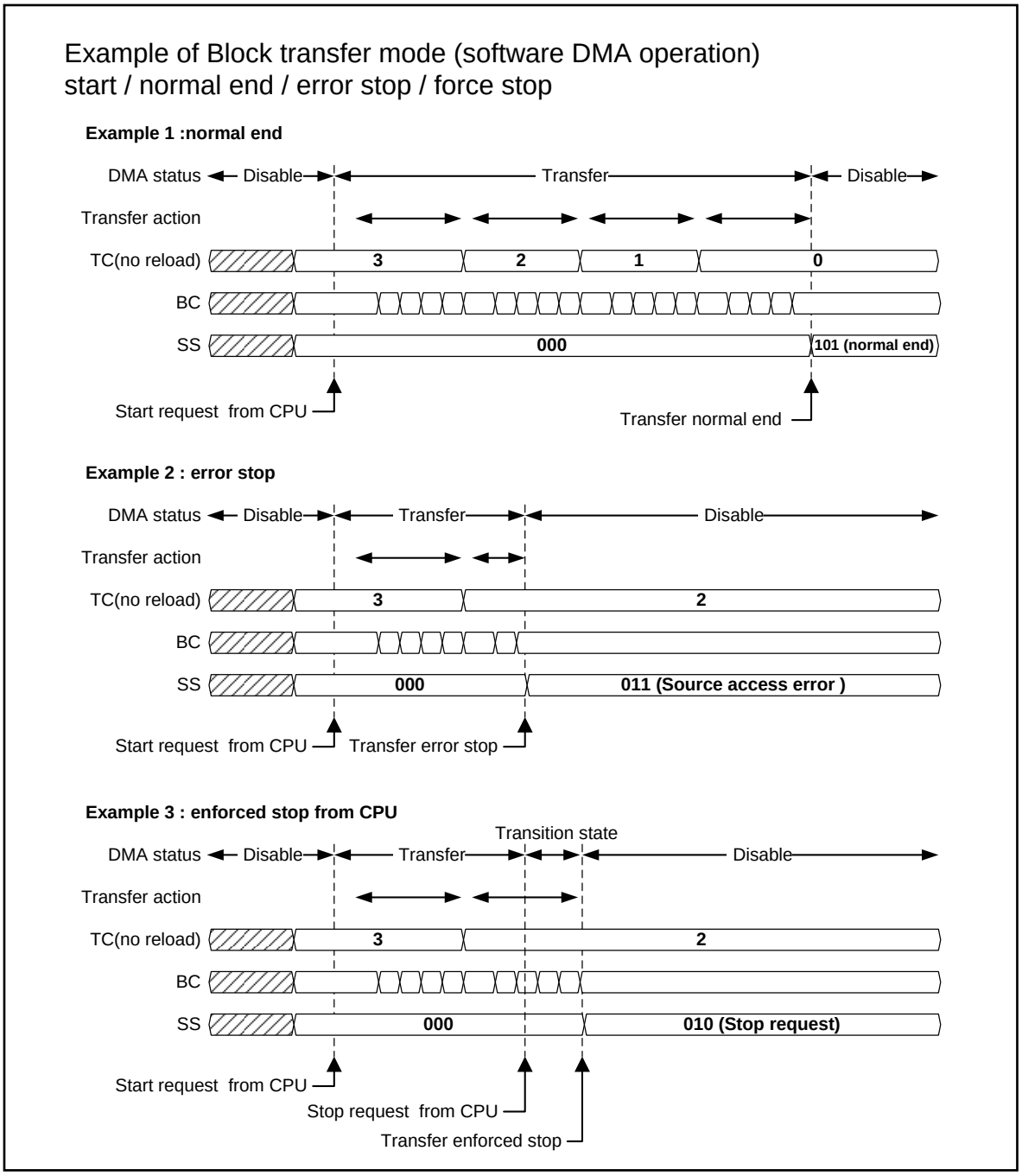
3. Transfer state

   When the channel in Transfer state becomes enabled to access the system bus, it performs a transfer according to the transfer content (it may take time to start the transfer, depending on the status of other channels). In the case of Block transfer, a Transfer Gap is generated every time TC is updated. In the case of Burst transfer, no Transfer Gap is generated. During the transfer operation, BC, TC, DMACSA and DMACDA indicate the remaining number of transfers and the transfer address at that time point. The transfer status can be checked by reading from CPU. The specified transfer content cannot be changed via CPU to the channel in Transfer state (rewriting to DMACSA, DMACDA, DMACA[29:0], DMACB[31:1]). (However, EB, PB and EM can be rewritten.)

4. Transfer state => Disable state / Successful completion of transfer

   When transfers are successfully completed for the number of times calculated by (BC+1) × (TC+1), the channel in Transfer state clears EB, PB and ST and moves to Disable state. It sets SS=101 to provide the notification of the successful completion. See Example 1 in Figure 4-2. If successful transfer completion interrupt has been enabled by CI, an interrupt occurs. If reload has been specified to BC, TC, DMACSA and DMACDA, such reload is executed according to the specified transfer content.

Figure 4-2 Example of Operation of Software-Block Transfer



5. Transfer state => Disable state / Transfer error stop

   The channel in Transfer state suspends the transfer process, if an address overflow, transfer source access error or transfer destination access error occurs. It clears EB, PB and ST and moves to Disable state. It sets the value that indicates the error content to SS[2:0] to give the notification of the error stop. See Example 2 in Figure 4-2. If unsuccessful transfer completion

interrupt has been enabled by EI, an interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set at the time of the transfer suspension.
Normally, a transfer error occurs, when an attempt is made to access an address area that does not exist in the system bus or an address area that prohibits access from DMAC. No such error occurs in general applications.

6. Transfer state, Pause state => Disable state / Forced transfer stop

   If an instruction to disable individual-channel operation or an instruction to disable all-channel operation is issued from CPU to a channel in Transfer state or Pause state, the transfer operation of that channel can be forced to stop (for the operation when an instruction to disable operation is issued to a channel in Disable state, see Step 11 in the software procedure).
   If an instruction is given from CPU, the relevant channel suspends its transfer process. It clears EB, PB and ST and moves to Disable state. It sets SS[2:0]=010 and gives the notification that the transfer of that channel has been forced to stop. If unsuccessful transfer completion interrupt has been enabled by EI, an interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set at the time of the transfer suspension.
   After instructed from CPU, the transfer stops at the timing when the relevant channel is not performing transfer (in Transfer Gap before the transfer starts), as shown in the Example 3 in Figure 4-2. In the case of a channel in Pause state, the transfer stops immediately. There is a time difference (Transition state) between the instruction and the stop. It may take some time, depending on the BC setting. As a new transfer cannot be set or started during this period, always make sure that the operation has stopped before setting the next transfer.
   In the case of an instruction to disable all-channel operation, the timing to stop varies depending on the channel. As DS is set when all of the channels are stopped, it can confirm that all of the channels have stopped.
   Even if instructed from CPU, the transfer may not be forced to stop, and instead, it may be successfully completed due to factors such as transfer mode (Burst/Block/Demand) and transfer status (the number of transfers performed, the timing of instruction to disable the operation). Also, if a transfer error occurs before the transfer stops, error stop applies to the transfer.

7. Disable state / Post-transfer process

   SS is read from CPU to check the state of completion of the transfer. CPU clears SS to prepare for the next transfer. If interrupts have been enabled, the interrupt signal from DMAC is deasserted by clearing SS.
   In the case of successful completion, CPU resets the transfer content, as required. If each reload has been specified, the values set before the start of the transfer are reloaded to BC, TC, DMACSA and DMACDA. If each reload has not been specified, BC and TC are initialized to 0. DMACSA and DMACDA show the address for the next transfer.
   In the cases of error stop and forced stop, BC, TC, DMACSA and DMACDA must always be reset, because they may have the values set at the time of the suspension.
   If the transfer is stopped due to an instruction to disable all-channel operation, DE is set to 0; therefore, the next transfer will require an instruction to enable all-channel operation and an instruction to enable individual-channel operation.

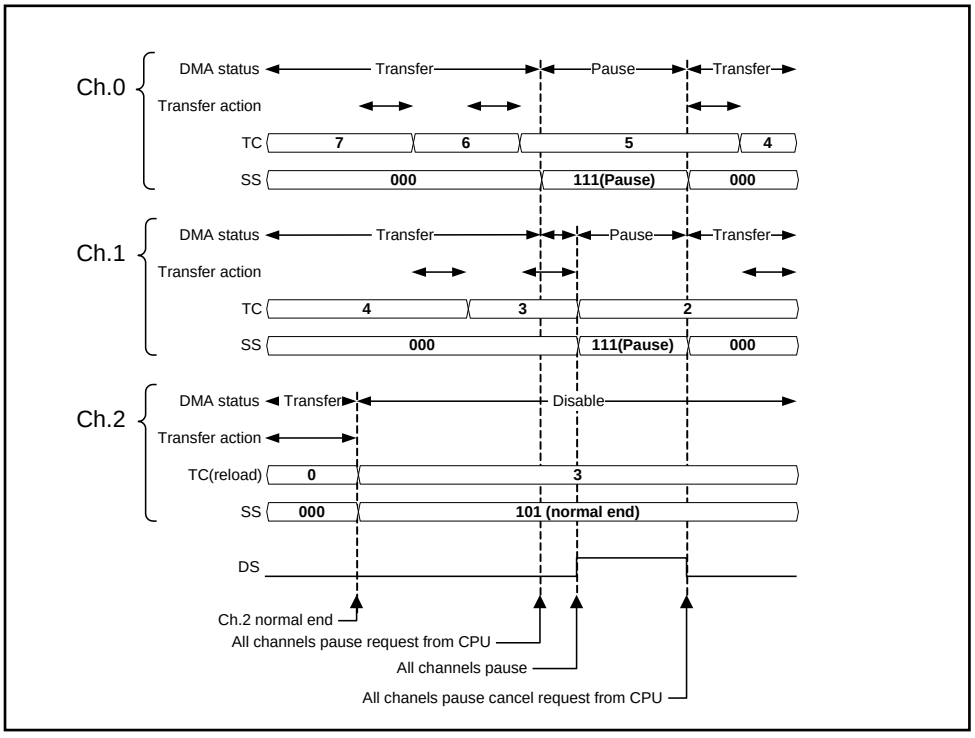
8. Transfer state / Transfer pause

   If an instruction to put individual-channel operation on pause or an instruction to put all-channel operation on pause is issued from CPU to a channel in Transfer state, the transfer operation of the relevant channel(s) can be put on pause (for the operation when an instruction to put the

operation on pause is issued to a channel in Disable state, see Step 11 in the software procedure).
If an instruction is given from CPU, the relevant channel(s) temporarily suspends the transfer process. It sets SS=111 and gives the notification that it is in Pause state. In this case, no interrupt can be generated.
After instructed from CPU, the transfer stops at the timing when the relevant channel is not performing transfer (in Transfer Gap before the start of the transfer). There is a time difference (Transition state) between the instruction and the stop. It may take some time, depending on the BC setting. See Figure 4-3.
In the case of an instruction to put all-channel operation on pause, the timing to stop varies depending on the channel. As DS is set when all of the channels are stopped, it can confirm that all of the channels have stopped. See Figure 4-3.

Even if instructed from CPU, the transfer may not be put on pause, and instead, it may be successfully completed due to factors such as transfer mode (Burst/Block/Demand) and transfer status (the number of transfers performed, the timing of instruction to put the operation on pause). Also, if a transfer error occurs before the transfer stops, error stop applies to the transfer.

Figure 4-3 Operation when All-channel Pause is Instructed



9. Pause state

SS is read from CPU to confirm the pause of the transfer. The SS of a channel in Pause state is 111. While in this state, it cannot be cleared from CPU.
Even during the pause, the transfer content cannot be specified or changed (rewriting DMACSA, DMACDA, DMACA[29:0] or DMACB[31:1]). Also, when a channel in Pause state is instructed to pause, it continues to remain in the Pause state.

10. Pause state / Cancellation of transfer pause

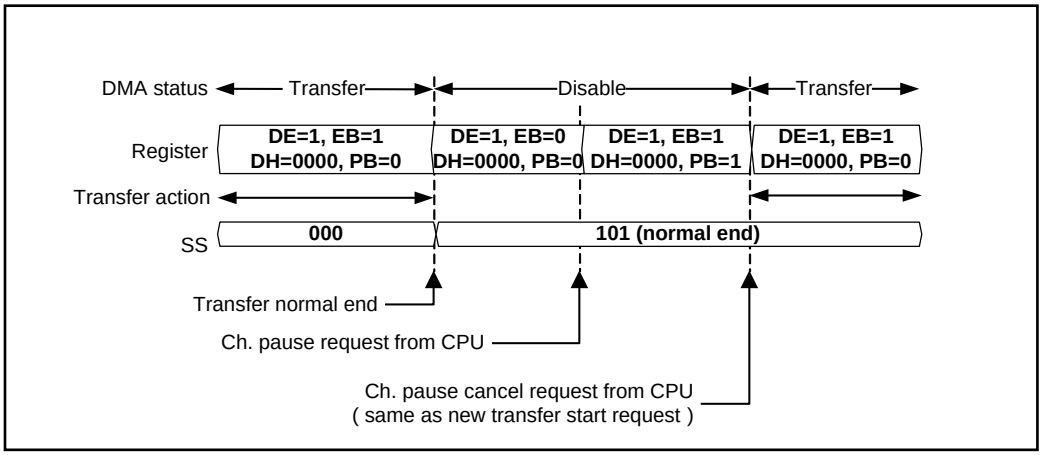
    If an instruction to enable individual-channel operation is issued to a channel that has been in Pause state due to an instruction to put individual-channel operation on pause, that channel returns to Transfer state. If an instruction to enable all-channel operation is issued to channels that have been in Pause state due to an instruction to put all-channel operation on pause, those channels return to Transfer state. If both of the pause instructions have been given, issue an instruction to cancel both of them.
    After the instruction, SS[2:0] is cleared to 000 via DMAC.
    If an instruction to enable individual-channel operation and an instruction to enable all-channel operation are issued in Pause state, they instruct the pause to be cancelled. If they are issued in Disable state, attention must be paid, as they may instruct a new transfer to be started. See Step 11 in the software procedure.
    Figure 4-3 shows an example of the case where an instruction to put all-channel operation on pause. The explanation of the figure is as follows.
    At the beginning, three channels, namely ch.0, ch.1 and ch.2, perform their transfer operations in Block transfer mode. ch.2 successfully completes its transfer, moves to Disable state and sets SS[2:0]=101. Then, ch.0 and ch.1 perform transfers alternately.
    If an instruction to put all-channel operation on pause is issued from CPU at this point, the following operation applies. As ch.0 is subject to the Transfer Gap timing, it immediately moves to Pause state and sets SS[2:0]=111. As ch.1 is in the middle of transfer operation, it performs the transfer until the timing of the next Transfer Gap, and then moves to Pause state and sets SS[2:0]=111. As ch.2 is in Disable state, it remains in the Disable state without changing SS. DS is set, when all of the channels stop their operations.
    Next, if an instruction to enable all-channel operation (instruction to cancel the pause) is issued from CPU, the following operation applies. ch.0 and ch.1 return to Transfer state and clear SS[2:0] to "000". As ch.2 is in Disable state (DE=1, EB=0), it remains in that state without starting the operation. Because the pause of all of the channels has been cancelled now, DS is reset.

11. Operation in Disable state

    A channel in Disable state remains in the Disable state, unless the conditions such as DE=1, DH=0000, EB=1, and PB=0 are established. Although in 1-2 of the software procedure, DE is set from the conditions of DE=0 and EB=0, and then, EB is set, there is no problem to set EB before DE. DE can be set last after all of the transfer settings of multiple channels subject to transfer are completed. In this case, an instruction can be issued to allow the multiple channels subject to transfer to start their transfer operations simultaneously. If such instruction for simultaneous start of transfers is issued, DMAC selects the channels to which transfers are to be started, according to the PR setting (PR can be set or changed, only when all-channel operation is disabled).
    If an instruction to disable individual-channel operation, an instruction to put individual-channel operation on pause, an instruction to disable all-channel operation or an instruction to put all-channel operation on pause is issued to a channel in Disable state, only the settings of DE, DH, EB and PB are changed, but the conditions of DE=1, DH=0000, EB=1 and PB=0 are not established. Therefore, the relevant channels do nothing and do not change SS[2:0]. If an instruction to put all-channel operation on pause is issued from CPU to a channel in Disable state, as shown in the example of ch.2 operation in Figure 4-3, that channel does not change its state with SS[2:0] indicating the completion of the previous transfer.

If an instruction to put individual- or all-channel operation on pause is issued to a channel in Disable state, it may be put in Disable state with DE=1, EB=1, (DH!=0000 or PB=1). Although the bit values in this state are the same as DE, EB, DH and PB, they can be distinguished because SS[2:0] has a different value. Figure 4-4 shows such an example.

Figure 4-4 Example of Operation when Instruction to Put Individual-channel Operation on Pause is Issued in Disabled State



A certain channel is performing transfer operation. CPU issues an instruction to put individual-channel operation on pause to that channel. The instruction is issued after the transfer is completed and it moves to Disable state (DE=1, DH=0000, EB=0, PB=0). This phenomenon can occur, because the channel currently performing transfer operation changes its state outside CPU's intention. In this case, the bit values of the relevant channel change to (DE=1, DH=0000, EB=1, PB=1) due to instruction from CPU, but SS[2:0] remains 101, the value set upon the completion. If the operation is stopped by a pause instruction, SS[2:0] will be 111; therefore, it will be possible to distinguish between the pause state and the state in which the transfer has been completed. It should be noted that if an instruction to cancel the pause is issued without checking the state of the channel by SS[2:0], a new transfer will accidentally start, as shown in Figure 4-4.

- Additional Matter 1

  As ST is cleared upon the completion of a transfer, the read value of ST is 0 after the completion of the transfer. In the case of software transfer, it should be noted that 1 must always be written to ST, regardless of its read value.

- Additional Matter 2

  An instruction to enable individual-channel operation cannot be issued during the period after the previous instruction to enable individual-channel operation instructs the start of transfer and before the completion of the transfer is confirmed. This is because the channel to be controlled may change its state outside CPU's intention and an instruction to start a new transfer may be issued when DMAC has moved to Disable state (EB=0). Even if the SS[2:0] value confirms that the channel to be controlled is in Transfer state, the channel to be controlled may move to Disable state during the period between that point and the write operation.

- Additional Matter 3

  The DE and DH values can only be rewritten from CPU and these registers are never cleared from DMAC. Therefore, there is no problem to write DE=1 and DH=0000 during the transfer operation.
  DH is not cleared, if an instruction to disable individual-channel operation is issued to a channel in all-channel Pause state (DE=1, DH!=0000, EB=1, PB=0). After the instruction, the relevant channel moves to Disable state (DE=1, DH!=0000, EB=0, PB=0). To start a new transfer of the relevant channel, write DE=1 and DH=0000. This indicates that the cancellation of the pause of all-channel operation is required in order to start a new transfer of the individual channel.

- Additional Matter 4

  The SS[2:0] value is set from DMAC upon the completion of a transfer and it is never rewritten from DMAC as long as it is in Disable state. Even if the SS[2:0] value is not cleared, the next transfer can be started. However, if it moves to Transfer state, the SS[2:0] value may be cleared from DMAC (or may not be cleared). When an interrupt from DMAC is used, it should be noted that the interrupt signal is deasserted at a timing which is not intended by CPU, if it moves to Transfer state without clearing SS[2:0].

4.3 DMAC Operation and Control Procedure for Hardware (EM=0) Transfer

This section explains DMAC operation and control procedure for hardware (EM=0) transfer.

Figure 4-5 Transitional Diagram of Hardware (EM=0) Transfer State

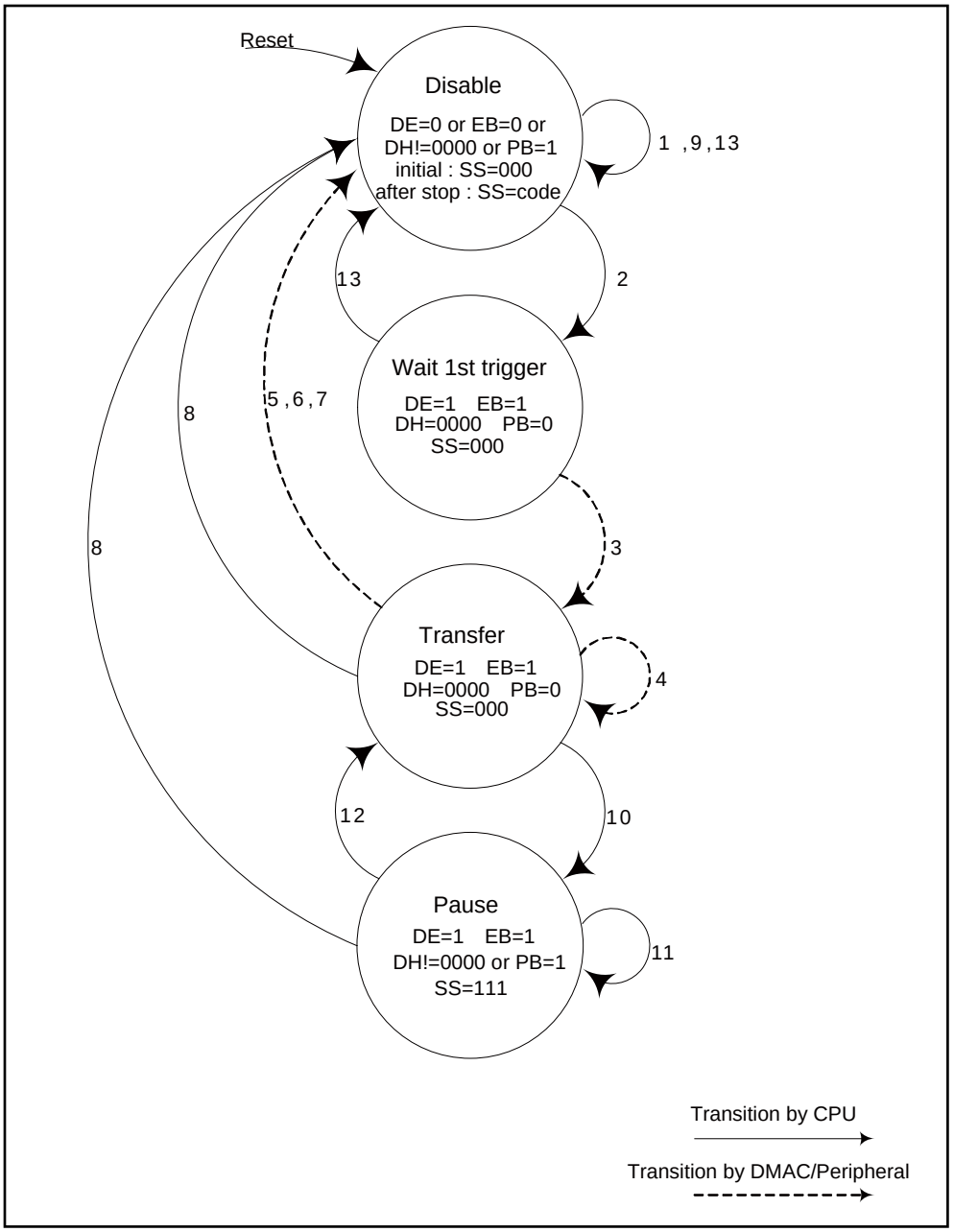


Figure 4-5 shows a transitional diagram of the states of the channel to be controlled for hardware (EM=0) transfer. The numbers next to the transitional lines in the figure correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC/Peripheral operation.

Some parts of the explanation below state "See the software transfer procedure". This means that where the same control as in the software transfer procedure applies, no special mentioning is required; therefore, such redundant explanation has been omitted. In this example, the explanation assumes that EM=0 is set.

Description of Each State

- Disable state

  See Explanation of Control Procedure in "4.2 DMAC Operation and Control Procedure for Software Transfer".

- Wait-1st-trigger state

  In this state, the channel to be controlled is enabled to perform transfer. A channel in this state waits for the first transfer request from a Peripheral to be asserted. It also changes its state upon instruction from CPU.

- Transfer state

  In this state, the channel to be controlled has received the first transfer request from the Peripheral. A channel in this state performs transfer operation as specified. Once all the transfer operation is completed, it returns to Disable state. It also changes its state upon instruction from CPU.

- Pause state

  See Explanation of Control Procedure in "4.2 DMAC Operation and Control Procedure for Software Transfer".

Explanation of Control Procedure

1. Disable state / Preparation for transfer

   See Step 1 in the software transfer procedure.
   The following restrictions apply to hardware transfer.
   Decide in advance on which Peripheral's interrupt signal to be used as the transfer request signal to DMAC using the interrupt controller block (See 4.1 DMA Request Selection Register (DRQSEL) in Chapter Interrupt.). Set ST=0 and specify which Peripheral's transfer request to be processed at the channel that will perform the transfer, by IS at the same time. Multiple channels cannot process transfer request of the same Peripheral. In the case of Demand transfer mode, set BC=0. This section explains the operation when EM=0 is set.

2. Disable state => Wait-1st-trigger state / Transfer enabled

   An instruction to enable individual-channel operation is issued from CPU. When DE=1, EB=1, DH=0000 and PB=0 are set, the channel to be controlled moves to Wait-1st-trigger state.

3. Wait-1st-trigger state / Start of transfer

   The channel in Wait-1st-trigger state is waiting for the transfer request signal to be asserted from the Peripheral or for an instruction from CPU. When the first transfer request signal is asserted, it moves to Transfer state.

4. Transfer state

   See Step 3 in the software transfer procedure.
   In the case of hardware transfer, a channel in Transfer state performs transfer operation by the transfer request signal from a Peripheral, as described in Sections 3.3 Hardware-Demand Transfer and 3.4 Hardware-Block Transfer & Burst Transfer. In each mode, match the number of transfer requests from the Peripheral with the number of transfer requests required by DMAC. Below is the explanation for the operation when the number of transfer requests goes over or below the requirement in each operation mode.
   Figure 4-6 shows a case of Demand transfer. In the case of Demand transfer, the number of transfer requests required to complete the transfer is TC+1. Unless the number of transfer

requests goes over or below the requirement, CPU does not have to intervene (Example 1 in Figure 4-6).
If the number of transfer requests generated from the Peripheral exceeds the DMAC's setting for the number of transfers, DMAC moves to Disable state after the completion of the specified number of transfers. In the Disable state, no further transfers are executed. Also, as the excessive transfer request signals are not cleared from DMAC, the asserted state continues (Example 2 in Figure 4-6).
If the number of transfer requests generated from the Peripheral is smaller than DMAC's setting for the number of transfers, DMAC waits for the remaining number of transfer requests in Transfer state (Example 3 in Figure 4-6).
It is supposed that DMAC's transfer processing may be too slow to catch up with the generation interval of transfer requests from Peripheral. In the case of Demand transfer, the transfer request signal remains asserted; therefore, as many as TC+1 of transfers can be performed (Example 4 in Figure 4-6).

Figure 4-6 Operation of Hardware-Demand Transfer

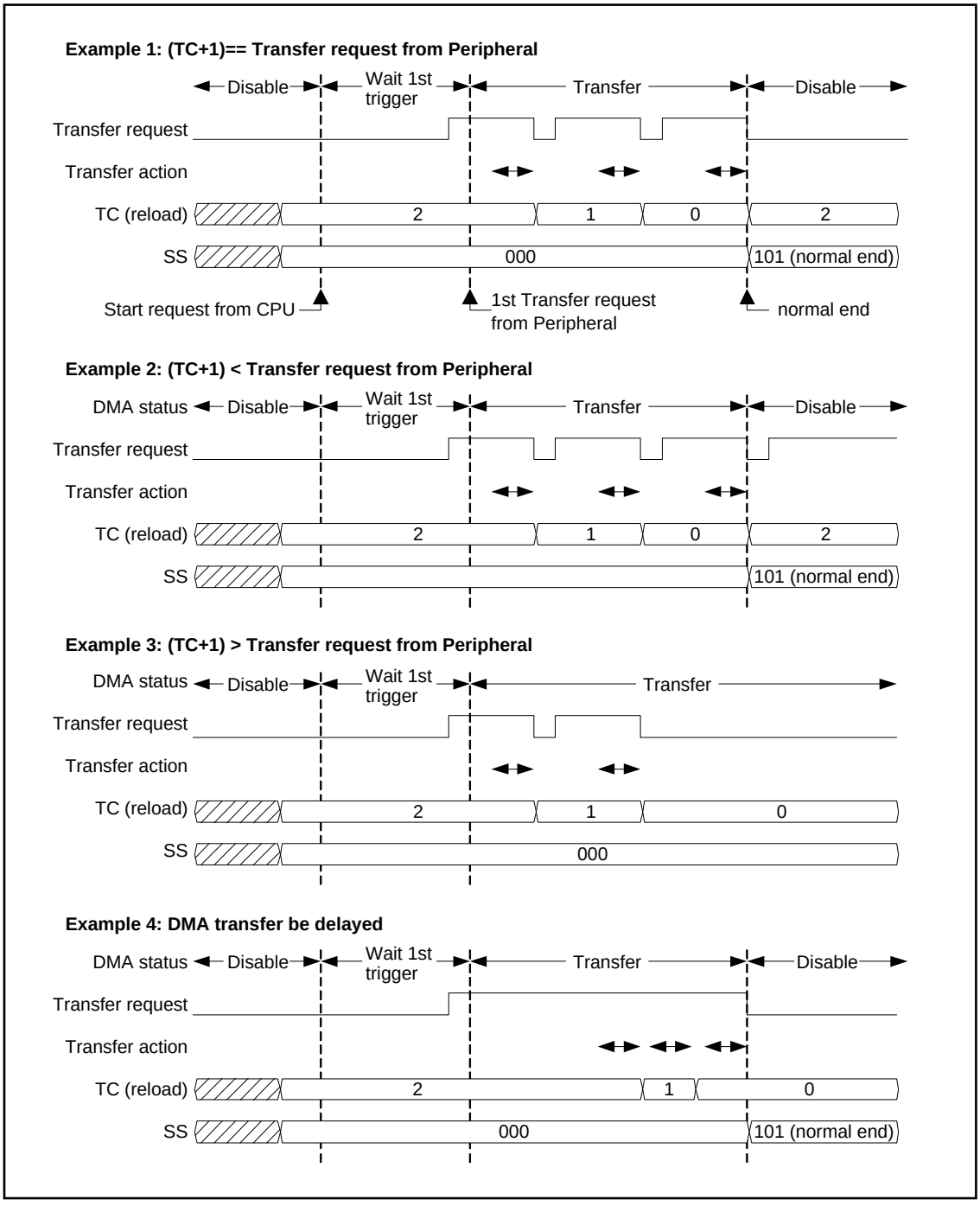
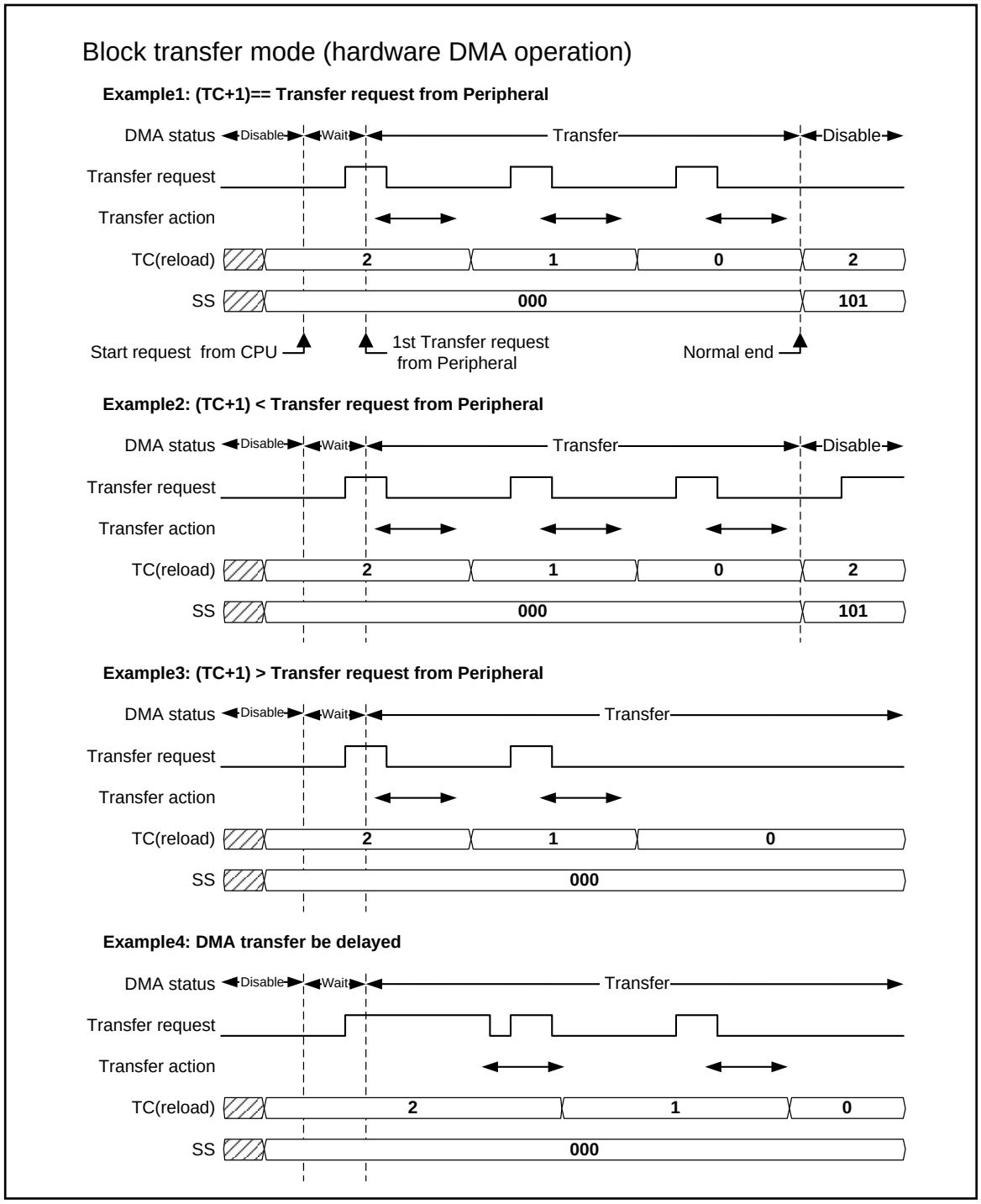


Figure 4-7 shows a case of Block transfer. In the case of Block transfer, the number of transfer requests required to complete the transfer is TC+1. Unless the number of transfer requests goes over or below the requirement, CPU does not have to intervene (Example 1 in Figure 4-7).

Figure 4-7 Operation of Hardware-Block Transfer



If the number of transfer requests generated from the Peripheral exceeds the DMAC's setting for the number of transfers, DMAC moves to Disable state after the completion of the specified number of transfers. In the Disable state, no further transfers are executed. Also, as the excessive transfer request signals are not cleared from DMAC, the asserted state continues, In this case, deassert the transfer request signal from CPU (Example 2 in Figure 4-7).
If the number of transfer requests generated from the Peripheral is smaller than DMAC's setting for the number of transfers, DMAC waits for the remaining number of transfer requests in Transfer state (Example 3 in Figure 4-7).
It is supposed that DMAC's transfer processing may be too slow to catch up with the generation interval of transfer requests from Peripheral. In the case of Block transfer, if DMAC's transfer processing is delayed from the transfer request from the Peripheral, the rising edge of the next transfer request signal during the transfer operation is ignored. Also, the transfer request signal asserted during the transfer operation is cleared from DMAC. Then, DMAC waits for the remaining transfer requests in Transfer state (Example 4 in Figure 4-7).
In the case of Burst transfer, all of the (BC+1) × (TC+1) of transfers are performed when it becomes accessible to the system bus after the first transfer request is received. The required number of transfer requests from the Peripheral is only the first one. If the number of transfer request signals generated exceeds the requirement, it is ignored in Disable state, just like Block transfer.

5. Transfer state => Disable state / Successful completion of transfer

   See Step 4 in the software transfer procedure.

6. Transfer state => Disable state / Transfer error stop

   See Step 5 in the software transfer procedure.

7. Transfer state => Disable state / End of Peripheral stop request

   The channel in Transfer state suspends its transfer processing, if the transfer stop request signal is asserted from the Peripheral. It clears EB, PB and ST and moves to Disable state. It sets "010" to SS[2:0] and gives the notification of the error stop. If interrupts have been enabled by EI, an unsuccessful transfer completion interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set during the suspension of the transfer. Attention must be paid to the SS[2:0] value, which is the same as the stop request from software.

8. Transfer state, Pause state => Disable state / Forced termination of transfer

   See Step 6 in the software transfer procedure.

9. Disable state / Post-transfer processing

   See Step 7 in the software transfer procedure.
   Normally, in the cases of stop request from Peripherals, forced termination from software and transfer error stop, the transfer request signal remains asserted, because the number of transfers processed is smaller than the number of transfer requests from the Peripheral. Instruct from CPU the Peripheral to deassert the transfer request signal. In the case of stop request from Peripherals, the transfer request signal is masked as long as the stop request signal is asserted. Also deassert the transfer stop request signal.
   Even if DMAC has successfully completed the specified number of transfers, the transfer request signal may remain asserted or may be reasserted, depending on Peripheral's settings. Attention must be paid to the possibility that this may affect the next transfer.

10. Transfer state, Pause state / Transfer pause

    See Step 8 in the software transfer procedure.

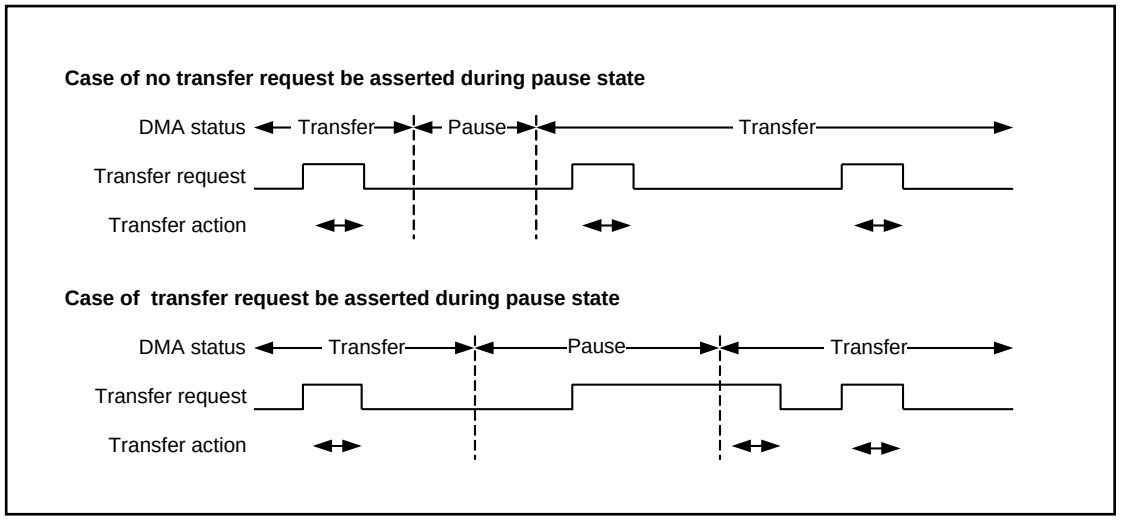
11. Pause state

    See Step 9 in the software transfer procedure.
    The channel in Pause state does not execute transfer, even if the transfer request signal from the Peripheral is asserted. It does not clear the transfer request signal either.

12. Pause state / Cancellation of transfer pause

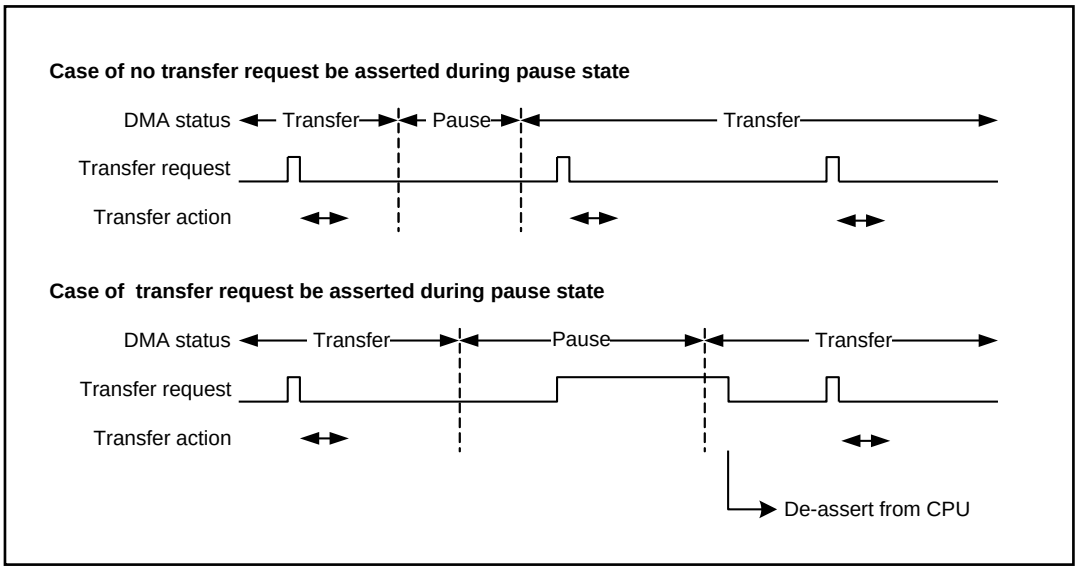
    See Step 10 in the software transfer procedure.
    When an instruction to cancel the pause is issued while it is in Pause state, it returns to Transfer state. If the transfer request signal was asserted in the previous Pause state, the operation to follow varies as shown below, depending on the transfer mode.
    In the case of Demand transfer mode, the transfer request signal remains asserted from the Pause state. Therefore, the transfer is resumed when DMAC returns to Transfer state, and the transfer request signal is cleared as normal. See Figure 4-8.

Figure 4-8 Operation of Demand Transfer in Pause State



In the case of Block transfer mode, the transfer request signal remains asserted. Even when it returns to Transfer state, the rising edge of the transfer request signal is not detected, and the transfer is not resumed. Therefore, the transfer request is ignored during Pause state. Also, the transfer request signal is not cleared from DMAC. To resume the transfer which has been put on pause, instruct from CPU the Peripheral to deassert the transfer request signal after an instruction to cancel the pause is issued to DMAC. After that, the transfer will be resumed when the next transfer request is generated from the Peripheral. In this case, attention must be paid to the difference between the number of transfer requests output from the Peripheral and the number of transfer requests received by DMAC. See Figure 4-9.

Figure 4-9 Operation of Block Transfer in Pause State



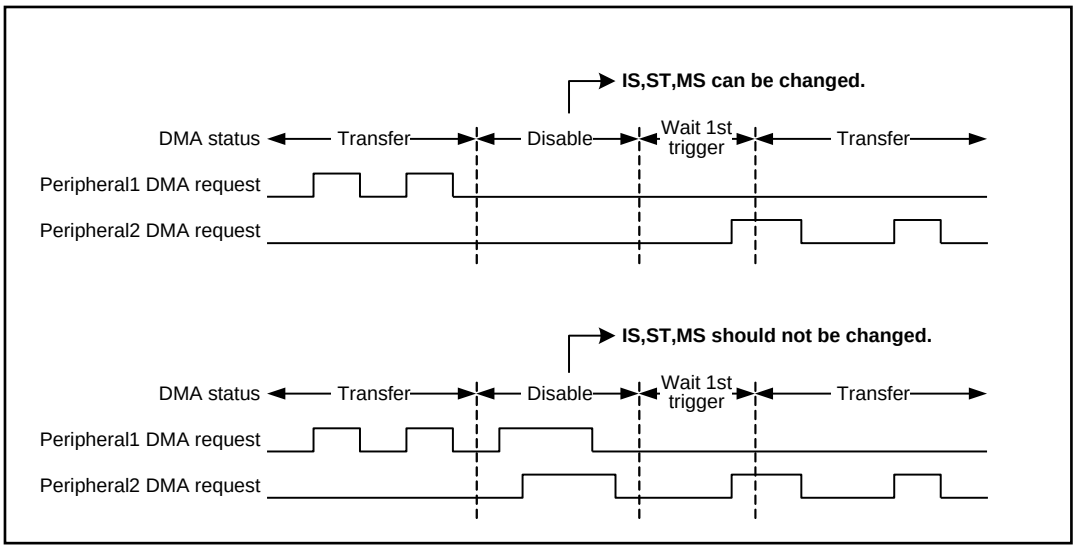
13. Operation in Disable state and Wait-1st-trigger state

See Step 11 in the software transfer procedure.

If the transfer request signal is not asserted to the channel in Disable state, the specifications of the transfer content can be changed freely (rewriting to registers DMACSA, DMACDA, DMACA[29:0], and DMACB).

If the transfer request signal is asserted or may be asserted to the channel in Disable state, the specifications of IS, ST and MS in the transfer content cannot be changed. If an attempt is made to change these settings, DMAC may perform unexpected behaviors. To change the settings of IS, ST and MS, first clear the transfer request signal to both of the Peripherals (used before and after the change) from CPU, and then always change the settings while the transfer request signal is deasserted. See Figure 4-10.

Figure 4-10 Changing IS, ST and MS Settings



The specifications of the transfer content cannot be changed to the channel in Wait-1st-trigger state from CPU

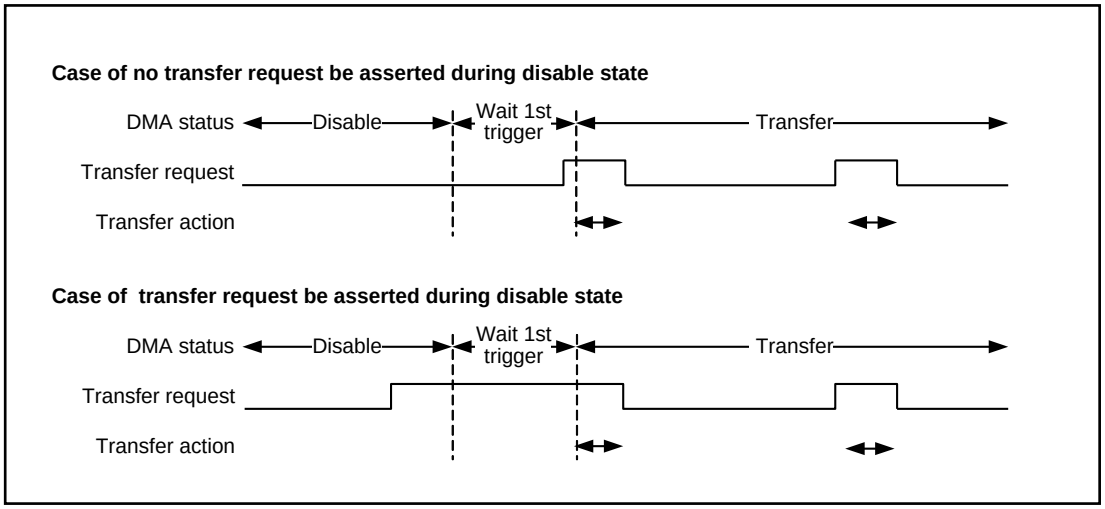
If the transfer request signal is not asserted to the channel in Wait-1st-trigger state, it moves to Disable state when CPU issues an instruction to disable individual- or all-channel operation or an instruction to put individual- or all-channel operation on pause. In this case, it is considered that the enabled transfer has been cancelled. In any case, SS does not change.

If the transfer request signal may possibly be asserted to the channel in Wait-1st-trigger state, it should be noted that DMAC has already started or completed the transfer before the attempted cancellation of the enabled transfer from CPU.

In Disable state, DMAC does not start the transfer or clear the transfer request, even if the transfer request signal is asserted. If it moves to Wait-1st-trigger state by instruction from CPU while the transfer request signal is asserted, the following operation applies (only when the settings of IS, ST and MS are not intended to be changed, as explained earlier).

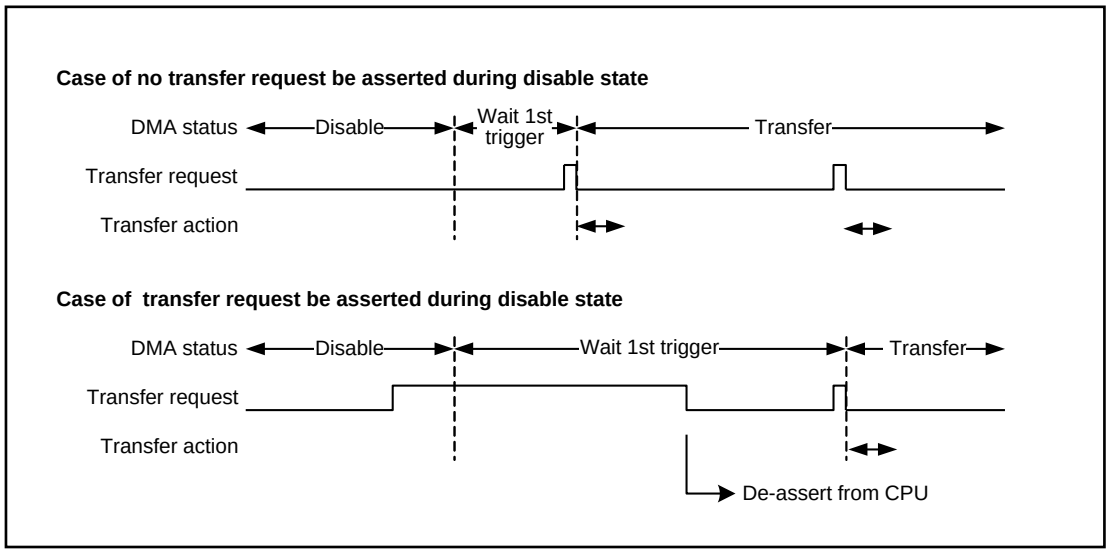
In the case of Demand transfer mode, DMAC immediately moves to Transfer state and starts the transfer, because the transfer request signal remains asserted. The transfer request signal is cleared from DMAC as normal. See Figure 4**-11**.

Figure 4-11 Operation of Demand Transfer in Disable State



In the case of Block transfer mode, the transfer request signal remains asserted. Even when it moves to Wait-1st-trigger state, the rising edge of the transfer request signal is not detected, and the transfer is not resumed. Therefore, the transfer request is ignored during Disable state. Also, the transfer request signal is not cleared from DMAC. To resume the transfer, instruct DMAC to move to Wait-1st-trigger state, and then instruct from CPU the Peripheral to deassert the transfer request signal. After that, it will move to Transfer state and the transfer will be resumed when the next transfer request is generated from the Peripheral. In this case, attention must be paid to the difference between the number of transfer requests output from the Peripheral and the number of transfer requests received by DMAC. See Figure 4-12.

Figure 4-12 Operation of Block Transfer in Disable State



-
- Additional Matter 1

  See Additional Matter 1 in the software transfer procedure.
  In the case of hardware transfer, always write 0 to ST.
- Additional Matter 2

  See Additional Matter 2 in the software transfer procedure.
- Additional Matter 3

  See Additional Matter 3 in the software transfer procedure.
- Additional Matter 4

  See Additional Matter 4 in the software transfer procedure.
- Additional Matter 5

  If the transfer request signal (interrupt signal) from the Peripheral needs to be deasserted, the following method is available. Normally, the interrupt signal from the Peripheral is the interrupt factor flag masked (logic AND) by the interrupt enable flag. The interrupt signal can be deasserted by resetting either of the flags. When the interrupt enable flag is reset and then set, the rising edge occurs to the interrupt signal. Following this procedure can notify DMAC of the transfer request for Block transfer again. For details, check the manual for each Peripheral.

4.4 DMAC Operation and Control Procedure for Hardware (EM=1) Transfer

This section explains DMAC operation and control procedure for hardware (EM=1) transfer.

Figure 4-13 Transitional Diagram of Hardware (EM=1) Transfer State

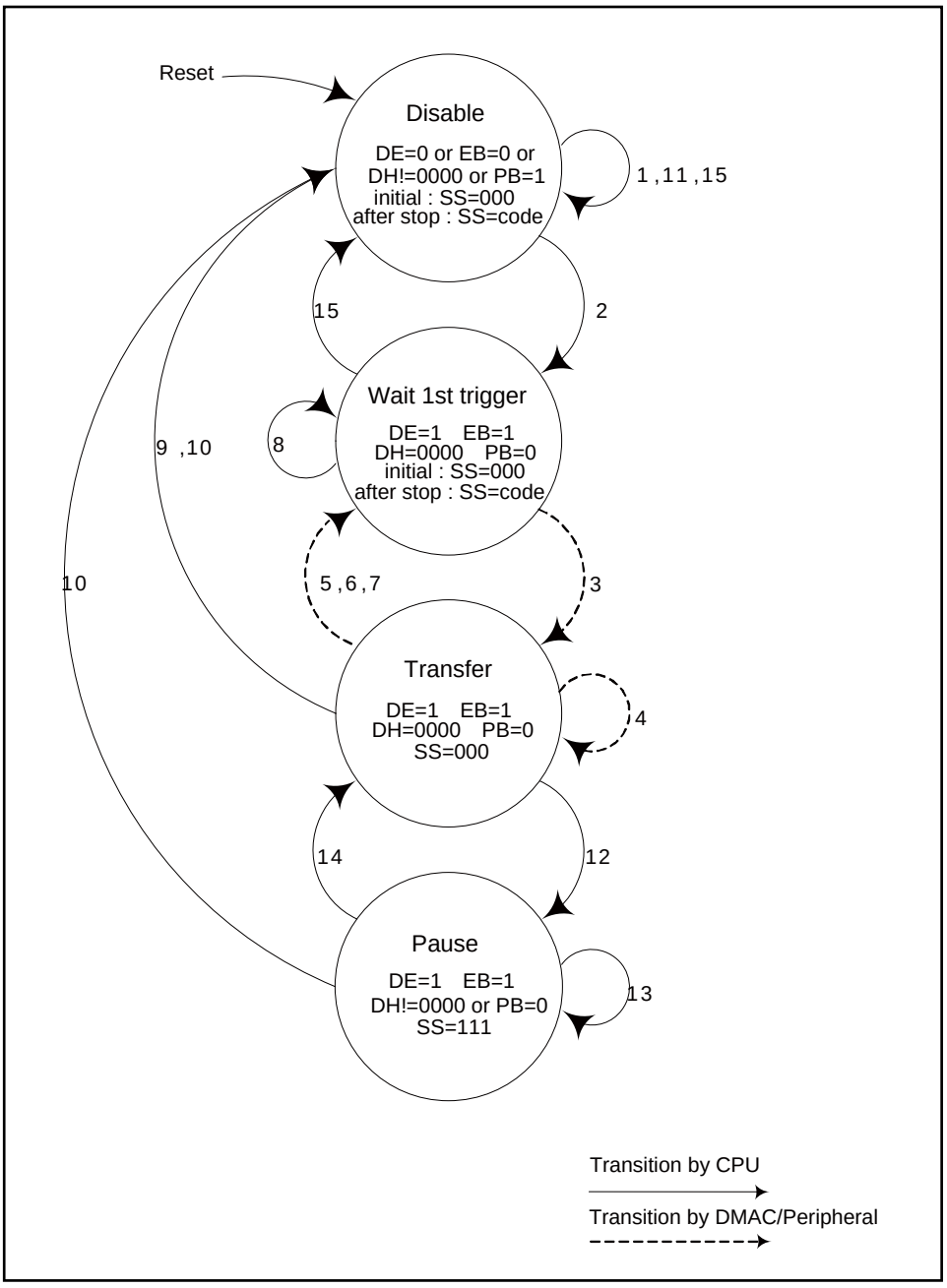


Figure 4-13 shows a transitional diagram of the states of the channel to be controlled for hardware (EM=1) transfer. The numbers next to the transitional lines in Figure 4-13 correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state

instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC/Peripheral operation.

EM (Enable bit clear mask) is a bit that masks EB clear upon the completion of transfer of the channel to be controlled. EM=1 enables the same transfer process to be repeated without giving instructions from CPU.

Description of Each State

- Disable state

  See the hardware transfer (EM=0) procedure.

- Wait-1st-trigger state

  See the hardware transfer (EM=0) procedure.

- Transfer state

  In this state, the channel to be controlled has received the first transfer request from the Peripheral. A channel in this state performs transfer operation as specified. In the case of EM=1, it moves to Wait-1st-trigger state, once all the transfer operation is completed. It also changes its state upon instruction from CPU.

- Pause state

  See the hardware transfer (EM=0) procedure.

Explanation of Control Procedure

1. Disable state / Preparation for transfer

   See Step 1 in the hardware transfer (EM=0) procedure.
   To set EM=1, set all of the reload specifications for the transfer content (RC, RS, RD) in order to prevent data transfer in an unintended address area. Also, CI is not set, because it is meaningless to generate a successful transfer completion interrupt from DMAC. EI is set to generate an unsuccessful transfer completion interrupt from DMAC.

2. Disable state => Wait-1st-trigger state / Enabling transfer

   See Step 2 in the hardware transfer (EM=0) procedure.

3. Wait-1st-trigger state / Start of transfer

   See Step 3 in the hardware transfer (EM=0) procedure.

4. Transfer state

   See Step 4 in the hardware transfer (EM=0) procedure.

5. Transfer state => Wait-1st-trigger state / Successful completion of transfer

   When transfers are successfully completed for the number of times calculated by (BC+1) × (TC+1), the channel in Transfer state does not clear EB but does clear PB and ST and moves to Wait-1st-trigger. It sets SS[2:0]=101 to provide the notification of the successful completion. As CI is not set, no successful transfer completion interrupt is generated. Since RC, RS and RD are set, the specifications of the transfer content of BC, TC, DMACSA and DMACDA are reloaded.

6. Transfer state => Wait-1st-trigger state / Transfer error end

   See Step 6 in the hardware transfer (EM=0) procedure.
   In the case of EM=1, EB is not cleared even if the transfer ends due to an error. It clears PB and ST, moves to Wait-1st-trigger state and waits for the next transfer request. Therefore, it is recommended not to use DMA transfer with EM=1 in an address area where a transfer error may occur.

7. Transfer state =>Wait-1st-trigger state /End of Peripheral stop request

   See Step 7 in the hardware transfer (EM=0) procedure.
   In the case of EM=1, EB is not cleared even if a stop request is issued from the Peripheral. It clears PB and ST and moves to Wait-1st-trigger state. Since RC, RS and RD are set, the

specifications of the transfer content of BC, TC, DMACSA and DMACDA are reloaded. As EI is set, an unsuccessful transfer completion interrupt is generated.

8. Wait-1st-trigger state / Post-transfer process

   In the case of EM=1, EB is not cleared upon the completion of the transfer. (DE=1, EB=1, DH=0000, PB=0) is set and it moves to Wait-1st-trigger state. When the next transfer request is generated from the Peripheral, therefore, the next transfer starts without an instruction from CPU. If it moves to Wait-1st-trigger state due to a stop request from the Peripheral, an unsuccessful completion interrupt occurs and that state can be confirmed. Also, the transfer request signal is masked while the stop request signal is asserted. Even if the next transfer request signal is asserted from the Peripheral, it will not be recognized and the channel to be controlled will remain in Wait-1st-trigger state, waiting for an instruction from CPU.
   In the above case, SS[2:0] is read from CPU to check the state of the transfer completion. The interrupt signal is deasserted by clearing SS[2:0] from CPU. CPU clears EB and it returns to Disable state (this operation is the operation shown in Step 15 of the hardware transfer (EM=1) procedure). The transfer request signal and the stop request signal from the Peripheral are deasserted, as shown in Step 7 of the hardware transfer (EM=0) procedure

9. Transfer state => Disable state / Completion of transfer by EM=0

   The operation can exit from the loop of Wait-1st-trigger state and Transfer state by writing EM=0 from CPU. At the timing when the transfer stops after the instruction, EB, ST and PB are cleared and the Transfer state changes to Disable state (DE=1, EB=0, DH=0000, PB=0) to successfully complete the transfer. In this case, no successful transfer completion interrupt is generated, as CI is not set.

10. Transfer state, Pause state => Disable state / Forced termination of transfer

    See Step 8 in the hardware transfer (EM=0) procedure.
    The operation can exit from the loop of Wait-1st-trigger state and Transfer state by an operation disable instruction. When an instruction to disable individual-channel operation is issued, the relevant channel moves to Disable state (DE=1, EB=0, DH=0000, PB=0) and stops the operation. When an instruction to enable all-channel operation is issued, it moves to Disable state (DE=0, EB=1, DH=0000, PB=0) and stops the operation. In the case of an instruction to disable all-channel operation, EB is not cleared either; therefore, attention must be paid.
    When the operation exits from Transfer state, an unsuccessful transfer completion interrupt occurs because it is unsuccessful completion due to the forced stop. When it exits from Wait-1st-trigger state, the enabled transfer is cancelled (this operation is the operation shown in Step 15 of the hardware transfer (EM=1) procedure).

11. Disable state / Post-transfer processing

    See Step 9 in the hardware transfer (EM=0) procedure.

12. Transfer state, Pause state / Transfer pause

    See Step 10 in the hardware transfer (EM=0) procedure.

13. Pause state

    See Step 11 in the hardware transfer (EM=0) procedure.

14. Pause state / Cancellation of transfer pause

    See Step 12 in the hardware transfer (EM=0) procedure.

15. Operation in Disable state and Wait-1st-trigger state

    See Step 13 in the hardware transfer (EM=0) procedure.
    In the case of EM=1, the Transfer state changes directly to Wait-1st-trigger state. Therefore, the specifications of the transfer content cannot be rewritten during the repeated transfer operation (rewriting the registers DMACSA, DMACDA, DMACB[31:1] and DMACA[28:0]).

-
- Additional Matter 1

  See Additional Matter 1 in the hardware transfer (EM=0) procedure.

- Additional Matter 2

  See Additional Matter 2 in the hardware transfer (EM=0) procedure.
  In the case of EM=1, Additional Matter 2 does not apply, because EB is not cleared during the transfer operation.

- Additional Matter 3

  See Additional Matter 3 in the hardware transfer (EM=0) procedure.

- Additional Matter 4

  See Additional Matter 4 in the hardware transfer (EM=0) procedure.
  The following explains what must be noted when setting interrupts from DMAC with EM=1. As the target channel does not change from Wait-1st-trigger state due to an unsuccessful completion interrupt by a stop request from the Peripheral, the interrupt signal is not deasserted until it is cleared from CPU. Similarly, as the target channel moves to Disable state due to an unsuccessful transfer completion interrupt by a stop request from software, the interrupt signal is not deasserted until it is cleared from CPU. Other successful transfer completion interrupts and unsuccessful transfer completion interrupts may be deasserted at a timing that is not intended by CPU, if the relevant channel moves to Transfer state. Therefore, attention must be paid.

- Additional Matter 5

  See Additional Matter 5 in the hardware transfer (EM=0) procedure.

5. Registers of DMAC

This section explains each register function of DMAC.

5.1. List of Registers
5.2. Entire DMAC Configuration Register (DMACR)
5.3. Configuration A Register (DMACA)
5.4. Configuration B Register (DMACB)
5.5. Transfer Source Address Register (DMACSA)
5.6. Transfer Destination Address Register (DMACDA)

5.1 List of Registers

Table 5-1 shows a list of DMAC control registers.

Table 5-1 List of DMAC Control Registers

Abbreviation	Ch. Controlled	Register name	Reference
DMACR	All	Entire DMAC configuration register	5.2
DMACA0	ch.0	Configuration A register	5.3
DMACB0		Configuration B register	5.4
DMACSA0		Transfer source address register	5.5
DMACDA0		Transfer destination address register	5.6
DMACA1	ch.1	Configuration A register	5.3
DMACB1		Configuration B register	5.4
DMACSA1		Transfer source address register	5.5
DMACDA1		Transfer destination address register	5.6
DMACA2	ch.2	Configuration A register	5.3
DMACB2		Configuration B register	5.4
DMACSA2		Transfer source address register	5.5
DMACDA2		Transfer destination address register	5.6
DMACA3	ch.3	Configuration A register	5.3
DMACB3		Configuration B register	5.4
DMACSA3		Transfer source address register	5.5
DMACDA3		Transfer destination address register	5.6
DMACA4	ch.4	Configuration A register	5.3
DMACB4		Configuration B register	5.4
DMACSA4		Transfer source address register	5.5
DMACDA4		Transfer destination address register	5.6
DMACA5	ch.5	Configuration A register	5.3
DMACB5		Configuration B register	5.4
DMACSA5		Transfer source address register	5.5
DMACDA5		Transfer destination address register	5.6
DMACA6	ch.6	Configuration A register	5.3
DMACB6		Configuration B register	5.4
DMACSA6		Transfer source address register	5.5
DMACDA6		Transfer destination address register	5.6
DMACA7	ch.7	Configuration A register	5.3
DMACB7		Configuration B register	5.4
DMACSA7		Transfer source address register	5.5
DMACDA7		Transfer destination address register	5.6

5.2 Entire DMAC Configuration Register (DMACR)

This section explains entire DMAC configuration register (DMACR).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DE	DS	Reserved	PR	DH[3:0]				Reserved							
Attribute	R/W	R/W	-	R/W	R/W				-							
Initial Value	0	0	-	0	0000				-							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															
Attribute	-															
Initial Value	-															

[bit31] DE : DMA Enable (all-channel operation enable bit)

This bit controls the enabling and disabling of transfer operations for all of the channels.
When "1" is set to this bit, the operations of all of the channels are enabled and each channel operates according to its settings.
When "0" is set to this bit, the operations of all of the channels are disabled, and no transfer is performed until "1" is set to the bit. Also, a channel in the middle of its transfer operation is forced to stop the transfer.

This bit can be used to force all of the channels that are currently performing a transfer to stop it and reset the configuration register.

bit	Function
0	Disables the operations of all of the channels. (Initial value)
1	Enables the operations of all of the channels.

[bit30] DS : DMA Stop

This bit indicates the transfer state of all of the channels.
If either of the following conditions is established during transfer operation, the bit is set to "1" by DMAC.

- When "0" is written to the DMACR:DE bit and then the transfers of all of the channels are completed.
- When a value other than "0000" is written to the DMACR:DH bit and then the transfers of all of the channels pause.

When DMACR:DE=1 and DMACR:DH=0000 are set and all of the channels become enabled to operate, this bit is set to "0" by DMAC.

Although the attribute of this bit is R/W, writing to it by CPU does not affect DMAC's operation. If, however, the DMACR register needs to be updated without affecting the state of this bit, first read from this bit and then rewrite the same value.

bit	Function
0	Clears the disabling of all-channel operation or the setting of all-channel pause. (Initial value)
1	The transfers of all of the channels have stopped due to the disabling of all-channel operation or the setting of all-channel pause.

[bit29] Reserved: Reserved bit

"0" is read out from this bit.
When writing this bit, set it to "0".

[bit28] PR : Priority Rotation

This bit controls the order of transfer priority among channels.
When this bit is set to "0", the priority order is fixed for all of the channels.
When this bit is set to "1", the priority order is determined in a rotation method for all of the channels.

bit	Function
0	Fixes the priority order. (ch.0>ch.1>ch.2>ch.3>ch.4>ch.5>ch.6>ch.7) (Initial value)
1	Applies the rotation method to the priority order.

For selection of the transfer priority order, see Section "3.5 Channel Priority Control".

[bit27:24] DH : DMA Halt (All-channel pause bit)

This bit controls the pause/cancellation of transfer operations for all of the channels.
When this bit is set to a value other than "0000", all of the channels that are currently performing a transfer are put on pause. When it is set to "0000", the transfers are resumed.

Even if a transfer request from an external/peripheral device is asserted, the channels in Pause state ignore the transfer request. In the cases of Block transfer and Burst transfer, the relevant channel does not start a transfer, even if the pause is cleared. In order to complete a transfer when a pause is set during the transfer, an additional transfer request is required after the pause is cancelled.

This bit can be used to put a transfer on pause without resetting the configuration registers of all of the channels.

bit27:24	Function
0000	Cancels the pause of transfers for all of the channels. (Initial value)
Other than 0000	Puts the transfers of all of the channels on pause.

[bit23:0] Reserved: Reserved bits

"0" is read out from these bits.
When writing these bits, set them to "0".

5.3 Configuration A Register (DMACA)

This section explains configuration A register (DMACA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	EB	PB	ST	IS[5:0]						Reserved			BC[3:0]			
Attribute	R/W	R/W	R/W	R/W						-			R/W			
Initial Value	0	0	0	000000						-			0000			

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TC[15:0]															
Attribute	R/W															
Initial Value	0x0000															

[bit31] EB : Enable bit (individual-channel operation enable bit)

This bit controls the enabling and disabling of the transfer operation of an individual channel.
When this bit is set to "1", the relevant channel is enabled to operate and waits for a trigger to start its transfer operation (the DMACR:DE must be set to "1").

If the EM bit (DMACB[0]) is not set to "1", DMAC clears this bit to "0" upon the completion of the transfer.

When this bit is set to "0", the relevant channel is disabled to operate and does not perform transfer operation until it is set to "1". Also, if it is in the middle of transfer operation, it is forced to stop the transfer. This bit can be used to force the relevant channel that is currently in transfer operation to stop it and reset the configuration register.

bit	Function
0	The operation of the relevant channel is disabled. (Initial value)
1	The operation of the relevant channel is enabled.

[bit30] PB : Pause bit (individual-channel pause bit)

This bit controls the pause/cancellation of the transfer operation of an individual channel.
When this bit is set to "1" and the relevant channel is currently in transfer operation, it puts the transfer on pause. When this bit is set to "0", it resumes the transfer.

This bit is cleared to "0", when the transfer operation of the channel is completed.

Even if a transfer request from an external/peripheral device is asserted, the channels in Pause state ignore the transfer request. In the cases of Block transfer and Burst transfer, the relevant channel does not start a transfer, even if the pause is cleared. In order to complete a transfer when a pause is set during the transfer, an additional transfer request is required after the pause is cancelled.

This bit can be used to put a transfer on pause without resetting the configuration register of the relevant channel.

bit	Function
0	Cancels the pause of the transfer of the relevant channel.
1	Puts the transfer of the relevant channel on pause.

Notes:

- *In this case of setting this bit during DMACB.RC="1", DMACA.BC and DMACA.TC must be set to reload value along with this bit by word access.*

[bit29] ST : Software Trigger

This bit is used to generate a software transfer request for an individual channel.

When this bit is set to "1", a trigger is generated by the software transfer request and the relevant channel starts its transfer. After the completion of the transfer, DMAC clears this bit to "0".
When this bit is set to "0" during the transfer, the transfer stops.

bit	Function
0	No software transfer request (Initial value)
1	Software transfer request available

[bit28:23] IS[5:0] : Input Select

These bits select the trigger for transfer requests.

When the transfer trigger is set to software request (ST=1), set the IS[5:0] bits to "000000".
When the transfer trigger is set to hardware request, specify which Peripheral's interrupt signal to be used to start transfer. Any Peripheral can be selected for all of the channels.

The hardware transfer request signal to be connected to DMAC varies depending on the product used. Check the transfer request signal to be connected in "2.2 I/O Signals of DMAC" before setting the selection.

bit28:23	Function
000000	Software (Initial value)
100000	IDREQ[0]
100001	IDREQ[1]
100010	IDREQ[2]
100011	IDREQ[3]
100100	IDREQ[4]
100101	IDREQ[5]
100110	IDREQ[6]
100111	IDREQ[7]
101000	IDREQ[8]
101001	IDREQ[9]
101010	IDREQ[10]
101011	IDREQ[11]
101100	IDREQ[12]
101101	IDREQ[13]
101110	IDREQ[14]
101111	IDREQ[15]
110000	IDREQ[16]
110001	IDREQ[17]
110010	IDREQ[18]
110011	IDREQ[19]

bit28:23	Function
110100	IDREQ[20]
110101	IDREQ[21]
110110	IDREQ[22]
110111	IDREQ[23]
111000	IDREQ[24]
111001	IDREQ[25]
111010	IDREQ[26]
111011	IDREQ[27]
111100	IDREQ[28]
111101	IDREQ[29]
111110	IDREQ[30]
111111	IDREQ[31]
Setting other than above	Setting is prohibited.

[bit22:20] Reserved: Reserved bits

"0" is read out from these bits.
When writing these bits, set them to "0".

[bit19:16] BC[3:0] : Block Count

These bits specify the number of blocks for Block/Burst transfer.

When the transfer mode is set to Demand transfer, set BC[3:0] to "0000".
Set the value "BC[3:0]=Number of blocks - 1". The maximum allowed number of blocks is 16.
The value of these bits can be read during a transfer. Normally, as one transfer source access or one transfer destination access is completed successfully, BC[3:0] is decreased by 1.

In the case of DMACB:RC=1, the value set when the transfer started is reloaded upon the completion of the transfer.
In the case of DMACB:RC=0, the value is set to "0" upon successful completion of the transfer, while the value remains the same value as set during the transfer suspension upon unsuccessful completion of the transfer.

bit19:16	Function
	Number of transfer blocks (Initial value : 0x0)

[bit15:0] TC[15:0] : Transfer Count

These bits specify the number of transfers for Block/Burst/Demand transfer.

Set the value "TC = Number of transfers - 1". The maximum allowed number of transfers is 65536.

The value of these bits can be read during a transfer. Normally, as the transfer of one block is completed, TC is decreased by 1.

In the case of DMACB:RC=1, the value set when the transfer started is reloaded upon the completion of the transfer.
In the case of DMACB:RC=0, the value is set to "0" upon successful completion of the transfer, while the

value remains the same value as set during the transfer suspension upon unsuccessful completion of the transfer.

bit15:0	Function
	Number of transfers (Initial value : 0x0000)

5.4 Configuration B Register (DMACB)

This section explains configuration B register (DMACB).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved		MS[1:0]		TW[1:0]		FS	FD	RC	RS	RD	EI	CI	SS[2:0]		
Attribute	R/W		R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
Initial Value	00		00		00		0	0	0	0	0	0	0	000		

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															EM
Attribute	R/W															R/W
Initial Value	000000000000000															0

[bit31:30] Reserved: Reserved bits

"0" is read out from these bits.

When writing these bits, set them to "0".

[bit29:28] MS[1:0] : Mode Select

These bits select the transfer mode.

bit	Function
00	Block transfer mode (Initial value)
01	Burst transfer mode
10	Demand transfer mode
11	Reserved

[bit27:26] TW[1:0] : Transfer Width

These bits specify the bit width of transfer data.

bit	Function
00	Byte (8 bits) (Initial value)
01	Half-word (16 bits)
10	Word (32 bits)
11	Reserved

[bit25] FS : Fixed Source

This bit specifies whether to increment or fix the transfer source address.

bit	Function
0	Increments the transfer source address according to TW[1:0]. (Initial value)
1	Fixes the transfer source address.

[bit24] FD : Fixed Destination

This bit specifies whether to increment or fix the transfer destination address.

bit	Function
0	Increments the transfer destination address according to TW[1:0]. (Initial value)
1	Fixes the transfer destination address.

[bit23] RC : Reload Count (BC/TC reload)

This bit controls the reload function of BC[3:0] and TC[15:0].
When this bit is set to "1", the value set when the transfer started is reloaded to BC[3:0] and TC[15:0] upon completion of the transfer.

bit	Function
0	Disables the reload function of BC/TC. (Initial value)
1	Enables the reload function of BC/TC.

[bit22] RS : Reload Source

This bit controls the reload function of the transfer source address.
When this bit is set to "1", the value set when the transfer started is reloaded to DMACSA upon completion of the transfer.

bit	Function
0	Disables the reload function of the transfer source address. (Initial value)
1	Enables the reload function of the transfer source address.

[bit21] RD : Reload Destination

This bit controls the reload function of the transfer destination address (DMACDA).
When this bit is set to "1", the value set when the transfer started is reloaded to DMACDA upon completion of the transfer.

bit	Function
0	Disables the reload function of the transfer destination address. (Initial value)
1	Enables the reload function of the transfer destination address.

[bit20] EI :Error Interrupt (unsuccessful transfer completion interrupt enable)

This bit enables or disables the notification of an interrupt when a transfer has been unsuccessfully completed.
When this bit is set to "1", an interrupt is issued if SS[2:0] is in the following status upon completion of the transfer.

- Address overflow
- Stop by transfer stop request from a Peripheral, or the disabling of transfer by the EB/DE bit
- Transfer source access error
- Transfer destination access error

bit	Function
0	Disables an interrupt to be issued upon unsuccessful completion of transfer. (Initial value)
1	Enables an interrupt to be issued upon unsuccessful completion of transfer.

[bit19] CI :Completion Interrupt : (successful transfer completion interrupt enable)

This bit enables or disables the notification of an interrupt when a transfer has been successfully completed.

When this bit is set to "1", an interrupt is generated, if SS[2:0] is set to successful completion upon completion of the transfer.

bit	Function
0	Disables an interrupt to be issued upon successful completion of transfer. (Initial value)
1	Enables an interrupt to be issued upon successful completion of transfer.

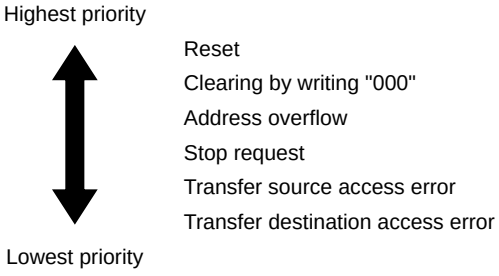
[bit18:16] SS[2:0] : Stop Status (stop status notification)

These bits represent a code that indicates the stop status or completion status of a transfer.

The following table shows the available codes.
If a successful transfer completion interrupt or unsuccessful transfer completion interrupt is issued, the interrupt signal is deasserted by writing "000" to these bits.

bit18:16	Description
000	Initial value
001	Termination by transfer error (address overflow)
010	Termination by transfer stop request (stop by transfer stop request for Peripheral or the disabling of transfer by the EB/DE bit)
011	Termination by transfer error (transfer source access error)
100	Termination by transfer error (transfer destination access error)
101	Successful transfer completion
110	Reserved
111	Transfer on pause

If various errors occur simultaneously, the termination code is indicated according to the following priority.



[bit15:1] Reserved: Reserved bits

"0" is read out from these bits.
When writing these bits, set them to "0".

[bit0] EM : Enable bit Mask (EB bit clear mask)

This bit is used to mask the clear of the EB bit (DMACA[31]) from DMAC upon completion of the transfer.

In the case of EM=0, DMAC clears the EB bit (DMACA[31]) to "0" upon completion of the transfer.
In the case of EM=1, it does not clear the EB bit upon completion of the transfer. This function allows transfers to be repeated without instruction from CPU.

This function can only be used for hardware transfer. To use the function, enable the reload function of RC, RS and RD bits.

bit	Function
0	Clears DMACA:EB to 0 upon completion of the transfer. (Initial value)
1	Does not clear DMACA:EB upon completion of the transfer.

5.5 Transfer Source Address Register (DMACSA)

This section explains transfer source address register (DMACSA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DMACSA[31:16]															
Attribute	R/W															
Initial Value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DMACSA[15:0]															
Attribute	R/W															
Initial Value	0x0000															

[bit31:0] DMACSA[31:0] : DMAC Source Address

These bits specify the transfer start address of the transfer source.

It is not possible to set unaligned address to transfer data width (TW[1:0]). The value of these bits can be read during the transfer.

In the case of DMACB:FS=1, the transfer source address is set to a fixed value and no change occurs.

In the cases of DMACB:FS=0 and DMACB:RS=0, the value is incremented according to TW[1:0].
Upon successful transfer completion, it is the next address after the transfer completion address.
Upon unsuccessful transfer completion, it is the value set during the suspension.

In the cases of DMACB:FS=0 and DMACB:RS=1, it is incremented according to TW[1:0] during the transfer. Upon completion of the transfer, the value set when the transfer started is reloaded.

	Function
bit31:0	Specifies the transfer source address from which the transfer starts. (Initial value: 0x00000000)

5.6 Transfer Destination Address Register (DMACDA)

This section explains transfer destination address register (DMACDA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DMACDA[31:16]															
Attribute	R/W															
Initial Value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DMACDA[15:0]															
Attribute	R/W															
Initial Value	0x0000															

[bit31:0] DMACDA[31:0] : DMAC Destination Address

These bits specify the transfer start address of the transfer destination.
It is not possible to set unaligned address to transfer data width (TW[1:0]). The value of these bits can be read during the transfer.

In the case of DMACB:FD=1, the transfer destination address is set to a fixed value and no change occurs.

In the cases of DMACB:FD=0 and DMACB:RD=0, the value is incremented according to TW[1:0].
Upon successful transfer completion, it is the next address after the transfer completion address.
Upon unsuccessful transfer completion, it is the value set during the suspension.

In the cases of DMACB:FD=0 and DMACB:RD=1, it is incremented according to TW[1:0] during the transfer.
Upon completion of the transfer, the value set when the transfer started is reloaded.

	Function
bit31:0	Transfer destination address from which DMA transfer starts (Initial value: 0x00000000)

6. Usage Precautions

This section explains the precautions on using DMAC.

Precautions on Register Setting

When setting DMAC register, please note the following matters.

- The DMACR, DMACA, DMACB, DMACSA and DMACDA registers can be accessed by byte, half-word and word.
- The register address in DMAC cannot be set to the DMACSA or DMACDA register.
- Channel setting registers cannot be changed during DMA transfer, except the DE/DH bits of DMACR, the EB/PB bits of DMACA and the EM bit of DMACB.

Precautions on Stop and Timer Mode Transition

When transiting to Stop mode and Timer mode, make sure to stop the operation of all channels of the DMAC and confirm the stop of the DMAC by DS flag. If the transition is made to Stop mode and Timer mode while DMAC is operating, an unexpected operation can be executed when returning to Run mode.

CHAPTER 10-1: I/O Port

This chapter explains the I/O port.

1. Overview
2. Configuration, Block Diagram, and Operation
3. Setup Procedure Example
4. Registers
5. Usage Precautions

CODE: 9BFGPIO-FM0-E03.0

1. Overview

This section provides an overview of the I/O port.

The I/O port of this series provides the following features.

- The I/O port of this series shares the following functions.
  - GPIO

    General-purpose I/O ports, which can read an input level and set an output level from the CPU.
  - Fast GPIO

    Fast GPIO, which can read an input level and set an output level from the CPU by 1 cycle. For the details, refer to the "CHAPTER: Fast GPIO"
  - Peripheral input/output

    Digital input/output signal ports of peripheral functions.
  - Special I/O ports
    - Analog input port
      An analog input port of an A/D converter and LCD controller.
    - Analog output port
      An analog output port of a D/A converter and LCD controller.
    - USB port
    - Oscillation port
- The followings settings can be made for each pin.
  - You can set whether the I/O port will be used as a GPIO, a digital pin of peripheral functions, or a special pin.
  - You can set whether the I/O port will be used as an input port or an output port.
  - You can enable or disable pull-up.
  - Peripheral functions are assigned to two or more I/O ports with input/output of the same function. You can set to which I/O port the function can be allocated (relocation function).
  - By setting registers, you can set the I/O port to Hi-Z status while the CPU is in standby mode.

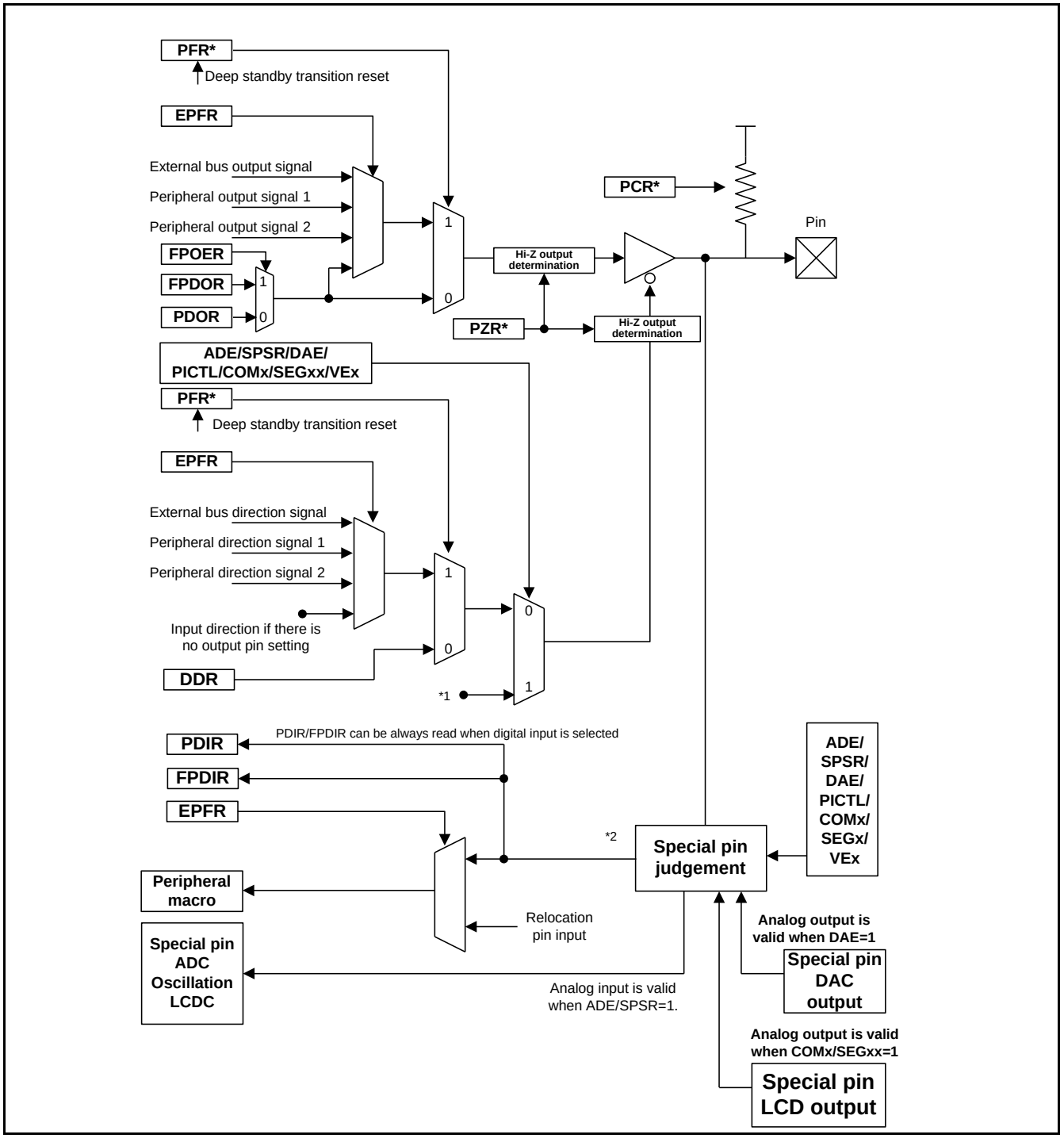
2. Configuration, Block Diagram, and Operation

This section explains the configuration, block diagram, and operation of the I/O port.

2.1 Configuration of the I/O Port

By setting registers of the I/O port, select Input/Output direction and select GPIO/peripheral. Figure 2-1 shows the details of the I/O port.

Figure 2-1 Block Diagram of the I/O Port



*1: When one of the followings is set, I/O port is set to input direction.

- ADE/SPSR=1
- DAE=1
- PICTL=0
- PICTL=1 and COMx/SEGxx=1
- VEx=1

*2: When one of the followings is set, the input value is fixed to "0".

Otherwise, the pin is set as the digital input pin.

- ADE/SPSR=1
- DAE=1
- PICTL=0
- PICTL=1 and COMx/SEGxx=1
- VEx=1

Notes:

- *For some products, 5V tolerant I/O does not have a pull-up resistor.*
- *USB pin does not have pull-up resistor.*
- *If it does not have a pull-up resistor, the PCR register setting is null.*
- *PZR register function is implemented only in some specific pins.*
  Only pins described as "PZR register control is enabled" in the remarks column of "I/O CIRCUIT TYPE" of the Data Sheet can control this feature.
- *For details of DAE bit, "5.1. D/A Control Register (DACR)" in "10-bit D/A CONVERTER" in "Analog Macro Part".*
- *For details of PICTL/COMx/SEGxx/VEx bit, "5.3 LCDC Control Register 3 (LCDCC3)", "5.5 LCDC COM Output Enable Register (LCDC_COMEN)" and "5.6 LCDC SEG Output Enable Register 1/2 (LCDC_SEG1/2)" of "LCD CONTROLLER" in "Analog Macro Part".*
- *FPDIR/FPDOR/FPOER are registers for Fast GPIO. For the details, refer to the "CHAPTER: Fast GPIO".*

Table 2-1 describes register function.

- The PFR, DDR, PDIR, PDOR, and PCR register have 1-bit control register for each I/O port and select a function for the I/O port.
- The ADE register has 1-bit control register for each I/O port which doubles as an analog input pin and selects a function for the I/O port.
- The SPSR register selects a function for the I/O port as a USB pin or an oscillation pin.
- The EPFR register has control register for each I/O pin of peripheral functions and selects to which I/O port an I/O pin of peripheral functions will be relocated.
- PZR register sets open drain control in pseudo mode by the Hi-Zing I/O port when outputting the High level of a particular pin.

Table 2-1 Register Function Descriptions

Register Name	Function Description
ADE	A register to set whether the I/O port will be used as a special pin (an analog input pin) or a digital input/output pin.
SPSR	A register to set whether the I/O port will be used as a special pin (USB or oscillation) or a digital input/output pin.
PFR	A register to set whether the I/O port will be used as an input/output pin of GPIO function or an input/output pin of peripheral functions.
PCR	A register to set whether a pull-up resistor of the I/O port will be connected or disconnected if the I/O port is used as a digital input pin or a digital bidirectional pin.
DDR	A register to set whether the I/O port will be used as an input pin or an output pin if the I/O port is used as a GPIO function pin. Note: If a pin is selected as an I/O pin of peripheral functions, a setting value is invalid.
PDIR	A register to read the level status of the I/O port. − If the I/O port is used as a digital input pin, it reads input level. − If the I/O port is used as a digital output pin, it reads output level. − If the I/O port is used as an analog input pin, it always reads "0".
PDOR	A register to set output level if the I/O port is used as an output pin of GPIO function. − When "0" is set, it outputs Low level. − When "1" is set, it outputs High level. Note: If a pin is selected as GPIO input or input/output of peripheral functions, a setting value is invalid.
EPFR	A register to select a function for an input/output of peripheral functions and set relocation function. − Setting a peripheral output pin It sets whether to produce output for the I/O port or not. In addition, it can also set to which I/O port a pin of peripheral functions will be relocated for each pin. − Setting a peripheral input pin It can set to which I/O port a pin of peripheral functions will be relocated for each pin. − Setting a peripheral bidirectional pin It can set to which I/O port a pin of peripheral functions will be relocated for each pin
PZR	This register sets open the drain control of the I/O port. − Set the I/O port to Low output when the I/O port is outputting Low level (pull-up disconnection regardless of PCR setting value) − Set open drain control in pseudo mode by setting the I/O port on Hi-Z status when the I/O port outputs High level (pull-up disconnection regardless of PCR setting value) − Set the I/O port on Hi-Z status when the I/O port is used for input (pull-up disconnection regardless of PCR setting value) Note: This function is implemented only in some specific pins. Only pins described as "PZR register control is enabled" in the remarks column of "I/O CIRCUIT TYPE" of the Data Sheet can control the open drain.

Table 2-2 lists pin functions which availability depends on selected I/O port functions and register setting values.

Table 2-2 I/O Port Functions and Register Setting Values

I/O Port Function		ADE/ SPSR/ DAE/ COMx/ SEGx/ VEx	PFR	DDR	PZR	PCR	EPFR
Available Main Function	**Available Sub Function**						
Special pin (Analog input, Analog output, USB, Oscillation)	N/A	1	-	-	-	Disconnect	*0
GPIO function input pin	Peripheral function input pin*5	0	0	0	0	Valid	*1
				0	1	Disconnect	
GPIO function output pin	GPIO function input pin (FB) Peripheral function input pin (FB)			1	0	Disconnect	
				1	1	Disconnect	
Peripheral function output pin	GPIO function input pin (FB) Peripheral function input pin (FB)		1	-	0	Disconnect	*2
					1	Disconnect	
Peripheral function bidirectional pin	GPIO function input pin (FB) Peripheral function input pin (FB)				0	Valid	*3
					1	Disconnect	
Peripheral function input pin	GPIO function input pin				0	Valid	*4
					1	Disconnect	

Legends

-: Indicates that a register setting value does not affect pin functions.

Valid: Indicates that a pull-up resistor is disconnected if PCR register value is 0.
Indicates that a pull-up resistor is connected if PCR register value is 1.

Disconnect: Indicates that a pull-up resistor is disconnected regardless of PCR register value.

(FB): Indicates that an output signal of the I/O port provides feedback and the level of the I/O port can be read from PDIR. The signal can be also used as input for peripheral functions.

*0: If the input pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the output pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the bidirectional pin of peripheral functions is selected for the I/O port, the setting is invalid.

*1: If the input pin of peripheral functions is selected for the I/O port, the setting is valid.
If the output pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the bidirectional pin of peripheral functions is selected for the I/O port, the setting is invalid.

*2: Indicates that the output pin of peripheral functions is selected for the I/O port.

*3: Indicates that the bidirectional pin of peripheral functions is selected for the I/O port.

*4: Indicates that neither the output pin nor the bidirectional pin of peripheral functions is selected for the I/O port.

*5: When NMIX pin is used, set NMIS="1" and PFR="1".

2.2 Initially Selected Functions for the I/O Port

Table 2-3 describes initially selected functions for each I/O port after reset is released.

Table 2-3 Initially Selected Functions for Each I/O Port after Reset Is Released

No	Pin	Initially Selected Function
1	SWCLK, SWDIO	Serial Wire Debug (SWD) pin is selected. Pull-up is enabled.
2	ANxx	Can be used as an analog input pin. Digital input is cut off and "0" is input.
3	X0,X1,X0A, X1A	Can be used as an oscillation pin. Digital input is cut off and "0" is input.
4	All GPIO pins other than the above pins	Digital input. Output is Hi-Z.

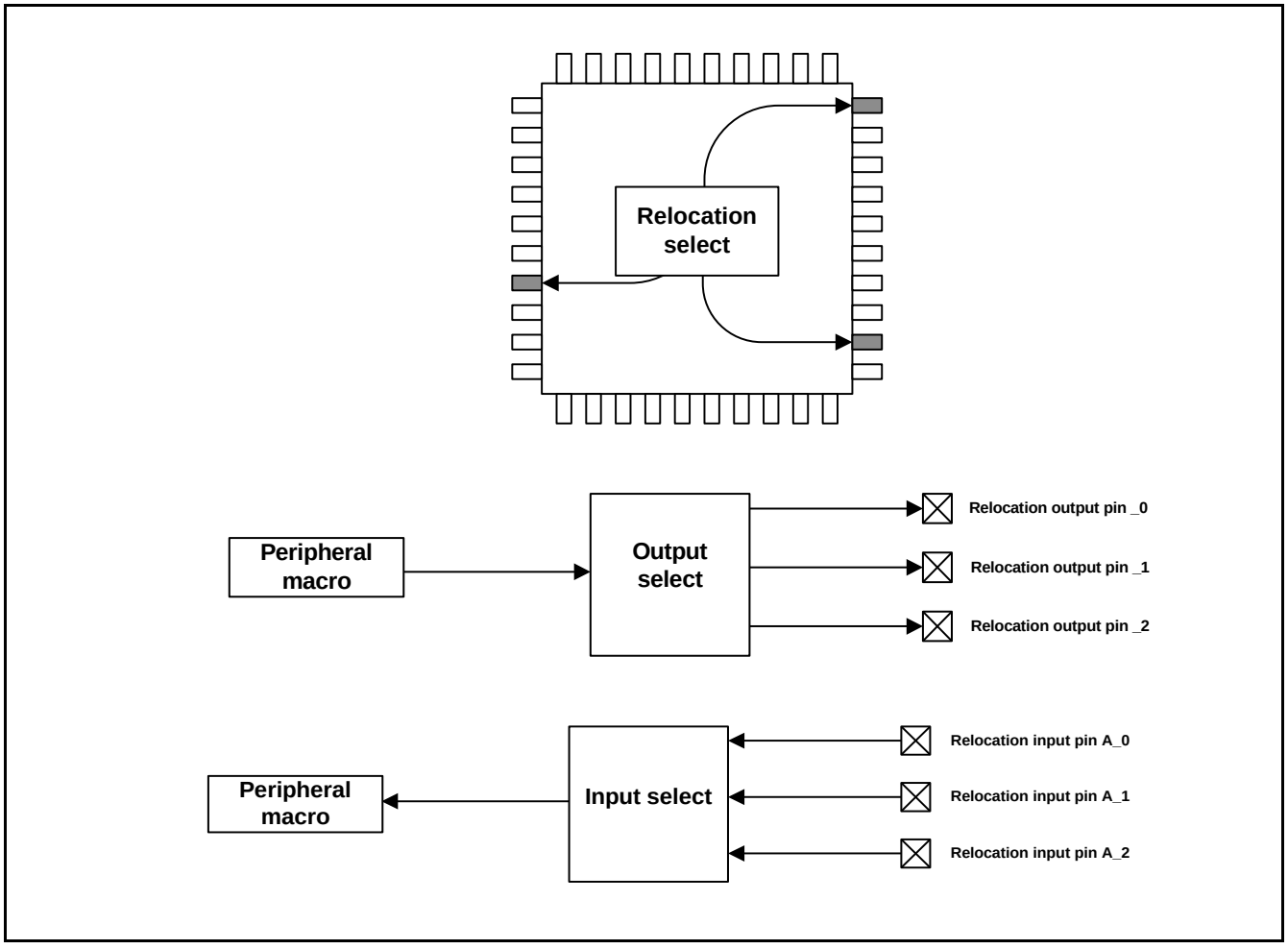
Note: For the status of pins other than GPIO (MD pins, a reset pin), see "Data Sheet" of the product used.

All the output selection values of EPFR during reset are "no output".

2.3 Relocation Function

- Some input/output of peripheral functions have more than one pin (relocation pin).
  One of the pins can be selected by setting EPFR. Figure 2-2 show the schematic view of relocation function.

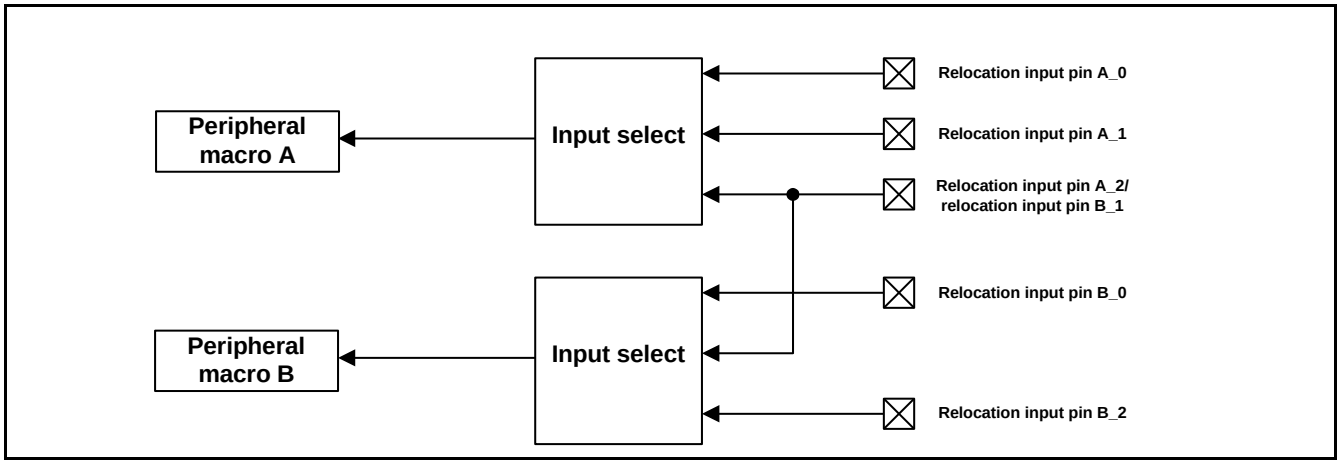
Figure 2-2 Schematic View of Relocation Function



Note: Which peripheral function is allocated to which pin depends on products.
See the pin function list of "Data Sheet" of the product used.

- Even if the input of one I/O port is connected to two or more peripheral functions, all peripheral inputs can be used by setting EPFR. For example, in Figure 2-3, by selecting input for both "Relocation input pin A_2" and "Relocation input pin B_1", simultaneous usage is possible. In this way, it is possible to use external interrupt and a multi-function serial input pin shared by one I/O port simultaneously.

Figure 2-3 Multiple Peripheral Inputs



- Even if an I/O pin is set as output, it can work as an input pin because input is not masked. For example, timer output can be used as external interrupt input which is shared.

2.4 Fixed Priority of EPFR Outputs

Only one output pin function among two or more outputs is allocated to one I/O port.
By setting the EPFR register, if more than one output is set, fixed priority is applied and output pins are selected. Figure 2-4 shows output pins and fixed priority.

Figure 2-4 Output Pins and Fixed Priority

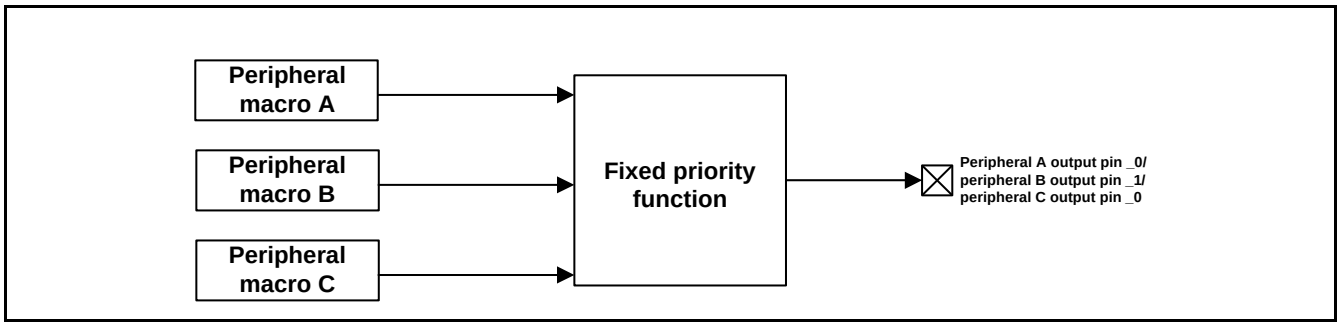


Table 2-4 describes the fixed priority of EPFR.

Table 2-4 Fixed Priority of EPFR

Order of Priority	Peripheral Function	Applied Pin
Priority Higher	Special input	Serial Wire Debug input, NMI input
↓	Serial Wire Debug	I/O pin
↓	HDMI-CEC	I/O pin*2
↓	I2C SLAVE	I/O pin
↓	USB (HCONX)	Output pin
↓	MFSI2S	Output pin
↓	Multi-function serial	Output pin, I/O pin*1
↓	Base timer output	I/O pin
↓	Multi-function timer	Output pin
↓	Smart Card	Output pin
↓	Built-in high-speed CR-oscillation clock output	Output pin
↓	RTCCO Output	Output pin
Priority Lower	SUBOUT Output	Output pin

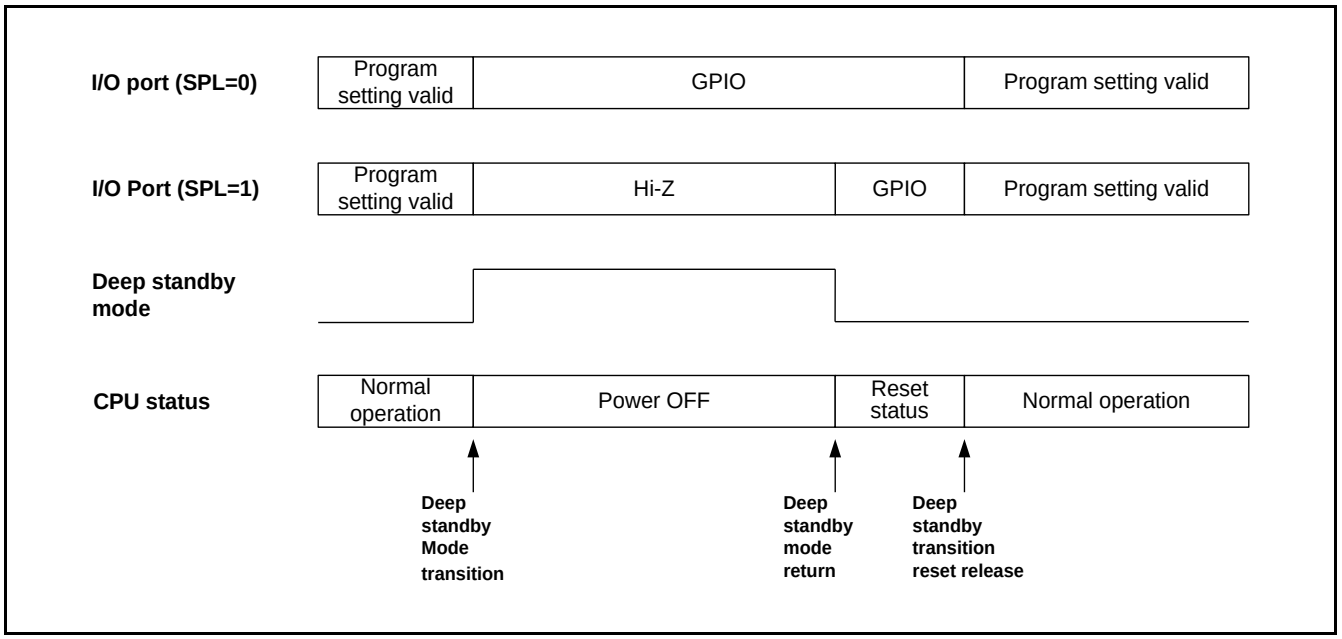
Note: The fixed priority is only applicable when "output" is set for more than one function. In case of "input", there is no fixed priority.
However, "Special input" has a higher priority than any other "output" setting. When "Special input" is set, the "output" setting allocated to the same port is invalid.
*1: The priority is SOT(serial data output)> SCS(chip select output) in TYPE1-M0+ product.
*2: In TYPE3-M0+ product, HDMI-CEC pin is not controlled by EPFR register.

- Due to output setting on the lower part of the priority, the EPFR register always includes "no output" setting.
- If you are going to use a pin as an external input pin of peripheral functions, disable all shared output settings. If every output of a pin is not selected by the EPFR register, the pin works as an external input pin.

2.5 Operation in Deep Standby Mode (TYPE2-M0+)

GPIO function is selected in deep standby mode. Figure 2-5 shows I/O port operation in deep standby mode.

Figure 2-5 I/O Port Operation in Deep Standby Mode (TYPE2-M0+)



Note:

- *GPIO Function is not selected in a few ports. For the state of each pin in deep standby mode, refer to the pin state table in the "Data Sheet" of the product used.*

2.6 Operation in Deep Standby Mode (TYPE3-M0+)

At the beginning of Deep Standby mode, GPIO is selected as I/O port function. After that, the state of I/O port is kept in the latch circuit included in I/O circuit block by CHOLDX.

Figure 2-6 shows an overview of I/O circuit block.

Figure 2-6 I/O Circuit Block (TYPE3-M0+)

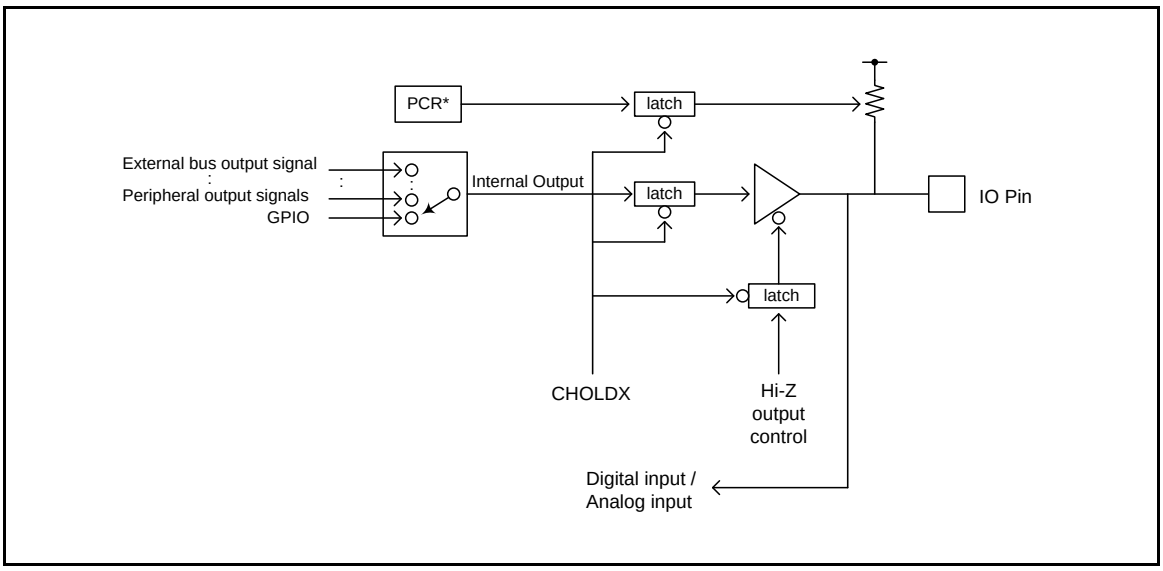


Figure 2-7 shows the I/O port operation in Deep Standby Mode in case of WIOLC_CTL:CONTX=1.In this case, the state of I/O port becomes the state of GPIO function in duration of Deep Standby Mode. After returning from Deep Standby Mode, the state of I/O port is initialized as shown in Table 2-3.

Figure 2-7 I/O Port Operation in Deep Standby Mode (CONTX=1,TYPE3-M0+)

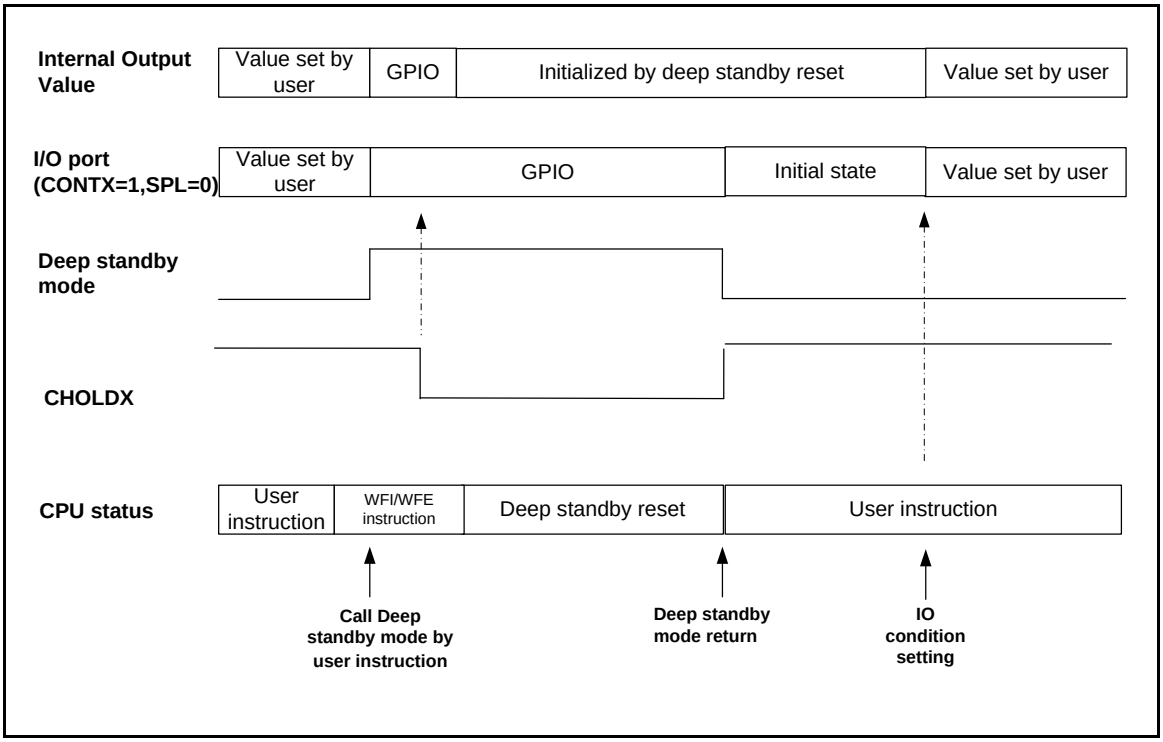
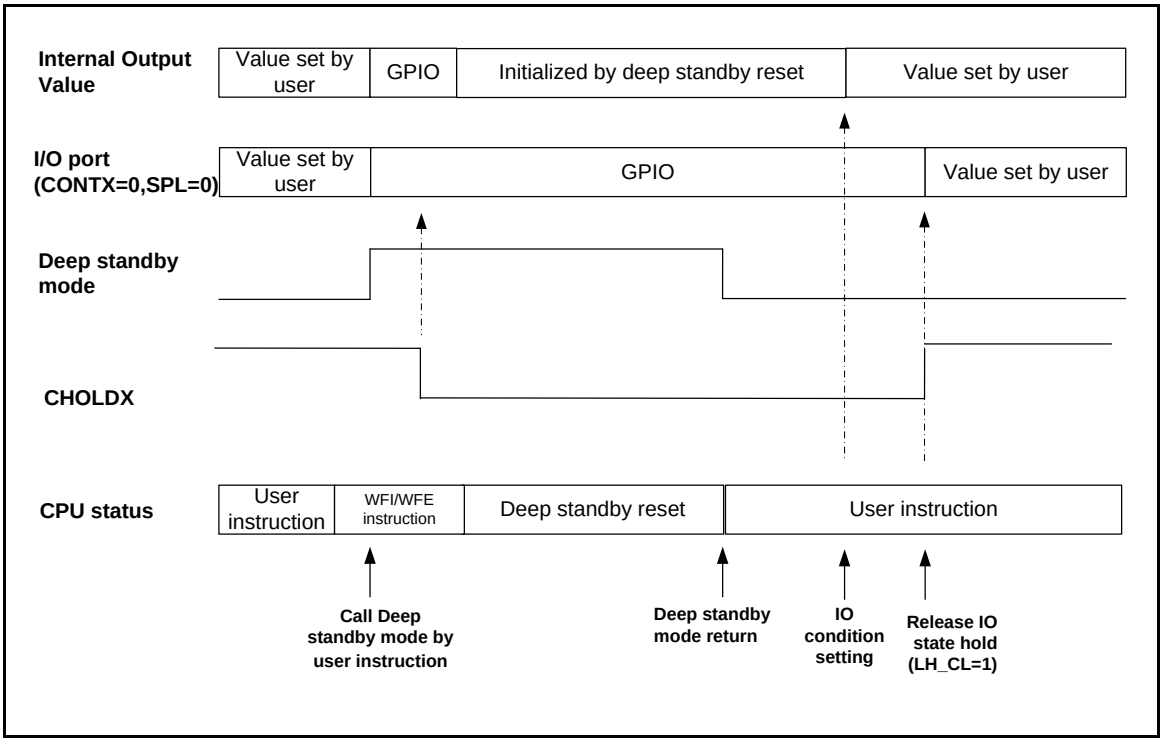


Figure 2-8 shows the I/O port operation in Deep Standby Mode in case of WIOLC_CTL:CONTX=0.

Figure 2-8 I/O Port Operation in Deep Standby Mode (CONTX=0,TYPE3-M0+)



In this case, the state of I/O port becomes GPIO state in duration of Deep Standby Mode. After returning from Deep Standby Mode, the state of I/O port is kept as it is until releasing the I/O latch. All control registers for GPIO are initialized by deep standby transition reset. The CPU needs to do initial setting to control registers for GPIO before releasing the I/O latch. After that, the CPU releases the I/O latch. This can be done by writing 1 to WIOLC_CTL:LH_CL. Changing the control registers for GPIO before releasing the I/O latch does not affect the state of I/O port.

For detail of WIOLC_CTL register, see “6-1 Low Power Consumption Mode”.

Note:

- *GPIO Function is not selected in a few ports. For the state of each pin in deep standby mode, refer to the pin state table in the "Data Sheet" of the product used.*
- *If Fast-GPIO function is used, when deep standby transition, Fast-GPIO setting is applied to the state of I/O port instead of GPIO setting.*
- *When using deep standby mode, write “00” to the MAINXC bit and SUBXC bit before setting these pins to suitable pin functions (clock oscillation pin or digital I/O pin). This is required only once after power-on. Please note that the pin state may become unstable instantaneously after writing “00” to these bits.*

2.7 Deep Standby Transition Reset

Table 2-5 shows which register is not initialized or initialized by Deep standby transition reset.

Table 2-5 Deep Standby transition Reset

Abbreviation	TYPE2-M0+	TYPE3-M0+
PFRx	Initialized *1 *2	Initialized *3
PCRx	Not initialized	Initialized after the I/O port is latched
DDRx	Not initialized	Initialized after the I/O port is latched
PDIRx	Not initialized	Initialized after the I/O port is latched
PDORx	Not initialized	Initialized after the I/O port is latched
ADE	Not initialized	Initialized after the I/O port is latched
EPFRx	Not initialized	Initialized after the I/O port is latched
SPSR	Not initialized	Initialized after the I/O port is latched
PZRx	Not initialized	Initialized after the I/O port is latched
LVDIE	Not initialized	N/A
FPDIRx, M_FPDIRx *4	Not initialized	Initialized after the I/O port is latched
FPDORx, M_FPDORx *4	Not initialized	Initialized after the I/O port is latched
FPOERx *4	Not initialized	Initialized after the I/O port is latched

*1: bit[4:0] of PFR0 register is not initialized.

*2: following bits are not initialized in each condition

bit11 of PFR0 register is not initialized when CEC0_1 is used.

bit0 of PFR6 register is not initialized when CEC1_0 is used.

bit12 of PFR4 register is not initialized when CEC0_0 is used.

bit6 of PFR5 register is not initialized when CEC1_1 is used.

*3 bit[1], bit[3] of PFR0 register is initialized after the I/O port is latched

Other than above register is initialized before the I/O port is latched

*4 These register are explained in chapter to “Fast GPIO”.

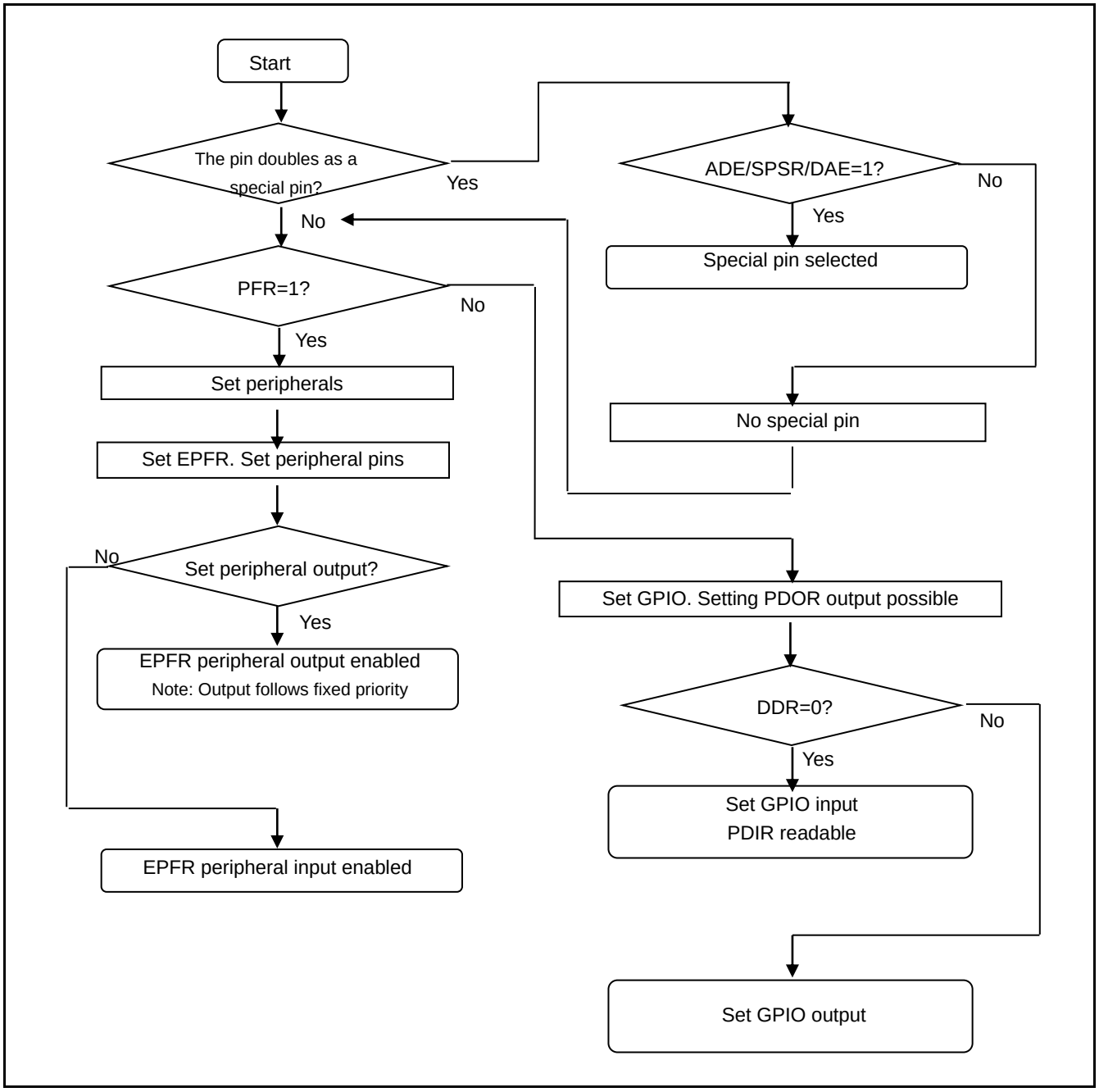
3. Setup Procedure Example

This section explains a procedure example of setting up the I/O port.

3.1 Setup of the I/O Port

By setting registers of the I/O port, select I/O direction and select GPIO/peripheral.
Figure 3-1 shows a setup procedure example.

Figure 3-1 Setup Procedure Example of the I/O Port



3.2 How to use I/O State Hold function (TYPE3-M0+)

Figure 3-2 shows how to enable/disable I/O state hold function. This setting should be done before moving to Deep Standby mode.

Figure 3-2 I/O State Hold function Setting Procedure Example1 (TYPE3-M0+)

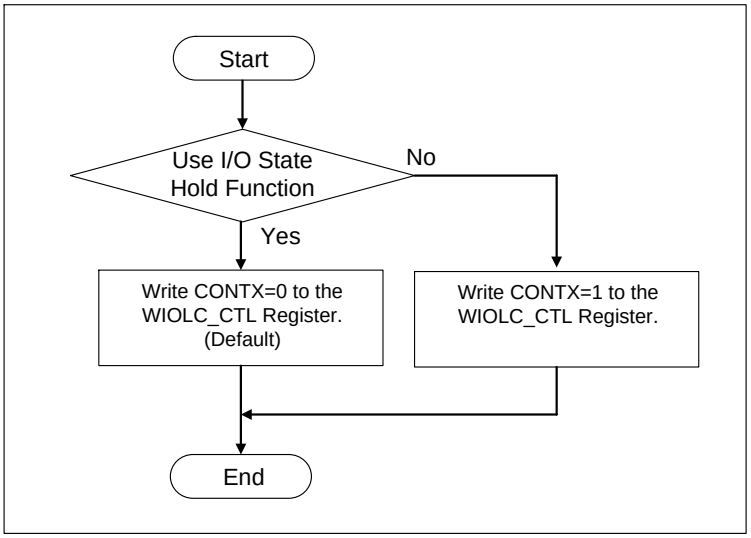
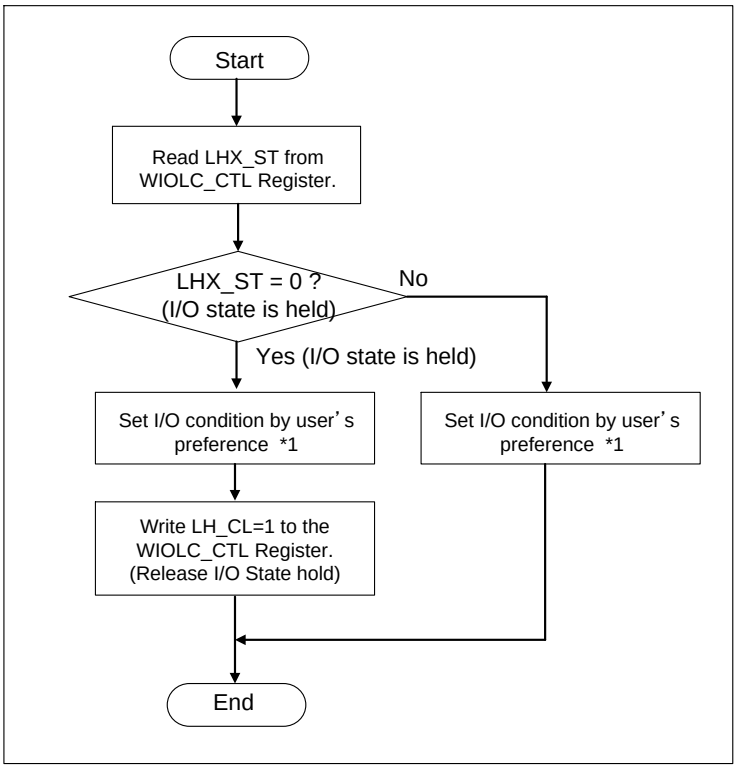


Figure 3-3 shows how to use I/O state hold function after returning from Deep Standby mode.

Figure 3-3 I/O State Hold function Setting Procedure Example2 (TYPE3-M0+)



*1 : See Figure 3-1 Setup Procedure Example of the I/O Port.

4. Registers

This section provides the register list of the I/O port.

Table 4-1 provides the register list.

Table 4-1 Register List of the I/O Port

Abbreviation	Register Name	Reference
PFR0	Port function setting register 0	4.1
PFR1	Port function setting register 1	
PFR2	Port function setting register 2	
PFR3	Port function setting register 3	
PFR4	Port function setting register 4	
PFR5	Port function setting register 5	
PFR6	Port function setting register 6	
PFR7	Port function setting register 7	
PFR8	Port function setting register 8	
PFR9	Port function setting register 9	
PFRA	Port function setting register A	
PFRB	Port function setting register B	
PFRC	Port function setting register C	
PFRD	Port function setting register D	
PFRE	Port function setting register E	
PFRF	Port function setting register F	
PCR0	Pull-up setting register 0	4.2
PCR1	Pull-up setting register 1	
PCR2	Pull-up setting register 2	
PCR3	Pull-up setting register 3	
PCR4	Pull-up setting register 4	
PCR5	Pull-up setting register 5	
PCR6	Pull-up setting register 6	
PCR7	Pull-up setting register 7	
PCR9	Pull-up setting register 9	
PCRA	Pull-up setting register A	
PCRB	Pull-up setting register B	
PCRC	Pull-up setting register C	
PCRD	Pull-up setting register D	
PCRE	Pull-up setting register E	
PCRF	Pull-up setting register F	

Abbreviation	Register Name	Reference
DDR0	Port input/output direction setting register 0	4.3
DDR1	Port input/output direction setting register 1	
DDR2	Port input/output direction setting register 2	
DDR3	Port input/output direction setting register 3	
DDR4	Port input/output direction setting register 4	
DDR5	Port input/output direction setting register 5	
DDR6	Port input/output direction setting register 6	
DDR7	Port input/output direction setting register 7	
DDR8	Port input/output direction setting register 8	
DDR9	Port input/output direction setting register 9	
DDRA	Port input/output direction setting register A	
DDRB	Port input/output direction setting register B	
DDRC	Port input/output direction setting register C	
DDRD	Port input/output direction setting register D	
DDRE	Port input/output direction setting register E	
DDRF	Port input/output direction setting register F	
PDIR0	Port input data register 0	4.4
PDIR1	Port input data register 1	
PDIR2	Port input data register 2	
PDIR3	Port input data register 3	
PDIR4	Port input data register 4	
PDIR5	Port input data register 5	
PDIR6	Port input data register 6	
PDIR7	Port input data register 7	
PDIR8	Port input data register 8	
PDIR9	Port input data register 9	
PDIRA	Port input data register A	
PDIRB	Port input data register B	
PDIRC	Port input data register C	
PDIRD	Port input data register D	
PDIRE	Port input data register E	
PDIRF	Port input data register F	

Abbreviation	Register Name	Reference
PDOR0	Port output data register 0	4.5
PDOR1	Port output data register 1	
PDOR2	Port output data register 2	
PDOR3	Port output data register 3	
PDOR4	Port output data register 4	
PDOR5	Port output data register 5	
PDOR6	Port output data register 6	
PDOR7	Port output data register 7	
PDOR8	Port output data register 8	
PDOR9	Port output data register 9	
PDORA	Port output data register A	
PDORB	Port output data register B	
PDORC	Port output data register C	
PDORD	Port output data register D	
PDORE	Port output data register E	
PDORF	Port output data register F	
ADE	Analog input setting register	4.6
EPFR	Extended pin function setting register	4.7
EPFR00	Extended pin function setting register 00	4.8
EPFR01	Extended pin function setting register 01	4.9
EPFR02	Extended pin function setting register 02	4.10
EPFR03	Extended pin function setting register 03	4.11
EPFR04	Extended pin function setting register 04	4.12
EPFR05	Extended pin function setting register 05	4.13
EPFR06	Extended pin function setting register 06	4.14
EPFR07	Extended pin function setting register 07	4.15
EPFR08	Extended pin function setting register 08	4.16
EPFR09	Extended pin function setting register 09	4.17
EPFR12	Extended pin function setting register 12	4.18
EPFR13	Extended pin function setting register 13	4.19
EPFR14	Extended pin function setting register 14	4.20
EPFR15	Extended pin function setting register 15	4.21
EPFR16	Extended pin function setting register 16	4.22
EPFR17	Extended pin function setting register 17	4.23
EPFR18	Extended pin function setting register 18	4.24
EPFR21	Extended pin function setting register 21	4.25
EPFR22	Extended pin function setting register 22	4.26
EPFR23	Extended pin function setting register 23	4.27
EPFR31	Extended pin function setting register 31	4.28
EPFR33	Extended pin function setting register 33	4.29
EPFR34	Extended pin function setting register 34	4.30
EPFR37	Extended pin function setting register 37	4.31
EPFR38	Extended pin function setting register 38	4.32
SPSR	Special Port Setting Register	4.33

Abbreviation	Register Name	Reference
PZR0	Port pseudo open drain setting register 0	4.34
PZR1	Port pseudo open drain setting register 1	
PZR2	Port pseudo open drain setting register 2	
PZR3	Port pseudo open drain setting register 3	
PZR4	Port pseudo open drain setting register 4	
PZR5	Port pseudo open drain setting register 5	
PZR6	Port pseudo open drain setting register 6	
PZR7	Port pseudo open drain setting register 7	
PZR8	Port pseudo open drain setting register 8	
PZR9	Port pseudo open drain setting register 9	
PZRA	Port pseudo open drain setting register A	
PZRB	Port pseudo open drain setting register B	
PZRC	Port pseudo open drain setting register C	
PZRD	Port pseudo open drain setting register D	
PZRE	Port pseudo open drain setting register E	
PZRF	Port pseudo open drain setting register F	
LVDIE	LVDI input enable setting register	4.35

4.1 Port Function Setting Register (PFRx)

The PFRx register selects usage of a pin.

List of PFR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	PFR0	0x000A	R/W	P0F to P00
Reserved	PFR1	0x0000	R/W	P1F to P10
Reserved	PFR2	0x0000	R/W	P2F to P20
Reserved	PFR3	0x0000	R/W	P3F to P30
Reserved	PFR4	0x0000	R/W	P4F to P40
Reserved	PFR5	0x0000	R/W	P5F to P50
Reserved	PFR6	0x0000	R/W	P6F to P60
Reserved	PFR7	0x0000	R/W	P7F to P70
Reserved	PFR8	0x0000	R/W	P8F to P80
Reserved	PFR9	0x0000	R/W	P9F to P90
Reserved	PFRA	0x0000	R/W	PAF to PA0
Reserved	PFRB	0x0000	R/W	PBF to PB0
Reserved	PFRC	0x0000	R/W	PCF to PC0
Reserved	PFRD	0x0000	R/W	PDF to PD0
Reserved	PFRE	0x0000	R/W	PEF to PE0
Reserved	PFRF	0x0000	R/W	PFF to PF0

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	PFRx

Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] PFRx: Port Function Setting Register x

Selects usage of a pin.

bit15:0		Description
Reading		Can read out the setting value of the register.
Writing	0	Uses a pin as a GPIO pin.
	1	Uses a pin as an input/output pin of peripheral functions.

Notes:

- *The "x" of PFRx is a wildcard. PFRx indicates PFR0, PFR1, PFR2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *Functions can be set for 16 ports from PxF to Px0.*
- *Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PFR0 sets P0F, the 14th bit of PFR0 sets P0E, and the 0th bit of PFR0 sets P00.*

- *As a Serial Wire Debug pin is selected for P01 and P03, the initial value is "1".*
- *Initial value is just an example for TYPE1-M0+, please check Appendix for different product TYPE.*
- *For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.*

4.2 Pull-Up Setting Register (PCRx)

The PCRx register sets pull-up of a pin.

List of PCR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	PCR0	0x000A	R/W	P0F to P00
Reserved	PCR1	0x0000	R/W	P1F to P10
Reserved	PCR2	0x0000	R/W	P2F to P20
Reserved	PCR3	0x0000	R/W	P3F to P30
Reserved	PCR4	0x0000	R/W	P4F to P40
Reserved	PCR5	0x0000	R/W	P5F to P50
Reserved	PCR6	0x0000	R/W	P6F to P60
Reserved	PCR7	0x0000	R/W	P7F to P70
Reserved	PCR8	0x0000	R/W	P8F to P80
Reserved	PCR9	0x0000	R/W	P9F to P90
Reserved	PCRA	0x0000	R/W	PAF to PA0
Reserved	PCRB	0x0000	R/W	PBF to PB0
Reserved	PCRC	0x0000	R/W	PCF to PC0
Reserved	PCRD	0x0000	R/W	PDF to PD0
Reserved	PCRE	0x0000	R/W	PEF to PE0
Reserved	PCRF	0x0000	R/W	PFF to PF0

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	PCRx

Register Function

[bit31:16] Reserved: Register bits

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] PCRx: Pull-up Setting Register x

Sets pull-up of a pin

bit15:0		Description
Reading		Can read out the setting value of the register.
Writing	0	Disconnects the pull-up resistor of a pin.
	1	When a pin is in input status (for both GPIO and peripheral functions), the pull-up resistor is connected. When a pin is in output status, the pull-up resistor is disconnected.

Notes:

- *The "x" of PCRx is a wildcard. PCRx indicates PCR0, PCR1, PCR2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *One register allows setting 16 pull-ups from PxF to Px0.*
- *Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PCR0 sets P0F, the 14th bit of PCR0 sets P0E, and the 0th bit of PCR0 sets P00.*
- *As a Serial Debug pin is selected for P01 and P03, the initial value is "1".*
- *When using I^2C function, use external pull-up by setting PCRx=0.*
- *PCR8 is not available.*
- *For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.*
- *PE0, PE1 ports do not have a pull-up resistor. Because of this, writing a value to PE register is invalid. An initial value or a write value is read in this register.*
- *Initial value is just an example for TYPE1-M0+, please check Appendix for different product TYPE.*

4.3 Port Input/Output Direction Setting Register (DDRx)

The DDRx register sets input/output direction of a pin.

List of DDR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	DDR0	0x0000	R/W	P0F to P00
Reserved	DDR1	0x0000	R/W	P1F to P10
Reserved	DDR2	0x0000	R/W	P2F to P20
Reserved	DDR3	0x0000	R/W	P3F to P30
Reserved	DDR4	0x0000	R/W	P4F to P40
Reserved	DDR5	0x0000	R/W	P5F to P50
Reserved	DDR6	0x0000	R/W	P6F to P60
Reserved	DDR7	0x0000	R/W	P7F to P70
Reserved	DDR8	0x0000	R/W	P8F to P80
Reserved	DDR9	0x0000	R/W	P9F to P90
Reserved	DDRA	0x0000	R/W	PAF to PA0
Reserved	DDRB	0x0000	R/W	PBF to PB0
Reserved	DDRC	0x0000	R/W	PCF to PC0
Reserved	DDRD	0x0000	R/W	PDF to PD0
Reserved	DDRE	0x0000	R/W	PEF to PE0
Reserved	DDRF	0x0000	R/W	PFF to PF0

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	DDRx

Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] DDRx: Port input/output Direction Setting Register x

Sets input/output direction of a pin.

bit15:0		Description
Reading		Can read out the setting value of the register.
Writing	0	Uses GPIO in input direction. If a pin is selected as an input/output pin of peripheral functions, this setting value is invalid.
	1	Uses GPIO in output direction. If a pin is selected as an input/output pin of peripheral functions, this setting value is invalid.

Notes:

- *The "x" of DDRx is a wildcard. DDRx indicates DDR0, DDR1, DDR2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *One register allows setting the input/output direction of 16 ports from PxF to Px0.*
- *Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of DDR0 sets P0F, the 14th bit of DDR0 sets P0E, and the 0th bit of DDR0 sets P00.*
- *If the output RTO of a multifunction timer is selected, in an emergency stop due to DTTIX signal, a DDR controls pin status. For more information, see the chapter "Multifunction Timer" in "Timer Part".*
- *For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.*
- *Initial value is just an example for TYPE1, please check Appendix for different product TYPE.*

4.4 Port Input Data Register (PDIRx)

The PDIRx register indicates input data of a pin.

List of PDIR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	PDIR0	0xXXXX	R	P0F to P00
Reserved	PDIR1	0xXXXX	R	P1F to P10
Reserved	PDIR2	0xXXXX	R	P2F to P20
Reserved	PDIR3	0xXXXX	R	P3F to P30
Reserved	PDIR4	0xXXXX	R	P4F to P40
Reserved	PDIR5	0xXXXX	R	P5F to P50
Reserved	PDIR6	0xXXXX	R	P6F to P60
Reserved	PDIR7	0xXXXX	R	P7F to P70
Reserved	PDIR8	0xXXXX	R	P8F to P80
Reserved	PDIR9	0xXXXX	R	P9F to P90
Reserved	PDIRA	0xXXXX	R	PAF to PA0
Reserved	PDIRB	0xXXXX	R	PBF to PB0
Reserved	PDIRC	0xXXXX	R	PCF to PC0
Reserved	PDIRD	0xXXXX	R	PDF to PD0
Reserved	PDIRE	0xXXXX	R	PEF to PE0
Reserved	PDIRF	0xXXXX	R	PFF to PF0

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	PDIRx

Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] PDIRx: Port Input Data Register x

Reads out input data of a pin.

bit15:0		Description
Reading	0	Regardless of pin function settings (PFR/EPFR/DDR/PDOR), it indicates that a pin is in the status of "L" level input or "L" level output. When a special pin is selected by ADE/SPSR, as input is cut off, 0 is always read out.
	1	Regardless of pin function settings (PFR/EPFR/DDR/PDOR), it indicates that a pin is in the status of "H" level input or "H" level output.
Writing		Writing does not affect anything.

Notes:

- *The "x" of PDIRx is a wildcard. PDIRx indicates PDIR0, PDIR1, PDIR2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *One register allows reading input data of 16 ports from PxF to Px0.*
- *Each bit in the register indicates the status of each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PDIR0 indicates P0F, the 14th bit of PDIR0 indicates P0E, and the 0th bit of PDIR0 indicates P00.*
- *"0" is always read for a bit value of the pin which is not available in your product.*
- *Initial value is just an example for TYPE1-M0+, please check Appendix for different product TYPE.*

4.5 Port Output Data Register x (PDORx)

The PDORx register sets output data to a pin.

List of PDOR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	PDOR0	0x0000	R/W	P0F to P00
Reserved	PDOR1	0x0000	R/W	P1F to P10
Reserved	PDOR2	0x0000	R/W	P2F to P20
Reserved	PDOR3	0x0000	R/W	P3F to P30
Reserved	PDOR4	0x0000	R/W	P4F to P40
Reserved	PDOR5	0x0000	R/W	P5F to P50
Reserved	PDOR6	0x0000	R/W	P6F to P60
Reserved	PDOR7	0x0000	R/W	P7F to P70
Reserved	PDOR8	0x0000	R/W	P8F to P80
Reserved	PDOR9	0x0000	R/W	P9F to P90
Reserved	PDORA	0x0000	R/W	PAF to PA0
Reserved	PDORB	0x0000	R/W	PBF to PB0
Reserved	PDORC	0x0000	R/W	PCF to PC0
Reserved	PDORD	0x0000	R/W	PDF to PD0
Reserved	PDORE	0x0000	R/W	PEF to PE0
Reserved	PDORF	0x0000	R/W	PFF to PF0

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	PDORx

Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] PDORx: Port Output Data Register x

Sets output data of a pin.

bit15:0		Description
Reading		Reads out the register value.
Writing	0	Outputs "L" level to GPIO. If a pin is selected as I/O input or peripheral functions input/output, a setting value is invalid.
	1	Outputs "H" level to GPIO. If a pin is selected as I/O input or peripheral functions input/output, a setting value is invalid.

Notes:

- *The "x" of PDORx is a wildcard. PDORx indicates PDOR0, PDOR1, PDOR2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *One register allows setting output data of 16 ports from PxF to Px0.*
- *Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PDOR0 sets P0F, the 14th bit of PDOR0 sets P0E, and the 0th bit of PDOR0 sets P00.*
- *For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.*
- *Initial value is just an example for TYPE1-M0+, please check Appendix for different product TYPE.*

4.6 Analog Input Setting Register (ADE)

The ADE register sets an external pin as an analog signal input pin of ADC.

Register Configuration

bit	31															0
Field	ADE															
Attribute	R/W															
Initial value	0xFFFFFFFF															

Register Function

[bit31:0] ADE: Analog Input Setting Register

Sets as an analog signal input pin.

bit31:0		Description
Reading		Reads out the register value.
Writing	0	Uses an external pin not as analog input but digital input/output.
	1	Uses an external pin as analog input. (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)

Notes:

- *This register sets analog input pins from AN31 to AN00.*
- *Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 31st bit of ADE sets AN31, the 14th bit of ADE sets AN14, and the 0th bit of ADE sets AN00. The port position of ANxx differs by each product.*
  For correspondence, refer to the "Data Sheet" of the product used.
- *In case of the port that is shared with WKUPx pin and ANx pin, The setting as (WIER:WUEx=1) & (ADEx=1)) is prohibited.*

4.7 Extended Pin Function Setting Register (EPFRx)

The EPFRx register assigns functions to a pin if there is more than one function.

List of EPFRx Register Configuration

bit 31 ... 0	Initial value	Attribute	Corresponding function
EPFR00	0x00010000	R/W	System function
EPFR01	0x00000000	R/W	Multi-function timer
EPFR02	0x00000000	R/W	
EPFR03	0x00000000	R/W	
EPFR04	0x00000000	R/W	Base timer
EPFR05	0x00000000	R/W	
EPFR06	0x00000000	R/W	External interrupt
EPFR07	0x00000000	R/W	Multi-function serial
EPFR08	0x00000000	R/W	
EPFR09	0x00000000	R/W	ADC trigger/QPRC
EPFR12	0x00000000	R/W	Base timer
EPFR13	0x00000000	R/W	
EPFR14	0x00000000	R/W	QPRC
EPFR15	0x00000000	R/W	External interrupt
EPFR16	0x00000000	R/W	Multi-function serial
EPFR17	0x00000000	R/W	
EPFR18	0x00000000	R/W	HDMI-CEC/Remote reception
EPFR21	0x00000000	R/W	QPRC
EPFR22	0x00000000	R/W	Multi-function serial
EPFR23	0x00000000	R/W	Multi-function serial
EPFR31	0x00000000	R/W	I2C Slave Wakeup
EPFR33	0x00000000	R/W	ICCard
EPFR34	0x00000000	R/W	Multi-function serial. Flow Control
EPFR37	0x00000000	R/W	Multi-function serial.I2S
EPFR38	0x00000000	R/W	Multi-function serial.I2S

EPFRx register is different depending on product TYPE.

For the correspondence between EPFRx register existence and product TYPE, SEE Table 4-2.

Notes:

- *Initial value is just an example for TYPE1-M0+, please check Appendix for different product TYPE.*

Table 4-2 EPFRx Register Product TYPE Correspondence Table

	TYPE1-M0+	TYPE2-M0+	TYPE3-M0+
EPFR00	❍	❍	❍
EPFR01	❍	❍	-
EPFR02	-	-	-
EPFR03	-	-	-
EPFR04	❍	❍	❍
EPFR05	-	❍	❍
EPFR06	❍	❍	❍
EPFR07	❍	❍	❍
EPFR08	-	❍	❍
EPFR09	❍	❍	❍
EPFR12	-	-	-
EPFR13	-	-	-
EPFR14	-	-	-
EPFR15	-	❍	-
EPFR16	-	❍	-
EPFR17	-	-	-
EPFR18	-	❍	-
EPFR21	❍	-	-
EPFR22	❍	-	❍
EPFR23	-	❍	❍
EPFR31	-	-	❍
EPFR33	-	❍	❍
EPFR34	-	❍	-
EPFR37	-	❍	❍
EPFR38	-	❍	❍

4.8 Extended Pin Function Setting Register 00 (EPFR00)

The EPFR00 register assigns functions to a pin if there is more than one function.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							SWDEN
Attribute	-							R/W
Initial value	-							1

bit	15	14	13	12	11	10	9	8
Field	Reserved						USBP0E	Reserved
Attribute	-						R/W	-
Initial value	-						0	-

bit	7	6	5	4	3	2	1	0
Field	SUBOUTE		RTCCOE		Reserved	CROUTE		NMIS
Attribute	R/W		R/W		-	R/W		R/W
Initial value	00		00		-	00		0

Register Function

[bit31:17] Reserved: Reserved bits

"0" is read out from these bits.

When writing these bits, set them to "0".

[bit16] SWDEN: Serial Wire Debug Function Select bit 0

Selects the function for SWCLK and SWDIO pins.

Bit		Description
Reading		Reads out the register value.
Writing	0	Does not use two pins of SWCLK and SWDIO. (A shared pin is available.)
	1	Uses two pins of SWCLK and SWDIO. [Initial value]

Note:

- *To enable SWD, DBG_EN bit of DEBUG_SW_CTL register has to be set to 1, in addition to enabling SWDEN.*

[bit15:10] Reserved: Reserved bits

"0" is read out from these bits.
When writing these bits, set them to "0".

[bit9] USBP0E: USB ch.0 Function Select bit 1

Selects a function for USB ch.0.

Bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output D+ resistor control signal (HCONTX) for USB ch.0. [Initial value] (A shared pin is available.)
	1	Produces output D+ resistor control signal (HCONTX) for USB ch.0.

[bit8:0] Reserved: Reserved bits

"0" is read out from these bits.
When writing these bits, set them to "0".

[bit7:6] SUBOUTE: Sub clock divide output function select bit

Selects sub clock divide output.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Sub clock divide output is not executed. [initial value]
	01	SUBOUT_0 is used as the sub clock divide output pin.
	10	SUBOUT_1 is used as the sub clock divide output pin.
	11	SUBOUT_2 is used as the sub clock divide output pin.

[bit5:4] RTCCOE: RTC clock output select bit

Selects a RTC clock output.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	RTC clock output is not executed. [initial value]
	01	RTCCOE_0 is used as the RTC clock output pin.
	10	RTCCOE_1 is used as the RTC clock output pin.
	11	RTCCOE_2 is used as the RTC clock output pin.

[bit3] Reserved: Reserved bit

"0" is read out from this bit.
When writing this bit, set it to "0".

[bit2:1] CROUTE: Internal high-speed CR Oscillation Output Function Select bit

Selects internal high-speed CR oscillation output.

bit2:1		Description
Reading		Reads out the register value.
Writing	00	Does not produce internal high-speed CR oscillation output. [Initial value]
	01	Uses CROUT_0 at the internal high-speed CR oscillation output pin.
	10	Uses CROUT_1 at the internal high-speed CR oscillation output pin.
	11	Uses CROUT_2 at the internal high-speed CR oscillation output pin.

[bit0] NMIS: NMIX Function Select bit

Selects a function for the NMIX pin.

Bit		Description
Reading		Reads out the register value.
Writing	0	Does not use the NMIX pin. [Initial value]
	1	Uses the NMIX pin.

Note:

- *When NMIX pin is used, set NMIS="1" and PFR="1".*

In case of the TYPE1-M0+, TYPE2-M0+ products.

- *When the I/O port which is mapped to NMI input pin is changed to GPIO or other peripheral function from NMI (write EPFR00.NMIS = 1), input level of the I/O port should be held high level, and change the I/O port. Internal NMI signal is tied to high level in case of the I/O port is selected to GPIO or other peripheral function. Therefore, when input level of the I/O port is low, to change the I/O port from GPIO or other peripheral function to NMI, is caused to change of internal NMI signal high to low. So, falling edge will be detected, NMI request occurred.*

In case of the TYPE3-M0+ product.

- *When the I/O port which is mapped to NMI input pin is changed to GPIO or other peripheral function from NMI (write EPFR00.NMIS = 1), It should be changed in the state that is disabled the NMI with the Non Maskable Interrupt Enable Register (NMIENR=0). After changing GPIO, clear NMIRR with NMICL. And enable the NMI with (NMIENR=1). For details, see chapter of "External Interrupt and NMI Control Sections".*

4.9 Extended Pin Function Setting Register 01 (EPFR01)

The EPFR01 register assigns functions to a pin of the multifunction timer Unit0.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	IC03S			IC02S			IC01S	
Attribute	R/W			R/W			R/W	
Initial value	000			000			00	

bit	23	22	21	20	19	18	17	16
Field	IC01S	IC00S			FRCK0S		DTTI0S	
Attribute	R/W	R/W			R/W		R/W	
Initial value	0	000			00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved		IGTRG0	DTTI0C	RTO05E		RTO04E	
Attribute	-		R/W	R/W	R/W		R/W	
Initial value	-		0	0	00		00	

bit	7	6	5	4	3	2	1	0
Field	RTO03E		RTO02E		RTO01E		RTO00E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:29] IC03S: IC03 Input Select bits

Selects input for IC03.

bit31:29		Description
Reading		Reads out the register value.
Writing	000	Uses IC03_0 at the input pin of the input capture IC03. [Initial value]
	001	Same as Writing 000.
	010	Uses IC03_1 at the input pin of the input capture IC03.
	011	Uses IC03_2 at the input pin of the input capture IC03.
	100	Uses internal macro MFS ch.3 LSYN for input of the input capture IC03.
	101	Uses internal macro MFS ch.7 LSYN for input of the input capture IC03.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit28:26] IC02S: IC02 Input Select bits

Selects input for IC02.

bit28:26		Description
Reading		Reads out the register value.
Writing	000	Uses IC02_0 at the input pin of the input capture IC02. [Initial value]
	001	Same as Writing 000.
	010	Uses IC02_1 at the input pin of the input capture IC02.
	011	Uses IC02_2 at the input pin of the input capture IC02.
	100	Uses internal macro MFS ch.2 LSYN for input of the input capture IC02.
	101	Uses internal macro MFS ch.6 LSYN for input of the input capture IC02.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit25:23] IC01S: IC01 Input Select bits

Selects input for IC01.

bit25:23		Description
Reading		Reads out the register value.
Writing	000	Uses IC01_0 at the input pin of the input capture IC01. [Initial value]
	001	Same as Writing 000.
	010	Uses IC01_1 at the input pin of the input capture IC01.
	011	Uses IC01_2 at the input pin of the input capture IC01.
	100	Uses internal macro MFS ch.1 LSYN for input of the input capture IC01.
	101	Uses internal macro MFS ch.5 LSYN for input of the input capture IC01.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit22:20] IC00S: IC00 Input Select bits

Selects input for IC00.

bit22:20		Description
Reading		Reads out the register value.
Writing	000	Uses IC00_0 at the input pin of the input capture IC00. [Initial value]
	001	Same as Writing 000.
	010	Uses IC00_1 at the input pin of the input capture IC00.
	011	Uses IC00_2 at the input pin of the input capture IC00.
	100	Uses internal macro MFS ch.0 LSYN for input of the input capture IC00.
	101	Uses internal macro MFS ch.4 LSYN for input of the input capture IC00.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit19:18] FRCK0S: FRCK0 Input Select bits

Selects input for FRCK0.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses FRCK0_0 at the input pin of the free-run timer FRCK0. [Initial value]
	01	Same as Writing 00.
	10	Uses FRCK0_1 at the input pin of the free-run timer FRCK0.
	11	Uses FRCK0_2 at the input pin of the free-run timer FRCK0.

[bit17:16] DTTI0S: DTTI0X Input Select bits

Selects input for DTTI0X.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses DTTI0X_0 at the input pin of the waveform generator DTTI0X. [Initial value]
	01	Same as Writing 00.
	10	Uses DTTI0X_1 at the input pin of the waveform generator DTTI0X.
	11	Uses DTTI0X_2 at the input pin of the waveform generator DTTI0X.

[bit15:14] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit13] IGTRG0: IGTRG0 Input Select bit

Selects input for IGTRG0.

bit13		Description
Reading		Reads out the register value.
Writing	0	Uses IGTRG0_0 at the input pin of the PPG IGTRG. [Initial value]
	1	Uses IGTRG0_1 at the input pin of the PPG IGTRG.

[bit12] DTTI0C: DTTI0X Function Select bit

Selects a function for DTTI0X.

bit12		Description
Reading		Reads out the register value.
Writing	0	Does not switch GPIO by DTTIF0 for output of pins RTO00 to RTO05. [Initial value]
	1	Switches GPIO by DTTIF0 for output of pins RTO00 to RTO05.

[bit11:10] RTO05E: RTO05 Output Select bits

Selects output for RTO05.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO05. [Initial value]
	01	Uses RTO05_0 at the output pin of the waveform generator RTO05.
	10	Uses RTO05_1 at the output pin of the waveform generator RTO05.
	11	Setting is prohibited.

[bit9:8] RTO04E: RTO04 Output Select bits

Selects output for RTO04.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO04. [Initial value]
	01	Uses RTO04_0 at the output pin of the waveform generator RTO04.
	10	Uses RTO04_1 at the output pin of the waveform generator RTO04.
	11	Setting is prohibited.

[bit7:6] RTO03E: RTO03 Output Select bits

Selects output for RTO03.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO03. [Initial value]
	01	Uses RTO03_0 at the output pin of the waveform generator RTO03.
	10	Uses RTO03_1 at the output pin of the waveform generator RTO03.
	11	Setting is prohibited.

[bit5:4] RTO02E: RTO02 Output Select bits

Selects output for RTO02.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO02. [Initial value]
	01	Uses RTO02_0 at the output pin of the waveform generator RTO02.
	10	Uses RTO02_1 at the output pin of the waveform generator RTO02.
	11	Setting is prohibited.

[bit3:2] RTO01E: RTO01 Output Select bits

Selects output for RTO01.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO01. [Initial value]
	01	Uses RTO01_0 at the output pin of the waveform generator RTO01.
	10	Uses RTO01_1 at the output pin of the waveform generator RTO01.
	11	Setting is prohibited.

[bit1:0] RTO00E: RTO00 Output Select bits

Selects output for RTO00.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO00. [Initial value]
	01	Uses RTO00_0 at the output pin of the waveform generator RTO00.
	10	Uses RTO00_1 at the output pin of the waveform generator RTO00.
	11	Setting is prohibited.

4.10 Extended Pin Function Setting Register 02 (EPFR02)

The EPFR02 register assigns functions to a pin of the multifunction timer Unit1.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	IC13S			IC12S			IC11S	
Attribute	R/W			R/W			R/W	
Initial value	000			000			00	

bit	23	22	21	20	19	18	17	16
Field	IC11S	IC10S			FRCK1S		DTTI1S	
Attribute	R/W	R/W			R/W		R/W	
Initial value	0	000			00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved			DTTI1C	RTO15E		RTO14E	
Attribute	-			R/W	R/W		R/W	
Initial value	-			0	00		00	

bit	7	6	5	4	3	2	1	0
Field	RTO13E		RTO12E		RTO11E		RTO10E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:29] IC13S: IC13 Input Select bits

Selects input for IC13.

bit31:29		Description
Reading		Reads out the register value.
Writing	000	Uses IC13_0 at the input pin of the input capture IC13. [Initial value]
	001	Same as Writing 000.
	010	Uses IC13_1 at the input pin of the input capture IC13.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.3 LSYN for input of the input capture IC13.
	101	Uses internal macro MFS ch.7 LSYN for input of the input capture IC13.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit28:26] IC12S: IC12 Input Select bits

Selects input for IC12.

bit28:26		Description
Reading		Reads out the register value.
Writing	000	Uses IC12_0 at the input pin of the input capture IC12. [Initial value]
	001	Same as Writing 000.
	010	Uses IC12_1 at the input pin of the input capture IC12.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.2 LSYN for input of the input capture IC12.
	101	Uses internal macro MFS ch.6 LSYN for input of the input capture IC12.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit25:23] IC11S: IC11 Input Select bits

Selects input for IC11.

bit25:23		Description
Reading		Reads out the register value.
Writing	000	Uses IC11_0 at the input pin of the input capture IC11. [Initial value]
	001	Same as Writing 000.
	010	Uses IC11_1 at the input pin of the input capture IC11.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.1 LSYN for input of the input capture IC11.
	101	Uses internal macro MFS ch.5 LSYN for input of the input capture IC11.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit22:20] IC10S: IC10 Input Select bits

Selects input for IC10.

bit22:20		Description
Reading		Reads out the register value.
Writing	000	Uses IC10_0 at the input pin of the input capture IC10. [Initial value]
	001	Same as Writing 000.
	010	Uses IC10_1 at the input pin of the input capture IC10.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.0 LSYN for input of the input capture IC10.
	101	Uses internal macro MFS ch.4 LSYN for input of the input capture IC10.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit19:18] FRCK1S: FRCK1 Input Select bits

Selects input for FRCK1.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses FRCK1_0 at the input pin of the free-run timer FRCK1. [Initial value]
	01	Same as Writing 00.
	10	Uses FRCK1_1 at the input pin of the free-run timer FRCK1.
	11	Setting is prohibited.

[bit17:16] DTTI1S: DTTI1X Input Select bits

Select input for DTTI1X.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses DTTI1X_0 at the input pin of the waveform generator DTTI1X. [Initial value]
	01	Same as Writing 00.
	10	Uses DTTI1X_1 at the input pin of the waveform generator DTTI1X.
	11	Setting is prohibited.

[bit15:13] Reserved: Reserved bits

"0b000" is read out from these bits.

When writing these bits, set them to "0b000".

[bit12] DTTI1C: DTTI1X Function Select bit

Selects a function for DTTI1X.

Bit		Description
Reading		Reads out the register value.
Writing	0	Does not switch GPIO by DTTIF1 for output of pins RTO10 to RTO15. [Initial value]
	1	Switches GPIO by DTTIF1 for output of pins RTO10 to RTO15.

[bit11:10] RTO15E: RTO15 Output Select bits

Selects output for RTO15.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO15. [Initial value]
	01	Uses RTO15_0 at the output pin of the waveform generator RTO15.
	10	Uses RTO15_1 at the output pin of the waveform generator RTO15.
	11	Setting is prohibited.

[bit9:8] RTO14E: RTO14 Output Select bits

Selects output for RTO14.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO14. [Initial value]
	01	Uses RTO14_0 at the output pin of the waveform generator RTO14.
	10	Uses RTO14_1 at the output pin of the waveform generator RTO14.
	11	Setting is prohibited.

[bit7:6] RTO13E: RTO13 Output Select bits

Selects output for RTO13.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO13. [Initial value]
	01	Uses RTO13_0 at the output pin of the waveform generator RTO13.
	10	Uses RTO13_1 at the output pin of the waveform generator RTO13.
	11	Setting is prohibited.

[bit5:4] RTO12E: RTO12 Output Select bits

Selects output for RTO12.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO12. [Initial value]
	01	Uses RTO12_0 at the output pin of the waveform generator RTO12.
	10	Uses RTO12_1 at the output pin of the waveform generator RTO12.
	11	Setting is prohibited.

[bit3:2] RTO11E: RTO11 Output Select bits

Selects output for RTO11.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO11. [Initial value]
	01	Uses RTO11_0 at the output pin of the waveform generator RTO11.
	10	Uses RTO11_1 at the output pin of the waveform generator RTO11.
	11	Setting is prohibited.

[bit1:0] RTO10E: RTO10 Output Select bits

Selects output for RTO10.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO10. [Initial value]
	01	Uses RTO10_0 at the output pin of the waveform generator RTO10.
	10	Uses RTO10_1 at the output pin of the waveform generator RTO10.
	11	Setting is prohibited.

4.11 Extension Function Pin Setting Register 03 (EPFR03)

EPFR03 register sets the function assignment to the multi-function timer Unit2 pin.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	IC23S			IC22S			IC21S	
Attribute	R/W			R/W			R/W	
Initial value	000			000			00	

bit	23	22	21	20	19	18	17	16
Field	IC21S	IC20S			FRCK2S		DTTI2S	
Attribute	R/W	R/W			R/W		R/W	
Initial value	0	000			00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved			DTTI2C	RTO25E		RTO24E	
Attribute	-			R/W	R/W		R/W	
Initial value	-			0	00		00	

bit	7	6	5	4	3	2	1	0
Field	RTO23E		RTO22E		RTO21E		RTO20E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:29] IC23S: IC23 input select bits

Selects IC23 input.

bit31:29		Description
Reading		Reads out the register value.
Writing	000	Use IC23_0 as the input pin of input capture IC23. [initial value]
	001	Same as when writing "000"
	010	Use IC23_1 as the input pin of input capture IC23.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.3 LSYN as input of input capture IC23.
	101	Use internal macro MFS ch.7 LSYN as input of input capture IC23.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit28:26] IC22S: IC22 input select bits

Selects IC22 input.

bit28:26		Description
Reading		Reads out the register value.
Writing	000	Use IC22_0 as the input pin of input capture IC22. [initial value]
	001	Same as when writing "000"
	010	Use IC22_1 as the input pin of input capture IC22.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.2 LSYN as input of input capture IC22.
	101	Use internal macro MFS ch.6 LSYN as input of input capture IC22.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit25:23] IC21S: IC21 input select bits

Selects IC21 input.

bit25:23		Description
Reading		Reads out the register value.
Writing	000	Use IC21_0 as the input pin of input capture IC21. [initial value]
	001	Same as when writing "000"
	010	Use IC21_1 as the input pin of input capture IC21.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.1 LSYN as input of input capture IC21.
	101	Use internal macro MFS ch.5 LSYN as input of input capture IC21.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit22:20] IC20S: IC20 input select bits

Selects IC20 input.

bit22:20		Description
Reading		Reads out the register value.
Writing	000	Use IC20_0 as the input pin of input capture IC20. [initial value]
	001	Same as when writing "000"
	010	Use IC20_1 as the input pin of input capture IC20.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.0 LSYN as input of input capture IC20.
	101	Use internal macro MFS ch.4 LSYN as input of input capture IC20.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit19:18] FRCK2S: FRCK2 Input Select bits

Selects input for FRCK2.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Use FRCK2_0 as the input pin of free-run timer FRCK2. [Initial value]
	01	Same as Writing 00.
	10	Use FRCK2_1 a as the input pin of free-run timer FRCK2.
	11	Setting is prohibited.

[bit17:16] DTTI2S: DTTI2X Input Select bits

Selects input for DTTI2X.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Use DTTI2X_0 as the input pin of waveform generator DTTI2X. [Initial value]
	01	Same as Writing 00.
	10	Use DTTI2X_1 as the input pin of waveform generator DTTI2X.
	11	Setting is prohibited.

[bit15:13] Reserved: Reserved bits

"0b000" is read from these bits.

When writing, set them to "0b000".

[bit12] DTTI2C: DTTI2X Function Select bit

Selects the function of DTTI2X.

Bit		Description
Reading		Reads out the register value.
Writing	0	Does not switch GPIO by DTTIF2 for outputs of the pins from RTO20 to RTO25. [Initial value]
	1	Switches GPIO by DTTIF2 for outputs of the pins from RTO20 to RTO25.

[bit11:10] RTO25E: RTO25 Output Select bits

Selects the output of RTO25.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO25. [Initial value]
	01	Use RTO25_0 as the output pin of waveform generator RTO25.
	10	Use RTO25_1 as the output pin of waveform generator RTO25.
	11	Setting is prohibited.

[bit9:8] RTO24E: RTO24 Output Select bits

Selects output for RTO24.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO24. [Initial value]
	01	Use RTO24_0 as the output pin of waveform generator RTO24.
	10	Use RTO24_1 as the output pin of waveform generator RTO24.
	11	Setting is prohibited.

[bit7:6] RTO23E: RTO23 Output Select bits

Selects output for RTO23.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO23. [Initial value]
	01	Use RTO23_0 as the output pin of waveform generator RTO23.
	10	Use RTO23_1 as the output pin of waveform generator RTO23.
	11	Setting is prohibited.

[bit5:4] RTO22E: RTO22 Output Select bits

Selects output for RTO22.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO22. [Initial value]
	01	Use RTO22_0 as the output pin of waveform generator RTO22.
	10	Use RTO22_1 as the output pin of waveform generator RTO22.
	11	Setting is prohibited.

[bit3:2] RTO21E: RTO21 Output Select bits

Selects output for RTO21.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO21. [Initial value]
	01	Use RTO21_0 as the output pin of waveform generator RTO21.
	10	Use RTO21_1 as the output pin of waveform generator RTO21.
	11	Setting is prohibited.

[bit1:0] RTO20E: RTO20 Output Select bits

Selects output for RTO20.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO20. [Initial value]
	01	Use RTO20_0 as the output pin of waveform generator RTO20.
	10	Use RTO20_1 as the output pin of waveform generator RTO20.
	11	Setting is prohibited.

4.12 Extended Pin Function Setting Register 04 (EPFR04)

The EPFR04 register assigns functions to pins of ch.0, ch.1, ch.2, and ch.3 of the base timer.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB3S		TIOA3E		TIOA3S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB2S		TIOA2E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB1S		TIOA1E		TIOA1S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved	TIOB0S			TIOA0E		Reserved	
Attribute	-	R/W			R/W		-	
Initial value	-	000			00		-	

Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit29:28] TIOB3S: TIOB3 Input Select bits

Selects input for TIOB3.

bit29:28		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB3_0 at the input pin of BT ch.3 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB3_1 at the input pin of BT ch.3 TIOB.
	11	Uses TIOB3_2 at the input pin of BT ch.3 TIOB.

[bit27:26] TIOA3E: TIOA3 Output Select bits

Selects output for TIOA3.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for BT ch.3 TIOA. [Initial value]
	01	Uses TIOA3_0 at the output pin of BT ch.3 TIOA.
	10	Uses TIOA3_1 at the output pin of BT ch.3 TIOA.
	11	Uses TIOA3_2 at the output pin of BT ch.3 TIOA.

[bit25:24] TIOA3S: TIOA3 Input Select bits

Selects input for TIOA3.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA3_0 at the input pin of BT ch.3 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA3_1 at the input pin of BT ch.3 TIOA.
	11	Uses TIOA3_2 at the input pin of BT ch.3 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit21:20] TIOB2S: TIOB2 Input Select bits

Selects input for TIOB2.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB2_0 at the input pin of BT ch.2 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB2_1 at the input pin of BT ch.2 TIOB.
	11	Uses TIOB2_2 at the input pin of BT ch.2 TIOB.

[bit19:18] TIOA2E: TIOA2 Output Select bits

Selects output for TIOA2.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for BT ch.2 TIOA. [Initial value]
	01	Uses TIOA2_0 at the output pin of BT ch.2 TIOA.
	10	Uses TIOA2_1 at the output pin of BT ch.2 TIOA.
	11	Uses TIOA2_2 at the output pin of BT ch.2 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.
When writing these bits, set them to "0b0000".

[bit13:12] TIOB1S: TIOB1 Input Select bits

Selects input for TIOB1.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB1_0 at the input pin of BT ch.1 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB1_1 at the input pin of BT ch.1 TIOB.
	11	Uses TIOB1_2 at the input pin of BT ch.1 TIOB.

[bit11:10] TIOA1E: TIOA1 Output Select bits

Selects output for TIOA1.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for BT ch.1 TIOA. [Initial value]
	01	Uses TIOA1_0 at the output pin of BT ch.1 TIOA.
	10	Uses TIOA1_1 at the output pin of BT ch.1 TIOA.
	11	Uses TIOA1_2 at the output pin of BT ch.1 TIOA.

[bit9:8] TIOA1S: TIOA1 Input Select bits

Selects input for TIOA1.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA1_0 at the input pin of BT ch.1 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA1_1 at the input pin of BT ch.1 TIOA.
	11	Uses TIOA1_2 at the input pin of BT ch.1 TIOA.

[bit7] Reserved: Reserved bit

"0b0" is read out from this bit.
When writing this bit, set it to "0b0".

[bit6:4] TIOB0S: TIOB0 Input Select bits

Selects input for TIOB0.

bit6:4		Description
Reading		Reads out the register value.
Writing	000	Uses TIOB0_0 at the input pin of BT ch.0 TIOB. [Initial value]
	001	Same as Writing 000.
	010	Uses TIOB0_1 at the input pin of BT ch.0 TIOB.
	011	Uses TIOB0_2 at the input pin of BT ch.0 TIOB.
	100	Setting is prohibited.
	101	Setting is prohibited.
	110	Uses SUBOUT at the input pin of BT ch.0 TIOB.
	111	Uses at the pin for measuring trimming of the high-speed CR frequency division clock.

[bit3:2] TIOA0E: TIOA0 Output Select bits

Selects output for TIOA0.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Produces output for BT ch.0 TIOA. [Initial value]
	01	Uses TIOA0_0 at the output pin of BT ch.0 TIOA.
	10	Uses TIOA0_1 at the output pin of BT ch.0 TIOA.
	11	Uses TIOA0_2 at the output pin of BT ch.0 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

Notes:

- *TIOA*
  Even channels are for output only.
  Odd channels are for both input and output.
- *TIOB*
  Input only.
- *TIOA1, TIOA3, TIOA5, TIOA7(odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.*
  When output is chosen for odd TIOA channel, input setting will be ignored.

  Example1: Use TIOA1 as an output pin:
  When TIOA1 is output to TIOA1_0, select EPFR04:TIOA1E = 01.
  When TIOA1 is output to TIOA1_1, select EPFR04:TIOA1E = 10.
  When TIOA1 is output to TIOA1_2, select EPFR04:TIOA1E = 11.
  Settings for EPFR04:TIOA1S will be ignored.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

Example2: When TIOA1 is used as an input pin:

Select EPFR04:TIOA1E = 00.

When TIOA1 is input from TIOA1_0, select EPFR04:TIOA1S = 00 or 01.

When TIOA1 is input from TIOA1_1, select EPFR04:TIOA1S = 10.

When TIOA1 is input from TIOA1_2, select EPFR04:TIOA1S = 11.

Select ADE=0, PFR=1 for selected pins (DDR will be ignored).

All the output of other peripheral function pins which are also used by selected pins must be OFF.

** When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.*

4.13 Extended Pin Function Setting Register 05 (EPFR05)

The EPFR05 register assigns functions to pins of ch.4, ch.5, ch.6, and ch.7 of the base timer.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB7S		TIOA7E		TIOA7S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB6S		TIOA6E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB5S		TIOA5E		TIOA5S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved		TIOB4S		TIOA4E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit29:28] TIOB7S: TIOB7 Input Select bits

Selects input for TIOB7.

bit29:28		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB7_0 at the input pin of BT ch.7 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB7_1 at the input pin of BT ch.7 TIOB.
	11	Uses TIOB7_2 at the input pin of BT ch.7 TIOB.

[bit27:26] TIOA7E: TIOA7 Output Select bits

Selects output for TIOA7.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.7 TIOA. [Initial value]
	01	Uses TIOA7_0 at the output pin of BT ch.7 TIOA.
	10	Uses TIOA7_1 at the output pin of BT ch.7 TIOA.
	11	Uses TIOA7_2 at the output pin of BT ch.7 TIOA.

[bit25:24] TIOA7S: TIOA7 Input Select bits

Selects input for TIOA7.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA7_0 at the input pin of BT ch.7 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA7_1 at the input pin of BT ch.7 TIOA.
	11	Uses TIOA7_2 at the input pin of BT ch.7 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

■ [bit21:20] TIOB6S: TIOB6 Input Select bits

Selects input for TIOB6.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB6_0 at the input pin of BT ch.6 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB6_1 at the input pin of BT ch.6 TIOB.
	11	Uses TIOB6_2 at the input pin of BT ch.6 TIOB.

[bit19:18] TIOA6E: TIOA6 Output Select bits

Selects output for TIOA6.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.6 TIOA. [Initial value]
	01	Uses TIOA6_0 at the output pin of BT ch.6 TIOA.
	10	Uses TIOA6_1 at the output pin of BT ch.6 TIOA.
	11	Uses TIOA6_2 at the output pin of BT ch.6 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.
When writing these bits, set them to "0b0000".

[bit13:12] TIOB5S: TIOB5 Input Select bits

Selects input for TIOB5.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB5_0 at the input pin of BT ch.5 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB5_1 at the input pin of BT ch.5 TIOB.
	11	Uses TIOB5_2 at the input pin of BT ch.5 TIOB.

[bit11:10] TIOA5E: TIOA5 Output Select bits

Selects output for TIOA5.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.5 TIOA. [Initial value]
	01	Uses TIOA5_0 at the output pin of BT ch.5 TIOA.
	10	Uses TIOA5_1 at the output pin of BT ch.5 TIOA.
	11	Uses TIOA5_2 at the output pin of BT ch.5 TIOA.

[bit9:8] TIOA5S: TIOA5 Input Select bits

Selects input for TIOA5.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA5_0 at the input pin of BT ch.5 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA5_1 at the input pin of BT ch.5 TIOA.
	11	Uses TIOA5_2 at the input pin of BT ch.5 TIOA.

[bit7:6] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit5:4] TIOB4S: TIOB4 Input Select bits

Selects input for TIOB4.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB4_0 at the input pin of BT ch.4 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB4_1 at the input pin of BT ch.4 TIOB.
	11	Uses TIOB4_2 at the input pin of BT ch.4 TIOB.

[bit3:2] TIOA4E: TIOA4 Output Select bits

Selects output for TIOA4.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.4 TIOA. [Initial value]
	01	Uses TIOA4_0 at the output pin of BT ch.4 TIOA.
	10	Uses TIOA4_1 at the output pin of BT ch.4 TIOA.
	11	Uses TIOA4_2 at the output pin of BT ch.4 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

Notes:

- *TIOA*
  Even channels are for output only.
  Odd channels are for both input and output.
- *TIOB*
  Input only.
- *TIOA1, TIOA3, TIOA5, TIOA7(odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.*
  When output is chosen for odd TIOA channel, input setting will be ignored.

  Example1: Use TIOA1 as an output pin:
  When TIOA1 is output to TIOA1_0, select EPFR04:TIOA1E = 01.
  When TIOA1 is output to TIOA1_1, select EPFR04:TIOA1E = 10.
  When TIOA1 is output to TIOA1_2, select EPFR04:TIOA1E = 11.
  Settings for EPFR04:TIOA1S will be ignored.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

  Example2: When TIOA1 is used as an input pin:
  Select EPFR04:TIOA1E = 00.
  When TIOA1 is input from TIOA1_0, select EPFR04:TIOA1S = 00 or 01.
  When TIOA1 is input from TIOA1_1, select EPFR04:TIOA1S = 10.
  When TIOA1 is input from TIOA1_2, select EPFR04:TIOA1S = 11.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).

All the output of other peripheral function pins which are also used by selected pins must be OFF.

** When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.*

4.14 Extended Pin Function Setting Register 06 (EPFR06)

The EPFR06 register assigns functions to external interrupt pins.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	EINT15S		EINT14S		EINT13S		EINT12S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	EINT11S		EINT10S		EINT09S		EINT08S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	EINT07S		EINT06S		EINT05S		EINT04S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	EINT03S		EINT02S		EINT01S		EINT00S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:30] EINT15S: External Interrupt Input Select bits

Selects input for EINT15.

bit31:30		Description
Reading		Reads out the register value.
Writing	00	Uses INT15_0 at the input pin of EINT ch.15. [Initial value]
	01	Same as Writing 00.
	10	Uses INT15_1 at the input pin of EINT ch.15.
	11	Uses INT15_2 at the input pin of EINT ch.15.

[bit29:28] EINT14S: External Interrupt Input Select bits

Selects input for EINT14.

bit29:28		Description
Reading		Reads out the register value.
Writing	00	Uses INT14_0 at the input pin of EINT ch.14. [Initial value]
	01	Same as Writing 00.
	10	Uses INT14_1 at the input pin of EINT ch.14.
	11	Uses INT14_2 at the input pin of EINT ch.14.

[bit27:26] EINT13S: External Interrupt Input Select bits

Selects input for EINT13.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Uses INT13_0 at the input pin of EINT ch.13. [Initial value]
	01	Same as Writing 00
	10	Uses INT13_1 at the input pin of EINT ch.13.
	11	Uses INT13_2 at the input pin of EINT ch.13.

[bit25:24] EINT12S: External Interrupt Input Select bits

Selects input for EINT12.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses INT12_0 at the input pin of EINT ch.12. [Initial value]
	01	Same as Writing 00.
	10	Uses INT12_1 at the input pin of EINT ch.12.
	11	Uses INT12_2 at the input pin of EINT ch.12.

[bit23:22] EINT11S: External Interrupt Input Select bits

Selects input for EINT11.

bit23:22		Description
Reading		Reads out the register value.
Writing	00	Uses INT11_0 at the input pin of EINT ch.11. [Initial value]
	01	Same as Writing 00.
	10	Uses INT11_1 at the input pin of EINT ch.11.
	11	Uses INT11_2 at the input pin of EINT ch.11.

[bit21:20] EINT10S: External Interrupt Input Select bits

Selects input for EINT10.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses INT10_0 at the input pin of EINT ch.10. [Initial value]
	01	Same as Writing 00.
	10	Uses INT10_1 at the input pin of EINT ch.10.
	11	Uses INT10_2 at the input pin of EINT ch.10.

[bit19:18] EINT09S: External Interrupt Input Select bits

Selects input for EINT09.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses INT09_0 at the input pin of EINT ch.9. [Initial value]
	01	Same as Writing 00.
	10	Uses INT09_1 at the input pin of EINT ch.9.
	11	Uses INT09_2 at the input pin of EINT ch.9.

[bit17:16] EINT08S: External Interrupt Input Select bits

Selects input for EINT08.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses INT08_0 at the input pin of EINT ch.8. [Initial value]
	01	Same as Writing 00.
	10	Uses INT08_1 at the input pin of EINT ch.8.
	11	Uses INT08_2 at the input pin of EINT ch.8.

[bit15:14] EINT07S: External Interrupt Input Select bits

Selects input for EINT07.

bit15:14		Description
Reading		Reads out the register value.
Writing	00	Uses INT07_0 at the input pin of EINT ch.7. [Initial value]
	01	Same as Writing 00.
	10	Uses INT07_1 at the input pin of EINT ch.7.
	11	Uses INT07_2 at the input pin of EINT ch.7.

[bit13:12] EINT06S: External Interrupt Input Select bits

Selects input for EINT06.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses INT06_0 at the input pin of EINT ch.6. [Initial value]
	01	Same as Writing 00.
	10	Uses INT06_1 at the input pin of EINT ch.6.
	11	Uses INT06_2 at the input pin of EINT ch.6.

[bit11:10] EINT05S: External Interrupt Input Select bits

Selects input for EINT05.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses INT05_0 at the input pin of EINT ch.5. [Initial value]
	01	Same as Writing 00.
	10	Uses INT05_1 at the input pin of EINT ch.5.
	11	Uses INT05_2 at the input pin of EINT ch.5.

[bit9:8] EINT04S: External Interrupt Input Select bits

Selects input for EINT04.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses INT04_0 at the input pin of EINT ch.4. [Initial value]
	01	Same as Writing 00.
	10	Uses INT04_1 at the input pin of EINT ch.4.
	11	Uses INT04_2 at the input pin of EINT ch.4.

[bit7:6] EINT03S: External Interrupt Input Select bits

Selects input for EINT03.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses INT03_0 at the input pin of EINT ch.3. [Initial value]
	01	Same as Writing 00.
	10	Uses INT03_1 at the input pin of EINT ch.3.
	11	Uses INT03_2 at the input pin of EINT ch.3.

[bit5:4] EINT02S: External Interrupt Input Select bits

Selects input for EINT02.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses INT02_0 at the input pin of EINT ch.2. [Initial value]
	01	Same as Writing 00.
	10	Uses INT02_1 at the input pin of EINT ch.2.
	11	Uses INT02_2 at the input pin of EINT ch.2.

[bit3:2] EINT01S: External Interrupt Input Select bits

Selects input for EINT01.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Uses INT01_0 at the input pin of EINT ch.1. [Initial value]
	01	Same as Writing 00.
	10	Uses INT01_1 at the input pin of EINT ch.1.
	11	Uses INT01_2 at the input pin of EINT ch.1.

[bit1:0] EINT00S: External Interrupt Input Select bits

Selects input for EINT00.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Uses INT00_0 at the input pin of EINT ch.0. [Initial value]
	01	Same as Writing 00.
	10	Uses INT00_1 at the input pin of EINT ch.0.
	11	Uses INT00_2 at the input pin of EINT ch.0.

4.15 Extended Pin Function Setting Register 07 (EPFR07)

The EPFR07 register assigns functions of multi-function serial ch.0 to ch.3.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved				SCK3B		SOT3B	
Attribute	-				R/W		R/W	
Initial value	-				00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN3S		SCK2B		SOT2B		SIN2S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK1B		SOT1B		SIN1S		SCK0B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SOT0B		SIN0S		Reserved			
Attribute	R/W		R/W		-			
Initial value	00		00		-			

Register Function

[bit31:28] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

[bit27:26] SCK3B: SCK3 Input/Output Select bits

Selects input/output for SCK3.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Uses SCK3_0 at the input pin of MFS ch.3 SCK. Does not produce output. [Initial value]
	01	Uses SCK3_0 at the input pin of MFS ch.3 SCK. Uses SCK3_0 at the output pin.
	10	Uses SCK3_1 at the input pin of MFS ch.3 SCK. Uses SCK3_1 at the output pin.
	11	Uses SCK3_2 at the input pin of MFS ch.3 SCK. Uses SCK3_2 at the output pin.

[bit25:24] SOT3B: SOT3 Input/Output Select bits

Selects input/output for SOT3.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses SOT3_0 at the input pin of MFS ch.3 SOT. Does not produce output. [Initial value]
	01	Uses SOT3_0 at the input pin of MFS ch.3 SOT. Uses SOT3_0 at the output pin.
	10	Uses SOT3_1 at the input pin of MFS ch.3 SOT. Uses SOT3_1 at the output pin.
	11	Uses SOT3_2 at the input pin of MFS ch.3 SOT. Uses SOT3_2 at the output pin.

[bit23:22] SIN3S: SIN3 Input Select bits

Selects input for SIN3.

bit23:22		Description
Reading		Reads out the register value.
Writing	00	Uses SIN3_0 at the input pin of MFS ch.3 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN3_1 at the input pin of MFS ch.3 SIN.
	11	Uses SIN3_2 at the input pin of MFS ch.3 SIN.

[bit21:20] SCK2B: SCK2 Input/Output Select bits

Selects input/output for SCK2.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses SCK2_0 at the input pin of MFS ch.2 SCK. Does not produce output. [Initial value]
	01	Uses SCK2_0 at the input pin of MFS ch.2 SCK. Uses SCK2_0 at the output pin.
	10	Uses SCK2_1 at the input pin of MFS ch.2 SCK. Uses SCK2_1 at the output pin.
	11	Uses SCK2_2 at the input pin of MFS ch.2 SCK. Uses SCK2_2 at the output pin.

[bit19:18] SOT2B: SOT2 Input/Output Select bits

Selects input/output for SOT2.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses SOT2_0 at the input pin of MFS ch.2 SOT. Does not produce output. [Initial value]
	01	Uses SOT2_0 at the input pin of MFS ch.2 SOT. Uses SOT2_0 at the output pin.
	10	Uses SOT2_1 at the input pin of MFS ch.2 SOT. Uses SOT2_1 at the output pin.
	11	Uses SOT2_2 at the input pin of MFS ch.2 SOT. Uses SOT2_2 at the output pin.

[bit17:16] SIN2S: SIN2 Input Select bits

Selects input for SIN2.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses SIN2_0 at the input pin of MFS ch.2 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN2_1 at the input pin of MFS ch.2 SIN.
	11	Uses SIN2_2 at the input pin of MFS ch.2 SIN.

[bit15:14] SCK1B: SCK1 Input/Output Select bits

Selects input/output for SCK1.

bit15:14		Description
Reading		Reads out the register value.
Writing	00	Uses SCK1_0 at the input pin of MFS ch.1 SCK. Does not produce output. [Initial value]
	01	Uses SCK1_0 at the input pin of MFS ch.1 SCK. Uses SCK1_0 at the output pin.
	10	Uses SCK1_1 at the input pin of MFS ch.1 SCK. Uses SCK1_1 at the output pin.
	11	Uses SCK1_2 at the input pin of MFS ch.1 SCK. Uses SCK1_2 at the output pin.

[bit13:12] SOT1B: SOT1 Input/Output Select bits

Selects input/output for SOT1.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses SOT1_0 at the input pin of MFS ch.1 SOT. Does not produce output. [Initial value]
	01	Uses SOT1_0 at the input pin of MFS ch.1 SOT. Uses SOT1_0 at the output pin.
	10	Uses SOT1_1 at the input pin of MFS ch.1 SOT. Uses SOT1_1 at the output pin.
	11	Uses SOT1_2 at the input pin of MFS ch.1 SOT. Uses SOT1_2 at the output pin.

[bit11:10] SIN1S: SIN1 Input Select bits

Selects input for SIN1.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses SIN1_0 at the input pin of MFS ch.1 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN1_1 at the input pin of MFS ch.1 SIN.
	11	Uses SIN1_2 at the input pin of MFS ch.1 SIN.

[bit9:8] SCK0B: SCK0 Input/Output Select bits

Selects input/output for SCK0.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses SCK0_0 at the input pin of MFS ch.0 SCK. Does not produce output. [Initial value]
	01	Uses SCK0_0 at the input pin of MFS ch.0 SCK. Uses SCK0_0 at the output pin.
	10	Uses SCK0_1 at the input pin of MFS ch.0 SCK. Uses SCK0_1 at the output pin.
	11	Uses SCK0_2 at the input pin of MFS ch.0 SCK. Uses SCK0_2 at the output pin.

[bit7:6] SOT0B: SOT0 Input/Output Select bits

Selects input/output for SOT0.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses SOT0_0 at the input pin of MFS ch.0 SOT. Does not produce output. [Initial value]
	01	Uses SOT0_0 at the input pin of MFS ch.0 SOT. Uses SOT0_0 at the output pin.
	10	Uses SOT0_1 at the input pin of MFS ch.0 SOT. Uses SOT0_1 at the output pin.
	11	Uses SOT0_2 at the input pin of MFS ch.0 SOT. Uses SOT0_2 at the output pin.

[bit5:4] SIN0S: SIN0 Input Select bits

Selects input for SIN0.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses SIN0_0 at the input pin of MFS ch.0 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN0_1 at the input pin of MFS ch.0 SIN.
	11	Uses SIN0_2 at the input pin of MFS ch.0 SIN.

[bit3:0] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

4.16 Extended Pin Function Setting Register 08 (EPFR08)

The EPFR08 register assigns functions of multi-function serial ch.4 to ch.7.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved				SCK7B		SOT7B	
Attribute	-				R/W		R/W	
Initial value	-				00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN7S		SCK6B		SOT6B		SIN6S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK5B		SOT5B		SIN5S		SCK4B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SOT4B		SIN4S		CTS4S		RTS4E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:28] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

[bit27:26] SCK7B: SCK7 Input/Output Select bits

Selects input/output for SCK7.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Uses SCK7_0 at the input pin of MFS ch.7 SCK. Does not produce output. [Initial value]
	01	Uses SCK7_0 at the input pin of MFS ch.7 SCK. Uses SCK7_0 at the output pin.
	10	Uses SCK7_1 at the input pin of MFS ch.7 SCK. Uses SCK7_1 at the output pin.
	11	Uses SCK7_2 at the input pin of MFS ch.7 SCK. Uses SCK7_2 at the output pin.

[bit25:24] SOT7B: SOT7 Input/Output Select bits

Selects input/output for SOT7.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses SOT7_0 at the input pin of MFS ch.7 SOT. Does not produce output. [Initial value]
	01	Uses SOT7_0 at the input pin of MFS ch.7 SOT. Uses SOT7_0 at the output pin.
	10	Uses SOT7_1 at the input pin of MFS ch.7 SOT. Uses SOT7_1 at the output pin.
	11	Uses SOT7_2 at the input pin of MFS ch.7 SOT. Uses SOT7_2 at the output pin.

[bit23:22] SIN7S: SIN7 Input Select bits

Selects input for SIN7.

bit23:22		Description
Reading		Reads out the register value.
Writing	00	Uses SIN7_0 at the input pin of MFS ch.7 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN7_1 at the input pin of MFS ch.7 SIN.
	11	Uses SIN7_2 at the input pin of MFS ch.7 SIN.

[bit21:20] SCK6B: SCK6 Input/Output Select bits

Selects input/output for SCK6.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses SCK6_0 at the input pin of MFS ch.6 SCK. Does not produce output. [Initial value]
	01	Uses SCK6_0 at the input pin of MFS ch.6 SCK. Uses SCK6_0 at the output pin.
	10	Uses SCK6_1 at the input pin of MFS ch.6 SCK. Uses SCK6_1 at the output pin.
	11	Uses SCK6_2 at the input pin of MFS ch.6 SCK. Uses SCK6_2 at the output pin.

[bit19:18] SOT6B: SOT6 Input/Output Select bits

Selects input/output for SOT6.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses SOT6_0 at the input pin of MFS ch.6 SOT. Does not produce output. [Initial value]
	01	Uses SOT6_0 at the input pin of MFS ch.6 SOT. Uses SOT6_0 at the output pin.
	10	Uses SOT6_1 at the input pin of MFS ch.6 SOT. Uses SOT6_1 at the output pin.
	11	Uses SOT6_2 at the input pin of MFS ch.6 SOT. Uses SOT6_2 at the output pin.

[bit17:16] SIN6S: SIN6 Input Select bits

Selects input for SIN6.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses SIN6_0 at the input pin of MFS ch.6 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN6_1 at the input pin of MFS ch.6 SIN.
	11	Uses SIN6_2 at the input pin of MFS ch.6 SIN.

[bit15:14] SCK5B: SCK5 Input/Output Select bits

Selects input/output for SCK5.

bit15:14		Description
Reading		Reads out the register value.
Writing	00	Uses SCK5_0 at the input pin of MFS ch.5 SCK. Does not produce output. [Initial value]
	01	Uses SCK5_0 at the input pin of MFS ch.5 SCK. Uses SCK5_0 at the output pin.
	10	Uses SCK5_1 at the input pin of MFS ch.5 SCK. Uses SCK5_1 at the output pin.
	11	Uses SCK5_2 at the input pin of MFS ch.5 SCK. Uses SCK5_2 at the output pin.

[bit13:12] SOT5B: SOT5 Input/Output Select bits

Selects input/output for SOT5.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses SOT5_0 at the input pin of MFS ch.5 SOT. Does not produce output. [Initial value]
	01	Uses SOT5_0 at the input pin of MFS ch.5 SOT. Uses SOT5_0 at the output pin.
	10	Uses SOT5_1 at the input pin of MFS ch.5 SOT. Uses SOT5_1 at the output pin.
	11	Uses SOT5_2 at the input pin of MFS ch.5 SOT. Uses SOT5_2 at the output pin.

[bit11:10] SIN5S: SIN5 Input Select bits

Selects input for SIN5.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses SIN5_0 at the input pin of MFS ch.5 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN5_1 at the input pin of MFS ch.5 SIN.
	11	Uses SIN5_2 at the input pin of MFS ch.5 SIN.

[bit9:8] SCK4B: SCK4 Input/Output Select bits

Selects input/output for SCK4.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses SCK4_0 at the input pin of MFS ch.4 SCK. Does not produce output. [Initial value]
	01	Uses SCK4_0 at the input pin of MFS ch.4 SCK. Uses SCK4_0 at the output pin.
	10	Uses SCK4_1 at the input pin of MFS ch.4 SCK. Uses SCK4_1 at the output pin.
	11	Uses SCK4_2 at the input pin of MFS ch.4 SCK. Uses SCK4_2 at the output pin.

[bit7:6] SOT4B: SOT4 Input/Output Select bits

Selects input/output for SOT4.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses SOT4_0 at the input pin of MFS ch.4 SOT. Does not produce output. [Initial value]
	01	Uses SOT4_0 at the input pin of MFS ch.4 SOT. Uses SOT4_0 at the output pin.
	10	Uses SOT4_1 at the input pin of MFS ch.4 SOT. Uses SOT4_1 at the output pin.
	11	Uses SOT4_2 at the input pin of MFS ch.4 SOT. Uses SOT4_2 at the output pin.

[bit5:4] SIN4S: SIN4 Input Select bits

Selects input for SIN4.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses SIN4_0 at the input pin of MFS ch.4 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN4_1 at the input pin of MFS ch.4 SIN.
	11	Uses SIN4_2 at the input pin of MFS ch.4 SIN.

[bit3:2] CTS4S: CTS4 Input Select bits

Selects input for CTS4.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Uses CTS4_0 at the input pin of MFS ch.4 CTS. [Initial value]
	01	Same as Writing 00.
	10	Uses CTS4_1 at the input pin of MFS ch.4 CTS.
	11	Uses CTS4_2 at the input pin of MFS ch.4 CTS.

[bit1:0] RTS4E: RTS4 Output Select bits

Selects output for RTS4.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for MFS ch.4 RTS. [Initial value]
	01	Uses RTS4_0 at the output pin of MFS ch.4 RTS.
	10	Uses RTS4_1 at the output pin of MFS ch.4 RTS.
	11	Uses RTS4_2 at the output pin of MFS ch.4 RTS.

4.17 Extended Pin Function Setting Register 09 (EPFR09)

The EPFR09 register assigns functions to ADC trigger, and QPRC peripheral pins.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	ADTRG2S				ADTRG1S			
Attribute	R/W				R/W			
Initial value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	ADTRG0S				QZIN1S		QBIN1S	
Attribute	R/W				R/W		R/W	
Initial value	0000				00		00	

bit	7	6	5	4	3	2	1	0
Field	QAIN1S		QZIN0S		QBIN0S		QAIN0S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:24] Reserved: Reserved bits

"0b00000000" is read out from these bits.

When writing these bits, set them to "0b00000000".

[bit23:20] ADTRG2S: ADTRG2 Input Select bits

Selects input for ADTRG2.

bit23:20		Description
Reading		Reads out the register value.
Writing	0000	Uses ADTG_0 at the input pin of ADC unit 2's startup trigger. [Initial value]
	0001	Same as Writing 0000.
	0010	Uses ADTG_1 at the input pin of ADC unit 2's startup trigger.
	0011	Uses ADTG_2 at the input pin of ADC unit 2's startup trigger.
	0100	Uses ADTG_3 at the input pin of ADC unit 2's startup trigger.
	0101	Uses ADTG_4 at the input pin of ADC unit 2's startup trigger.
	0110	Uses ADTG_5 at the input pin of ADC unit 2's startup trigger.
	0111	Uses ADTG_6 at the input pin of ADC unit 2's startup trigger.
	1000	Uses ADTG_7 at the input pin of ADC unit 2's startup trigger.

	1001	Uses ADTG_8 at the input pin of ADC unit 2's startup trigger.
Writing other data		Setting is prohibited.

[bit19:16] ADTRG1S: ADTRG1 Input Select bits

Selects input for ADTRG1.

bit19:16		Description
Reading		Reads out the register value.
Writing	0000	Uses ADTG_0 at the input pin of ADC unit 1's startup trigger. [Initial value]
	0001	Same as Writing 0000.
	0010	Uses ADTG_1 at the input pin of ADC unit 1's startup trigger.
	0011	Uses ADTG_2 at the input pin of ADC unit 1's startup trigger.
	0100	Uses ADTG_3 at the input pin of ADC unit 1's startup trigger.
	0101	Uses ADTG_4 at the input pin of ADC unit 1's startup trigger.
	0110	Uses ADTG_5 at the input pin of ADC unit 1's startup trigger.
	0111	Uses ADTG_6 at the input pin of ADC unit 1's startup trigger.
	1000	Uses ADTG_7 at the input pin of ADC unit 1's startup trigger.
	1001	Uses ADTG_8 at the input pin of ADC unit 1's startup trigger.
Writing other data		Setting is prohibited.

[bit15:12] ADTRG0S: ADTRG0 Input Select bits

Selects input for ADTRG0.

bit15:12		Description
Reading		Reads out the register value.
Writing	0000	Uses ADTG_0 at the input pin of ADC unit 0's startup trigger. [Initial value]
	0001	Same as Writing 0000.
	0010	Uses ADTG_1 at the input pin of ADC unit 0's startup trigger.
	0011	Uses ADTG_2 at the input pin of ADC unit 0's startup trigger.
	0100	Uses ADTG_3 at the input pin of ADC unit 0's startup trigger.
	0101	Uses ADTG_4 at the input pin of ADC unit 0's startup trigger.
	0110	Uses ADTG_5 at the input pin of ADC unit 0's startup trigger.
	0111	Uses ADTG_6 at the input pin of ADC unit 0's startup trigger.
	1000	Uses ADTG_7 at the input pin of ADC unit 0's startup trigger.
	1001	Uses ADTG_8 at the input pin of ADC unit 0's startup trigger.
Writing other data		Setting is prohibited.

[bit11:10] QZIN1S: QZIN1S Input Select bits

Selects input for QPRC ZIN1.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses ZIN1_0 at the input pin of QPRC ch.1's ZIN. [Initial value]
	01	Same as Writing 00.
	10	Uses ZIN1_1 at the input pin of QPRC ch.1's ZIN.
	11	Uses ZIN1_2 at the input pin of QPRC ch.1's ZIN.

[bit9:8] QBIN1S: QBIN1S Input Select bits

Selects input for QPRC BIN1.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses BIN1_0 at the input pin of QPRC ch.1's BIN. [Initial value]
	01	Same as Writing 00.
	10	Uses BIN1_1 at the input pin of QPRC ch.1's BIN.
	11	Uses BIN1_2 at the input pin of QPRC ch.1's BIN.

[bit7:6] QAIN1S: QAIN1S Input Select bits

Selects input for QPRC AIN1.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses AIN1_0 at the input pin of QPRC ch.1's AIN. [Initial value]
	01	Same as Writing 00.
	10	Uses AIN1_1 at the input pin of QPRC ch.1's AIN.
	11	Uses AIN1_2 at the input pin of QPRC ch.1's AIN.

[bit5:4] QZIN0S: QZIN0S Input Select bits

Selects input for QPRC ZIN0. If ZIN0_3 is used, set this bit together with QZIN0S[2] bit in Extended Pin Function Setting Register 21 (EPFR21).

EPFR21[2],bit5:4		Description
Reading		Reads out the register value.
Writing	000	Uses ZIN0_0 at the input pin of QPRC ch.0's ZIN. [Initial value]
	001	Same as Writing 00.
	010	Uses ZIN0_1 at the input pin of QPRC ch.0's ZIN.
	011	Uses ZIN0_2 at the input pin of QPRC ch.0's ZIN.
	100	Uses ZIN0_3 at the input pin of QPRC ch.0's ZIN.
	Others	Setting is prohibited.

[bit3:2] QBIN0S: QBIN0S Input Select bits

Selects input for QPRC BIN0. If BIN0_3 is used, set this bit together with QBIN0S[2] bit in Extended Pin Function Setting Register 21 (EPFR21).

EPFR21[1],bit3:2		Description
Reading		Reads out the register value.
Writing	000	Uses BIN0_0 at the input pin of QPRC ch.0's BIN. [Initial value]
	001	Same as Writing 00.
	010	Uses BIN0_1 at the input pin of QPRC ch.0's BIN.
	011	Uses BIN0_2 at the input pin of QPRC ch.0's BIN.
	100	Uses BIN0_3 at the input pin of QPRC ch.0's BIN.
	Others	Setting is prohibited

[bit1:0] QAIN0S: QAIN0S Input Select bits

Selects input for QPRC AIN0. If AIN0_3 is used, set this bit together with QAIN0S[2] bit in Extended Pin Function Setting Register 21 (EPFR21)

EPFR21[0],bit1:0		Description
Reading		Reads out the register value.
Writing	000	Uses AIN0_0 at the input pin of QPRC ch.0's AIN. [Initial value]
	001	Same as Writing 00.
	010	Uses AIN0_1 at the input pin of QPRC ch.0's AIN.
	011	Uses AIN0_2 at the input pin of QPRC ch.0's AIN.
	100	Uses AIN0_3 at the input pin of QPRC ch.0's AIN.
	Others	Setting is prohibited.

4.18 Extended Pin Function Setting Register 12 (EPFR12)

The EPFR12 register assigns functions to pins of ch.8, ch.9, ch.10, and ch.11 of the base timer.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB11S		TIOA11E		TIOA11S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB10S		TIOA10E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB9S		TIOA9E		TIOA9S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved		TIOB8S		TIOA8E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit29:28] TIOB11S: TIOB11 Input Select bits

Selects input for TIOB11.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB11_0 at the input pin of BT ch.11 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB11_1 at the input pin of BT ch.11 TIOB.
	11	Uses TIOB11_2 at the input pin of BT ch.11 TIOB.

[bit27:26] TIOA11E: TIOA11 Output Select bits

Selects output for TIOA11.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.11 TIOA. [Initial value]
	01	Uses TIOA11_0 at the output pin of BT ch.11 TIOA.
	10	Uses TIOA11_1 at the output pin of BT ch.11 TIOA.
	11	Uses TIOA11_2 at the output pin of BT ch.11 TIOA.

[bit25:24] TIOA11S: TIOA11 Input Select bits

Selects input for TIOA11.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA11_0 at the input pin of BT ch.11 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA11_1 at the input pin of BT ch.11 TIOA.
	11	Uses TIOA11_2 at the input pin of BT ch.11 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit21:20] TIOB10S: TIOB10 Input Select bits

Selects input for TIOB10.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB10_0 at the input pin of BT ch.10 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB10_1 at the input pin of BT ch.10 TIOB.
	11	Uses TIOB10_2 at the input pin of BT ch.10 TIOB.

[bit19:18] TIOA10E: TIOA10 Output Select bits

Selects output for TIOA10.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.10 TIOA. [Initial value]
	01	Uses TIOA10_0 at the output pin of BT ch.10 TIOA.
	10	Uses TIOA10_1 at the output pin of BT ch.10 TIOA.
	11	Uses TIOA10_2 at the output pin of BT ch.10 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.

When writing these bits, set them to "0b0000".

[bit13:12] TIOB9S: TIOB9 Input Select bits

Selects input for TIOB9.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB9_0 at the input pin of BT ch.9 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB9_1 at the input pin of BT ch.9 TIOB.
	11	Uses TIOB9_2 at the input pin of BT ch.9 TIOB.

[bit11:10] TIOA9E: TIOA9 Output Select bits

Selects output for TIOA9.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.9 TIOA. [Initial value]
	01	Uses TIOA9_0 at the output pin of BT ch.9 TIOA.
	10	Uses TIOA9_1 at the output pin of BT ch.9 TIOA.
	11	Uses TIOA9_2 at the output pin of BT ch.9 TIOA.

[bit9:8] TIOA9S: TIOA9 Input Select bits

Selects input for TIOA9.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA9_0 at the input pin of BT ch.9 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA9_1 at the input pin of BT ch.9 TIOA.
	11	Uses TIOA9_2 at the input pin of BT ch.9 TIOA.

[bit7:6] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit5:4] TIOB8S: TIOB8 Input Select bits

Selects input for TIOB8.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB8_0 at the input pin of BT ch.8 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB8_1 at the input pin of BT ch.8 TIOB.
	11	Uses TIOB8_2 at the input pin of BT ch.8 TIOB.

[bit3:2] TIOA8E: TIOA8 Output Select bits

Selects output for TIOA8.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.8 TIOA. [Initial value]
	01	Uses TIOA8_0 at the output pin of BT ch.8 TIOA.
	10	Uses TIOA8_1 at the output pin of BT ch.8 TIOA.
	11	Uses TIOA8_2 at the output pin of BT ch.8 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

Notes:

- *TIOA*
  Even channels are for output only.
  Odd channels are for both input and output.
- *TIOB*
  Input only.
- *TIOA9, TIOA11, TIOA13, TIOA15 (odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.*
  When output is chosen for odd TIOA channel, input setting will be ignored.

  Example1: Use TIOA11 as an output pin:
  When TIOA11 is output to TIOA11_0, select EPFR12:TIOA11E = 01.
  When TIOA11 is output to TIOA11_1, select EPFR12:TIOA11E = 10.
  When TIOA11 is output to TIOA11_2, select EPFR12:TIOA11E = 11.
  Settings for EPFR12:TIOA11S will be ignored.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

  Example2: When TIOA11 is used as an input pin:
  Select EPFR12:TIOA11E = 00.

When TIOA11 is input from TIOA11_0, select EPFR12:TIOA11S = 00 or 01.
When TIOA11 is input from TIOA11_1, select EPFR12:TIOA11S = 10.
When TIOA11 is input from TIOA11_2, select EPFR12:TIOA11S = 11.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

** When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.*

4.19 Extended Pin Function Setting Register 13 (EPFR13)

The EPFR13 register assigns functions to pins of ch.12, ch.13, ch.14, and ch.15 of the base timer.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB15S		TIOA15E		TIOA15S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB14S		TIOA14E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB13S		TIOA13E		TIOA13S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved		TIOB12S		TIOA12E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit29:28] TIOB15S: TIOB15 Input Select bits

Selects input for TIOB15.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB15_0 at the input pin of BT ch.15 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB15_1 at the input pin of BT ch.15 TIOB.
	11	Uses TIOB15_2 at the input pin of BT ch.15 TIOB.

[bit27:26] TIOA15E: TIOA15 Output Select bits

Selects output for TIOA15.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.15 TIOA. [Initial value]
	01	Uses TIOA15_0 at the output pin of BT ch.15 TIOA.
	10	Uses TIOA15_1 at the output pin of BT ch.15 TIOA.
	11	Uses TIOA15_2 at the output pin of BT ch.15 TIOA.

[bit25:24] TIOA15S: TIOA15 Input Select bits

Selects input for TIOA15.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA15_0 at the input pin of BT ch.15 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA15_1 at the input pin of BT ch.15 TIOA.
	11	Uses TIOA15_2 at the input pin of BT ch.15 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit21:20] TIOB14S: TIOB14 Input Select bits

Selects input for TIOB14.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB14_0 at the input pin of BT ch.14 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB14_1 at the input pin of BT ch.14 TIOB.
	11	Uses TIOB14_2 at the input pin of BT ch.14 TIOB.

[bit19:18] TIOA14E: TIOA14 Output Select bits

Selects output for TIOA14.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.14 TIOA. [Initial value]
	01	Uses TIOA14_0 at the output pin of BT ch.14 TIOA.
	10	Uses TIOA14_1 at the output pin of BT ch.14 TIOA.
	11	Uses TIOA14_2 at the output pin of BT ch.14 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.

When writing these bits, set them to "0b0000".

[bit13:12] TIOB13S: TIOB13 Input Select bits

Selects input for TIOB13.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB13_0 at the input pin of BT ch.13 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB13_1 at the input pin of BT ch.13 TIOB.
	11	Uses TIOB13_2 at the input pin of BT ch.13 TIOB.

[bit11:10] TIOA13E: TIOA13 Output Select bits

Selects output for TIOA13.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.13 TIOA. [Initial value]
	01	Uses TIOA13_0 at the output pin of BT ch.13 TIOA.
	10	Uses TIOA13_1 at the output pin of BT ch.13 TIOA.
	11	Uses TIOA13_2 at the output pin of BT ch.13 TIOA.

[bit9:8] TIOA13S: TIOA13 Input Select bits

Selects input for TIOA13.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA13_0 at the input pin of BT ch.13 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA13_1 at the input pin of BT ch.13 TIOA.
	11	Uses TIOA13_2 at the input pin of BT ch.13 TIOA.

[bit7:6] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

[bit5:4] TIOB12S: TIOB12 Input Select bits

Selects input for TIOB12.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB12_0 at the input pin of BT ch.12 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB12_1 at the input pin of BT ch.12 TIOB.
	11	Uses TIOB12_2 at the input pin of BT ch.12 TIOB.

[bit3:2] TIOA12E: TIOA12 Output Select bits

Selects output for TIOA12.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.12 TIOA. [Initial value]
	01	Uses TIOA12_0 at the output pin of BT ch.12 TIOA.
	10	Uses TIOA12_1 at the output pin of BT ch.12 TIOA.
	11	Uses TIOA12_2 at the output pin of BT ch.12 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.

When writing these bits, set them to "0b00".

Notes:

- *TIOA*
  Even channels are for output only.
  Odd channels are for both input and output.
- *TIOB*
  Input only.
- *TIOA9, TIOA11, TIOA13, TIOA15 (odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.*
  When output is chosen for odd TIOA channel, input setting will be ignored.

  Example1: Use TIOA11 as an output pin:
  When TIOA11 is output to TIOA11_0, select EPFR12:TIOA11E = 01.
  When TIOA11 is output to TIOA11_1, select EPFR12:TIOA11E = 10.
  When TIOA11 is output to TIOA11_2, select EPFR12:TIOA11E = 11.
  Settings for EPFR12:TIOA11S will be ignored.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

  Example2: When TIOA11 is used as an input pin:
  Select EPFR12:TIOA11E = 00.

When TIOA11 is input from TIOA11_0, select EPFR12:TIOA11S = 00 or 01.
When TIOA11 is input from TIOA11_1, select EPFR12:TIOA11S = 10.
When TIOA11 is input from TIOA11_2, select EPFR12:TIOA11S = 11.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

** When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.*

4.20 Extended Pin Function Setting Register 14 (EPFR14)

EPFR14 register sets the function assignment to QPRC pins.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved		QZIN2S		QBIN2S		QAIN2S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

Register Function

[bit31:6] Reserved: Reserved bits

"0" is read out from these bits.

When writing these bits, set them to "0".

[bit5:4] QZIN2S: QPRC-ch.2 ZIN Input Pin bits

Selects input for QPRC-ch.2 as ZIN.

Bit		Description
Reading		Reads out the register value.
Writing	00	ZIN2_0 is used as ZIN, the input pin of QPRC ch.2. [Initial value]
	01	ZIN2_0 is used as ZIN, the input pin of QPRC ch.2.
	10	ZIN2_1 is used as ZIN, the input pin of QPRC ch.2.
	11	ZIN2_2 is used as ZIN, the input pin of QPRC ch.2.

[bit3:2] QBIN2S: QPRC-ch.2 BIN Input Pin bits

Selects input for QPRC-ch.2 as BIN.

Bit		Description
Reading		Reads out the register value.
Writing	00	BIN2_0 is used as BIN, the input pin of QPRC ch.2. [Initial value]
	01	BIN2_0 is used as BIN, the input pin of QPRC ch.2.
	10	BIN2_1 is used as BIN, the input pin of QPRC ch.2.
	11	BIN2_2 is used as BIN, the input pin of QPRC ch.2.

[bit1:0] QAIN2S: QPRC-ch.2 AIN Input Pin bits

Selects input for QPRC-ch.2 as AIN.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	AIN2_0 is used as AIN, the input pin of QPRC ch.2. [Initial value]
	01	AIN2_0 is used as AIN, the input pin of QPRC ch.2.
	10	AIN2_1 is used as AIN, the input pin of QPRC ch.2.
	11	AIN2_2 is used as AIN, the input pin of QPRC ch.2.

4.21 Extended Pin Function Setting Register 15 (EPFR15)

EPFR15 register sets the function assignment to external interrupt pins.

Register Configuration

bit	31 30	29 28	27 26	25 24
Field	EINT31S	EINT30S	EINT29S	EINT28S
Attribute	R/W	R/W	R/W	R/W
Initial value	00	00	00	00

bit	23 22	21 20	19 18	17 16
Field	EINT27S	EINT26S	EINT25S	EINT24S
Attribute	R/W	R/W	R/W	R/W
Initial value	00	00	00	00

bit	15 14	13 12	11 10	9 8
Field	EINT23S	EINT22S	EINT21S	EINT20S
Attribute	R/W	R/W	R/W	R/W
Initial value	00	00	00	00

bit	7 6	5 4	3 2	1 0
Field	EINT19S	EINT18S	EINT17S	EINT16S
Attribute	R/W	R/W	R/W	R/W
Initial value	00	00	00	00

Register Function

[bit31:30] EINT31S: External Interrupt Input Select bits

Selects input for EINT31.

bit31:30		Description
Reading		Reads out the register value.
Writing	00	Uses INT31_0 at the input pin of EINT ch.31. [Initial value]
	01	Same as Writing 00.
	10	Uses INT31_1 at the input pin of EINT ch.31.
	11	Uses INT31_2 at the input pin of EINT ch.31.

[bit29:28] EINT30S: External Interrupt Input Select bits

Selects input for EINT30.

bit29:28		Description
Reading		Reads out the register value.
Writing	00	Uses INT30_0 at the input pin of EINT ch.30. [Initial value]
	01	Same as Writing 00.
	10	Uses INT30_1 at the input pin of EINT ch.30.
	11	Uses INT30_2 at the input pin of EINT ch.30.

[bit27:26] EINT29S: External Interrupt Input Select bits

Selects input for EINT29.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Uses INT29_0 at the input pin of EINT ch.29. [Initial value]
	01	Same as Writing 00.
	10	Uses INT29_1 at the input pin of EINT ch.29.
	11	Uses INT29_2 at the input pin of EINT ch.29.

[bit25:24] EINT28S: External Interrupt Input Select bits

Selects input for EINT28.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses INT28_0 at the input pin of EINT ch.28. [Initial value]
	01	Same as Writing 00.
	10	Uses INT28_1 at the input pin of EINT ch.28.
	11	Uses INT28_2 at the input pin of EINT ch.28.

[bit23:22] EINT27S: External Interrupt Input Select bits

Selects input for EINT27.

bit23:22		Description
Reading		Reads out the register value.
Writing	00	Uses INT27_0 at the input pin of EINT ch.27. [Initial value]
	01	Same as Writing 00.
	10	Uses INT27_1 at the input pin of EINT ch.27.
	11	Uses INT27_2 at the input pin of EINT ch.27.

[bit21:20] EINT26S: External Interrupt Input Select bits

Selects input for EINT26.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses INT26_0 at the input pin of EINT ch.26. [Initial value]
	01	Same as Writing 00.
	10	Uses INT26_1 at the input pin of EINT ch.26.
	11	Uses INT26_2 at the input pin of EINT ch.26.

[bit19:18] EINT25S: External Interrupt Input Select bits

Selects input for EINT25.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses INT25_0 at the input pin of EINT ch.25. [Initial value]
	01	Same as Writing 00.
	10	Uses INT25_1 at the input pin of EINT ch.25.
	11	Uses INT25_2 at the input pin of EINT ch.25.

[bit17:16] EINT24S: External Interrupt Input Select bits

Selects input for EINT24.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses INT24_0 at the input pin of EINT ch.24. [Initial value]
	01	Same as Writing 00.
	10	Uses INT24_1 at the input pin of EINT ch.24.
	11	Uses INT24_2 at the input pin of EINT ch.24.

[bit15:14] EINT23S: External Interrupt Input Select bits

Selects input for EINT23.

bit15:14		Description
Reading		Reads out the register value.
Writing	00	Uses INT23_0 at the input pin of EINT ch.23. [Initial value]
	01	Same as Writing 00.
	10	Uses INT23_1 at the input pin of EINT ch.23.
	11	Uses INT23_2 at the input pin of EINT ch.23.

[bit13:12] EINT22S: External Interrupt Input Select bits

Selects input for EINT22.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses INT22_0 at the input pin of EINT ch.22. [Initial value]
	01	Same as Writing 00.
	10	Uses INT22_1 at the input pin of EINT ch.22.
	11	Uses INT22_2 at the input pin of EINT ch.22.

[bit11:10] EINT21S: External Interrupt Input Select bits

Selects input for EINT21.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses INT21_0 at the input pin of EINT ch.21. [Initial value]
	01	Same as Writing 00.
	10	Uses INT21_1 at the input pin of EINT ch.21.
	11	Uses INT21_2 at the input pin of EINT ch.21.

[bit9:8] EINT20S: External Interrupt Input Select bits

Selects input for EINT20.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses INT20_0 at the input pin of EINT ch.20. [Initial value]
	01	Same as Writing 00.
	10	Uses INT20_1 at the input pin of EINT ch.20.
	11	Uses INT20_2 at the input pin of EINT ch.20.

[bit7:6] EINT19S: External Interrupt Input Select bits

Selects input for EINT19.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses INT19_0 at the input pin of EINT ch.19. [Initial value]
	01	Same as Writing 00.
	10	Uses INT19_1 at the input pin of EINT ch.19.
	11	Uses INT19_2 at the input pin of EINT ch.19.

[bit5:4] EINT18S: External Interrupt Input Select bits

Selects input for EINT18.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses INT18_0 at the input pin of EINT ch.18. [Initial value]
	01	Same as Writing 00.
	10	Uses INT18_1 at the input pin of EINT ch.18.
	11	Uses INT18_2 at the input pin of EINT ch.18.

[bit3:2] EINT17S: External Interrupt Input Select bits

Selects input for EINT17.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Uses INT17_0 at the input pin of EINT ch.17. [Initial value]
	01	Same as Writing 00.
	10	Uses INT17_1 at the input pin of EINT ch.17.
	11	Uses INT17_2 at the input pin of EINT ch.17.

[bit1:0] EINT16S: External Interrupt Input Select bits

Selects input for EINT16.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Uses INT16_0 at the input pin of EINT ch.16. [Initial value]
	01	Same as Writing 00.
	10	Uses INT16_1 at the input pin of EINT ch.16.
	11	Uses INT16_2 at the input pin of EINT ch.16.

4.22 Extended Pin Function Setting Register 16 (EPFR16)

The EPFR16 register assigns functions of multi-function serial channel 8, channel 9, channel 10, and channel 11.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved				SCK11B		SOT11B	
Attribute	-				R/W		R/W	
Initial value	-				00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN11S		SCK10B		SOT10B		SIN10S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK9B		SOT9B		SIN9S		SCK8B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SOT8B		SIN8S		Reserved			
Attribute	R/W		R/W		-			
Initial value	00		00		-			

Register Function

[bit31:28] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

[bit27:26] SCK11B: SCK11 Input/Output Select bits

Selects input/output for SCK11.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Uses SCK11_0 at the input pin of MFS ch.11 SCK. Does not produce output. [Initial value]
	01	Uses SCK11_0 at the input pin of MFS ch.11 SCK. Uses SCK11_0 at the output pin.
	10	Uses SCK11_1 at the input pin of MFS ch.11 SCK. Uses SCK11_1 at the output pin.
	11	Uses SCK11_2 at the input pin of MFS ch.11 SCK. Uses SCK11_2 at the output pin.

[bit25:24] SOT11B: SOT11 Input/Output Select bits

Selects input/output for SOT11.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses SOT11_0 at the input pin of MFS ch.11 SOT. Does not produce output. [Initial value]
	01	Uses SOT11_0 at the input pin of MFS ch.11 SOT. Uses SOT11_0 at the output pin.
	10	Uses SOT11_1 at the input pin of MFS ch.11 SOT. Uses SOT11_1 at the output pin.
	11	Uses SOT11_2 at the input pin of MFS ch.11 SOT. Uses SOT11_2 at the output pin.

[bit23:22] SIN11S: SIN11 Input Select bits

Selects input for SIN11.

bit23:22		Description
Reading		Reads out the register value.
Writing	00	Uses SIN11_0 at the input pin of MFS ch.11 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN11_1 at the input pin of MFS ch.11 SIN.
	11	Uses SIN11_2 at the input pin of MFS ch.11 SIN.

[bit21:20] SCK10B: SCK10 Input/Output Select bits

Selects input/output for SCK10.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses SCK10_0 at the input pin of MFS ch.10 SCK. Does not produce output. [Initial value]
	01	Uses SCK10_0 at the input pin of MFS ch.10 SCK. Uses SCK10_0 at the output pin.
	10	Uses SCK10_1 at the input pin of MFS ch.10 SCK. Uses SCK10_1 at the output pin.
	11	Uses SCK10_2 at the input pin of MFS ch.10 SCK. Uses SCK10_2 at the output pin.

[bit19:18] SOT10B: SOT10 Input/Output Select bits

Selects input/output for SOT10.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses SOT10_0 at the input pin of MFS ch.10 SOT. Does not produce output. [Initial value]
	01	Uses SOT10_0 at the input pin of MFS ch.10 SOT. Uses SOT10_0 at the output pin.
	10	Uses SOT10_1 at the input pin of MFS ch.10 SOT. Uses SOT10_1 at the output pin.
	11	Uses SOT10_2 at the input pin of MFS ch.10 SOT. Uses SOT10_2 at the output pin.

[bit17:16] SIN10S: SIN10 Input Select bits

Selects input for SIN10.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses SIN10_0 at the input pin of MFS ch.10 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN10_1 at the input pin of MFS ch.10 SIN.
	11	Uses SIN10_2 at the input pin of MFS ch.10 SIN.

[bit15:14] SCK9B: SCK9 Input/Output Select bits

Selects input/output for SCK9.

bit15:14		Description
Reading		Reads out the register value.
Writing	00	Uses SCK9_0 at the input pin of MFS ch.9 SCK. Does not produce output. [Initial value]
	01	Uses SCK9_0 at the input pin of MFS ch.9 SCK. Uses SCK9_0 at the output pin.
	10	Uses SCK9_1 at the input pin of MFS ch.9 SCK. Uses SCK9_1 at the output pin.
	11	Uses SCK9_2 at the input pin of MFS ch.9 SCK. Uses SCK9_2 at the output pin.

[bit13:12] SOT9B: SOT9 Input/Output Select bits

Selects input/output for SOT9.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses SOT9_0 at the input pin of MFS ch.9 SOT. Does not produce output. [Initial value]
	01	Uses SOT9_0 at the input pin of MFS ch.9 SOT. Uses SOT9_0 at the output pin.
	10	Uses SOT9_1 at the input pin of MFS ch.9 SOT. Uses SOT9_1 at the output pin.
	11	Uses SOT9_2 at the input pin of MFS ch.9 SOT. Uses SOT9_2 at the output pin.

[bit11:10] SIN9S: SIN9 Input Select bits

Selects input for SIN9.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses SIN9_0 at the input pin of MFS ch.9 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN9_1 at the input pin of MFS ch.9 SIN.
	11	Uses SIN9_2 at the input pin of MFS ch.9 SIN.

[bit9:8] SCK8B: SCK8 Input/Output Select bits

Selects input/output for SCK8.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses SCK8_0 at the input pin of MFS ch.8 SCK. Does not produce output. [Initial value]
	01	Uses SCK8_0 at the input pin of MFS ch.8 SCK. Uses SCK8_0 at the output pin.
	10	Uses SCK8_1 at the input pin of MFS ch.8 SCK. Uses SCK8_1 at the output pin.
	11	Uses SCK8_2 at the input pin of MFS ch.8 SCK. Uses SCK8_2 at the output pin.

[bit7:6] SOT8B: SOT8 Input/Output Select bits

Selects input/output for SOT8.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses SOT8_0 at the input pin of MFS ch.8 SOT. Does not produce output. [Initial value]
	01	Uses SOT8_0 at the input pin of MFS ch.8 SOT. Uses SOT8_0 at the output pin.
	10	Uses SOT8_1 at the input pin of MFS ch.8 SOT. Uses SOT8_1 at the output pin.
	11	Uses SOT8_2 at the input pin of MFS ch.8 SOT. Uses SOT8_2 at the output pin.

[bit5:4] SIN8S: SIN8 Input Select bits

Selects input for SIN4.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses SIN8_0 at the input pin of MFS ch.8 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN8_1 at the input pin of MFS ch.8 SIN.
	11	Uses SIN8_2 at the input pin of MFS ch.8 SIN.

[bit3:0] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

4.23 Extended Pin Function Setting Register 17 (EPFR17)

The EPFR17 register assigns functions of multi-function serial channel 12, channel 13, channel 14, and channel 15.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved				SCK15B		SOT15B	
Attribute	-				R/W		R/W	
Initial value	-				00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN15S		SCK14B		SOT14B		SIN14S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK13B		SOT13B		SIN13S		SCK12B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SOT12B		SIN12S		Reserved			
Attribute	R/W		R/W		-			
Initial value	00		00		-			

Register Function

[bit31:28] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

[bit27:26] SCK15B: SCK15 Input/Output Select bits

Selects input/output for SCK15.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Uses SCK15_0 at the input pin of MFS ch.15 SCK. Does not produce output. [Initial value]
	01	Uses SCK15_0 at the input pin of MFS ch.15 SCK. Uses SCK15_0 at the output pin.
	10	Uses SCK15_1 at the input pin of MFS ch.15 SCK. Uses SCK15_1 at the output pin.
	11	Uses SCK15_2 at the input pin of MFS ch.15 SCK. Uses SCK15_2 at the output pin.

[bit25:24] SOT15B: SOT15 Input/Output Select bits

Selects input/output for SOT15.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses SOT15_0 at the input pin of MFS ch.15 SOT. Does not produce output. [Initial value]
	01	Uses SOT15_0 at the input pin of MFS ch.15 SOT. Uses SOT15_0 at the output pin.
	10	Uses SOT15_1 at the input pin of MFS ch.15 SOT. Uses SOT15_1 at the output pin.
	11	Uses SOT15_2 at the input pin of MFS ch.15 SOT. Uses SOT15_2 at the output pin.

[bit23:22] SIN15S: SIN15 Input Select bits

Selects input for SIN15.

bit23:22		Description
Reading		Reads out the register value.
Writing	00	Uses SIN15_0 at the input pin of MFS ch.15 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN15_1 at the input pin of MFS ch.15 SIN.
	11	Uses SIN15_2 at the input pin of MFS ch.15 SIN.

[bit21:20] SCK14B: SCK14 Input/Output Select bits

Selects input/output for SCK14.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses SCK14_0 at the input pin of MFS ch.14 SCK. Does not produce output. [Initial value]
	01	Uses SCK14_0 at the input pin of MFS ch.14 SCK. Uses SCK14_0 at the output pin.
	10	Uses SCK14_1 at the input pin of MFS ch.14 SCK. Uses SCK14_1 at the output pin.
	11	Uses SCK14_2 at the input pin of MFS ch.14 SCK. Uses SCK14_2 at the output pin.

[bit19:18] SOT14B: SOT14 Input/Output Select bits

Selects input/output for SOT14.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Uses SOT14_0 at the input pin of MFS ch.14 SOT. Does not produce output. [Initial value]
	01	Uses SOT14_0 at the input pin of MFS ch.14 SOT. Uses SOT14_0 at the output pin.
	10	Uses SOT14_1 at the input pin of MFS ch.14 SOT. Uses SOT14_1 at the output pin.
	11	Uses SOT14_2 at the input pin of MFS ch.14 SOT. Uses SOT14_2 at the output pin.

[bit17:16] SIN14S: SIN14 Input Select bits

Selects input for SIN14.

bit17:16		Description
Reading		Reads out the register value.
Writing	00	Uses SIN14_0 at the input pin of MFS ch.14 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN14_1 at the input pin of MFS ch.14 SIN.
	11	Uses SIN14_2 at the input pin of MFS ch.14 SIN.

[bit15:14] SCK13B: SCK13 Input/Output Select bits

Selects input/output for SCK13.

bit15:14		Description
Reading		Reads out the register value.
Writing	00	Uses SCK13_0 at the input pin of MFS ch.13 SCK. Does not produce output. [Initial value]
	01	Uses SCK13_0 at the input pin of MFS ch.13 SCK. Uses SCK13_0 at the output pin.
	10	Uses SCK13_1 at the input pin of MFS ch.13 SCK. Uses SCK13_1 at the output pin.
	11	Uses SCK13_2 at the input pin of MFS ch.13 SCK. Uses SCK13_2 at the output pin.

[bit13:12] SOT13B: SOT13 Input/Output Select bits

Selects input/output for SOT13.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses SOT13_0 at the input pin of MFS ch.13 SOT. Does not produce output. [Initial value]
	01	Uses SOT13_0 at the input pin of MFS ch.13 SOT. Uses SOT13_0 at the output pin.
	10	Uses SOT13_1 at the input pin of MFS ch.13 SOT. Uses SOT13_1 at the output pin.
	11	Uses SOT13_2 at the input pin of MFS ch.13 SOT. Uses SOT13_2 at the output pin.

[bit11:10] SIN13S: SIN13 Input Select bits

Selects input for SIN13.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses SIN13_0 at the input pin of MFS ch.13 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN13_1 at the input pin of MFS ch.13 SIN.
	11	Uses SIN13_2 at the input pin of MFS ch.13 SIN.

[bit9:8] SCK12B: SCK12 Input/Output Select bits

Selects input/output for SCK12.

bit9:8		Description
Reading		Reads out the register value.
Writing	00	Uses SCK12_0 at the input pin of MFS ch.12 SCK. Does not produce output. [Initial value]
	01	Uses SCK12_0 at the input pin of MFS ch.12 SCK. Uses SCK12_0 at the output pin.
	10	Uses SCK12_1 at the input pin of MFS ch.12 SCK. Uses SCK12_1 at the output pin.
	11	Uses SCK12_2 at the input pin of MFS ch.12 SCK. Uses SCK12_2 at the output pin.

[bit7:6] SOT12B: SOT12 Input/Output Select bits

Selects input/output for SOT12.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses SOT12_0 at the input pin of MFS ch.12 SOT. Does not produce output. [Initial value]
	01	Uses SOT12_0 at the input pin of MFS ch.12 SOT. Uses SOT12_0 at the output pin.
	10	Uses SOT12_1 at the input pin of MFS ch.12 SOT. Uses SOT12_1 at the output pin.
	11	Uses SOT12_2 at the input pin of MFS ch.12 SOT. Uses SOT12_2 at the output pin.

[bit5:4] SIN12S: SIN12 Input Select bits

Selects input for SIN12.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses SIN12_0 at the input pin of MFS ch.12 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN12_1 at the input pin of MFS ch.12 SIN.
		Uses SIN12_2 at the input pin of MFS ch.12 SIN.

[bit3:0] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

4.24 Extended Pin Function Setting Register 18 (EPFR18)

The EPFR18 register assigns functions of HDMI-CEC pins.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved				CECR1B		CECR0B	
Attribute	-				R/W		R/W	
Initial value	-				00		00	

Register Function

[bit31:4] Reserved: Reserved bits

"0" is read from these bits.

When writing these bits, set them to "0".

[bit3:2] CECR1B: CEC1 input/output selection bits

It selects I/O for I/O pin CEC1 of HDMI-CEC/Remote Control Reception ch.1.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	HDMI-CEC/remote control reception ch.1 is not input/output. [Initial value]
	01	CEC1_0 is used, HDMI-CEC/remote control reception ch.1 is input/output.
	10	CEC1_1 is used, HDMI-CEC/remote control reception ch.1 is input/output.
	11	Setting is prohibited.

Notes:

- *In TYPE3-M0+ product, CECR1B does not work. Instead, use CEC_CTL register. See “Registers” part in Low power consumption section. When writing these bits, set them to "00".*

[bit1:0] CECR0B: CEC0 input/output selection bits

It selects I/O for I/O pin CEC0 of HDMI-CEC/Remote Control Reception ch.0.

bit1:0	Description
Reading	Reads out the register value.
Writing 00	HDMI-CEC/remote control reception ch.0 is not input/output. [Initial value]
Writing 01	CEC0_0 is used, HDMI-CEC/remote control reception ch.0 is input/output.
Writing 10	CEC0_1 is used, HDMI-CEC/remote control reception ch.0 is input/output.
Writing 11	Setting is prohibited.

Notes:

- *In TYPE3-M0+ product,CECR0B does not work. Instead, use CEC_CTL register. See “Registers” part in Low power consumption section. When writing these bits, set them to "00".*

4.25 Extended Pin Function Setting Register 21 (EPFR21)

EPFR21 register sets the function assignment to QPRC.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved					QZIN0S[2]	QBIN0S[2]	QAIN0S[2]
Attribute	-					R/W	R/W	R/W
Initial value	-					0	0	0

Register Function

[bit31:3] Reserved: Reserved bits

"0" is read out from these bits.

When writing these bits, set them to "0".

[bit2] QZIN0S[2]: QPRC-ch.0 ZIN Input Pin bits

Selects input for QPRC-ch.0 as ZIN. If ZIN0_3 is used, set this bit together with QZIN0S bit in Extended Pin Function Setting Register 9 (EPFR09).

Bit[2],EPFR09[5:4]		Description
Reading		Reads out the register value.
Writing	000	ZIN0_0 is used as ZIN, the input pin of QDU ch.0. [Initial value]
	001	ZIN0_0 is used as ZIN, the input pin of QDU ch.0.
	010	ZIN0_1 is used as ZIN, the input pin of QDU ch.0.
	011	ZIN0_2 is used as ZIN, the input pin of QDU ch.0.
	100	ZIN0_3 is used as ZIN, the input pin of QDU ch.0.
	Others	Setting is prohibited.

[bit1] QBIN0S[2]: QPRC-ch.0 BIN Input Pin bits

Selects input for QPRC-ch.0 as BIN. If BIN0_3 is used, set this bit together with QBIN0S bit in Extended Pin Function Setting Register 9 (EPFR09).

Bit[1],EPFR09[3:2]		Description
Reading		Reads out the register value.
Writing	000	BIN0_0 is used as BIN, the input pin of QDU ch.0. [Initial value]
	001	BIN0_0 is used as BIN, the input pin of QDU ch.0.
	010	BIN0_1 is used as BIN, the input pin of QDU ch.0.
	011	BIN0_2 is used as BIN, the input pin of QDU ch.0.
	100	BIN0_3 is used as BIN, the input pin of QDU ch.0.
	Others	Setting is prohibited.

[bit0] QAIN0S[2]: QPRC-ch.0 AIN Input Pin bits

Selects input for QPRC-ch.0 as AIN. If AIN0_3 is used, set this bit together with QAIN0S bit in Extended Pin Function Setting Register 9 (EPFR09)

Bit[0],EPFR09[1:0]		Description
Reading		Reads out the register value.
Writing	000	AIN0_0 is used as AIN, the input pin of QDU ch.0. [Initial value]
	001	AIN0_0 is used as AIN, the input pin of QDU ch.0.
	010	AIN0_1 is used as AIN, the input pin of QDU ch.0.
	011	AIN0_2 is used as AIN, the input pin of QDU ch.0.
	100	AIN0_3 is used as AIN, the input pin of QDU ch.0.
	Others	Setting is prohibited.

4.26 Extended Pin Function Setting Register 22 (EPFR22)

The EPFR22 register assigns functions of multi-function serial channel 0, channel 1, channel 2, channel 3.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	SCS31E		SCS30B		Reserved			
R/W	R/W		R/W		R/W			
00	00		00		00			

bit	7	6	5	4	3	2	1	0
Field	SCS11E		SCS10B		Reserved			
Attribute	R/W		R/W		-			
Initial value	00		00		-			

Register Function

[bit31:16] Reserved: Reserved bits

"0" is read from these bits.

When writing these bits, set them to "0".

[bit15:14] SCS31E: SCS31 Output Select bits.

Selects output for SCS31.

bit15:14		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for MFS ch.3 SCS31. [Initial value]
	01	Uses SCS31_0 at the output pin of MFS ch. 3 SCS31.
	10	Uses SCS31_1 at the output pin of MFS ch. 3 SCS31.
	11	Uses SCS31_2 at the output pin of MFS ch. 3 SCS31.

[bit13:12] SCS30B: SCS30 Input/Output Select bits.

Selects input/output for SCS30.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses SCS30_0 at the input pin of MFS ch.3 SCS30. Does not produce output. [Initial value]
	01	Uses SCS30_0 at the input pin of MFS ch.3 SCS30. Uses SCS30_0 at the output pin.
	10	Uses SCS30_1 at the input pin of MFS ch.3 SCS30. Uses SCS30_1 at the output pin.
	11	Uses SCS30_2 at the input pin of MFS ch.3 SCS30. Uses SCS30_2 at the output pin.

[bit11:8] Reserved: Reserved bits

"0" is read from these bits.

When writing these bits, set them to "0".

[bit7:6] SCS11E: SCS11 Output Select bits.

Selects output for SCS11.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for MFS ch.1 SCS11. [Initial value]
	01	Uses SCS11_0 at the output pin of MFS ch.1 SCS11.
	10	Uses SCS11_1 at the output pin of MFS ch.1 SCS11.
	11	Uses SCS11_2 at the output pin of MFS ch.1 SCS11.

[bit5:4] SCS10B: SCS10 Input/Output Select bits.

Selects input/output for SCS10.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses SCS10_0 at the input pin of MFS ch.1 SCS10. Does not produce output. [Initial value]
	01	Uses SCS10_0 at the input pin of MFS ch.1 SCS10. Uses SCS10_0 at the output pin.
	10	Uses SCS10_1 at the input pin of MFS ch.1 SCS10. Uses SCS10_1 at the output pin.
	11	Uses SCS10_2 at the input pin of MFS ch.1 SCS10. Uses SCS10_2 at the output pin.

[bit3:0] Reserved: Reserved bits

"0" is read from these bits.

When writing these bits, set them to "0".

4.27 Extended Pin Function Setting Register 23 (EPFR23)

The EPFR23 register assigns functions of multi-function serial ch.6 to ch.7.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	SCS73E		SCS72E		SCS71E		SCS70B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SCS63E		SCS62E		SCS61E		SCS60B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read from these bits.

When writing these bits, set them to "0x0000".

[bit15:14] SCS73E: SCS73 Output Select bits

Selects output for SCS73.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of MFS ch7 SCS73. [Initial value]
	01	Uses SCS73_0 at the output pin of MFS ch.7 SCS73.
	10	Uses SCS73_1 at the output pin of MFS ch.7 SCS73.
	11	Uses SCS73_2 at the output pin of MFS ch.7 SCS73.

[bit13:12] SCS72E: SCS72 Output Select bits

Selects output for SCS72.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of MFS ch7 SCS72. [Initial value]
	01	Uses SCS72_0 at the output pin of MFS ch.7 SCS72.
	10	Uses SCS72_1 at the output pin of MFS ch.7 SCS72.
	11	Uses SCS72_2 at the output pin of MFS ch.7 SCS72.

[bit11:10] SCS71E: SCS71 Output Select bits

Selects output for SCS71.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of MFS ch7 SCS71. [Initial value]
	01	Uses SCS71_0 at the output pin of MFS ch.7 SCS71.
	10	Uses SCS71_1 at the output pin of MFS ch.7 SCS71.
	11	Uses SCS71_2 at the output pin of MFS ch.7 SCS71.

[bit9:8] SCS70B: SCS70 Input/Output Select bits

Selects input/output for SCS70.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCS70_0 at the input pin of MFS ch.7 SCS70. Does not produce output. [Initial value]
	01	Uses SCS70_0 at the input pin of MFS ch.7 SCS70. Uses SCS70_0 at the output pin.
	10	Uses SCS70_1 at the input pin of MFS ch.7 SCS70. Uses SCS70_1 at the output pin.
	11	Uses SCS70_2 at the input pin of MFS ch.7 SCS70. Uses SCS70_2 at the output pin.

[bit7:6] SCS63E: SCS63 Output Select bits

Selects output for SCS63.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of MFS ch6 SCS63. [Initial value]
	01	Uses SCS63_0 at the output pin of MFS ch.6 SCS63.
	10	Uses SCS63_1 at the output pin of MFS ch.6 SCS63.
	11	Uses SCS63_2 at the output pin of MFS ch.6 SCS63.

[bit5:4] SCS62E: SCS62 Output Select bits

Selects output for SCS62.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of MFS ch6 SCS62. [Initial value]
	01	Uses SCS62_0 at the output pin of MFS ch.6 SCS62.
	10	Uses SCS62_1 at the output pin of MFS ch.6 SCS62.
	11	Uses SCS62_2 at the output pin of MFS ch.6 SCS62.

[bit3:2] SCS61E: SCS61 Output Select bits

Selects output for SCS61.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of MFS ch6 SCS61. [Initial value]
	01	Uses SCS61_0 at the output pin of MFS ch.6 SCS61.
	10	Uses SCS61_1 at the output pin of MFS ch.6 SCS61.
	11	Uses SCS61_2 at the output pin of MFS ch.6 SCS61.

[bit1:0] SCS60B: SCS60 Input/Output Select bits

Selects input/output for SCS60.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCS60_0 at the input pin of MFS ch.6 SCS60. Does not produce output. [Initial value]
	01	Uses SCS60_0 at the input pin of MFS ch.6 SCS60. Uses SCS60_0 at the output pin.
	10	Uses SCS60_1 at the input pin of MFS ch.6 SCS60. Uses SCS60_1 at the output pin.
	11	Uses SCS60_2 at the input pin of MFS ch.6 SCS60. Uses SCS60_2 at the output pin.

4.28 Extended Pin Function Setting Register 31 (EPFR31)

The EPFR31 register assigns functions of I2C Slave Wake-up.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved				SI2CSDA6B		SI2CSCL6B	
Attribute	R/W				R/W		R/W	
Initial value	0000				00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	R/W							
Initial value	00000000							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	R/W							
Initial value	00000000							

bit	7	6	5	4	3	2	1	0
Field	Reserved							
Attribute	R/W							
Initial value	00000000							

Register Function

[bit31:28] Reserved: Reserved bits

"0x0" is read from these bits.

When writing these bits, set them to "0x0".

[bit27:26] SI2CSDA6B: I2C Slave ch.6 SDA pin select bits

Selects ch.6 SDA pin.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses SI2CSDA6_0 at the input pin of I2CSLAVE ch.6 SDA. Does not produce output. [Initial value]
	01	Uses SI2CSDA6_0 at the input pin of I2CSLAVE ch.6 SDA. Uses SI2CSDA6_0 at the output pin.
	10	Uses SI2CSDA6_1 at the input pin of I2CSLAVE ch.6 SDA. Uses SI2CSDA6_1 at the output pin.
	11	Uses SI2CSDA6_2 at the input pin of I2CSLAVE ch.6 SDA. Uses SI2CSDA6_2 at the output pin.

[bit25:24] SI2CSCL6B: I2C Slave ch.6 SCL pin select bits

Selects ch.6 SCL pin.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses SI2CSCL6_0 at the input pin of I2CSLAVE ch.6 SCL. Does not produce output. [Initial value]
	01	Uses SI2CSCL6_0 at the input pin of I2CSLAVE ch.6 SCL. Uses SI2CSCL6_0 at the output pin.
	10	Uses SI2CSCL6_1 at the input pin of I2CSLAVE ch.6 SCL. Uses SI2CSCL6_1 at the output pin.
	11	Uses SI2CSCL6_2 at the input pin of I2CSLAVE ch.6 SCL. Uses SI2CSCL6_2 at the output pin.

[bit23:0] Reserved: Reserved bits

"0x000000" is read from these bits.

When writing these bits, set them to "0x000000".

4.29 Extended Pin Function Setting Register 33 (EPFR33)

The EPFR33 register assigns functions of Smart Card

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		Reserved		CLK1E		VCC1E	
Attribute	-		-		R/W		R/W	
Initial value	-		-		00		00	

bit	23	22	21	20	19	18	17	16
Field	VPEN1E		RST1E		DATA1B		CIN1S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved		Reserved		CLK0E		VCC0E	
Attribute	-		-		R/W		R/W	
Initial value	-		-		00		00	

bit	7	6	5	4	3	2	1	0
Field	VPEN0E		RST0E		DATA0B		CIN0S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:28] Reserved: Reserved bits

"0x0000" is read from these bits.

When writing these bits, set them to "0x0000".

[bit27:26] CLK1E: CLK1 Output Select bits

Selects output for ICC CLK1.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of ICC CLK1. [Initial value]
	01	Uses CLK1_0 at the output pin of ICC CLK1.
	10	Uses CLK1_1 at the output pin of ICC CLK1.
	11	Setting is prohibited.

[bit25:24] VCC1E: VCC1 Output Select bits

Selects output for VCC1.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of VCC1. [Initial value]
	01	Uses VCC1_0 at the output pin of VCC1.
	10	Uses VCC1_1 at the output pin of VCC1.
	11	Setting is prohibited.

[bit23:22] VPEN1E: VCC1 Output Select bits

Selects output for ICC VPEN1.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of ICC VPEN1. [Initial value]
	01	Uses VPEN1_0 at the output pin of ICC VPEN1.
	10	Uses VPEN1_1 at the output pin of ICC VPEN1.
	11	Setting is prohibited.

[bit21:20] RST1E: VCC1 Output Select bits

Selects output for ICC RST1.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of ICC RST1. [Initial value]
	01	Uses RST1_0 at the output pin of ICC RST1.
	10	Uses RST1_1 at the output pin of ICC RST1.
	11	Setting is prohibited.

[bit19:18] DATA1B: DATA1 Bus Select bits

Selects input/output for DATA1.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses DATA1_0 at the input pin of ICC DATA1, DATA1_0 output is disable. [Initial value]
	01	Same as Writing 00.
	10	Uses DATA1_1 at the input/output pin of ICC DATA1.
	11	Setting is prohibited.

[bit17:16] CIN1S: CIN1 Input Select bits

Selects input for CIN1 card insert detection.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses CIN1_0 at the input pin of ICC CIN1. [Initial value]
	01	Same as Writing 00.
	10	Uses CIN1_1 at the input pin of ICC CIN1.
	11	Does not produce input of CIN1.

[bit11:10] CLK0E: CLK0 Output Select bits

Selects output for ICC CLK0.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of ICC CLK0. [Initial value]
	01	Uses CLK0_0 at the output pin of ICC CLK0.
	10	Uses CLK0_1 at the output pin of ICC CLK0.
	11	Setting is prohibited.

[bit9:8] VCC0E: VCC0 Output Select bits

Selects output for VCC0.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of VCC0. [Initial value]
	01	Uses VCC0_0 at the output pin of VCC0.
	10	Uses VCC0_1 at the output pin of VCC0.
	11	Setting is prohibited.

[bit7:6] VPEN0E: VPEN0 Output Select bits

Selects output for ICC VPEN0.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of ICC VPEN0. [Initial value]
	01	Uses VPEN0_0 at the output pin of ICC VPEN0.
	10	Uses VPEN0_1 at the output pin of ICC VPEN0.
	11	Setting is prohibited.

[bit5:4] RST0E: RST0 Output Select bits

Selects output for ICC RST0.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of ICC RST0. [Initial value]
	01	Uses RST0_0 at the output pin of ICC RST0.
	10	Uses RST0_1 at the output pin of ICC RST0.
	11	Setting is prohibited.

[bit3:2] DATA0B: DATA0 Bus Select bits

Selects input/output for DATA0.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses DATA0_0 at the input/output pin of ICC DATA0. [Initial value]
	01	Same as Writing 00.
	10	Uses DATA0_1 at the input/output pin of ICC DATA0.
	11	Setting is prohibited.

[bit1:0] CIN0S: CIN0 Input Select bits

Selects input for CIN0 card insert detection.

Bit		Description
Reading		Reads out the register value.
Writing	00	Uses CIN0_0 at the input pin of ICC CIN0. [Initial value]
	01	Same as Writing 00.
	10	Uses CIN0_1 at the input pin of ICC CIN0.
	11	Does not produce input of CIN1.

4.30 Extended Pin Function Setting Register 34 (EPFR34)

The EPFR34 register assigns functions of MFS flow control.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	CTS1S		RTS1E		Reserved		Reserved	
Attribute	R/W		R/W		-		-	
Initial value	00		00		-		-	

Register Function

[bit31:8] Reserved: Reserved bits

"0x0" is read from these bits.

When writing these bits, set them to "0x0".

[bit7:6] CTS1S CTS1 Input Select bits

Selects input for CTS1.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the input pin of MFS ch.1 CTS. [Initial value]
	01	Same as Writing 00.
	10	Uses CTS1_1 at the input pin of MFS ch.1 CTS.
	11	Uses CTS1_2 at the input pin of MFS ch.1 CTS.

[bit5:4] RTS1E: RTS1 Output Select bits

Selects output for RTS1.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for MFS ch.1 RTS. [Initial value]
	01	Same as Writing 00.
	10	Uses RTS1_1 at the output pin of MFS ch.1 RTS.
	11	Uses RTS1_2 at the output pin of MFS ch.1 RTS.

[bit3:0] Reserved: Reserved bits

"0x0" is read from these bits.

When writing these bits, set them to "0x0".

4.31 Extended Pin Function Setting Register 37 (EPFR37)

The EPFR37 register assigns functions of MFSI2S with MFS ch.4, ch.5.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		Reserved		SDO5E		SDI5S	
Attribute	-		-		R/W		R/W	
Initial value	-		-		00		00	

bit	23	22	21	20	19	18	17	16
Field	WS5B		SCK5B		MCK5E		MCK5S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved		Reserved		SDO4E		SDI4S	
Attribute	-		-		R/W		R/W	
Initial value	-		-		00		00	

bit	7	6	5	4	3	2	1	0
Field	WS4B		SCK4B		MCK4E		MCK4S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:28] Reserved: Reserved bits

"0x0" is read from these bits.

When writing these bits, set them to "0x0".

[bit27:26] SDO5E: MFSI2S SDO5 Output Select bits

Selects output for I2SDO5.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.5 I2SDO. [Initial value]
	01	Uses I2SDO5_0 at the output pin of I2S ch.5 I2SDO.
	10	Uses I2SDO5_1 at the output pin of I2S ch.5 I2SDO.
	11	Uses I2SDO5_2 at the output pin of I2S ch.5 I2SDO.

[bit25:24] SDI5S: MFSI2S I2SDI5 Input Select bits

Selects input for I2SDI5.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce input of I2S ch.5 I2SDI. [Initial value]
	01	Uses I2SDI5_0 at the input pin of I2S ch.5 I2SDI.
	10	Uses I2SDI5_1 at the input pin of I2S ch.5 I2SDI.
	11	Uses I2SDI5_2 at the input pin of I2S ch.5 I2SDI.

[bit23:22] MFSI2S WS5B: I2SWS5 Output Select bits

Selects output for I2SWS5.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.5 I2SWS. [Initial value]
	01	Uses I2SWS5_0 at the output pin of I2S ch.5 I2SWS.
	10	Uses I2SWS5_1 at the output pin of I2S ch.5 I2SWS.
	11	Uses I2SWS5_2 at the /output pin of I2S ch.5 I2SWS.

[bit21:20] MFSI2S SCK5B: I2SCK5 Output Select bits

Selects output for I2SCK5.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.5 I2SCK. [Initial value]
	01	Uses I2SCK5_0 at the output pin of I2S ch.5 I2SCK.
	10	Uses I2SCK5_1 at the output pin of I2S ch.5 I2SCK.
	11	Uses I2SCK5_2 at the output pin of I2S ch.5 I2SCK.

[bit19:18] MFSI2S MCLK5E: I2SMCLK5 Output Select bits

Selects output for I2SMCLK5.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.5 I2SMCLK. [Initial value]
	01	Uses I2SMCLK5_0 at the output pin of I2S ch.5 I2SMCLK.
	10	Uses I2SMCLK5_1 at the output pin of I2S ch.5 I2SMCLK.
	11	Uses I2SMCLK5_2 at the output pin of I2S ch.5 I2SMCLK.

[bit17:16] MFSI2S MCLK5S: I2SMCLK5 Input Select bits

Selects input for I2SMCLK5.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce input of I2S ch.5 I2SMCLK. [Initial value]
	01	Uses I2SMCLK5_0 at the input pin of I2S ch.5 I2SMCLK.
	10	Uses I2SMCLK5_1 at the input pin of I2S ch.5 I2SMCLK.
	11	Uses I2SMCLK5_2 at the input pin of I2S ch.5 I2SMCLK.

[bit15:12] Reserved: Reserved bits

"0x0" is read from these bits.

When writing these bits, set them to "0x0".

[bit11:10] SDO4E: MFSI2S SDO4 Output Select bits

Selects output for I2SDO4.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.4 I2SDO. [Initial value]
	01	Uses I2SDO4_0 at the output pin of I2S ch.4 I2SDO.
	10	Uses I2SDO4_1 at the output pin of I2S ch.4 I2SDO.
	11	Uses I2SDO4_2 at the output pin of I2S ch.4 I2SDO.

[bit9:8] SDI4S: MFSI2S I2SDI4 Input Select bits

Selects input for I2SDI4.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce input of I2S ch.4 I2SDI. [Initial value]
	01	Uses I2SDI4_0 at the input pin of I2S ch.4 I2SDI.
	10	Uses I2SDI4_1 at the input pin of I2S ch.4 I2SDI.
	11	Uses I2SDI4_2 at the input pin of I2S ch.4 I2SDI.

[bit7:6] MFSI2S WS4B: I2SWS4 Output Select bits

Selects output for I2SWS4.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.4 I2SWS. [Initial value]
	01	Uses I2SWS4_0 at the output pin of I2S ch.4 I2SWS.
	10	Uses I2SWS4_1 at the output pin of I2S ch.4 I2SWS.
	11	Uses I2SWS4_2 at the output pin of I2S ch.4 I2SWS.

[bit5:4] MFSI2S SCK4B: I2SCK4 Output Select bits

Selects output for I2SCK4.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.4 I2SCK. [Initial value]
	01	Uses I2SCK4_0 at the output pin of I2S ch.4 I2SCK.
	10	Uses I2SCK4_1 at the output pin of I2S ch.4 I2SCK.
	11	Uses I2SCK4_2 at the output pin of I2S ch.4 I2SCK.

[bit3:2] MFSI2S MCLK4E: I2SMCLK4 Output Select bits

Selects output for I2SMCLK4.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.4 I2SMCLK. [Initial value]
	01	Uses I2SMCLK4_0 at the output pin of I2S ch.4 I2SMCLK.
	10	Uses I2SMCLK4_1 at the output pin of I2S ch.4 I2SMCLK.
	11	Uses I2SMCLK4_2 at the output pin of I2S ch.4 I2SMCLK.

[bit1:0] MFSI2S MCLK4S: I2SMCLK4 Input Select bits

Selects input for I2SMCLK4.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce input of I2S ch.4 I2SMCLK. [Initial value]
	01	Uses I2SMCLK4_0 at the input pin of I2S ch.4 I2SMCLK.
	10	Uses I2SMCLK4_1 at the input pin of I2S ch.4 I2SMCLK.
	11	Uses I2SMCLK4_2 at the input pin of I2S ch.4 I2SMCLK.

4.32 Extended Pin Function Setting Register 38 (EPFR38)

The EPFR38 register assigns functions of MFSI2S with MFS ch6.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved		Reserved		SDO6E		SDI6S	
Attribute	-		-		R/W		R/W	
Initial value	-		-		00		00	

bit	7	6	5	4	3	2	1	0
Field	WS6B		SCK6B		MCK6E		MCK6S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Register Function

[bit31:12] Reserved: Reserved bits

"0" is read from these bits.

When writing these bits, set them to "0".

[bit11:10] SDO6E: MFSI2S SDO6 Output Select bits

Selects output for I2SDO6.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.6 I2SDO. [Initial value]
	01	Uses I2SDO6_0 at the output pin of I2S ch.6 I2SDO.
	10	Uses I2SDO6_1 at the output pin of I2S ch.6 I2SDO.
	11	Uses I2SDO6_2 at the output pin of I2S ch.6 I2SDO.

[bit9:8] SDI6S: MFSI2S I2SDI6 Input Select bits

Selects input for I2SDI6.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce input of I2S ch.6 I2SDI. [Initial value]
	01	Uses I2SDI6_0 at the input pin of I2S ch.6 I2SDI.
	10	Uses I2SDI6_1 at the input pin of I2S ch.6 I2SDI.
	11	Uses I2SDI6_2 at the input pin of I2S ch.6 I2SDI.

[bit7:6] MFSI2S WS6B: I2SWS6 Output Select bits

Selects output for I2SWS6.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.6 I2SWS. [Initial value]
	01	Uses I2SWS6_0 at the output pin of I2S ch.6 I2SWS.
	10	Uses I2SWS6_1 at the output pin of I2S ch.6 I2SWS.
	11	Uses I2SWS6_2 at the output pin of I2S ch.6 I2SWS.

[bit5:4] MFSI2S SCK6B: I2SCK6 Output Select bits

Selects output for I2SCK6.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.6 I2SCK. [Initial value]
	01	Uses I2SCK6_0 at the output pin of I2S ch.6 I2SCK.
	10	Uses I2SCK6_1 at the output pin of I2S ch.6 I2SCK.
	11	Uses I2SCK6_2 at the output pin of I2S ch.6 I2SCK.

[bit3:2] MFSI2S MCLK6E: I2SMCLK6 Output Select bits

Selects output for I2SMCLK6.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of I2S ch.6 I2SMCLK. [Initial value]
	01	Uses I2SMCLK6_0 at the output pin of I2S ch.6 I2SMCLK.
	10	Uses I2SMCLK6_1 at the output pin of I2S ch.6 I2SMCLK.
	11	Uses I2SMCLK6_2 at the output pin of I2S ch.6 I2SMCLK.

[bit1:0] MFSI2S MCLK6S: I2SMCLK6 Input Select bits

Selects input for I2SMCLK6.

Bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce input of I2S ch.6 I2SMCLK. [Initial value]
	01	Uses I2SMCLK6_0 at the input pin of I2S ch.6 I2SMCLK.
	10	Uses I2SMCLK6_1 at the input pin of I2S ch.6 I2SMCLK.
	11	Uses I2SMCLK6_2 at the input pin of I2S ch.6 I2SMCLK.

4.33 Special Port Setting Register (SPSR)

The SPSR register sets a pin as a signal pin of special functions.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved			USB0C	MAINXC		SUBXC	
Attribute	-			R/W	R/W		R/W	
Initial value	-			0	01		01	

Register Function

[bit31:5] Reserved: Reserved bits

"0" is read from these bits.

When writing these bits, set them to "0".

[bit4] USB0C: USB (ch.0) Pin Setting Register

This bit sets a pin as a USB pin.

bit4		Description
Reading		Reads out the register value.
Writing	0	Does not use two pins of UDM0 and UDP0 as USB pins but as digital input/output pins. [Initial value]
	1	Uses two pins of UDM0 and UDP0 as USB pins. (An I/O cell will be in a state of input direction and input cut-off.)

[bit3:2] MAINXC: Main Clock (Oscillation) Pin Setting Register

This bit sets a pin as a main clock (oscillation) pin.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Does not use two pins of X0 and X1 as main clock (oscillation) pins but as digital input/output pins.
	01	Uses two pins of X0 and X1 as main clock (oscillation) pins. [Initial value] (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)
	10	Setting is prohibited.
	11	Uses X0 pin as an external clock input pin. Uses X1 pin as a digital input/output.

Notes:

- *Only writing "01" to the MAINXC bit does not make a main clock start oscillation.*
  To start oscillation, enable oscillation by the MOSCE bit of the System Clock Mode Control Register (SCM_CTL), which is described in the chapter "Clock", after writing "01" to the MAINXC bit.
- *To use external clock, refer to "Using an external clock" of "HANDLING PRECAUTIONS " in Datasheet of each product.*
- *When using deep standby mode, write “00” to the MAINXC bit and SUBXC bit before setting these pins to suitable pin functions (clock oscillation pin or digital I/O pin). This is required only once after power-on. Please note that the pin state may become unstable instantaneously after writing “00” to these bits.*

[bit1:0] SUBXC: Sub Clock (Oscillation) Pin Setting Register

This bit sets a pin as a sub clock (oscillation) pin in TYPE1-M0+ products.

In TYPE2-M0+ product, see chapter “VBAT domain”

In TYPE3-M0+ product, see chapter “Low Power Consumption Mode”

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Does not use two pins of X0A and X1A as sub clock (oscillation) pins but as digital input/output pins.
	01	Uses two pins of X0A and X1A as sub clock (oscillation) pins. [Initial value] (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)
	10	Setting is prohibited.
	11	Uses X0A pin as an external clock input pin. Uses X1A pin as a digital input/output.

Notes:

- *Only writing "01" to the SUBXC bit does not make a sub clock start oscillation.*
  To start oscillation, enable oscillation by the SOSCE bit of the System Clock Mode Control Register (SCM_CTL), which is described in the chapter "Clock", after writing "01" to the SUBXC bit.
- *To use external clock, refer to "Using an external clock" of "HANDLING PRECAUTIONS " in Datasheet of each product.*

- *When using deep standby mode, write “00” to the MAINXC bit and SUBXC bit before setting these pins to suitable pin functions (clock oscillation pin or digital I/O pin). This is required only once after power-on. Please note that the pin state may become unstable instantaneously after writing “00” to these bits.*

4.34 Port Pseudo Open Drain Setting Register (PZRx)

PZRx register makes I/O port Hi-Z when output is High level and sets pseudo open drain control.

List of PZR Register Configuration

bit 31 to 16	bit 15 to 0	Initial value	Attribute
Reserved	PZR0	0x0000	R/W
Reserved	PZR1	0x0000	R/W
Reserved	PZR2	0x0000	R/W
Reserved	PZR3	0x0000	R/W
Reserved	PZR4	0x0000	R/W
Reserved	PZR5	0x0000	R/W
Reserved	PZR6	0x0000	R/W
Reserved	PZR7	0x0000	R/W
Reserved	PZR8	0x0000	R/W
Reserved	PZR9	0x0000	R/W
Reserved	PZRA	0x0000	R/W
Reserved	PZRB	0x0000	R/W
Reserved	PZRC	0x0000	R/W
Reserved	PZRD	0x0000	R/W
Reserved	PZRE	0x0000	R/W
Reserved	PZRF	0x0000	R/W

Details of Register Configuration

bit	31 to 16	15 to 0
Field	Reserved	PZRx

Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] PZRx: Port Pseudo Open Drain Setting Register x

Sets the pseudo open drain of the pin.

bit15:0		Description
Reading		Reads out the register value.
Writing	0	Set the pin to High level when outputting digital High level by GPIO or peripheral macro.
	1	Set the pin to Hi-Z when outputting digital High level by GPIO or peripheral macro. Disconnect the pull-up resistor regardless of the PCR setting.

Notes:

- *The "x" description of PZRx is wildcard. It shows PZR0, PZR1, PZR2, and so on.*
- *The function of the PZR register is implemented only in some specific pins.*
  Only pins described as "PZR register control is enabled" in remarks column of "I/O circuit type" of Data Sheet can control open drain.

- *PZR register does not exist in all pins. However, even the pins that do not have PZR registers can control pseudo open drain by the setting of DDR register if they are used as GPIO.*
  In such a case, after setting PFR = 0 (GPIO setting) and PDOR = 0,
  When setting L output: used as DDR = 1 (output direction).
  When setting Hi-Z output: used as DDR = 0 (input direction).

- *However, in open drain by the GPIO setting, you cannot apply voltage that exceeds VCC at Hi-Z.*

- *Initial value is just an example for TYPE1-M0+, please check Appendix for different product TYPE.*

4.35 LVDI Input Setting Register (LVDIE)

The LVDIE register sets to enable input LVDI if the product has LVDI pin.

Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved							LVDIE
Attribute	-							R/W
Initial value	-							0

Register Function

[bit31:1] Reserved: Reserved bits

"0x00000000" is read from these bits.
When writing these bits, set them to "0".

[bit0] LVDIE: LVD Input Enable Setting Register

LVDIE register sets to enable input LVDI.

bit15:0		Description
Reading		Reads out the register value.
Writing	0	Set the pin to High level when outputting digital High level by GPIO or peripheral macro.
	1	Set the pin to Hi-Z when outputting digital High level by GPIO or peripheral macro. Disconnect the pull-up resistor regardless of the PCR setting.

Notes:

- *This register does not exist in TYPE1-M0+ and TYPE3-M0+ products.*
- *This register is not initialized by deep standby transition reset.*
- *For more detail about LVDIE, refer chapter 5-3.*

5. Usage Precautions

This section describes precautions for using the I/O port.

■ ON/OFF of the Pull-up Resistance When SPL=1

SPL is a signal for turning a pin into Hi-Z state during standby mode.

- When SPL=0 Normal operations
- When SPL=1 Pin Hi-Z, input cut-off, pull-up disconnection
  However, the SPL bit cannot be used for setting external interrupts, NMIX, SWD pins.

For details of the SPL bit, see Chapter "Low Power Consumption Mode".

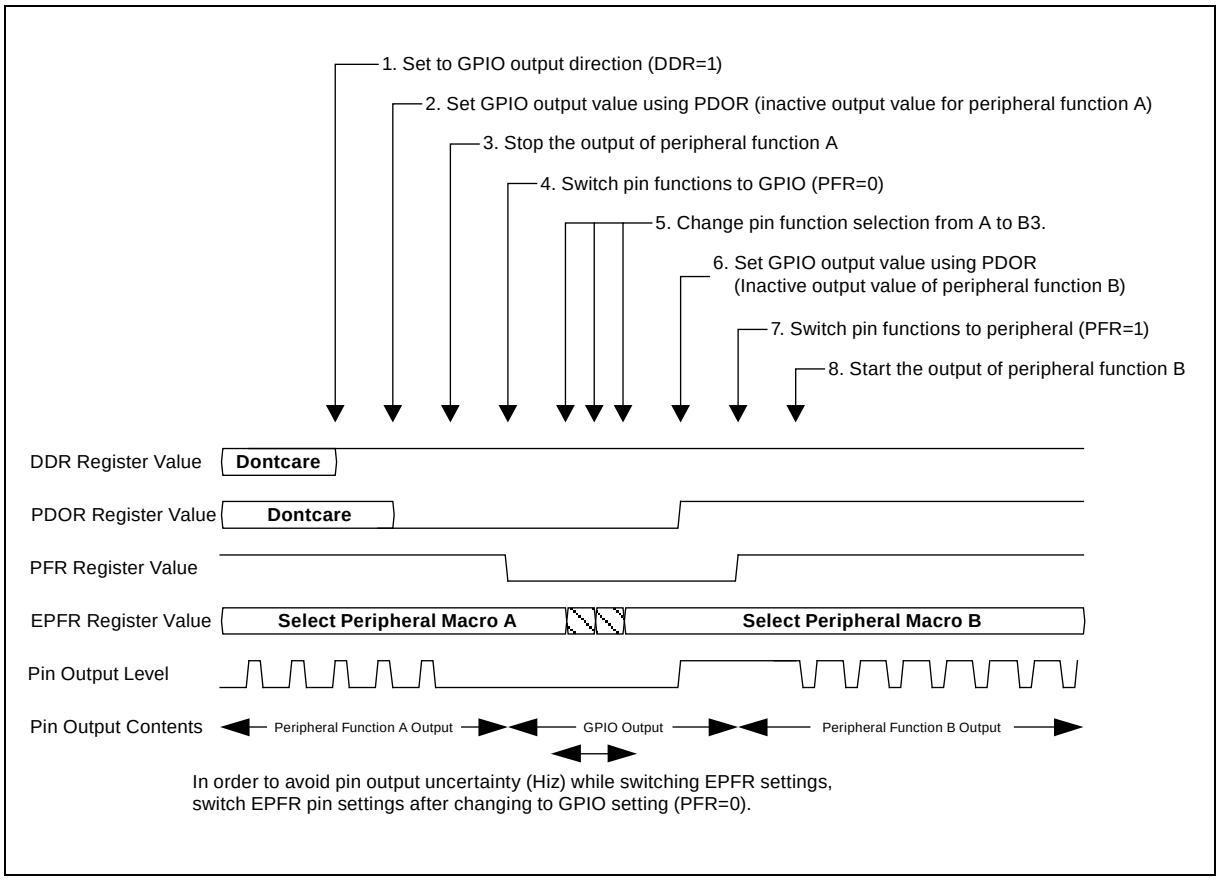
■ DTTIX Input

DTTI input is an input signal for switching the dual-purpose motor control PWM output (RTO) setting output pin to its other GPIO pin setting to address a motor stop demand in an emergency.
To use this function, enable switching by EPFR.

■ Procedures for Switching Pin Functions

When switching the outputs for peripheral functions using the EPFR register, to prevent pin uncertain output (Hi-Z), switch settings according to the procedures similar to the switching example shown in the following Figure 5-1.

Figure 5-1 Procedures for Switching Pin Functions



■ Reserved bit

This bit is read out as "0" except for that of ADE register. When writing, always write "0". The ADE reserved bit is read out as "1". When writing, always write "1".

■ Multi-function Serial Pin Group

When there are some multi-function serial inputs/outputs, set each input/output to the port of the same group. "The port of the same group" means that relocate function numbers attached to the pin name are the same, just like "xxx_0" or "yyy_1".

Table 5-1 shows an example setting.

Table 5-1 Multi-Function Serial Interface Example Setting

Serial Data Output	Serial Clock Input/Output	Serial Data Input	Effective Port
Pin SOT1_0 (Port 0)	Pin SCK1_0 (Port 0)	Pin SIN1_0 (Port 0)	Port 0
		Pin SIN1_1 (Port 1)	Setting is prohibited.
	Pin SCK1_1 (Port 1)	Pin SIN1_0 (Port 0)	
		Pin SIN1_1 (Port 1)	
Pin SOT1_1 (Port 1)	Pin SCK1_0 (Port 0)	Pin SIN1_0 (Port 0)	
		Pin SIN1_1 (Port 1)	
	Pin SCK1_1 (Port 1)	Pin SIN1 (Port 0)	
		Pin SIN1_1 (Port 1)	Port 1

■ Peripheral Function Output

As output pins for peripheral functions are uniquely determined by EPFR settings, Output for peripheral functions cannot be assigned to separate pins.

(Disabled example) Assign multifunction serial output SOT1_0 and SOT1_1 to the same output.

■ Pin Settings and Operation Mode

For SWDsettings, see Chapter "Debug Interface".

For state of each pin during standby mode or reset, see "Data Sheet" of the product used.

■ Product Specifications and Peripheral Function Pin Assignment

Functions which are assigned to pins (GPO, peripheral I/O and special I/O) vary in different products. Please see the pin function table of "Data Sheet" to confirm the pin function of each product. Do not select a function for a pin which is not available in your product by using the EPFR register setting.

■ When PE0 pin is used as GPIO

To use PE0 pin, the following settings are required.

Input: By reading PDIR, the value is read.

Output: Only L output is available because I/O of PE0 pin is Nch open drain pin.
PFR=0 (Used as GPIO.)
DDR=1(Used as output)
PDOR=0 (Output data is "0".)
SPL=0 (GPIO status is retained in STOP mode.)

■ External Interrupt Pin Settings in Standby Mode

When the mode is transferred to the Standby mode under the setting of SPL=1, set PFR=1 and select peripheral functions to enable the external interrupt assignment pin for returning.

If the setting of a pin used for external interrupt is remained PFR=0, unintended operation occurs.

CHAPTER 10-2: Fast GPIO

This chapter explains the Fast GPIO port.

1. Overview
2. Configuration
3. Setup Procedure Example
4. Registers
5. Bit Manipulation Base Address

CODE: 9AF_FastGPIO-E01.0

1. Overview

This section provides an overview of the Fast GPIO.

1 Cycle Access

- GPIO can access by 1 cycle is called Fast GPIO.
- Fast GPIO can read an input level and set an output level from the CPU within one cycle clock of HCLK.
- GPIO is selected in initial state. It is needed to select Fast GPIO by register setting to use Fast GPIO.

Pin Assignment

All I/O ports have both normal GPIO and Fast GPIO.

Bit Manipulation

The Fast GPIO support bit manipulation by dedicated circuit.
Even when bit operation, it can be accessed by 1cycle.

Simultaneous Access Function

Maximum 8 ports can be accessed together at the same time by the mirror register which bundles some ports at different group. It is enabling the effective access in each product.

2. Configuration

This section explains the configuration, block diagram, and operation of the Fast GPIO.

Configuration of Fast GPIO

Figure 2-1 shows the Fast GPIO block diagram.

I/O port can be switched to normal GPIO and Fast GPIO by FPOER.

Setting PCRx, DDRx, ADE is shared with normal GPIO.

Figure 2-1 Block Diagram of Fast GPIO Port

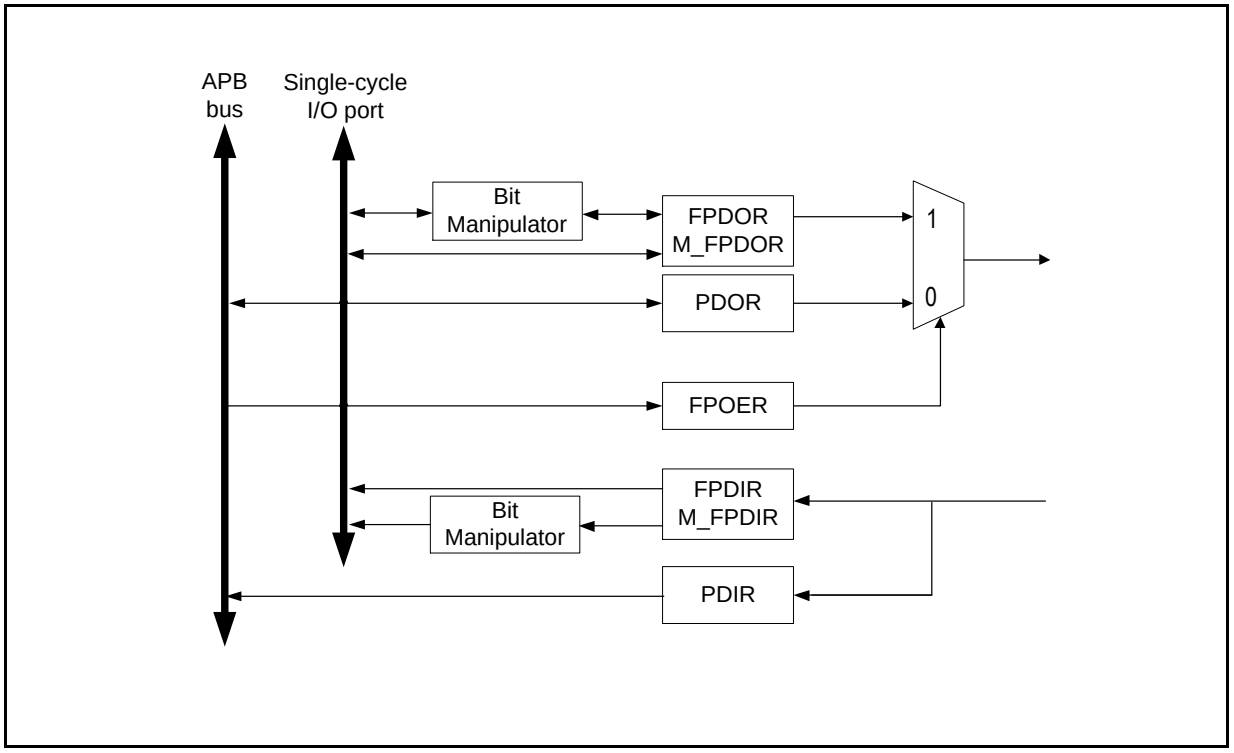


Table 2-1 Functions of Registers

Register Name	Function
FPDIR	A register to read the level status of I/O port. This is connected to Single-cycle I/O port, can be accessed by 1 cycle. − If the Fast GPIO port is used as a digital input pin, it reads input level. − If the Fast GPIO port is used as a digital output pin, it reads output level. − If the Fast GPIO port is used as a special pin, it always reads "0".
FPDOR	A register to set output level if the I/O port is used as Fast GPIO output pin. This is connected to Single-cycle I/O port, can be accessed by 1 cycle. − When "0" is set, it outputs Low level. − When "1" is set, it outputs High level. Notes: − If a pin is selected as input state or GPIO function or input/output of peripheral functions, the setting value is invalid. − Setting these registers will affect the value of related bit of M_FPDOR.
M_FPDIR	A Mirror Register to read the level status of FPDIR. Some Fast GPIO input value can be read simultaneously. − The register function is the same as FPDIR.
M_FPDOR	A Mirror Register to set output level of FPDOR. Some Fast GPIO output value can be set simultaneously. − The register function is the same as FPDOR. − Setting these register will affect the value of related bit of FPDOR.
FPOER	A register to select the output between normal GPIO and Fast GPIO. − When “0” select the output of normal GPIO. − When “1” select the output of Fast GPIO. − The setting of this register doesn’t affect the FPDIR/M_FPDIR.

Fast GPIO Access

- After configured the GPIO as Fast GPIO, each pin can be read by FPDIR and set output level by FPDOR. The usage is same as PDIR/PDOR.
- It is one-to-one correspondence between the bit of FPDIR/PDIR and FPDOR/PDOR.
- For example,

  If P00 is set as input state, access bit0 of FPDIR0 for reading the input value of P00.
  If P15 is set as Fast GPIO and output state, access bit5 of FPDOR1 for setting the output value of P15.

Fast GPIO Mirror Access (TYPE1- M0+, TYPE2-M0+)

- Some pins are re-assigned in a special group for flexibility.
- A group of some port assigned to mirror register(M_FPDIRx/M_FPDORx) can be accessed simultaneously .
- Table 2-2 shows the configuration of M_FPDIR0/M_FPDOR0. P10-15 and P22-23 can be accessed simultaneously. No need to read FPDIR1 and FPDIR2 separately. No need to write FPDOR1 and FPDOR2 separately.
- Table 2-3 shows the configuration of M_FPDIR1/M_FPDOR1. P3A-P3F and P46-47 can be accessed simultaneously. No need to read FPDIR3 and PFPDIR4 separately. No need to write FPDOR3 and FPDOR4 separately.

Table 2-2 M_FPDIR0/M_FPDOR0 Configuration (TYPE1-M0+ and TYPE2-M0+ Product)

Mirror Port Name	Port	Corresponding FPDIR Bit	Corresponding FPDOR Bit
M_FP00	P10	FPDIR1[0]	FPDOR1[0]
M_FP01	P11	FPDIR1[1]	FPDOR1[1]
M_FP02	P12	FPDIR1[2]	FPDOR1[2]
M_FP03	P13	FPDIR1[3]	FPDOR1[3]
M_FP04	P14	FPDIR1[4]	FPDOR1[4]
M_FP05	P15	FPDIR1[5]	FPDOR1[5]
M_FP06	P23	FPDIR2[3]	FPDOR2[3]
M_FP07	P22	FPDIR2[2]	FPDOR2[2]

Table 2-3 M_FPDIR1/M_FPDOR1 Configuration (TYPE1-M0+ and TYPE2-M0+ Product)

Mirror Port Name	Port	Corresponding FPDIR Bit	Corresponding FPDOR Bit
M_FP10	P3A	FPDIR3[10]	FPDOR3[10]
M_FP11	P3B	FPDIR3[11]	FPDOR3[11]
M_FP12	P3C	FPDIR3[12]	FPDOR3[12]
M_FP13	P3D	FPDIR3[13]	FPDOR3[13]
M_FP14	P3E	FPDIR3[14]	FPDOR3[14]
M_FP15	P3F	FPDIR3[15]	FPDOR3[15]
M_FP16	P46	FPDIR4[6]	FPDOR4[6]
M_FP17	P47	FPDIR4[7]	FPDOR4[7]

Notes:

- *Write the bit of M_FPDORx will affect the value in related bit of FPDORx. For example, after set the bit5 of M_FPDOR1 as "1", the value of bit15 of FPDOR3 is also changed to "1".*

Fast GPIO Mirror Access (TYPE3-M0+)

- Some pins are re-assigned in a special group for flexibility.
- A group of some port assigned to mirror register(M_FPDIR/M_FPDOR) can be accessed simultaneously .
- Table 2-4 shows the configuration of M_FPDIR0/M_FPDOR0. P10-15 and P22-23 can be accessed simultaneously. No need to read FPDIR1 and FPDIR2 separately.
- Table 2-5 shows the configuration of M_FPDIR1/M_FPDOR1. P05, P0F, P21, P46-47 and P50-52 can be accessed simultaneously. No need to read FPDIR0, FPDIR2, FPDIR4 and PFDIR5 separately.

Table 2-4 M_FPDIR0/M_FPDOR0 Configuration (TYPE3-M0+ Product)

Mirror Port Name	Port	Corresponding FPDIR Bit	Corresponding FPDOR Bit
M_FP00	P10	FPDIR1[0]	FPDOR1[0]
M_FP01	P11	FPDIR1[1]	FPDOR1[1]
M_FP02	P12	FPDIR1[2]	FPDOR1[2]
M_FP03	P13	FPDIR1[3]	FPDOR1[3]
M_FP04	P14	FPDIR1[4]	FPDOR1[4]
M_FP05	P15	FPDIR1[5]	FPDOR1[5]
M_FP06	P23	FPDIR2[3]	FPDOR2[3]
M_FP07	P22	FPDIR2[2]	FPDOR2[2]

Table 2-5 M_FPDIR1/M_FPDOR1 Configuration (TYPE3-M0+ Product)

Mirror Port Name	Port	Corresponding FPDIR Bit	Corresponding FPDOR Bit
M_FP10	P21	FPDIR2[1]	FPDOR2[1]
M_FP11	P05	FPDIR0[5]	FPDOR0[5]
M_FP12	P0F	FPDIR0[15]	FPDOR0[15]
M_FP13	P50	FPDIR5[0]	FPDOR5[0]
M_FP14	P51	FPDIR5[1]	FPDOR5[1]
M_FP15	P52	FPDIR5[2]	FPDOR5[2]
M_FP16	P46	FPDIR4[6]	FPDOR4[6]
M_FP17	P47	FPDIR4[7]	FPDOR4[7]

Notes:

- *Write the bit of M_FPDORx will affect the value in related bit of FPDORx. For example, after set the bit5 of M_FPDOR1 as "1", the value of bit2 of FPDOR5 is also changed to "1".*

Bit Manipulation

■ Overview

The Fast GPIO is mounted on Single-cycle I/O port. Single-cycle I/O port can access data with one-cycle . However, Single-cycle I/O port cannot support the bit-band operation.

The Fast GPIO supports the bit manipulation by special operation. Each bit of Fast GPIO can be set/clear independently. Similar with the bit-band, there is additional alias area for bit manipulation.

The bit manipulation just support to access the register FPDIR, FPDOR, M_FPDIR and M_FPDOR.

■ How to Use

This explains the address for bit manipulation.

- Alias address

  Alias address is accessed for bit manipulation. The formula is as following.

  Alias address = Base address + Offset address

- Base address

  Base address is decided by each FPDIR, FPDOR, M_FPDIR, M_FPDOR. Please refer to 5.Bit manipulation base address.

- Offset address

  Offset address is accessed to set access bit.

Table 2-6 Offset Address of Bit Manipulation

Access Bit	Offset Address
bit8	0x0004+1
bit9	0x0008+1
bit10	0x0010+1
bit11	0x0020+1
bit12	0x0040+1
bit13	0x0080+1
bit14	0x0100+1
bit15	0x0200+1
Setting is prohibited	Others

Access Bit	Offset Address
bit0	0x0004
bit1	0x0008
bit2	0x0010
bit3	0x0020
bit4	0x0040
bit5	0x0080
bit6	0x0100
bit7	0x0200
Setting is prohibited	Others

FPDIR read value is reflected to only "Access bit", other bits are "0". For example, when FPDIR8 is "0x07", "FPDIR8+Base address+0x0010" read value is "0x04".

Write to FPDOR is applicable to only "Access bit", other bits are ignored. For example, when FPDOR1 is "0x00" and write "0xFF" to "FPDOR1+Base address+0x0008", FPDOR1 is changed to "0x02".

■ Example of Alias address calculation

- When setting bit3 of FPDOR0

Access Register	Access Bit	Access Size	Base Address	Offset Address	Alias Address
FPDOR0	bit3	Byte	0xF801_4000	0x0020	0xF801_4020

- When setting bit13 of FPDOR0

Access Register	Access Bit	Access Size	Base Address	Offset Address	Alias Address
FPDOR0	bit13	Byte	0xF801_4000	0x0080+1	0xF801_4081

In the case of half word access, both upper byte and lower byte are masked.

- When setting bit13 and bit5 of FPDOR0

Access Register	Access Bit	Access Size	Base Address	Offset Address	Alias Address
FPDOR0	bit13, bit5	Half word	0xF801_4000	0x0080	0xF801_4080

In the case of word access, it is same with half word access.

3. Setup Procedure Example

This section explains an example of procedure for setting up the Fast GPIO.

Setting Up Fast GPIO

1. When Fast GPIO is used as output, choose the Fast GPIO by setting the FPOER register.
2. Setting the I/O port register as normal GPIO.

Figure 3-1 shows an example of procedure for setting up output.
Figure 3-2 shows an example of procedure for setting up input.

Figure 3-1 Example of Procedure for Setting Up Output of Fast GPIO

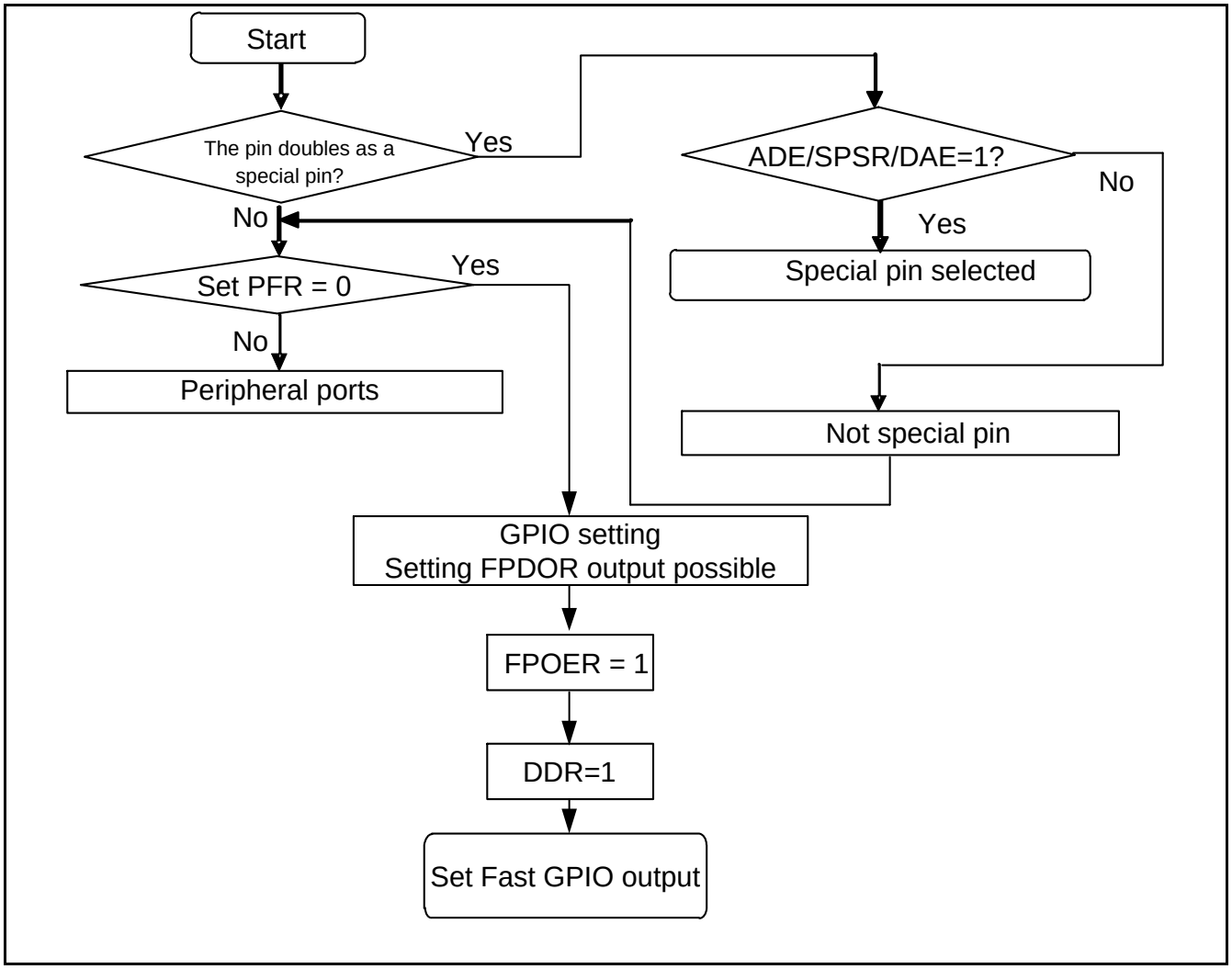
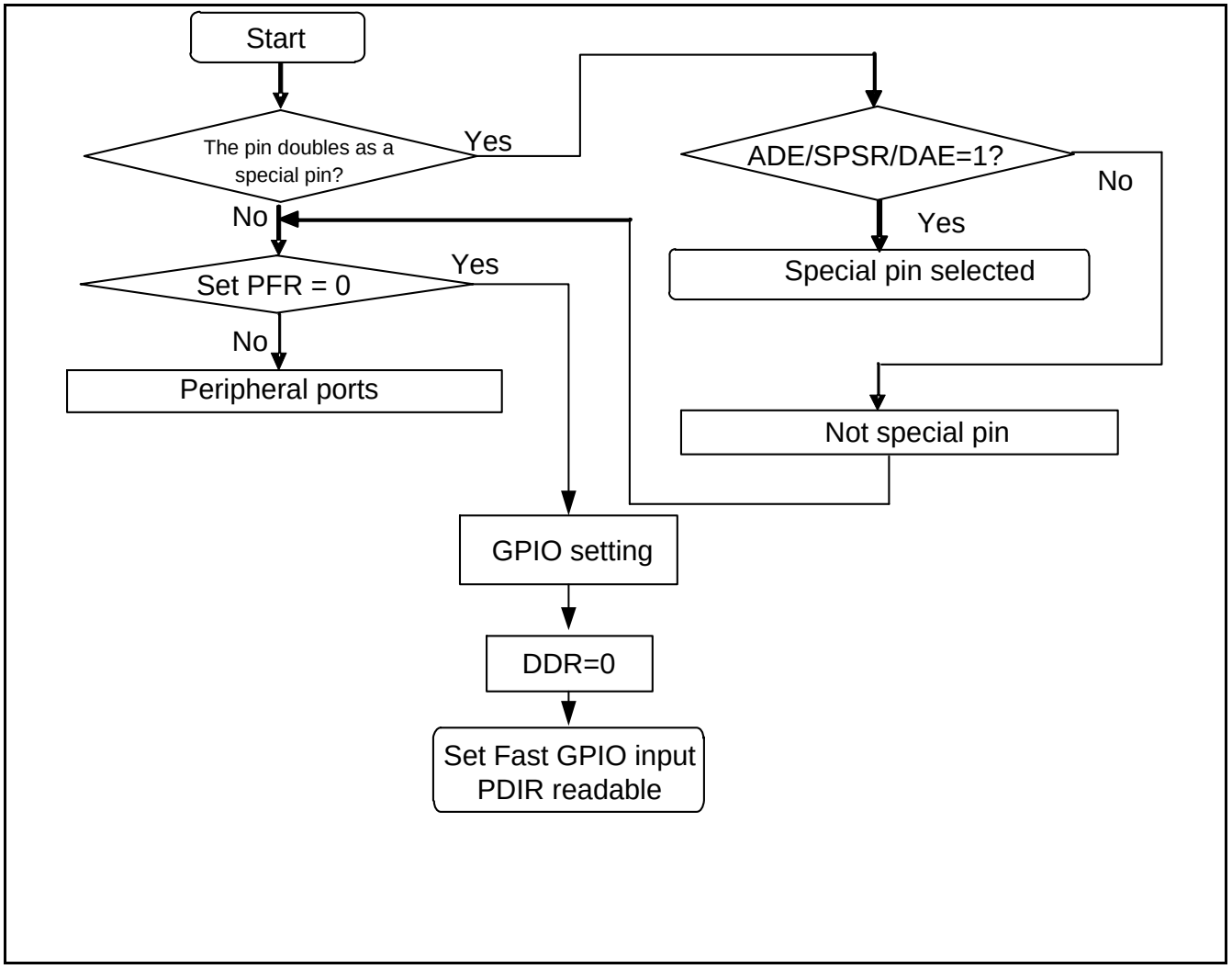


Figure 3-2 Example of Procedure for Setting Up Input of Fast GPIO



4. Registers

This section describes the registers of the Fast GPIO port.

List of Registers of Fast GPIO Port

Table 4-1 List of Registers of Fast GPIO

Abbreviation	Register Name	Reference
FPDIR0	Fast GPIO Input Data Register 0	4.1
FPDIR1	Fast GPIO Input Data Register 1	
FPDIR2	Fast GPIO Input Data Register 2	
FPDIR3	Fast GPIO Input Data Register 3	
FPDIR4	Fast GPIO Input Data Register 4	
FPDIR5	Fast GPIO Input Data Register 5	
FPDIR6	Fast GPIO Input Data Register 6	
FPDIR7	Fast GPIO Input Data Register 7	
FPDIR8	Fast GPIO Input Data Register 8	
FPDIR9	Fast GPIO Input Data Register 9	
FPDIRA	Fast GPIO Input Data Register A	
FPDIRB	Fast GPIO Input Data Register B	
FPDIRC	Fast GPIO Input Data Register C	
FPDIRD	Fast GPIO Input Data Register D	
FPDIRE	Fast GPIO Input Data Register E	
FPDIRF	Fast GPIO Input Data Register F	
FPDOR0	Fast GPIO Output Data Register 0	4.2
FPDOR1	Fast GPIO Output Data Register 1	
FPDOR2	Fast GPIO Output Data Register 2	
FPDOR3	Fast GPIO Output Data Register 3	
FPDOR4	Fast GPIO Output Data Register 4	
FPDOR5	Fast GPIO Output Data Register 5	
FPDOR6	Fast GPIO Output Data Register 6	
FPDOR7	Fast GPIO Output Data Register 7	
FPDOR8	Fast GPIO Output Data Register 8	
FPDOR9	Fast GPIO Output Data Register 9	
FPDORA	Fast GPIO Output Data Register A	
FPDORB	Fast GPIO Output Data Register B	
FPDORC	Fast GPIO Output Data Register C	
FPDORD	Fast GPIO Output Data Register D	
FPDORE	Fast GPIO Output Data Register E	
FPDORF	Fast GPIO Output Data Register F	

Abbreviation	Register Name	Reference
M_FPDIR0	Mirror of Fast GPIO Input Data Register 0	4.3
M_FPDIR1	Mirror of Fast GPIO Input Data Register 1	
M_FPDIR2	Mirror of Fast GPIO Input Data Register 2	
M_FPDIR3	Mirror of Fast GPIO Input Data Register 3	
M_FPDIR4	Mirror of Fast GPIO Input Data Register 4	
M_FPDIR5	Mirror of Fast GPIO Input Data Register 5	
M_FPDIR6	Mirror of Fast GPIO Input Data Register 6	
M_FPDIR7	Mirror of Fast GPIO Input Data Register 7	
M_FPDIR8	Mirror of Fast GPIO Input Data Register 8	
M_FPDIR9	Mirror of Fast GPIO Input Data Register 9	
M_FPDIRA	Mirror of Fast GPIO Input Data Register A	
M_FPDIRB	Mirror of Fast GPIO Input Data Register B	
M_FPDIRC	Mirror of Fast GPIO Input Data Register C	
M_FPDIRD	Mirror of Fast GPIO Input Data Register D	
M_FPDIRE	Mirror of Fast GPIO Input Data Register E	
M_FPDIRF	Mirror of Fast GPIO Input Data Register F	
M_FPDOR0	Mirror of Fast GPIO Output Data Register 0	4.4
M_FPDOR1	Mirror of Fast GPIO Output Data Register 1	
M_FPDOR2	Mirror of Fast GPIO Output Data Register 2	
M_FPDOR3	Mirror of Fast GPIO Output Data Register 3	
M_FPDOR4	Mirror of Fast GPIO Output Data Register 4	
M_FPDOR5	Mirror of Fast GPIO Output Data Register 5	
M_FPDOR6	Mirror of Fast GPIO Output Data Register 6	
M_FPDOR7	Mirror of Fast GPIO Output Data Register 7	
M_FPDOR8	Mirror of Fast GPIO Output Data Register 8	
M_FPDOR9	Mirror of Fast GPIO Output Data Register 9	
M_FPDORA	Mirror of Fast GPIO Output Data Register A	
M_FPDORB	Mirror of Fast GPIO Output Data Register B	
M_FPDORC	Mirror of Fast GPIO Output Data Register C	
M_FPDORD	Mirror of Fast GPIO Output Data Register D	
M_FPDORE	Mirror of Fast GPIO Output Data Register E	
M_FPDORF	Mirror of Fast GPIO Output Data Register F	

Abbreviation	Register Name	Reference
FPOER0	Fast GPIO Output Enable Register 0	4.5
FPOER1	Fast GPIO Output Enable Register 1	
FPOER2	Fast GPIO Output Enable Register 2	
FPOER3	Fast GPIO Output Enable Register 3	
FPOER4	Fast GPIO Output Enable Register 4	
FPOER5	Fast GPIO Output Enable Register 5	
FPOER6	Fast GPIO Output Enable Register 6	
FPOER7	Fast GPIO Output Enable Register 7	
FPOER8	Fast GPIO Output Enable Register 8	
FPOER9	Fast GPIO Output Enable Register 9	
FPOERA	Fast GPIO Output Enable Register A	
FPOERB	Fast GPIO Output Enable Register B	
FPOERC	Fast GPIO Output Enable Register C	
FPOERD	Fast GPIO Output Enable Register D	
FPOERE	Fast GPIO Output Enable Register E	
FPOERF	Fast GPIO Output Enable Register F	

4.1 Fast GPIO Input Data Register (FPDIRx)

The Fast GPIO Input Data Register (FPDIRx) indicates input data of a pin.

List of FPDIRx Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	FPDIR0	0xXXXX	R	P0F to P00
Reserved	FPDIR1	0xXXXX	R	P1F to P10
Reserved	FPDIR2	0xXXXX	R	P2F to P20
Reserved	FPDIR3	0xXXXX	R	P3F to P30
Reserved	FPDIR4	0xXXXX	R	P4F to P40
Reserved	FPDIR5	0xXXXX	R	P5F to P50
Reserved	FPDIR6	0xXXXX	R	P6F to P60
Reserved	FPDIR7	0xXXXX	R	P7F to P70
Reserved	FPDIR8	0xXXXX	R	P8F to P80
Reserved	FPDIR9	0xXXXX	R	P9F to P90
Reserved	FPDIRA	0xXXXX	R	PAF to PA0
Reserved	FPDIRB	0xXXXX	R	PBF to PB0
Reserved	FPDIRC	0xXXXX	R	PCF to PC0
Reserved	FPDIRD	0xXXXX	R	PDF to PD0
Reserved	FPDIRE	0xXXXX	R	PEF to PE0
Reserved	FPDIRF	0xXXXX	R	PFF to PF0

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	FPDIRx

Register Functions

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] FPDIRx: Fast GPIO Input Data Register x

Reads out input data of Fast GPIO.

bit15:0		Description
Reading	0	Regardless of pin function settings (PFR/EPFR/DDR/FPDOR/M_FPDOR), it indicates that a pin is in the status of "L" level input or "L" level output. When a special pin is selected by ADE/SPSR, as input is cut off, 0 is always read out.
	1	Regardless of pin function settings (PFR/EPFR/DDR/FPDOR/M_FPDOR), it indicates that a pin is in the status of "H" level input or "H" level output.
Writing		Writing does not affect anything.

Notes:

- *The "x" of FPDIRx is a wildcard. FPDIRx indicates FPDIR0, FPDIR1, FPDIR2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *One register allows reading input data of 16 ports from PxF to Px0.*
- *Each bit in the register indicates the status of each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of FPDIR0 indicates P0F, the 14th bit of FPDIR0 indicates P0E, and the 0th bit of FPDIR0 indicates P00.*
- *"0" is always read for a bit value of the pin which is not available in your product.*

In case of TYPE2-M0+ product

- *FPDIRx register is not initialized by deep standby transition reset*

In case of TYPE3-M0+ product

- *FPDIRx register is initialized after I/O port is latchd.by deep standby transition reset*

4.2 Fast GPIO Output Data Register x (FPDORx)

The Fast GPIO Output Data Register x (FPDORx) sets output data to a pin.

List of FPDORx Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	FPDOR0	0x0000	R/W	P0F to P00
Reserved	FPDOR1	0x0000	R/W	P1F to P10
Reserved	FPDOR2	0x0000	R/W	P2F to P20
Reserved	FPDOR3	0x0000	R/W	P3F to P30
Reserved	FPDOR4	0x0000	R/W	P4F to P40
Reserved	FPDOR5	0x0000	R/W	P5F to P50
Reserved	FPDOR6	0x0000	R/W	P6F to P60
Reserved	FPDOR7	0x0000	R/W	P7F to P70
Reserved	FPDOR8	0x0000	R/W	P8F to P80
Reserved	FPDOR9	0x0000	R/W	P9F to P90
Reserved	FPDORA	0x0000	R/W	PAF to PA0
Reserved	FPDORB	0x0000	R/W	PBF to PB0
Reserved	FPDORC	0x0000	R/W	PCF to PC0
Reserved	FPDORD	0x0000	R/W	PDF to PD0
Reserved	FPDORE	0x0000	R/W	PEF to PE0
Reserved	FPDORF	0x0000	R/W	PFF to PF0

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	FPDORx

Register Functions

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] FPDORx: Fast GPIO Output Data Register x

Sets output data of Fast GPIO.

bit15:0		Description
Reading		Reads out the register value.
Writing	0	Outputs "L" level to Fast GPIO. If a pin is selected as I/O input or normal GPIO or peripheral functions input/output, a setting value is invalid.
	1	Outputs "H" level to Fast GPIO. If a pin is selected as I/O input or normal GPIO or peripheral functions input/output, a setting value is invalid.

Notes:

- *The "x" of FPDORx is a wildcard. FPDORx indicates FPDOR0, FPDOR1, FPDOR2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *One register allows setting output data of 16 ports from PxF to Px0.*
- *Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of FPDOR0 sets P0F, the 14th bit of FPDOR0 sets P0E, and the 0th bit of FPDOR0 sets P00.*
- *For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.*

In case of TYPE2-M0+ product

- *FPDORx register is not initialized by deep standby transition reset*

In case of TYPE3-M0+ product

- *FPDORx register is initialized after I/O port is latchd.by deep standby transition reset*

4.3 Mirror Fast GPIO Input Data Register (M_FPDIRx)

The Mirror Fast GPIO Input Data Register (M_FPDIRx) indicates input data of a pin.

List of M_FPDIRx Register Configuration

bit 31 to 8	bit 7 to 0	Initial value	Attribute	Corresponding port
Reserved	M_FPDIR0	0xXX	R	M_FP07 to M_FP00
Reserved	M_FPDIR1	0xXX	R	M_FP17 to M_FP10
Reserved	M_FPDIR2	0xXX	R	M_FP27 to M_FP20
Reserved	M_FPDIR3	0xXX	R	M_FP37 to M_FP30
Reserved	M_FPDIR4	0xXX	R	M_FP47 to M_FP40
Reserved	M_FPDIR5	0xXX	R	M_FP57 to M_FP50
Reserved	M_FPDIR6	0xXX	R	M_FP67 to M_FP60
Reserved	M_FPDIR7	0xXX	R	M_FP77 to M_FP70
Reserved	M_FPDIR8	0xXX	R	M_FP87 to M_FP80
Reserved	M_FPDIR9	0xXX	R	M_FP97 to M_FP90
Reserved	M_FPDIRA	0xXX	R	M_FPA7 to M_FPA0
Reserved	M_FPDIRB	0xXX	R	M_FPB7 to M_FPB0
Reserved	M_FPDIRC	0xXX	R	M_FPC7 to M_FPC0
Reserved	M_FPDIRD	0xXX	R	M_FPD7 to M_FPD0
Reserved	M_FPDIRE	0xXX	R	M_FPE7 to M_FPE0
Reserved	M_FPDIRF	0xXX	R	M_FPF7 to M_FPF0

Detailed Register Configuration

Bit	31 to 8	7 to 0
Field	Reserved	M_FPDIRx

Register Functions

[bit31:8] Reserved: Reserved bits

"0x0000" is read out from these bits. When writing these bits, set them to "0x0000".

[bit7:0] M_FPDIRx: Mirror Fast GPIO Input Data Register x

Reads out input data of Fast GPIO.

bit7:0		Description
Reading	0	Regardless of pin function settings (PFR/EPFR/DDR/FPDOR/M_FPDOR), it indicates that a pin is in the status of "L" level input or "L" level output. When a special pin is selected by ADE/SPSR, as input is cut off, 0 is always read out.
	1	Regardless of pin function settings (PFR/EPFR/DDR/FPDOR/M_FPDOR), it indicates that a pin is in the status of "H" level input or "H" level output.
Writing		Writing does not affect anything.

Notes:

- *The "x" of M_FPDIRx is a wildcard. M_FPDIRx indicates M_FPDIR0, M_FPDIR1, etc.*
- *"0" is always read for a bit value of the pin which is not available in your product.*

In case of TYPE2-M0+ product

- *M_FPDIRx register is not initialized by deep standby transition reset*

In case of TYPE3-M0+ product

- *N_FPDIRx register is initialized after I/O port is latchd.by deep standby transition reset*

4.4 Mirror of Fast GPIO Output Data Register (M_FPDORx)

The Mirror of Fast GPIO Output Data Register (M_FPDORx) sets output data to a pin.

List of M_FPDORx Register Configuration

bit 31 to 8	bit 7 to 0	Initial value	Attribute	Corresponding port
Reserved	M_FPDOR0	0x00	R/W	M_FP07 to M_FP00
Reserved	M_FPDOR1	0x00	R/W	M_FP17 to M_FP10
Reserved	M_FPDOR2	0x00	R/W	M_FP27 to M_FP20
Reserved	M_FPDOR3	0x00	R/W	M_FP37 to M_FP30
Reserved	M_FPDOR4	0x00	R/W	M_FP47 to M_FP40
Reserved	M_FPDOR5	0x00	R/W	M_FP57 to M_FP50
Reserved	M_FPDOR6	0x00	R/W	M_FP67 to M_FP60
Reserved	M_FPDOR7	0x00	R/W	M_FP77 to M_FP70
Reserved	M_FPDOR8	0x00	R/W	M_FP87 to M_FP80
Reserved	M_FPDOR9	0x00	R/W	M_FP97 to M_FP90
Reserved	M_FPDORA	0x00	R/W	M_FPA7 to M_FPA0
Reserved	M_FPDORB	0x00	R/W	M_FPB7 to M_FPB0
Reserved	M_FPDORC	0x00	R/W	M_FPC7 to M_FPC0
Reserved	M_FPDORD	0x00	R/W	M_FPD7 to M_FPD0
Reserved	M_FPDORE	0x00	R/W	M_FPE7 to M_FPE0
Reserved	M_FPDORF	0x00	R/W	M_FPF7 to M_FPF0

Detailed Register Configuration

bit	31 to 8	7 to 0
Field	Reserved	M_FPDORx

Register Functions

[bit31:8] Reserved: Reserved bits

"0x0000" is read out from these bits. When writing these bits, set them to "0x0000".

[bit7:0] M_FPDORx: Mirror Fast GPIO Output Data Register x

Sets output data of Fast GPIO.

bit7:0		Description
Reading		Reads out the register value.
Writing	0	Outputs "L" level to Fast GPIO. If a pin is selected as I/O input or normal GPIO or peripheral functions input/output, a setting value is invalid.
	1	Outputs "H" level to Fast GPIO. If a pin is selected as I/O input or normal GPIO or peripheral functions input/output, a setting value is invalid.

Notes:

- *The "x" of M_FPDORx is a wildcard. M_FPDORx indicates M_FPDOR0, M_FPDOR1, etc.*
- *For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.*

In case of TYPE2-M0+ product

- *M_FPDORx register is not initialized by deep standby transition reset*

In case of TYPE3-M0+ product

- *N_FPDORx register is initialized after I/O port is latchd.by deep standby transition reset*

4.5 Fast GPIO Output Enable Register (FPOERx)

The Fast GPIO Output Enable Register (FPOERx) selects the output of the normal GPIO/Fast GPIO.

List of FPOER Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	FPOER0	0x0000	(*1)	P0F to P00
Reserved	FPOER1	0x0000	(*1)	P1F to P10
Reserved	FPOER2	0x0000	(*1)	P2F to P20
Reserved	FPOER3	0x0000	(*1)	P3F to P30
Reserved	FPOER4	0x0000	(*1)	P4F to P40
Reserved	FPOER5	0x0000	(*1)	P5F to P50
Reserved	FPOER6	0x0000	(*1)	P6F to P60
Reserved	FPOER7	0x0000	(*1)	P7F to P70
Reserved	FPOER8	0x0000	(*1)	P8F to P80
Reserved	FPOER9	0x0000	(*1)	P9F to P90
Reserved	FPOERA	0x0000	(*1)	PAF to PA0
Reserved	FPOERB	0x0000	(*1)	PBF to PB0
Reserved	FPOERC	0x0000	(*1)	PCF to PC0
Reserved	FPOERD	0x0000	(*1)	PDF to PD0
Reserved	FPOERE	0x0000	(*1)	PEF to PE0
Reserved	FPOERF	0x0000	(*1)	PFF to PF0

*1 In TYPE3-M0+ product, attribute is R/W. In type of other product, attribute is W.

Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	FPOERx

Register Functions

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.

When writing these bits, set them to "0x0000".

[bit15:0] FPOERx: Fast GPIO Output Enable Register x

Set the I/O port output to normal GPIO or Fast GPIO.

bit15:0		Description
Reading		The read value is undefined
Writing	0	Select the output of PDOR to the pin. If a pin is selected as I/O input or peripheral functions input/output, a setting value is invalid.
	1	Select the output of FPDOR to the pin. If a pin is selected as I/O input or peripheral functions input/output, a setting value is invalid.

Notes:

- *The FPOERx registers are mounted on APB bus.*
- *The "x" of FPOERx is a wildcard. FPOERx indicates FPOER0, FPOER1, FPOER2, etc.*
- *The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.*
- *For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.*
  No matter what the value of FPOERx, the value of the pin can be read by FPDIRx.

In case of TYPE1-M0+ , TYPE2-M0+ product

- *These registers don't support the bit-band access because these registers are write only.*
- *FPOERx register is not initialized by deep standby transition reset*

In case of TYPE3-M0+ product

- *These registers support the bit-band access because these registers are read/write.*
- *FPOERx register is initialized after I/O port is latchd.by deep standby transition reset*

5. Bit Manipulation Base Address

This section describes the base address for bit manipulation to FPDIRx, FPDORx, M_FPDIRx and M_FPDORx.

List of Base Address

Table 5-1 shows the base address for bit manipulation.

Table 5-1 List of Alias Area

Name	Address	Name	Address
FPDIR0 base address	0xF801_0000	FPDOR0 base address	0xF801_4000
FPDIR1 base address	0xF801_0400	FPDOR1 base address	0xF801_4400
FPDIR2 base address	0xF801_0800	FPDOR2 base address	0xF801_4800
FPDIR3 base address	0xF801_0C00	FPDOR3 base address	0xF801_4C00
FPDIR4 base address	0xF801_1000	FPDOR4 base address	0xF801_5000
FPDIR5 base address	0xF801_1400	FPDOR5 base address	0xF801_5400
FPDIR6 base address	0xF801_1800	FPDOR6 base address	0xF801_5800
FPDIR7 base address	0xF801_1C00	FPDOR7 base address	0xF801_5C00
FPDIR8 base address	0xF801_2000	FPDOR8 base address	0xF801_6000
FPDIR9 base address	0xF801_2400	FPDOR9 base address	0xF801_6400
FPDIRA base address	0xF801_2800	FPDORA base address	0xF801_6800
FPDIRB base address	0xF801_2C00	FPDORB base address	0xF801_6C00
FPDIRC base address	0xF801_3000	FPDORC base address	0xF801_7000
FPDIRD base address	0xF801_3400	FPDORD base address	0xF801_7400
FPDIRE base address	0xF801_3800	FPDORE base address	0xF801_7800
FPDIRF base address	0xF801_3C00	FPDORF base address	0xF801_7C00
M_FPDIR0 base address	0xF801_8000	M_FPDOR0 base address	0xF801_C000
M_FPDIR1 base address	0xF801_8400	M_FPDOR1 base address	0xF801_C400
M_FPDIR2 base address	0xF801_8800	M_FPDOR2 base address	0xF801_C800
M_FPDIR3 base address	0xF801_8C00	M_FPDOR3 base address	0xF801_CC00
M_FPDIR4 base address	0xF801_9000	M_FPDOR4 base address	0xF801_D000
M_FPDIR5 base address	0xF801_9400	M_FPDOR5 base address	0xF801_D400
M_FPDIR6 base address	0xF801_9800	M_FPDOR6 base address	0xF801_D800
M_FPDIR7 base address	0xF801_9C00	M_FPDOR7 base address	0xF801_DC00
M_FPDIR8 base address	0xF801_A000	M_FPDOR8 base address	0xF801_E000
M_FPDIR9 base address	0xF801_A400	M_FPDOR9 base address	0xF801_E400
M_FPDIRA base address	0xF801_A800	M_FPDORA base address	0xF801_E800
M_FPDIRB base address	0xF801_AC00	M_FPDORB base address	0xF801_EC00
M_FPDIRC base address	0xF801_B000	M_FPDORC base address	0xF801_F000
M_FPDIRD base address	0xF801_B400	M_FPDORD base address	0xF801_F400
M_FPDIRE base address	0xF801_B800	M_FPDORE base address	0xF801_F800
M_FPDIRF base address	0xF801_BC00	M_FPDORF base address	0xF801_FC00

CHAPTER 11: CRC (Cyclic Redundancy Check)

This chapter explains the CRC functions.

1. Overview of CRC
2. CRC Operations
3. CRC Registers

CODE: FS15-E02.3

1. Overview of CRC

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 that the data is correctly received.

CRC functions

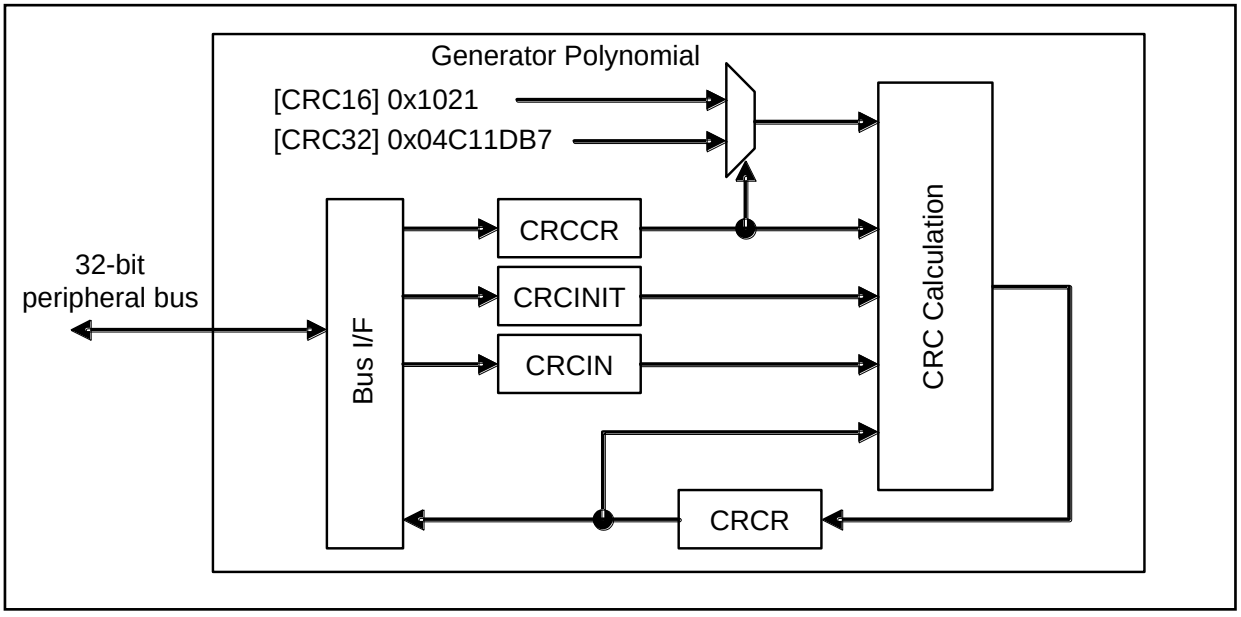
This module enables the calculation in both CCITT CRC16 and IEEE-802.3 CRC32. In this module, the generator polynomial is fixed to the numeric values for those two modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

- CCITT CRC16 generator polynomial: 0x1021
- IEEE-802.3 CRC32 generator polynomial: 0x04C11DB7

CRC Block Diagram

Figure 1-1 shows the CRC block diagram.

Figure 1-1 CRC Block Diagram



- CRCCR (CRC Control Register)
  Used to control CRC calculation.
- CRCINIT (CRC 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. CRC 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 Little endian (CRCCR:LSBFST=1), the sending sequence in bits is configured as shown below.

10100001 01001010 11010101 10000000
(Head) (End)

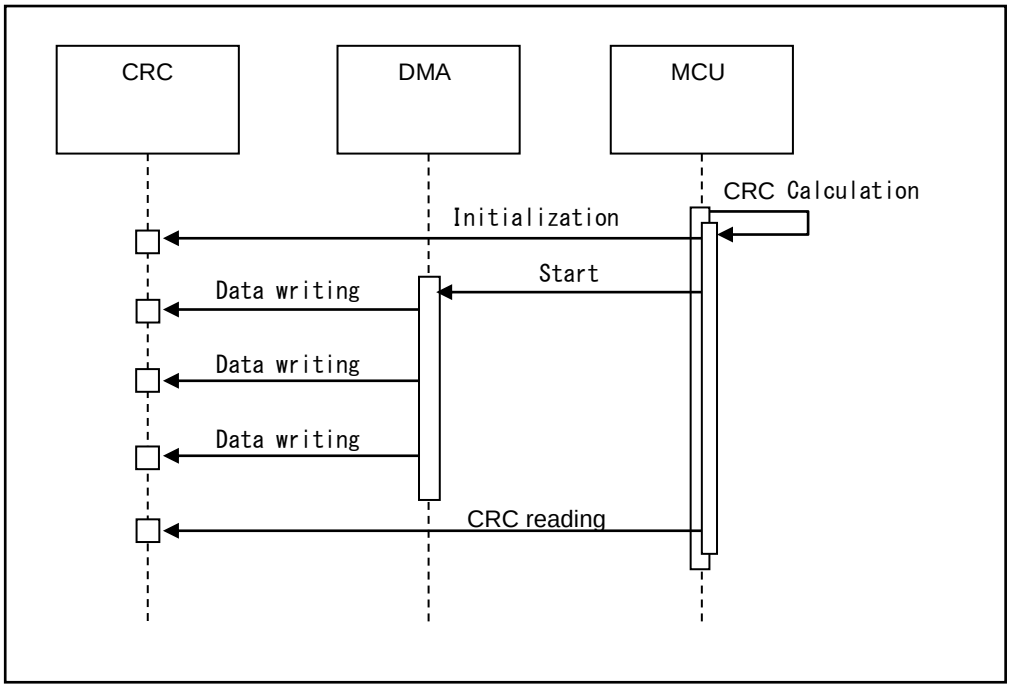
Note:

- *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 bit 31 to bit 16.

2.1 CRC 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 initial value bit (CRCCR.INIT). The value of the Initial Value Register (CRCINT) 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 CRC 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)

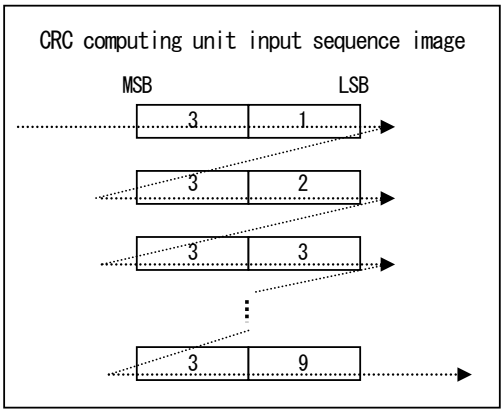
```
//*********************************************
//  CRC16 (CRC ITU-T)
//      polynomial:       0x1021
//      initial value:    0xFFFF
//      CRCCR.CRC32:      0 //CRC16
//      CRCCR.LTLEND:     0 //big endian
//      CRCCR.LSBFST:     0 //MSB First
//      CRCCR.CRCLTE:     0 //CRC big endian
//      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 CPU, CRC result byte order, CRCR (CRC Register) output position, and read address in CRC16 mode.

Table 2-1 CPU, CRC Result Byte Order, and CRCR Read Address

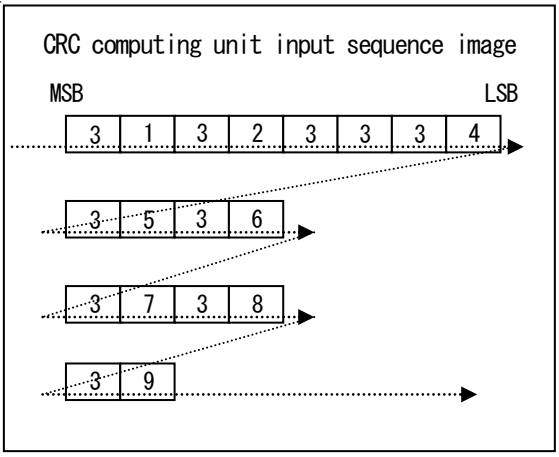
Core byte order	CRC result byte order	Output position to CRCR	CRCR H_READ address
Big endian	Big endian	bit 15 to bit 0	CRCR +2
Big endian	Little endian	bit 31 to bit 16	CRCR +0
Little endian	Big endian	bit 15 to bit 0	CRCR +0
Little endian	Little endian	bit 31 to bit 16	CRCR +2

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);
```

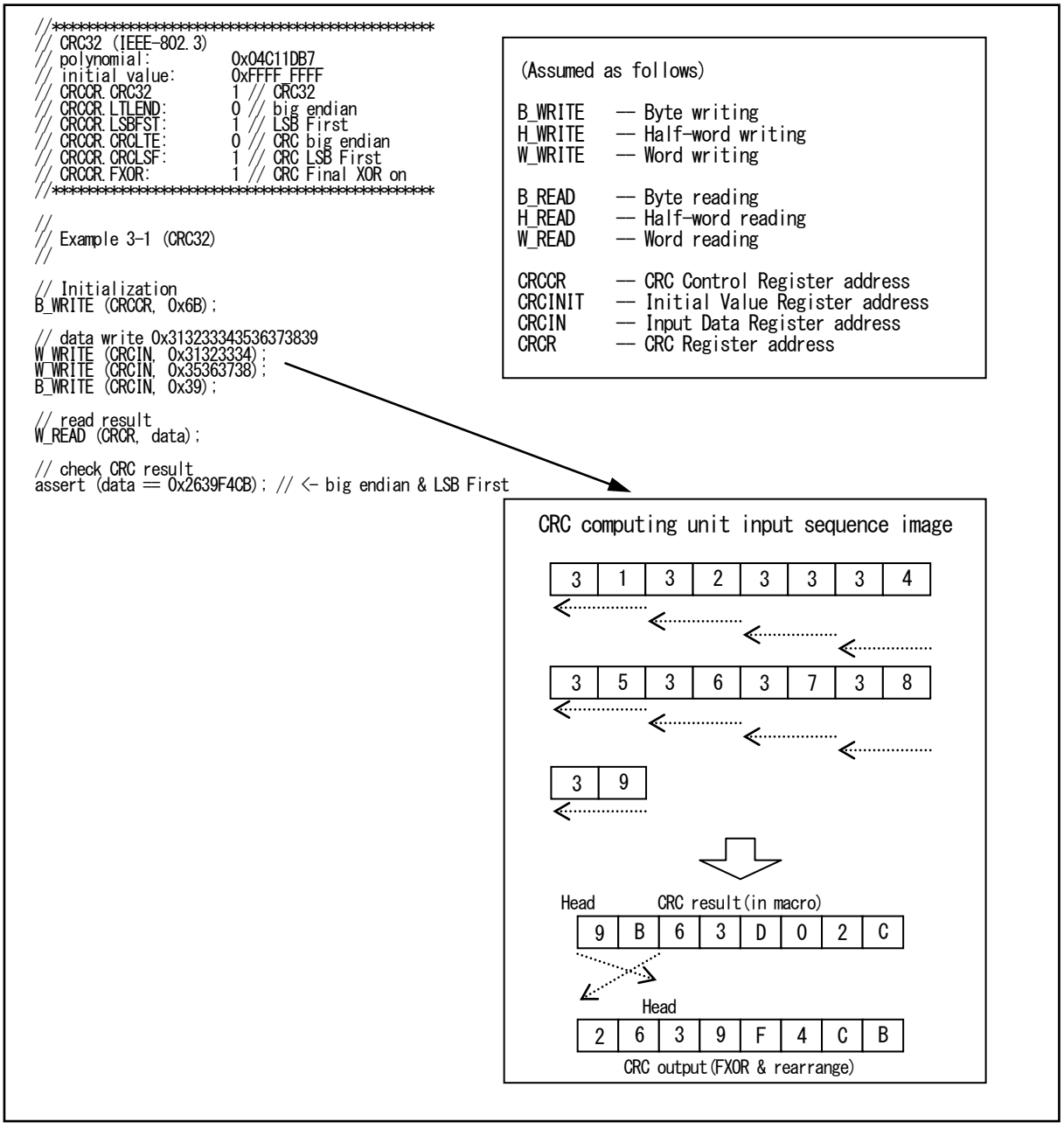
```
(Assumed as follows)

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 finally obtained in the word-base writing mode, both byte and half-word writings may be enabled.

Use Example 3 CRC32, Byte Order: Big Endian

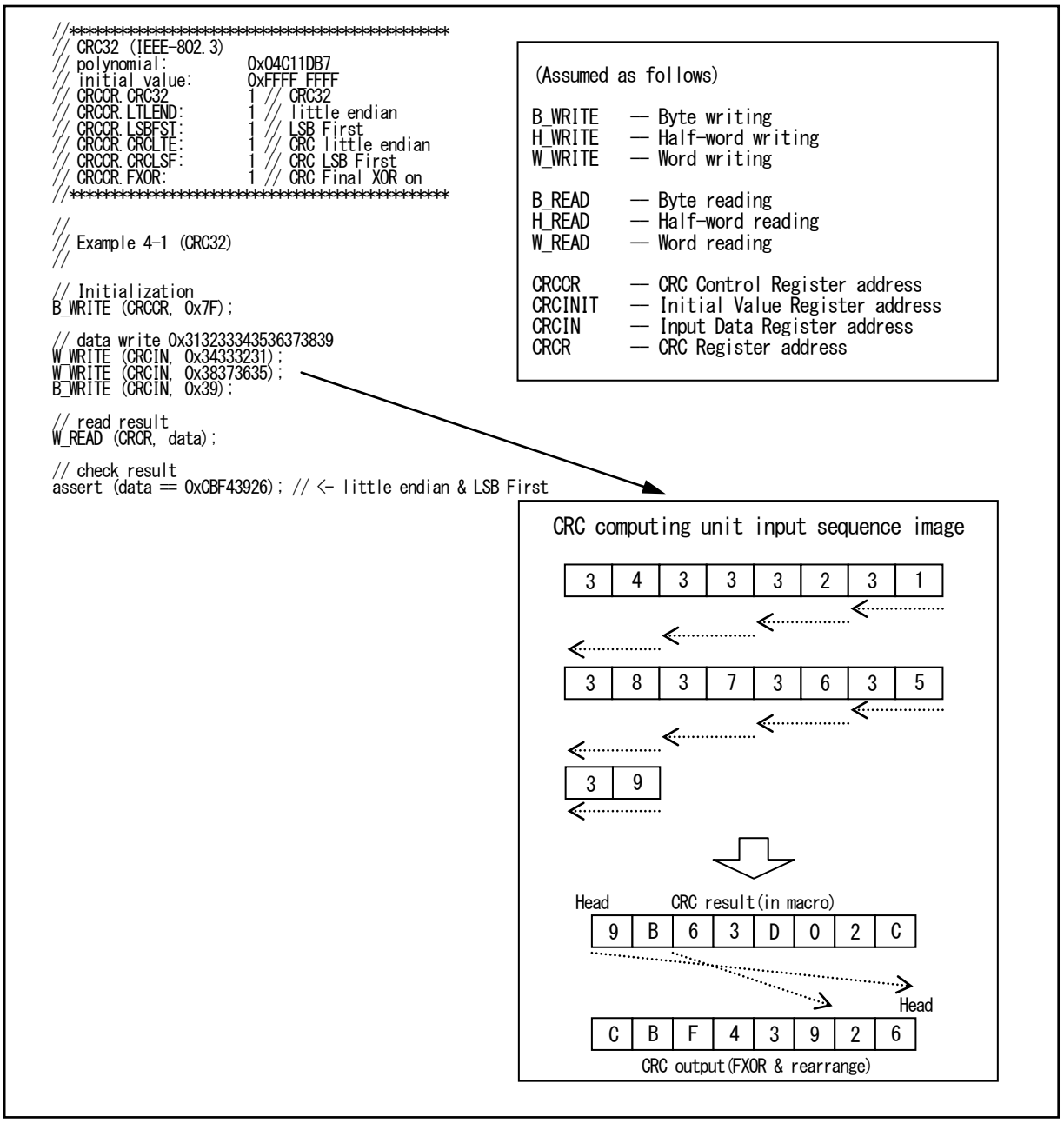
Figure 2-4 Use Example 3 (CRC32, Byte Order: Big Endian)



- 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. Figure 2-4 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. Figure 2-5 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.
  - Before calculation, Initialize with CRCCR=0x3F (CRCCR:FXOR=0, CRCCR:INIT=1).
  - After data was input, set the CRCCR=0x3E (CRCCR:FXOR=0, CRCCR:INIT=0).

3. CRC Registers

This section provides a list of CRC registers.

CRC Registers

Table 3-1 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 (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
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved bit

The read value is "0".
Be sure to write "0" to this bit.

[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 bit 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" to this bit 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	No operation	Always reads "0".
1	Initialization	

3.2 Initial Value Register (CRCINIT)

The Initial Value Register (CRCINIT) is used to save the initial values for CRC calculation.

bit	31	0
Field	D[31:0]	
Attribute	R/W	
Initial value	0xFFFFFFFF	

[bit31:0] D[31:0] : Initial value bits

These bits are used to save the initial values for CRC calculation.

Write the initial values for CRC calculation to this register.

(0xFFFFFFFF at resetting)

In CRC16 mode, D15 to D0 are used while D31 to D16 are ignored.

3.3 Input Data Register (CRCIN)

The Input Data Register (CRCIN) is used to set input data for CRC calculation.

bit	31	0
Field	D[31:0]	
Attribute	R/W	
Initial value	0x00000000	

[bit31:0] D[31:0] : Input data bits

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-bit, 16-bit, and 32-bit (byte, half word, word), 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 (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[31:0]															
Attribute	R															
Initial value	0xFFFFFFFF															

[bit31:0] D[31:0] : CRC bits

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 machine 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 (CRCLTE=0), the result is output to D15 to D0. When the byte order is set to little endian (CRCLTE=1), the result is output to D31 to D16.

CHAPTER 12: Debug Interface

This chapter explains the function and operation of the debug interface.

1. Overview
2. Pin Description

CODE: 9AFDEBUG-E01.0

1. Overview

This Family contains a Serial Wire Debug Port (SW-DP).

Connecting an ICE to the SW-DP allows system debugging.

This series also contains a Micro Trace Buffer (MTB) for recoding changes in a program flow.

This section describes the debugging interface.

For details on the SW-DP and system debug, see "Cortex-M0+ Technical Reference Manual".

Features

Two pins are assigned to the SW-DP.

The initial function of these two pins is serial wire debug.

2. Pin Description

This section explains pins.

2.1 Pins for Debug Purposes
2.2 Functions Initially Assigned to Pins
2.3 Internal Pull-Up of SW-DP Pins

2.1 Pins for Debug Purposes

Two pins (SWCLK and SWDIO) are assigned to the serial wire.

Table 2-1 shows a list of pin functions.

Table 2-1 SW-DP Pin Functions in Debug Mode

Pin	Function
SWCLK	Serial Wire Clock signal
SWDIO	Serial Wire Data Input/Output signal

2.2 Functions Initially Assigned to Pins

The two SW-DP pins are also used as GPIO.
The initial function of the SW-DP pins is debugging.

Note: For details on how to set the debug function, see Chapter "I/O Port"

Table 2-2 shows initial states after resets are cleared and the functions that can be changed by setting PFRs (Port function registers).

Note: For details on the PFRs, see chapter "I/O Port".

Table 2-2 Functions Initially Assigned to Pins for Debugging Purposes and Change of Functions

	Pin Name	Initially Assigned Pin Function	Change of Functions by Setting the PFRs
SW-DP pins	SWCLK	SWCLK	GPIO
	SWDIO	SWDIO	GPIO

2.3 Internal Pull-Up of SW-DP Pins

As specified in the Arm Standard, this Family provides the Debug pins that have internal pull-ups. The user can control pull-up by setting the appropriate registers in the GPIO.

Table 2-3 Enabled or Disabled State of Internal Pull-Up of SW-DP Pins

Pin Name	Pull-Up with Debug Pins Enabled*
SWCLK	Enabled
SWDIO	Enabled

*: Pull-up is enabled even at a reset.

CHAPTER 13: Micro Trace Buffer Data Watchpoint and Trace

This chapter explains the functions and operations of

the MTB_DWT(Micro Trace Buffer Data Watchpoint and Trace).

1. Overview
2. Block Diagram
3. Configuration and Setting Procedure Examples
4. Registers

CODE: 9AFEXTINT-E01.0_FW12-E0.14

1. Overview

The MTB_DWT is to generate TSTOP and TSTART signals of Arm Core-sight MTB. The MTB_DWT function monitors the processor address and data buses when accessing data phase, configurable watch points can be detected to trigger the appropriate response in the MTB recording.

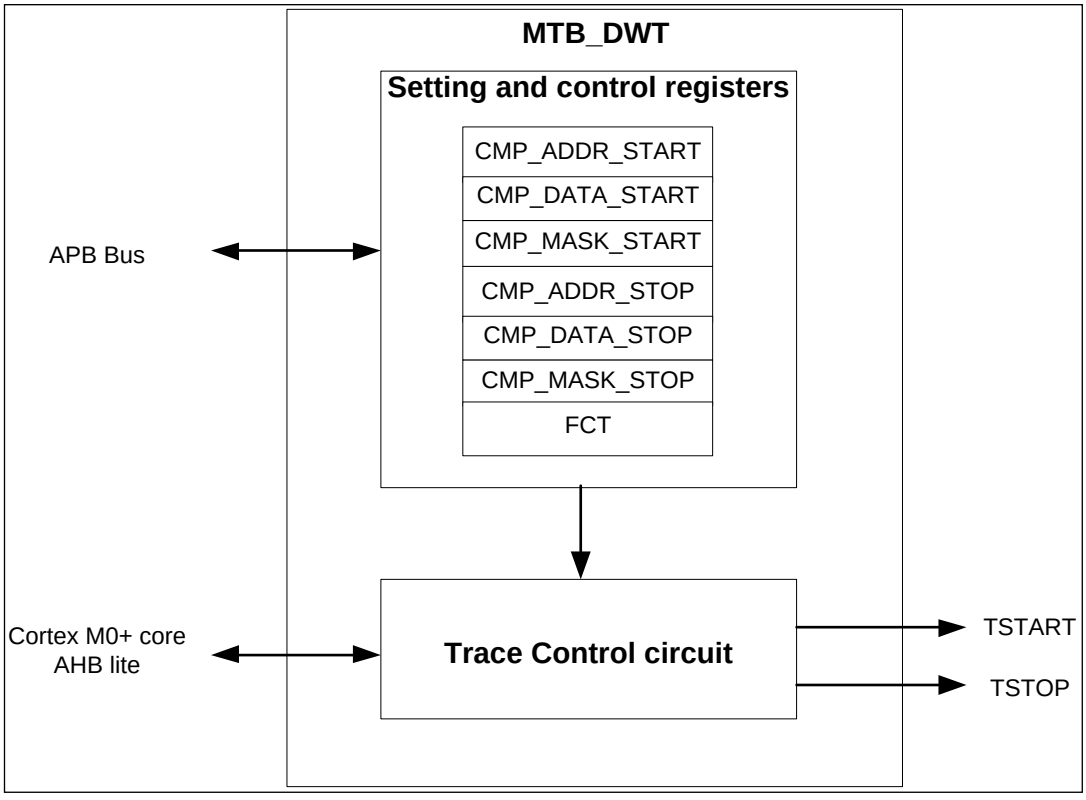
Features of MTB_DWT

- Support addresses and address + data programmable start/stop recording.
- Support masking each bit of data.
- Support write/read and read or write operations monitor.
- Support byte, half-word and word monitor.
- Support data phase monitor function.

2. Block Diagram

The following shows the block diagram of the MTB_DWT.

Figure 2-1 Block Diagram of MTB_DWT



3. Configuration and Setting Procedure Examples

This section explains configurations and setting procedure examples.

3.1 Configurations of MTB_DWT
3.2 Setting

3.1 Configurations of MTB_DWT

This section shows the configurations of the MTB_DWT

Overview of Configuration in MTB_DWT

MTB_DWT only supports little-endian data format. When the byte or half-word access is enabled, the unused bit of MTB_DWT Data Compare Start trace Register (CMP_DATA_START) and MTB_DWT Data Compare Stop trace Register (CMP_DATA_STOP) is shown as Table 3-1.The sign of "❍" in the table shows that it's valid data. The sign of "-" in the table shows that it's an invalid data.
Mask the corresponding bit with invalid data by setting CMP_MASK_START/CMP_MASK_STOP.

Table 3-1 AHB-Lite Byte Lane Definition

Access	Address Phase	Corresponding Data Phase			
	CMP_ADDRx*1 [1:0]	CMP_DATAx*2 [31:24]	CMP_DATAx*2 [23:16]	CMP_DATAx*2 [15:8]	CMP_DATAx*2 [7:0]
Byte	00	-	-	-	❍
	01	-	-	❍	-
	10	-	❍	-	-
	11	❍	-	-	-
Half-word	00	-	-	❍	❍
	10	❍	❍	-	-
Word	00	❍	❍	❍	❍

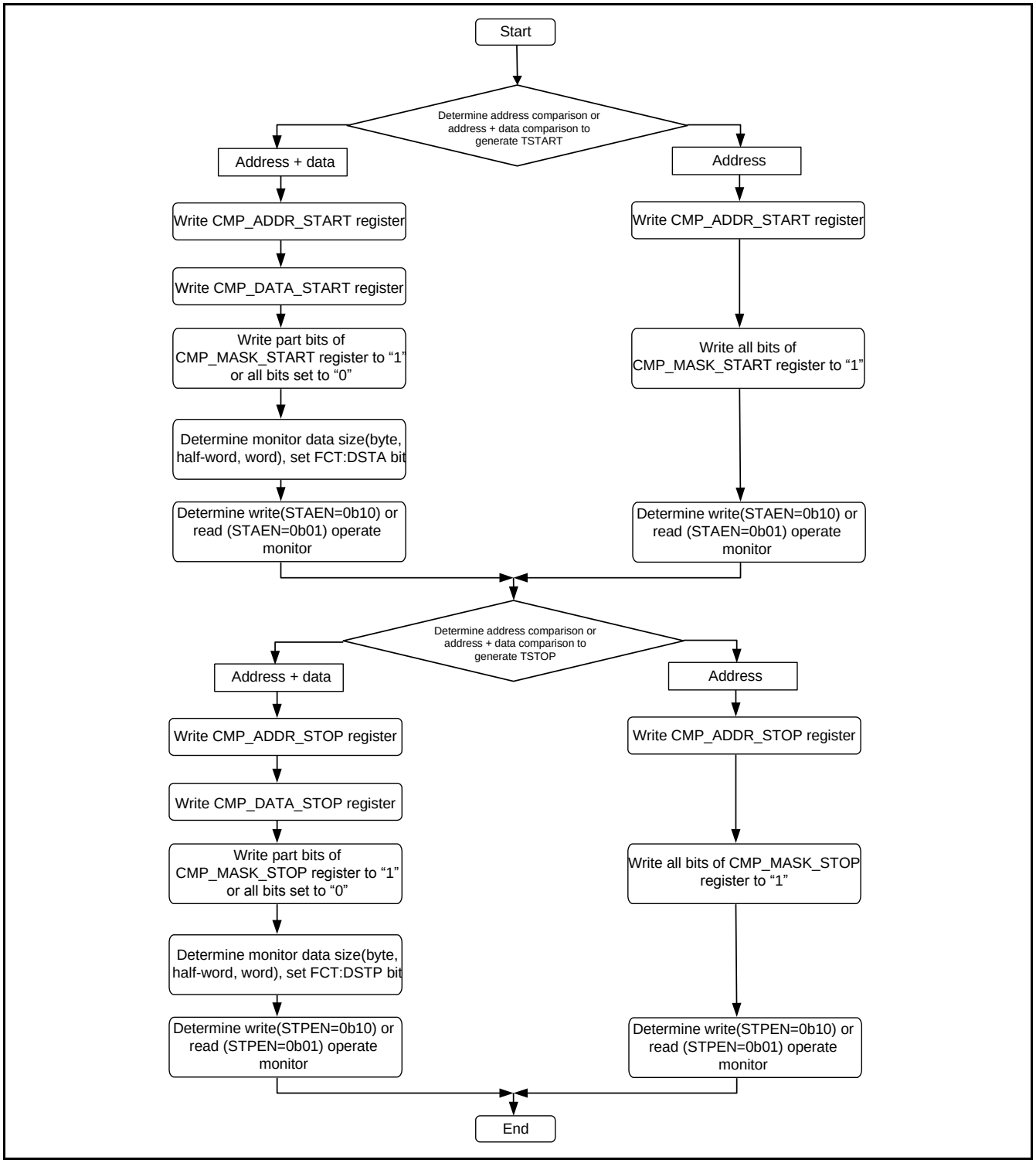
*1: CMP_ADDRx: CMP_ADDR_START register or CMP_ADDR_STOP register.

*2: CMP_DATAx: CMP_DATA_START register or CMP_DATA_STOP register.

3.2 Setting Procedure

For MTB_DWT setting procedure example, see Figure 3-1

Figure 3-1 MTB_DWT Setting Procedure



4. Registers

This section describes the registers of the MTB_DWT

List of Registers of the MTB_DWT

Table 4-1 List of Registers of the MTB_DWT

Abbreviation	Register Name	Reference
CMP_ADDR_START	MTB_DWT Address Compare Start trace Register	4.1
CMP_DATA_START	MTB_DWT Data Compare Start trace Register	4.2
CMP_MASK_START	MTB_DWT Mask Data Compare Start trace Register	4.3
CMP_ADDR_STOP	MTB_DWT Address Compare Stop trace Register	4.4
CMP_DATA_STOP	MTB_DWT Data Compare Stop trace Register	4.5
CMP_MASK_STOP	MTB_DWT Mask Data Compare Stop trace Register	4.6
FCT	MTB_DWT Function Register	4.7
PID4	Peripheral ID4 Register	4.8
PID5	Peripheral ID5 Register	
PID6	Peripheral ID6 Register	
PID7	Peripheral ID7 Register	
PID0	Peripheral ID0 Register	
PID1	Peripheral ID1 Register	
PID2	Peripheral ID2 Register	
PID3	Peripheral ID3 Register	
CID0	Component ID0 Register	4.9
CID1	Component ID1 Register	
CID2	Component ID2 Register	
CID3	Component ID3 Register	

4.1 MTB_DWT Address Compare Start Trace Register (CMP_ADDR_START)

The MTB_DWT Address Compare Start trace Register (CMP_ADDR_START) provide a reference address value for generating start trigger signal TSTART.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	ADCMP_STA[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ADCMP_STA[15:0]															
Attribute	R/W															
Initial value	0x0000															

Register Functions

[bit31:0] ADCMP_STA[31:0]: MTB_DWT address comparison start trace bits

Reference value for address comparison is to generate MTB start trigger.

Notes:

- *When word accessing, the bit[3:0] must set to one of 0x0, 0x4, 0x8 and 0xC.*
- *When half-word accessing, the bit[1:0] must set to one of 0x0 and 0x2.*
- *This register can be cleared by PRESET0. After PRESET0, this register should be configured again.*

4.2 MTB_DWT Data Compare Start Trace Register (CMP_DATA_START)

The MTB_DWT Data Compare Start trace Register (CMP_DATA_START) provides a reference data value for generating start trigger signal TSTART.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DTCMP_STA[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DTCMP_STA[15:0]															
Attribute	R/W															
Initial value	0x0000															

Register Functions

[bit31:0] DTCMP_STA[31:0]: MTB_DWT data comparison start trace bits

Reference value for data comparison is to generate MTB start trigger.

When all bits are masked by CMP_MASK_START, the MTB TSTART signal is only determined by comparative address result, otherwise the MTB TSTART signal is determined by both data CMP_DATA_START and CMP_ADDR_START comparative result.

Notes:

- *When byte or half-word accessing, set CMP_MASK_START register following Table 3-1 to ignore unused bit in CMP_DATA_START register.*
- *This register can be cleared by PRESET0. After PRESET0, this register should be configured again.*

4.3 MTB_DWT Mask Data Compare Start Trace Register (CMP_MASK_START)

The MTB_DWT Mask Data Compare Start trace Register (CMP_MASK_START) defines the ignored bit of MTB_DWT Data Compare Start trace Register (CMP_DATA_START).

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	MSK_STA[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	MSK_STA [15:0]															
Attribute	R/W															
Initial value	0x0000															

Register Functions

[bit31:0] MSK_STA[31:0]: MTB_DWT data compare start trace register mask bits

This register masks reference data value for starting MTB.

Bit	Function
0	No effect on operation [Initial value]
1	Mask corresponding bits

Notes:

- *When all bits of this register are set to “1”, the value of CMP_DATA_START register and the value of DSTA bit of FCT register will be ignored.*
- *This register can be cleared by PRESET0. After PRESET0, this register should be configured again.*

4.4 MTB_DWT Address Compare Stop Trace Register (CMP_ADDR_STOP)

The MTB_DWT Address Compare Stop trace Register (CMP_ADDR_STOP) provide a reference address value for generating stop trigger signal TSTOP.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	ADCMP_STO[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ADCMP_STO[15:0]															
Attribute	R/W															
Initial value	0x0000															

Register Functions

[bit31:0] ADCMP_STO[31:0]: MTB_DWT address comparison stop trace bits

Reference value for address comparison is to generate MTB stop trigger.

Notes:

- *When word accessing, the bit[3:0] must set to one of 0x0, 0x4, 0x8 and 0xC.*
- *When half-word accessing, the bit[1:0] must set to one of 0x0 and 0x2.*
- *This register can be cleared by PRESET0. After PRESET0, this register should be configured again.*

4.5 MTB_DWT Data Compare Stop Trace Register (CMP_DATA_STOP)

The MTB_DWT Data Compare Stop trace Register (CMP_DATA_STOP) provides a reference data value for generating stop trigger signal TSTOP.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DTCMP_STO[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DTCMP_STO[15:0]															
Attribute	R/W															
Initial value	0x0000															

Register Functions

[bit31:0] DTCMP_STO[31:0]: MTB_DWT data comparison stop trace bits

Reference value for data comparison is to generate MTB stop trigger.

When all bits are masked by MTB_DWT Mask data Compare Stop trace Register, the MTB TSTOP signal is only determined by comparative address result, otherwise the MTB TSTOP signal is determined by both data (MTB_DWT Data Compare Stop trace Register) and address (MTB_DWT Address Compare Stop trace Register) comparative result.

Notes:

- *When byte or half-word accessing, set CMP_MASK_STOP register following Table 3-1 <Not to ignore unused bit in CMP_DATA_STOP register.*
- *This register can be cleared by PRESET0. After PRESET0, this register should be configured again.*

4.6 MTB_DWT Mask Data Compare Stop Trace Register (CMP_MASK_STOP)

The MTB_DWT Mask Data Compare Stop Trace Register (CMP_MASK_STOP) defines the ignored bit of register MTB_DWT Data Compare Stop trace Register (CMP_DATA_STOP).

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	MSK_STO[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	MSK_STO[15:0]															
Attribute	R/W															
Initial value	0x0000															

Register Functions

[bit31:0] MSK_STO[31:0]: MTB_DWT data compare stop trace register mask bits

This register masks reference data value for stopping MTB.

Bit	Function
0	No effect on operation. [Initial value]
1	Mask corresponding bits.

Notes:

- *When all bits of this register are set to “1”, the value of CMP_DATA_STOP register and the value of DSTP bit of FCT register will be ignored.*
- *This register can be cleared by PRESET0. After PRESET0, this register should be configured again.*

4.7 MTB_DWT Function Register (FCT)

The MTB_DWT Function Register (FCT) controls read/write operation and data size.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	-															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								DSTP		DSTA		STPEN		STAEN	
Attribute	-								R/W		R/W		R/W		R/W	
Initial value	00000000								00		00		00		00	

Register Functions

[bit31:8] Reserved: Reserved bits

The read value is “0”. They have no effect in write mode.

[bit7:6] DSTP: Data size stop bits

These bits define the data value size to stop MTB function.

bit7	bit6	Function
0	0	Byte [Initial value]
0	1	Half-word
1	0	Word
1	1	Reserved.

[bit5:4] DSTA: Data size start bits

These bits define the data value size to start MTB function.

bit5	bit4	Function
0	0	Byte [Initial value]
0	1	Half-word
1	0	Word
1	1	Reserved.

[bit3:2] STPEN: Enable MTB_DWT stop MTB function bits

These bits enable MTB_DWT stop MTB function.

bit3	bit2	Function
0	0	Disabled MTB_DWT stop MTB function. [Initial value]
0	1	Data read operation to stop MTB.
1	0	Data write operation to stop MTB.
1	1	Data write or read operation to stop MTB.

[bit1:0] STAEN: Enable MTB_DWT start MTB function bits

These bits enable MTB_DWT start MTB function.

bit1	bit0	Function
0	0	Disabled MTB_DWT start MTB function. [Initial value]
0	1	Data read operation to start MTB.
1	0	Data write operation to start MTB.
1	1	Data write or read operation to start MTB.

Notes:

- *This register can be cleared by PRESET0. After PRESET0, this register should be configured again.*

4.8 Peripheral ID0-7 Register (PID0-7)

The Peripheral ID0-7 Registers (PID0-7) indicate the peripheral IDs.

Register Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	PERID[31:16]															
Attribute	R															
Initial Value	0xXXXX															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PERID[15:0]															
Attribute	R															
Initial Value	0xXXXX															

Register Functions

[bit31:0] PERID[31:0]: Peripheral ID bits

They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Bit	Function
When Read	A specified value is read. PID0: 0x00000016 PID1: 0x00000048 PID2: 0x00000008 PID3-7: 0x00000000
When Write	No effect on operation

4.9 Component ID0-3 Register (CID0-3)

The Component ID0-3 Registers (CID0-3) indicate the component IDs.

Register Configuration

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	CPNTID[31:16]															
Attribute	R															
Initial value	0xXXXX															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CPNTID[15:0]															
Attribute	R															
Initial value	0xXXXX															

Register Functions

[bit31:0] CPNTID[31:0]: Component ID bits

They are hardwired to specific values used during the auto-discovery process by an external debug agent.

Bit	Function
When Read	A specified value is read. CID0: 0x0000000D; CID1: 0x00000090; CID2: 0x00000005; CID3: 0x000000B1.
When Write	No effect on operation

CHAPTER 14: Flash Memory

For the flash memory, refer to the “FLASH PROGRAMMING MANUAL” of the product to be used.

CODE: 9xFLASHTOP-E01.1

CHAPTER 15: Unique ID Register

Functions and operations of Unique ID Register are explained as follows.

1. Overview
2. Registers

CODE: 9BFUNIQID-J01.0

1. Overview

Overview of this function is explained as follows.

41 bits of preset device unique values have been set to the Unique ID Register.

These values are different from each other in all of the devices which allow using these bits for various purposes such as security enhancement and product serial number.

This register is a read-only register which cannot be written by the user. Also, these values will not be changed due to reset or power on/off.

2. Registers

Configuration and functions of registers are explained as follows.

Registers List

Abbreviated Name	Register Name	Reference
UIDR0	Unique ID Register 0	2.1
UIDR1	Unique ID Register 1	2.2

2.1 Unique ID Register 0 (UIDR0: Unique ID Register 0)

Unique ID Register 0 is explained as follows.

bit	31	30	29	28	27	26	25	24
Field	UID[27:20]							
Attribute	R							

bit	23	22	21	20	19	18	17	16
Field	UID[19:12]							
Attribute	R							

bit	15	14	13	12	11	10	9	8
Field	UID[11:4]							
Attribute	R							

bit	7	6	5	4	3	2	1	0
Field	UID[3:0]				Reserved			
Attribute	R				-			

[bit31:4] UID[27:0] : Unique ID 27 through Unique ID 0

bit27 through bit0 of the unique ID.

[bit3:0] Reserved : Reserved bits

Reserved bits. Read values have no meaning.

2.2 Unique ID Register 1 (UIDR1: Unique ID Register 1)

Unique ID Register 1 is explained as follows.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved			UID[40:36]				
Attribute	-			R				

bit	7	6	5	4	3	2	1	0
Field	UID[35:28]							
Attribute	R							

[bit31:13] Reserved : Reserved bits

Reserved bits. Read values have no meaning.

[bit12:0] UID[40:28] : Unique ID 40 through Unique ID 28

bit 40 through bit28 of unique ID.

CHAPTER 16: DSTC

This chapter explains details of the DSTC (Descriptor System data Transfer Controller).

1. Overview of DSTC
2. DSTC Operations Overview and DSTC System Configuration
3. Functions and Operations of DSTC
4. Examples of DSTC Operations and Control
5. Registers and Descriptors of DSTC

1. Overview of DSTC

This section provides an overview of the DSTC (Descriptor System data Transfer Controller).

Overview

The DSTC (Descriptor System data Transfer Controller), like the DMAC, is a function block that can transfer data at high speed bypassing the CPU. Using the Descriptor (to be called DES later in this document) System Method, it directly accesses memory or a peripheral device according to the content specified in a DES created on the memory, and executes a data transfer operation.

One set of transfer control details (basic transfer settings, number of transfers, transfer source address, transfer destination address) is specified in one DES. The DSTC can multiple DES individually and can build up to 1024 transfer channels.

The data transfer operation can be started by one of the following three methods: direct start by the CPU (software start), start by an interrupt signal from a peripheral device (hardware start), and the Chain Start Function.

The chain start function executes a transfer according to the current DES, and then starts a new transfer according to the succeeding DES or according to the current DES again. It can be specified in the DES whether to use the Chain Start Function. With the chain start function, the DSTC can incorporate other types of transfer specified in multiple DES into a single start trigger (software start / hardware start) in the start DES, and execute such types of transfer together. In addition, it can also divide a transfer operation specified in a DES into several transfer operations and then execute them.

The DSTC has two reload functions for the transfer address and for the transfer count counter (the InnerReload Function that during a transfer makes the value return to the one at the start of the transfer, the OuterReload Function that at the end of a transfer makes the value return to the one at the start of the transfer). The two reload functions facilitate the control of repeating the same transfer operation.

The DSTC can notify the CPU of the normal end or abnormal end of a transfer operation as an interrupt.

It can control how an internal clock is stopped in a standby mode (low power consumption mode).

The DSTC has a dedicated bus, which is independent of the CPU bus, and has a configuration enabling it to execute a transfer operation when the CPU bus is being accessed.

The configuration of the dedicated bus conforms to the system bus (AHB) and supports a 32-bit address space (4 Gbyte).

Number of Channels Supported in Hardware Transfer of DSTC

For a product equipped with the DSTC, if the DSTC supports 256 channels, it can use all hardware transfer channels from channel 0 to channel 255. If the DSTC supports 128 channels, it cannot use channel 128 to channel 255. If the DSTC supports 64 channels, it cannot use channel 64 to channel 255.

2. DSTC Operations Overview and DSTC System Configuration

This section provides an overview of operations of the DSTC and explains the DSTC system configuration.

2.1 Operations Overview of DSTC

DES System

The DSTC executes a transfer operation according to the content specified in a DES built on the memory by the CPU. As shown in Table 2-1, a DES consists of seven settings, DES0 to DES6. The settings specify transfer basic settings, the number of transfers, the transfer source address and the transfer destination address in their respective areas. (For details of the bit assignment of each DES, see "5 Registers and Descriptors of DSTC".) These settings are stored in the memory area as a single group. DES0 to DES6 are all 32 bits (1 word) in size. DES4 to DES6 are optional DES. Their settings may not need to be specified depending on the content of a transfer.

Table 2-1 Types of DES and Settings

Storage address	Name	Details
DESP+0x00 (fixed)	DES0	This sets the basic settings of a transfer.
DESP+0x04 (fixed)	DES1	This sets the number of transfers.
DESP+0x08 (fixed)	DES2	This sets the transfer source address (SA) at which a transfer starts.
DESP+0x0C (fixed)	DES3	This sets the transfer destination address (DA) at which a transfer ends.
DESP+0x10 - (variable)	DES4	This controls the OuterReload of DES1 at the end of a transfer.
	DES5	This controls the OuterReload of DES2 at the end of a transfer.
	DES6	This controls the OuterReload of DES3 at the end of a transfer.

Figure 2-1 illustrates the configuration of the DES System Method of the DSTC. When using the DSTC, reserve adequate free memory area for storing the DES. Select a memory area that is readable and writable because the DSTC has to refer to and update the DES. Set (1. in Figure 2-1) the start address of the memory area using the DESTP (DES-Top-address) register of the DSTC. Multiple DES can be allocated to a 4096-word (16 Kbyte) area starting from the DESTP. Up to 1024 DES can be allocated in the area. The DSTC identifies a DES according to its address value (DESP: DES-pointer) relative to the DES0 area starting from the DESTP.

Start of the Transfer

After transfer information has been stored in a DES (2.in Figure 2-1), the DSTC transfer can be started by one of the following three start trigger methods.

- Software-Start (SW Start)

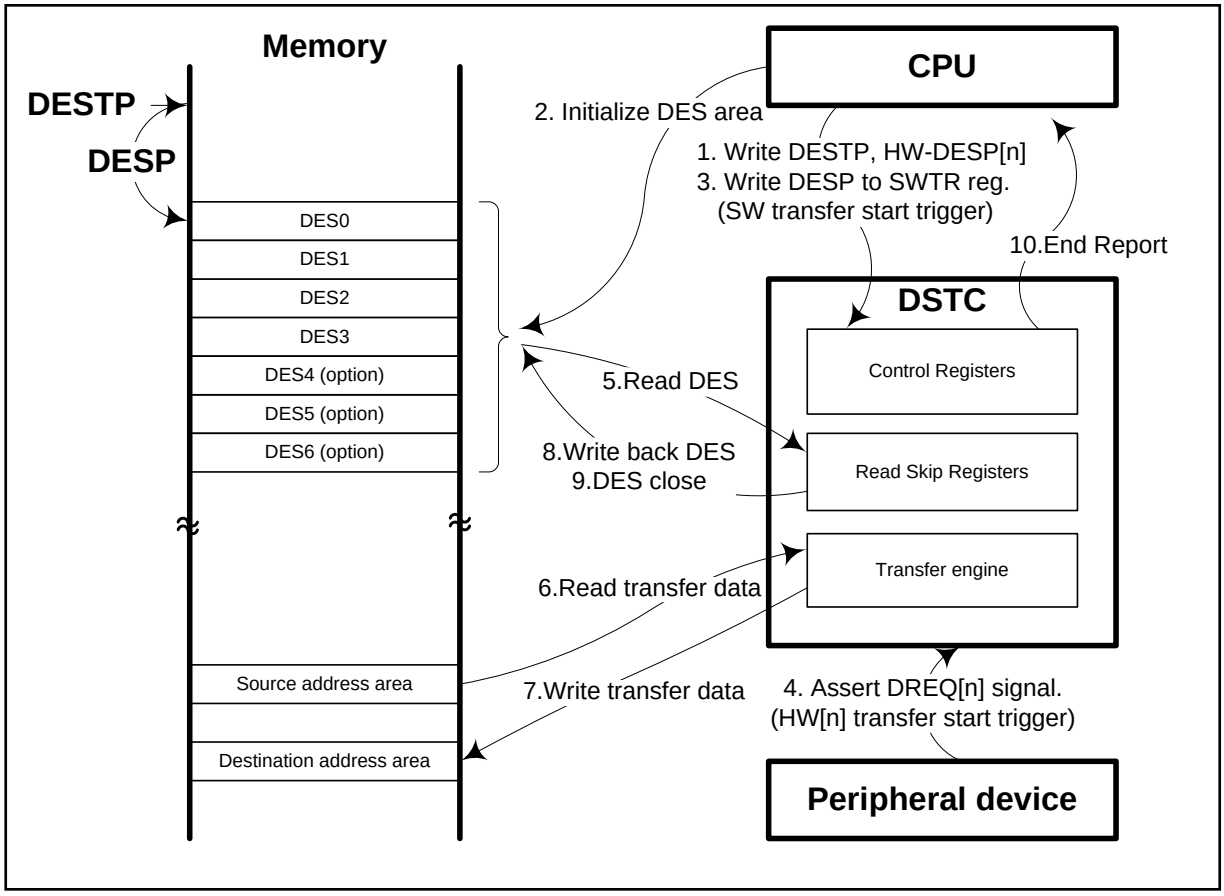
  The Software-Start is a start trigger method to start the DSTC transfer directly from the CPU. The transfer is started by writing the DESP of the DES to be used to the SWTR (Software Trigger) Register (3 in Figure 2-1).Software-Start is called SW Start, and the transfer of the DSTC by SW Start is called SW Transfer later in this document.
- Hardware-Start (HW Start)

  The Hardware-Start is a start trigger method to start the DSTC transfer with the interrupt signal from a peripheral as a transfer request signal. When an interrupt signal from a peripheral has been asserted (4 in Figure 2-1), the transfer of the DSTC starts bypassing the CPU. In advance, write the DESP of the DES of this transfer to the HWDESP register (Hardware DESP) corresponding to an HW channel in the DSTC (1 in Figure 2-1). Hardware-Start is called HW Start, and the transfer of the DSTC by HW Start is called HW Transfer later in this document.
- Chain Start

  Chain start is a start trigger method that is described in the DES. After the transfer in the DES ended, If the DES have Chain start trigger, the DSTC starts a new transfer according to the succeeding DES (or the same DES). The new DESP of DES for chain transfer is calculated from current DESP automatically.

In the following sections, "Start Trigger" represents all the above start triggers, SW Start, HW Start and Chain Start.

Figure 2-1 DES System Method Configuration Diagram



Operation of the Transfer

The DSTC refers to (5 in Figure 2-1) a DES in DESTP+DESP according to a Start Trigger mentioned above. The DSTC checks (DES Open Check) details of the DES it refers to, and executes a transfer (6 and 7 in Figure 2-1) if the details have no problem. In addition, if the DES has a Chain Start Trigger, the DSTC executes a transfer data according to the Chain Start trigger.

The number of transfers to be executed at one Start Trigger varies according to details of a DES and those of the Chain Trigger. Not all transfers specified in a DES may end at one Start Trigger. If that occurs, the number of transfers ended and updated transfer addresses are written back to a DES (8 in Figure 2-1). The DSTC waits for the next Start Trigger, and continues executing the transfer after receiving the next Start Trigger.

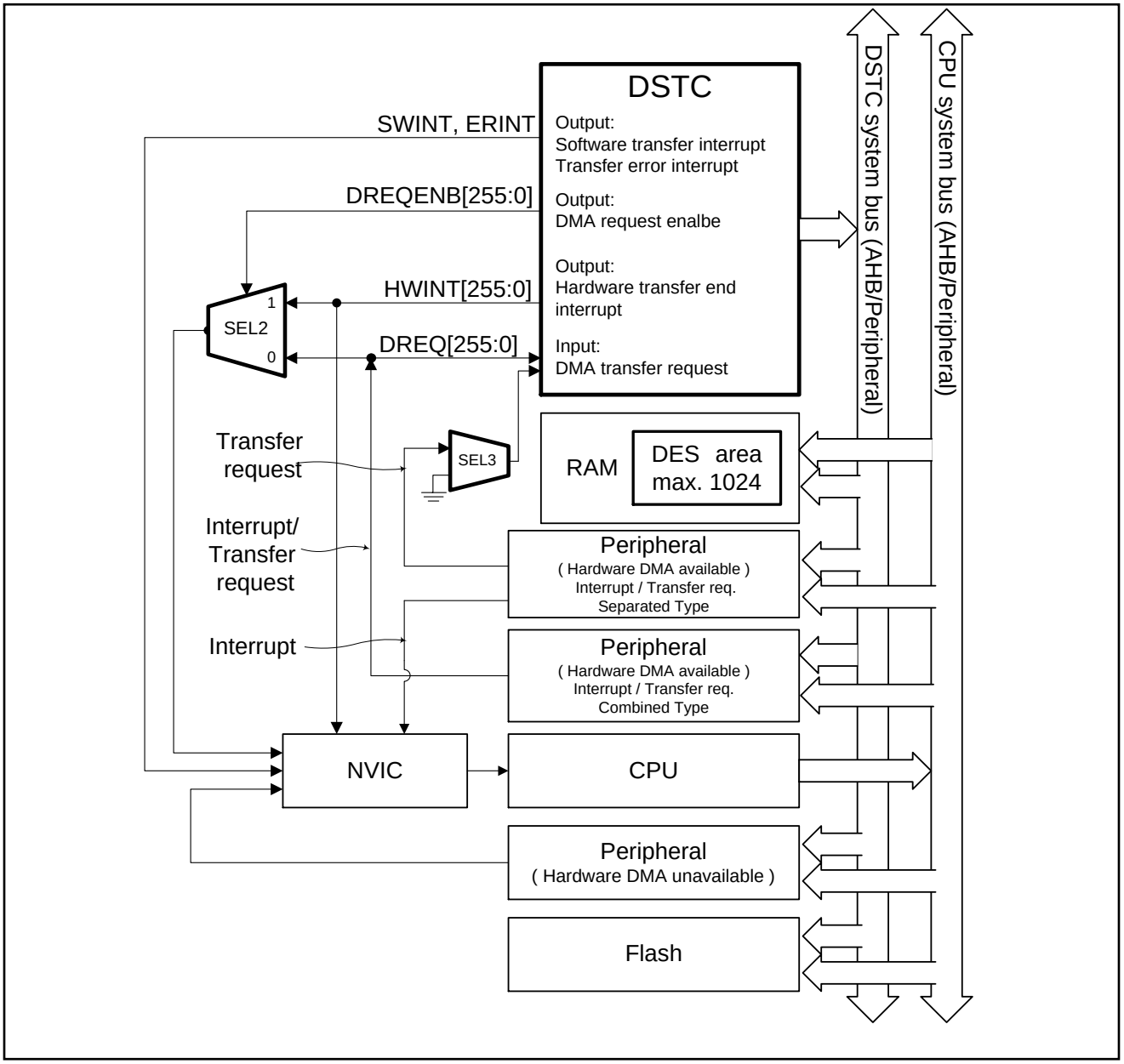
End of the Transfer

The DSTC executes the DES close process (9 in Figure 2-1) after all transfers specified in a DES have ended. The DES settings can prevent the DES close process from being executed. The DSTC can notify the CPU of the following events through an interrupt: i. the DSTC is waiting for a Start Trigger; ii. a transfer has ended normally; iii. a transfer has ended abnormally. (10 in Figure 2-1)

2.2 DSTC and System Configuration

Figure 2-2 shows the block diagram illustrating the DSTC and system configuration.

Figure 2-2 Block Diagram of DSTC and System Configuration



Connection with System

The system configuration diagram in Figure 2-2 has been simplified to facilitate explanation. For details of the system configuration, refer to chapter System Overview in Peripheral Manual. The DSTC is connected to the CPU, Flash, RAM and peripherals via the system bus. The DSTC has a dedicated bus, which is independent of the CPU bus, and has a configuration enabling it to execute a transfer operation when the CPU bus is being accessed. The DSTC accesses any address area on the system according to the specified transfer destination address and transfer source access of a channel, and executes data transfer between the memory and a peripheral. The DSTC cannot access certain areas. Refer to the memory map to check which areas the DSTC cannot access.

DREQENB[n] Register and Connection with DREQ[n] Signal and HWINT[n] Signal

The DSTC supports up to 256 hardware transfer request signal inputs. The interrupt signal from a peripheral supporting DSTC hardware transfer is connected to the DSTC. The DSTC can start a transfer operation with the interrupt signal from a peripheral as a DMA transfer request signal (DREQ[255:0]). The DSTC cannot start the DMA transfer of the DSTC with an interrupt signal from a peripheral not supporting DSTC hardware transfer. In the case of a peripheral having multiple channels and multiple interrupt sources, there are interrupts supporting and those not supporting the DMA transfer.

The settings of the DREQENB[255:0] determine whether hardware transfer requests from peripherals are valid. The specifications of the product equipped with the DSTC determine which bit out of 256 bits supports the interrupt signal of which peripheral. For details, see Chapter: Interrupts.

There are two types of peripheral functions using DMA transfer of DSTC; one is the interrupt signal and DMA transfer request signal is combined for sharing use,(This is abbreviated “Combined type”.) and another is handling them separately.(This is abbreviated “Separated type”.) The setting value of the DREQENB[255:0] register and the SE2, SEL3 in this figure switch the operations as follows.

■ Case of the Combined type:

When DREQENB[n]=0;

Interrupt signal from peripheral is inputted to the NVIC, notify interrupt.
Interrupt signal from peripheral is ignored by the DSTC.
HWINT[n] signals from DSTC are not inputted to NVIC. HWINT[n] is the output signal for interrupt from the DSTC to the CPU, and is used to for notification of a HW transfer completion of the DSTC.

When DREQENB[n]=1;

Interrupt signal from peripheral is not inputted to the NVIC.
Interrupt signal from peripheral is inputted to the DSTC, and the DSTC start the DMA transfer by this signals.
HWINT[n] signals from DSTC are inputted to NVIC, instead of interrupt signal from peripheral.

In the case of this type, the input port of NVIC is shared to use the interrupt from peripheral and HWINT[n] interrupt of the transfer completion from the DSTC. With this configuration, in the process of the NVIC, an interrupt from a peripheral, and a transfer completion interrupt from the DSTC jump to the same interrupt vector. Therefore, use the DREQENB[n] register to choose the interrupt to be processed.

■ Case of the Separated type:

When DREQENB[n]=0;

Interrupt signal from peripheral is inputted to the NVIC, notify interrupt.
Interrupt signal from peripheral is not inputted to the DSTC.
Transfer request signal from peripheral is not inputted to the NVIC
Transfer request signal from peripheral is ignored by the DSTC
HWINT[n] from the DSTC is inputted to the NVIC. (not asserted).

When DREQENB[n]=1;

Interrupt signal from peripheral is inputted to the NVIC, notify interrupt.
Interrupt signal from peripheral is not inputted to the DSTC.
Transfer request signal from peripheral is not inputted to the NVIC
Transfer request signal from peripheral is inputted to the DSTC, start the transfer.
HWINT[n] from the DSTC is inputted to the NVIC, notify transfer completion.

In the case of this type, the input port of NVIC is separated the interrupt from peripheral and HWINT[n] interrupt of the transfer completion from the DSTC. In the process of the NVIC, an interrupt from a peripheral, and a transfer completion interrupt from the DSTC does not jump to the same interrupt vector.

For details of peripheral types, refer to the list of interrupts and the list of interrupt signals input to DSTC in the Interrupts chapter.

Connection to Hardware Transfer Request Clear Signal

Among peripherals supporting the hardware transfer, there are some for which a transfer request signal (interrupt signal) has to be cleared after a transfer has ended. Though the clearing process is not mentioned in Figure 2-2, if the interrupt request signal of such peripheral is enabled by its corresponding DREQENB[n] register, the transfer request signal is cleared by the DSTC.

Connection to Hardware Transfer Stop Request Signal

The Multi-Function Serial Unit (to be called MFS later in this document) output signal for DMA transfer stop request. Though it is not mentioned in Figure 2-2, if the transfer stop request signal from these has been asserted, the transfer request signal is masked. Therefore, the DSTC does not perform the DMA transfer in the state of waiting for the DMA request signal, and no error response is generated from the DSTC.

Conditions that are asserted by MFS's transfer stop request signal show below.

− If received interrupts are enabled (SCR:RIE=1), a received interrupt occurs (SSR:PE bit, FRE bit, or ORE bit is set to 1).

− If chip select error interrupt are enabled (SACSR:CSEIE=1), a chip select error interrupt occurs (SACSR:CSE bit is set to 1).

Separately, the transfer stop request signal from MFS is sent to the CPU via NVIC as an interrupt. Terminate this current DMA transfer of the DSTC by the CPU with this interrupt. For details, see Chapter: Interrupts.

Interrupt Signal from DSTC

The transfer end interrupt for a transfer started by a software start is sent to the NVIC by the SWINT. The error interrupt generated due to the occurrence of a transfer error is sent to the NVIC by the ERINT.

3. Functions and Operations of DSTC

This section explains operations of the DSTC.

3.1. Settings of DES
3.2. Control Functions of DSTC
3.3. Operation Flows of DSTC

3.1 Settings of DES

This section explains setting details of the DES and operations of the DSTC.

3.1.1 Specifying Transfer Data Size

TW, IRM, IIN, ORM

The DSTC transfers data of the data width specified in TW in DES0 in a single transfer. There is a transfer number counter in the DSTC. The counter has a dual loop configuration consisting of an outer loop counter and an inner loop counter. ORM (outer loop remain) in DES1 indicates the remaining number of transfers of the outer loop counter, and IRM (inner loop remain) in the DES1 Register the remaining number of transfers of the inner loop counter. IIN (inner loop initial) in DES1 specifies the initial value of the inner loop counter.

At the start of a transfer, specify the transfer data width and different numbers of transfers in TW, IRM, IIN and ORM. For one DES, the DSTC transfers data of the total of TW×IIN×ORM.

Table 3-1 shows the method of specifying the transfer data width, the number of transfers and the transfer mode.

Table 3-1 Specifying Transfer Mode, Transfer Size and Number of Transfers

Area name	Name	Details
DES0	MODE	MODE selects a transfer mode. 0: Selects mode 0. 1: Selects mode 1.
	TW[1:0]	TW specifies the data width in a single transfer. 00: 8 bits (byte) 01: 16 bits (halfword) 10: 32 bits (word)
DES1	IIN	Specifies the initial value of the inner loop counter in the transfer number counter.
	IRM	Specifies the remain value of the inner loop counter in the transfer number counter.
	ORM	Specifies the remain value of the outer loop counter in the transfer number counter.

MODE

If 0 is written to MODE, the DSTC executes a transfer in mode 0. In this mode, one Start Trigger makes the DSTC transfer data of the bit width specified in TW for IIN times. After having executed transfers for the number specified in IIN, the DSTC executes the Chain Start or waits for the next Start Trigger. If Start Triggers of the amount specified in ORM are sent to the DSTC, the DSTC ends the transfer of the number (IIN×ORM) specified in the DES.

If 1 is written to MODE, the DSTC executes a transfer in mode 1. In this mode, one Start Trigger makes the DSTC transfer data of the bit width specified in TW once. After having executed one transfer, the DSTC executes the Chain Start or waits for the next Start Trigger. If Start Triggers of the amount equivalent to the result of IIN×ORM are sent to the DSTC, the DSTC ends IIN×ORM times of transfer specified in the DES.

In the SW transfer, both mode 0 and mode 1 can be used. In the HW transfer, select either mode 0 or mode 1 depending on the type of peripherals that generates a Start Trigger. To make the DSTC execute a transfer at a Start Trigger from a peripheral that has to have to a handshake with the DSTC at every data transfer, use mode 1. In the case of data transfer with the MFS, ADC and USB, since the DSTC transfers data to the FIFO in a peripheral, use mode 1. In the case of a Start Trigger from a peripheral that notifies the DSTC of the transfer start timing, such as the timer, the external interrupt block, etc., both mode 0 and mode 1 can be used.

In mode 0, specifies the settings of ORM and IIN. A value of 1 to 65536 inclusive can be specified in both ORM and IIN. The setting of IRM does not need to be specified. In the DSTC, the setting of IIN is copied to IRM.

In mode 1, specifies the settings of ORM, IIN and IRM. A value of 1 to 65536 inclusive can be specified in ORM. A value of 1 to 256 inclusive can be specified in IIN. Specify the same value in IRM and IIN.

3.1.2 Setting Transfer Addresses

SA, DA, SAC[2:0], DAC[2:0]

Set the start address of the transfer source area (SA) in DES2 and the start address of the transfer destination area (DA) in DES3. Align each transfer address to a specified data width (TW). The DSTC cannot execute an unaligned transfer.

Specify the transfer address update methods during a transfer in the SAC[2:0] bits and DAC[2:0] bits in DES0. The transfer address update method for SA and that for DA can be specified separately. Table 3-2 shows the methods of specifying transfer addresses.

Table 3-2 Specifying Transfer Addresses

Area name	bit	Details
DES0	SAC[2:0] DAC[2:0]	Select the respective update methods of transfer addresses SA and DA during a transfer. 000: The address is increased by TW×1 at every transfer without InnerReload. 001: The address is increased by TW×1 at every transfer with InnerReload. 010: The address is increased by TW×2 at every transfer without InnerReload. 011: The address is increased by TW×2 at every transfer with InnerReload. 100: The address is increased by TW×4 at every transfer without InnerReload. 101: The transfer address remains unchanged during a transfer. 110: The address is decreased by TW×1 at every transfer without InnerReload. 111: The address is decreased by TW×1 at every transfer with InnerReload.
DES2	SA[31:0]	Specify the start address of the transfer source area.
DES3	DA[31:0]	Specify the start address of the transfer destination area.

Updating Transfer Number Counter and Transfer Address

Figure 3-1 is an example showing how the transfer number counter and the transfer addresses change when the following settings are used: the number of outer loop transfers is 3, the number of inner loop transfers is 4 and the transfer address is increased. The horizontal axis of this figure is a time scale, indicating transfer progress.

The upper part of this figure shows the behavior of the transfer number counter. The inner loop counter remain value (IRM) downcounts at every transfer. The IRM reloads the inner loop counter initial value (IIN), when the transfers of the number specified in IIN have been executed. The outer loop counter remain value (ORM) downcounts at the timing when the inner loop counter reloads. The DSTC counts the number of transfers using this dual loop counter.

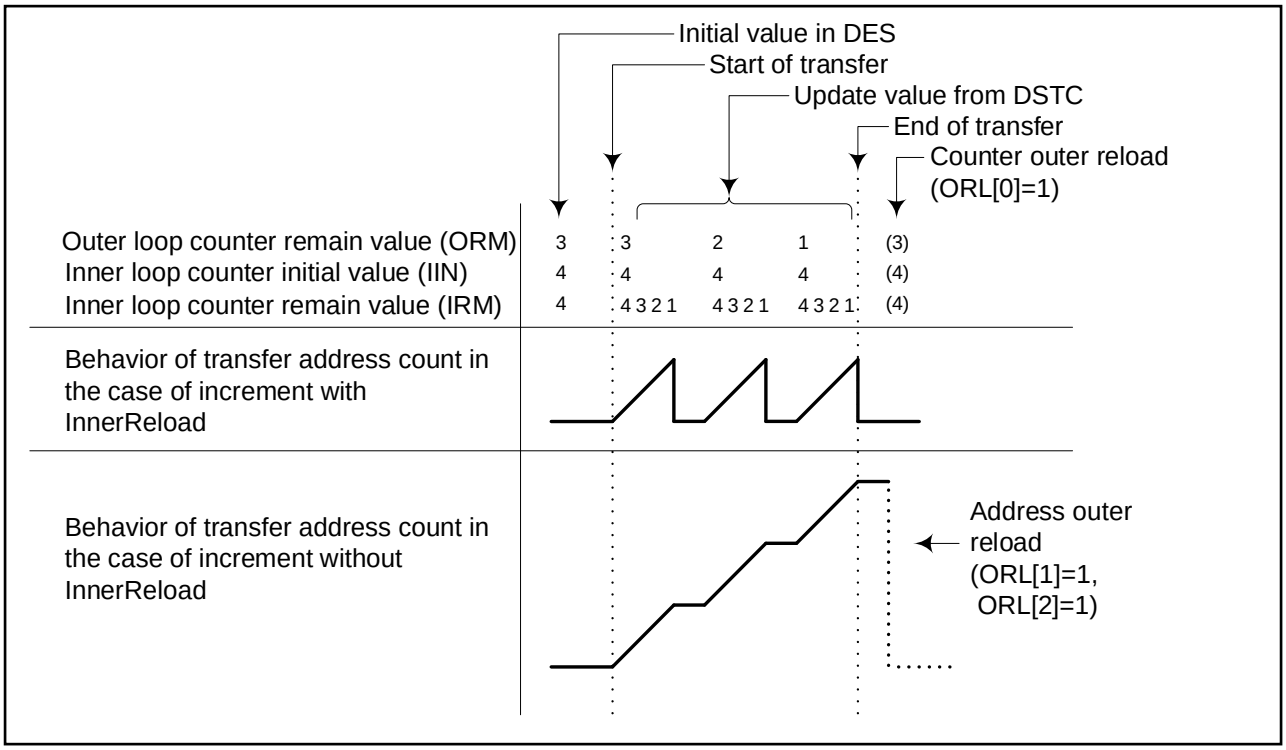
During a transfer, when the IRM reloads, it can be decided that whether the operation of returning a transfer address (SA/DA) to its initial value is executed. (InnerReload)

The middle part of this figure shows the update behavior of the transfer address for increment with InnerReload. As shown in this figure, if transfer address increment and InnerReload are selected, after a transfer starts, the transfer address increases, and at the timing of IRM reload, is reset to the value at the start of the transfer.

The lower part of this figure shows the update behavior of the transfer address for increment without InnerReload. The increment update of the transfer address continues for the timing of reloading the IRM.

Figure 3-1 illustrates also the operation of OuterReload. For its details, see the section on OuterReload.

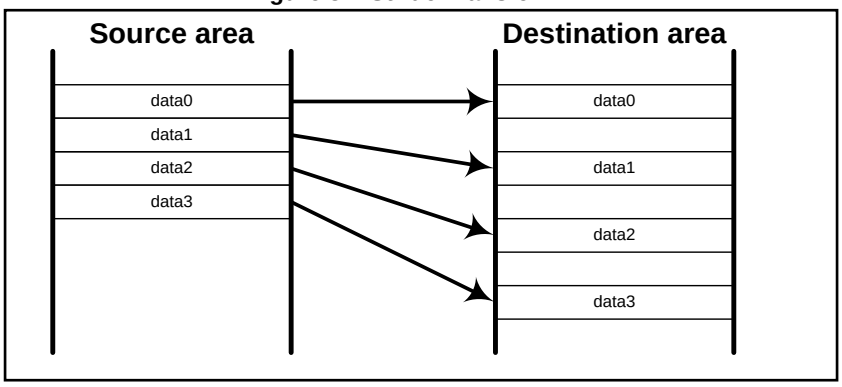
Figure 3-1 Operations of Transfer Number Counter and Transfer Address



Stride Transfer

If an increase of TW×2 is specified in the SAC[2:0] bits and an increase of TW×4 in the DAC[2:0] bits, a stride transfer is executed, in which at every transfer, the transfer address increases by TW×2 and by TW ×4 in turn. Figure 3-2 shows an example of executing a transfer with SAC = 000 (increase by TW×1) and DAC = 010 (increase by TW×2). Using the stride transfer and the chain transfer together facilitates memory data rearrangement. For its details, see 4.3 Transfer Operation Example 3.

Figure 3-2 Stride Transfer



3.1.3 Specifying OuterReload

ORL[2:0]

Table 3-3 shows the method of specifying the transfer number counter and transfer address for OuterReload.

As shown in Figure 3-1, after IIN×ORM times of transfer have ended, for the next transfer, the transfer number counter (ORM/IRM/IIN) of DES1, the transfer source address (SA) of DES2 and the transfer destination address (DA) of DES3 can be reset (OuterReload) to their respective values at the start of the transfer.

Before starting a transfer, set DES4 to the same value as DES1, DES5 as DES2, and DES6 as DES3, respectively. After IIN×ORM times of transfer have ended, the values of DES4, DES5 and DES6 are copied to DES1, DES2 and DES3 respectively, and are reset to the values before the start of the transfer. In the case of using the same details in the next transfer, using OuterReload can eliminate the need of rebuilding the DES via the CPU.

Table 3-3 Specifying OuterReload

Area name	bit	Details
DES0	ORL[0]	Selects whether to execute OuterReload for the transfer number counter (DES1). 0: OuterReload for DES1 is not to be executed. DES4 area is not required. 1: OuterReload for DES1 is to be executed. DES4 area is required.
	ORL[1]	Selects whether to execute OuterReload for the transfer source address (DES2). 0: OuterReload for DES2 is not to be executed. DES5 area is not required. 1: OuterReload for DES2 is to be executed. DES5 area is required.
	ORL[2]	Selects whether to execute OuterReload for the transfer destination address (DES3). 0: OuterReload for DES3 is not to be executed. DES6 area is not required. 1: OuterReload for DES3 is to be executed. DES6 area is required.

DES Size at Using OuterReload

DES0 to DES3 are areas always required. DES4 to DES6 are areas required only when OuterReload is to be executed. The DES size and the addresses of DES4 to DES6 are defined according to the value of the ORL[2:0] bits as shown in Table 3-4. The respective relative addresses from the DESP of DES4 to DES6 vary according to the value of ORL[2:0]. An area not required is considered nonexistent.

Table 3-4 DES Size and Storage Positions of DES4 to DES7 in OuterReload

ORL[2:0]	DES-SIZE (word)	DES4-address	DES5-address	DES6-address
000	4	No DES4	No DES5	No DES6
001	5	DESP+0x10	No DES5	No DES6
010	5	No DES4	DESP+0x10	No DES6
100	5	No DES4	No DES5	DESP+0x10
011	6	DESP+0x10	DESP+0x14	No DES6
101	6	DESP+0x10	No DES5	DESP+0x14
110	6	No DES4	DESP+0x10	DESP+0x14
111	7	DESP+0x10	DESP+0x14	DESP+0x18

DES Values after Transfer End

The DES area can be saved when the OuterReload function is not used. If InnerReload is enabled for the transfer address, OuterReload does not need to be enabled for the transfer address because the values at the start of the transfer are stored in the DES. If neither OuterReload nor InnerReload is enabled, after a transfer has ended, depending on the settings, values stored in the DES may be different from those stored at the start of the transfer. In this situation, since the transfer cannot be started with the same DES values, rebuild the DES via the CPU. The following explains how the values of the DES area are updated after a transfer has ended.

The values of DES0 after the end of the transfer are the same as those at the start of the transfer, except those of the DV[1:0] bits and ST[1:0] bits. In addition, the values of DES4, DES5 and DES6 after the end of the transfer are the same as those at the start of the transfer.

The values of DES1 (transfer number counter) after the end of the transfer are updated according to the values of MODE and ORL[0] at the start of the transfer as shown in Table 3-5. "X" in Table 3-5 indicates that value has no effect on operation.

The value of IIN after the end of the transfer is the same as that at the start of the transfer. In the case of ORL[0] = 0, the value of ORM is updated to 0x0001 and the value of IRM to "0x01" when a transfer ends regardless of their values at the start of the transfer. However, if at the start of the transfer, the value of ORM is not "0x0001" or the value of IRM is not 0x01, the value of ORM or the value of IRM is different from what it was at the start of the transfer. Therefore, before restarting the transfer, rebuild the DES that makes the DSTC rewrite ORM and IRM with necessary values via the CPU. Table 3-5 summarizes conditions under which rebuilding DES1 becomes necessary.

Table 3-5 Values of DES1 at Transfer End and Necessity of Rebuilding DES1

Values of DES0/DES1 at transfer start					Values of DES1 after transfer end			Necessity of rebuilding DES1
MODE	ORL[0]	ORM	IIN	IRM	ORM	IIN	IRM	
0	0	0x0001	X	-	0x0001	Value kept	-	Unnecessary
		Other than 0x0001	X	-				Necessary
1	0	0x0001	0x01	Same as IIN			0x01	Unnecessary
		X	Other than 0x01	Same as IIN				Necessary
		Other than 0x0001	X	Same as IIN				Necessary
0	1	X	X	-	Values of DES4 are copied.			Unnecessary
1	1			Same as IIN				Unnecessary

The values of DES2 (transfer source address) after the end of the transfer are updated according to the values of MODE, SAC[2:0] and ORL[1] of DES0 at the start of the transfer as shown in Table 3-6. "X" in Table 3-6 indicates that value has no effect on operation. If the start value of DES2 is different from the end value, rebuild DES2 before restart to transfer.

Table 3-6 Values of DES2 after Transfer End

Values of DES0 at transfer start			Values of DES2 after transfer end	Necessity of rebuilding DES2
MODE	SAC[2:0]	ORL[1]		
0	xx0	0	Transfer source address at final outer loop start	Necessary
0	xx1	0	Values at transfer start	Unnecessary
1	xx0	0	Transfer source address of final transfer	Necessary
1	xx1	0	Values at transfer start	Unnecessary
X	X	1	Values of DES5 are copied.	Unnecessary

The values of DES3 (transfer destination address) after the end of the transfer are updated according to the values of MODE, DAC[2:0] and ORL[2] of DES0 at the start of the transfer as shown in Table 3-7. "X" in Table 3-7 indicates that value has no effect on operation. If the start value of DES3 is different from the end value, rebuild DES3 before restart to transfer.

Table 3-7 Values of DES2 after Transfer End

Values of DES0 at transfer start			Values of DES3 after transfer end	Necessity of rebuilding DES3
MODE	DAC[2:0]	ORL[2]		
0	xx0	0	Transfer destination address at final outer loop start	Necessary
0	xx1	0	Values at transfer start	Unnecessary
1	xx0	0	Transfer destination address of final transfer	Necessary
1	xx1	0	Values at transfer start	Unnecessary
X	X	1	Values of DES6 are copied.	Unnecessary

The setting is that DES0.DV[1]=1 and DES1,2,3 is need to rebuild (DES1,2,3 is not returned to the start value), caused to notify a DES open error from the DSTC. For details, see 3.2.8 MONERS Register.

3.1.4 Setting Chain Start and Transfer End Interrupt Notification

CHRS[5:0], CHLK

The DSTC executes transfers for the number of times specified in each DES (IIN times if MODE = 0, 1 time if MODE = 1) after receiving a Start Trigger. After executing transfers, the DSTC the next process according to the value of CHRS[5:0] in DES0. Table 3-8 shows the method of setting the Chain Start and the transfer end interrupt notification.

Table 3-8 Details of CHRS[5:0]

Area name	Name	Details
DES0	CHRS[5:4]	These bits select how the DSTC operates after the transfer number counter remain value becomes (ORM == 1) && (IRM == 1). 00: No interrupt flag is set. There is no Chain Start. The DSTC ends the transfer. 01: An interrupt flag is set. There is no Chain Start. The DSTC ends the transfer. 10: No interrupt flag is set. The DSTC executes a Chain Start on the next DES. 11: Setting prohibited (A DES open error occurs.)
	CHRS[3:2]	These bits select how the DSTC operates after the transfer number counter remain value becomes (ORM! = 1) && (IRM == 1). 00: No interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 01: An interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 10: No interrupt flag is set. The DSTC executes a Chain Start on the next DES. 11: No interrupt flag is set. The DSTC executes a Chain Start again on the current DES.
	CHRS[1:0]	If MODE is "1", these bits select how the DSTC operates after the transfer number counter remain value becomes (IRM! = 1). 00: No interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 01: An interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 10: No interrupt flag is set. The DSTC executes a Chain Start on the next DES. 11: No interrupt flag is set. The DSTC executes a Chain Start again on the current DES. If MODE is "0", the above settings are meaningless. Write "00" to CHRS[1:0] if MODE is "0". (Writing a value other than "00" to CHRS[1:0] if MODE is "0" causes a DES open error.)
	CHLK	This bit selects the next transfer started by the Chain Start whether to execute immediately after the current transfer (Chain Lock) or to enable other transfers to be executed before the next transfer started by the Chain Start. 0: After the current transfer, other transfers can be executed before the Chain Start transfer. 1: The Chain Start transfer is executed immediately after the current transfer.

If the next process is the Chain Start on the next DES, the DSTC starts transferring data according to the next DES. If the next process is the Chain Start again on the current DES, the DSTC starts transferring data according to the current DES again. If the next process does not involve the Chain Start, the DSTC ends the transfer (or waits for the next Start Trigger). The status of the transfer number counter determines which of CHRS[5:4], CHRS[3:2] and CHRS[1:0] the DSTC follows when executing the next process after the current DES.

In the case of not executing the Chain Start, after an interrupt flag is set, the DSTC can notify the CPU of the fact that the DSTC has ended the transfer (or is waiting for the next Start Trigger). In the case of SW Transfer or Chain Start Transfer from SW Transfer, the DSTC set the SWST bit to the SWTR register to "1". In the case of HW Transfer or Chain Start Transfer from HW Transfer, the DSTC set the HWINT[n] register to "1".

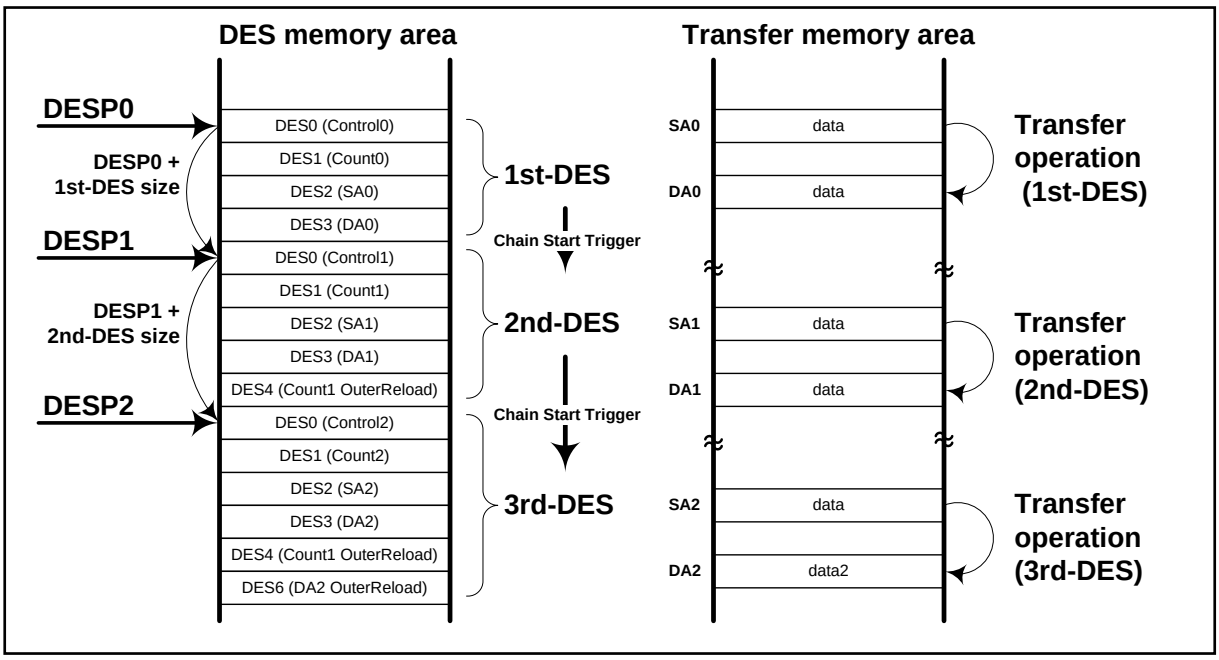
Operations of Chain Start

Using the Chain Start enables making a Start Trigger for different transfers set in multiple DES.

Figure 3-3 illustrates how the DES reference and transfer operation are executed when the DSTC executes a Chain Start on the next DES. 1st-DES is located at the position of DESP0. The size of 1st-DES is defined by the value of ORL[2:0] of 1st-DES. If there is a Chain Start Trigger in the DES after 1st-DES, the DSTC computes DESP1 of the succeeding 2nd-DES from the DESP0 and the size of 1st-DES. The DSTC starts a transfer specified in 2nd-DES after referring to details of 2nd-DES. Therefore, the succeeding 2nd-DES to be started by the Chain Start must be located next to 1st-DES started first.

Specifying the Chain Start in 2nd-DES can start the transfer of 3rd-DES. The DSTC can keep executing the Chain Start transfer until there is no more Chain Start Trigger within the maximum number of DES that can be built.

Figure 3-3 DES Reference in Chain Start Transfer



Chain Lock Function

The DSTC may execute other transfer first according to the transfer priority order if there is a transfer start request of other source at the timing of a Chain Start. If that occurs, the DSTC executes a specified Chain start transfer after executing other transfer. The Chain Lock Function ensures that in the situation described above, transfers started by a Chain Start are executed successively and that no other transfer request interrupts the transfers, regardless of the transfer priority order. The Chain Lock Function can be enabled by writing 1 to CHLK in a DES.

3.1.5 Other DES Settings

DV[1:0]

The DSTC refers to the DES area and updates it while executing a transfer operation. If the CPU updates a DES area that the DSTC is using, the DSTC may executes an illegal transfer operation, which is not set in a program. To prevent any illegal transfer operation, the DES area uses a mutually exclusive memory management system for the CPU and the DSTC. In DES0, there are the DV[1:0] bits (Descriptor Valid), indicating that the DES write update right ownership belongs to the CPU or the DSTC. Table 3-9 shows details of the DV and the related operations of the DSTC.

Table 3-9 Details of DV

Area name	Name	Details
DES0	DV[1:0]	DV specifies which of the CPU and the DSTC the ownership of the DES belongs to. DV specifies whether a transfer is executed after the DES open process DV specifies whether the DES close process is to be executed after transfer ended. 00: The owner is the CPU. No transfer is executed. No DES close process is executed. (If the DSTC read this value, the DSTC notifies a DES open error.) 01: The owner is the DSTC. A transfer is executed. The DES close process is executed. 10: The owner is the DSTC. No transfer is executed. The DES close process is executed. 11: The owner is the DSTC. A transfer is executed. No DES close process is executed.

That DV is 00 indicates that the CPU has the ownership of the DES area. That DV is 01, 10 or 11 indicates that the DSTC has the ownership of the DES area. After setting the initial values of a DES, the CPU sets DV to 01, 10 or 11 to notify that the ownership of that DES belongs to the DSTC. After receiving a Start Trigger, the DSTC checks the DV value of DES0 to determine subsequent operations. (The DSTC reading a DES upon a Start Trigger is called DES open process.)

In the case of DV set to 01, the DSTC executes transfers for the number of times (ORM×IIN) specified. While the transfers are in progress, the value of DV remains 01 and the DSTC keeps the ownership of the DES. When all specified times of transfer end, the DSTV updates the value of DV to 00 and the DSTC returns the ownership of the DES to the CPU. (The process that the DSTC updates the value of DV to 00 and the DSTC returns the ownership of the DES to the CPU is called DES close process.)

In the case of DV set to 11, the DSTC executes transfers for the number of times (ORM×IIN) specified. While the transfers are in progress, the value of DV remains 11 and the DSTC keeps the ownership of the DES. Even after all specified times of transfer have ended, the DSTC does not execute the DES close process and keeps having the ownership of the DES.

In the case of DV set to 10, the DSTC does not execute a transfer, but executes only the DES close process.

In the case of DV set to 00, the DSTC recognizes that the DES area is being updated by the CPU. It does not execute any transfer or update the DES. The DSTC notifies the CPU of a DES open error.

The program determines the value of DV as explained below based on the way of using a DES.

If details of a transfer defined in a DES are subject to change, and the CPU has to update details of the DES at every transfer, setting DV to 01 grants the ownership of the DES to the DSTC. In this situation, after checking that the value of DV is 00 and that the ownership of the DES has been returned to the CPU, the CPU can safely update the DES. The CPU transfers the ownership of the DES to the DSTC again after completing the initialization of the DES.

If details of a transfer defined in a DES are fixed, and the CPU has to reuse details of the DES, setting DV to"11" grants the ownership of the DES to the DSTC. In this situation, since the DES close process is not executed after the transfer, the process of CPU transferring the ownership of the DES again can be omitted. After specified times (ORM×IIN) of transfer have ended, if a new Start Trigger is issued, transfers of the same details as the previous ones are started.

With DV set to 11, since the DSTC keeps having the ownership of the DES and does not return it to the CPU, the transfer becomes an infinite loop process. To escape from the infinite loop of transfer, set DV to "10" to return to CPU the ownership of the DES that has been transferred to the DSTC when DV was set to "11".

The CPU can update the DES if DV is set to 00. If DV is set to 01, 10 or 11, the CPU cannot update details of the DES area other than DV while the DSTC is executing the transfer according to that DES.

If DV[1] is set to 1, the DSTC reuses the values of the DES after executing ORM×IIN times of transfer. Therefore, specific restrictions on reloading the transfer counter and transfer address are added. If DV[1] is set to 1 and the settings of the DES make the values of DES1, DES2 and DES3 not return to their respective values, the DSTC notifies the CPU a DES open error. For details, see “3.2.8 MONERS Register”.

ST[1:0]

ST(Status)[1:0] of DES0 is for the DSTC to notify the CPU of the transfer end status. Table 3-10 shows data the DSTC writes to ST in a DES close process.

Table 3-10 Content of ST Notification

Area name	Name	Details
DES0	ST[1:0]	After the transfer specified in a DES has ended, in a DES close process, the DSTC writes the end status value to ST. 00: The transfer has ended normally. 01: The transfer has ended abnormally because an error occurred at a transfer source access. 10: The transfer has ended abnormally because an error occurred at a transfer destination access. 11: The transfer has ended abnormally because a transfer compulsory stopped by standby transition command is issued from CPU.

If a transfer ends abnormally due to a DES access error or a DES open error, the DSTC does not execute a DES close process and does not write data to ST because the DSTC cannot access the DES area. Such error notifications are executed according to the MONERS Register of the DSTC but not ST of the DES. For details, see 3.2.8 MONERS Register.

PCHK[3:0]

PCHK[3:0] (Parity Check) sets the parity (to be called equation below) of the DES0 area.

PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4])

If data in the DES area is corrupted by an event like a runaway of the CPU, the DSTC may start an unintended transfer. The parity check function is installed in DES0 to prevent the above from occurring. The CPU sets the parity of DES0 to PCHK when building the DES for the first time. The DSTC checks the consistency between the values of PCHK and those of DES0 in the DES open process. If a parity error occurs, the DSTC notifies the CPU of a DES open error and does not execute the transfer. For details, see 3.2.8 MONERS Register.

ACK[1:0]

ACK[1:0] (Acknowledge) sets the value for adjusting the timing of DSTC outputting the DMA transfer request acknowledge signal to a peripheral device when the HW transfer is used.

If the HW transfer is used, set ACK to 01 for a DES to be directly started by the HW Start from a peripheral device. For other DES (the DES started by the Chain Start from the HW transfer, the DES used in the SW transfer, and the DES started by the Chain Start from the SW transfer), set ACK to 00.

DMSET

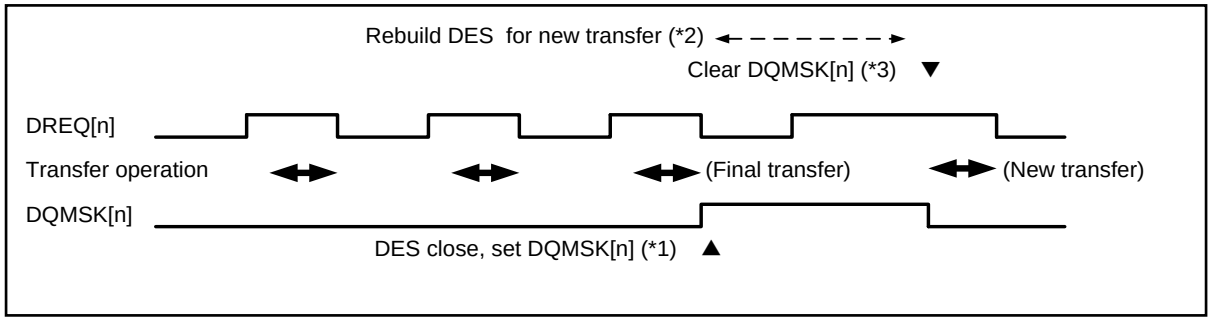
The DMSET (DMA request mask set) provides a function that sets the DQMSK[n] Register and masks a DMA transfer request signal from a peripheral during the period between the time at which the DSTC finishes the DES close process and the time at which the CPU finishes rebuilding the DES. For details, see " 3.2.4 Control of HW Transfer".

If the HW transfer is used, the DMA transfer request signal (DREQ) from a peripheral is negated by the DMA transfer request acknowledge signal (DACK) after the transfer has ended. However, depending on peripherals, the DREQ is asserted at the following transfer request regardless of the status of the DSTC. If the DREQ is asserted during the period between the time at which the DSTC finishes the DES close process and the time at which the start of the next transfer is ready (rebuilding the DES), the DSTC notifies the CPU of a DES open error because the start of the next transfer is not ready. In this situation, setting the DMSET bit in Descriptor 0 to 1 can prevent the DSTC from notifying the CPU of a DES open error, and can suppress the start of an HW Start transfer until the completion of rebuilding a DES.

Figure 3-4 shows an operation example.

If DMSET = 1 in the DES0 that is processed by the HW transfer directly from a peripheral, or that is started by the Chain Start from HW transfer, when the DES close process is executed, the bit corresponding to HW channel in the DQMSK[n] register is set to 1. (*1 in Figure 3-4) After that channel bit has been set to 1, the DSTC does not recognize the DREQ[n] signal, and does not notify the CPU of the DES open error. After the CPU has rebuilt the DES (*2 in Figure 3-4) and the next transfer is ready, the CPU clears the channel bit in the DQMSK[n] Register. (*3 in Figure 3-4) After the CPU has cleared the channel bit in the DQMSK[n] Register, the DSTC recognizes the DREQ[n] signal. A new transfer is started according to the DES rebuilt by the CPU.

Figure 3-4 Using DMSET to Suppress Transfer Start



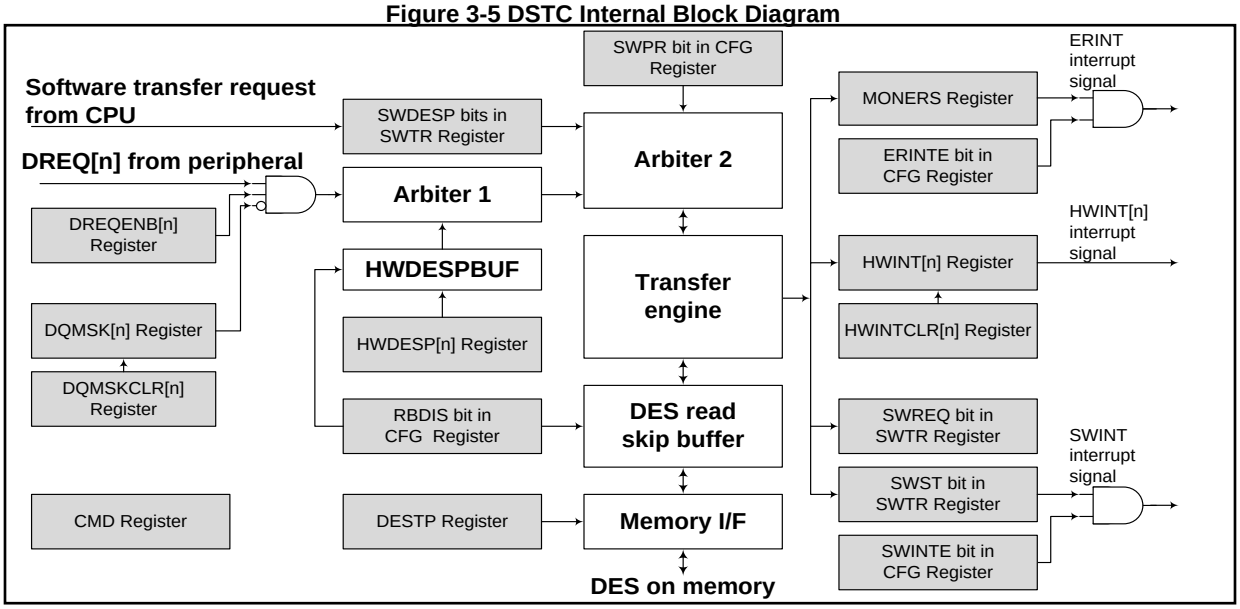
3.2 Control Functions of DSTC

This section explains the control functions of the DSTC.

3.2.1 DSTC internal Block Diagram

Figure 3-5 illustrates the connection between control blocks and control registers (shaded rectangles) in the DSTC that can be accessed from the CPU. The CPU starts DSTC transfer and controls end notifications via accesses to control registers. The following sections explain the operations of each block and the function overview of each register shown in the DSTC internal block diagram. For details of register functions, see "5 Registers and Descriptors of DSTC".

Figure 3-5 DSTC Internal Block Diagram



3.2.2 DESTP Register

The DESTP (DES top address) Register is a register specifying the start address of the DES area on the memory. Specify the start address when doing the initial settings. The DSTC refers to the DES located at the address of "DESTP + DESP" and executes a transfer.

3.2.3 Control of SW Transfer

To issue a Start Trigger of the SW Transfer, write the DESP value of the DES to be started to SWDESP (Software DES pointer) in the SWTR (Software trigger) Register. If a Chain Start is executed during the SW Transfer, SWDESP is updated by the DSTC to the value of DESP used in the Chain Start. The value of SWDESP is sent to Arbiter 2 in Figure 3-5 as a transfer request.

The SWREQ (Software request) bit in the SWTR Register is a read-only bit indicating whether the execution of the SW Transfer is pending, or the SW Transfer as well as the Chain Start transfer are being executed. A write access (Start Trigger) to the SWTR Register sets the SWREQ bit to 1. If the SW Transfer ends normally, abnormally, or is waiting for a Start Trigger, SWREQ is reset to 0.

A SW Start trigger can be issued only after the current SW Transfer has ended. If the SWREQ bit is 1, a write access to the SWTR Register is ignored.

The SWST (Software status) bit in the SWTR Register is a read-only bit for sending the SW transfer end notification to the CPU. In the interrupt flag set is specified in the CHRS in the DES of SW Transfer, or in the DES started by the Chain Start from the SW Transfer. If the SW Transfer ends normally, SWST is set to 1. SWST can be cleared to 0 by sending the SWCLR command to the CMD Register.

If the SWST bit has been set to 1, the SWINT interrupt can be enabled by writing 1 to the SWINTE bit in the CFG Register. In the case of (SWINTE==1) & (SWST==1), the SWINT interrupt signal for the NVIC is asserted.

3.2.4 Control of HW Transfer

If a peripheral makes a transfer request (assertion of DREQ[n]), the DSTC starts the HW Transfer. The DSTC controls the HW Transfer on a transfer channel using the following registers whose number corresponds to the number of transfer channels. The CPU does the initial settings of those registers before a peripheral makes a transfer request. In addition, the CPU clears registers according to the progress of a transfer.

DREQENB[n] Register

The DREQENB[n] (DMA request enable) Register determines whether HW channel n is used in the initial settings. Write 1 to the DREQENB[n] Register to use HW channel n. Write 0 to the DREQENB[n] Register to not use HW channel n. If the DREQENB[n] Register is 0, the interrupt signal (DREQ[n]) of a peripheral connected to the DSTC is ignored. The value of the DREQENB[n] Register is not modified by the DSTC. The value of the DREQENB[n] Register determines which of the interrupt signal from a peripheral and HWINT[n] from the DSTC is selected as an interrupt signal connected to the NVIC. For its details, see "2 DSTC Operations Overview and DSTC System Configuration".

DQMSK[n] Register and DQMSKCLR[n] Register

The DQMSK[n] (DMA request mask) Register is a read-only register. This register is 1 indicates that the HW Start request (DREQ[n]) to the DSTC is being suppressed. If one of the following conditions is met, the DSTC sets DQMSK[n] to 1 and suppresses the transfer request of the HW channel corresponding to DQMSK[n].

- A transfer error has occurred at a HW Transfer on HW channel n.
- The CPU has issued a standby transition command to the CMD Register.
- DMSET in the DES for the transfer on HW channel n is 1 and the DSTC has executed a DES close process.

After the CPU has rebuilt the DES and the HW transfer has become ready to start, the suppression of the HW Start transfer request to the DSTC can be released by the CPU. If 1 is written to the DQMSKCLR[n] (DMA request mask clear) Register, the DQMSK[n] Register is cleared to 0 and the succeeding HW transfer request (DREQ[n]) is recognized.

HWDESP[n] Register

The HWDESP[n] (Hardware DES pointer) Register sets the DESP of the DES that the DSTC refers to and executes at a transfer request of HW channel n. Set this register before making an HW transfer request.

If an HW Start trigger is issued, the DSTC starts a transfer referring to the DES of the DESP set in the HWDESP[n] Register. The DSTC stores the DESP value of the HWDESP[n] Register in HWDESPBUF in Figure 3-5 before using it. In a Chain Start, the value stored in HWDESPBUF is updated to the DESP value set after the Chain Start. The value of the HWDESP[n] Register cannot be modified by the DSTC. If HW Start requests of channel n are made successively, the DSTC uses the DESP value stored in HWDESPBUF, but not the DESP value of the HWDESP[n] Register. Therefore, if the values of the HWDESP[n] Register are modified via the CPU, invalidate the value stored in HWDESPBUF. The DESP value of HWDESPBUF can be invalidated by modifying the value of the RBDIS bit in the CFG Register. For its details, see 5.5 CFG Register.

HWINT[n] Register and HWINTCLR[n] Register

The HWINT[n] (Hardware transfer interrupt) Register is a read-only register for sending the HW transfer end notification to the CPU. The interrupt flag set is specified in the DES started by the HW Start, or CHRS in the DES started by the Chain Start after the DES started by the HW Start. If the HW transfer ends normally, HWINT[n] is set to 1. The HWINT[n] Register can be cleared to 0 by writing 1 to the HWINTCLR[n] Register. If the HWINT[n] Register is set to 1, the HW transfer completion interrupt signal from the DSTC (HWINT[n]) for the NVIC is asserted.

3.2.5 Arbitration of Transfer Requests

The DSTC arbitrates start triggers if multiple HW Start requests conflict with an SW Start request, and executes transfers sequentially. The arbitration of start requests are processed by two blocks, Arbiter 1 and Arbiter 2, shown in Figure 3-5. Below are details of arbitration.

Arbiter 1

The HW transfer request is arbitrated by Arbiter 1. If there are conflicting requests, Arbiter 1 uses the rotation method explained below to select a transfer start channel. After a bus reset, the smaller the channel number, the higher the priority is in the selection priority order.

highest priority 0,1,2,3,4,5,6,7,,,,,,254,255 lowest priority

According to this priority order, for instance, if a request from channel 5 and another from channel 6 are made simultaneously, channel 5 is selected. Once a transfer channel is selected, its priority is rotated to the lowest. In the above example, as channel 5 is selected, the priority order is updated to the one below.

highest priority 6,7,8,9,10,11,,,,,,254,255, 0,1,2,3,4,5 lowest priority

According to this priority order, for instance, if a request from channel 5 and another from channel 6 are made simultaneously, channel 6 is selected. The rotation method enables multiple HW transfer requests to be processed equally.

Arbiter 1 refers to HWDESP[n] of channel n selected and notifies Arbiter 2 of the DESP of the DES used. In addition, if the Chain Start is used in the HW transfer, Arbiter 1 notifies Arbiter 2 of the updated DESP. After all Chain transfers have ended and the DSTC has started to wait for the next Start Trigger, Arbiter 1 notifies Arbiter 2 of the transfer request of the channel n selected.

Arbiter 2

Arbiter 2 selects which of the HW transfer request selected by Arbiter 1 and the SW transfer request is to be executed. If there are conflicting transfer requests, the DSTC selects a transfer request according to the probability set in the SWPR (Software transfer priority) bits in the CFG Register and starts the transfer engine. Table 3-11 shows the settings of the SWPR bits in the CFG Register and the probability of the SW transfer acquiring the transfer right.

Table 3-11 Details of CFG:SWPR[2:0]

Area name	Name	Details
CFG	SWPR[2:0]	In the arbitration of Arbiter 2, if the SW transfer request conflicts with the HW transfer request, Arbiter 2 specifies the probability of the SW transfer acquiring the transfer right. 000: Sets the priority of the SW transfer to the highest priority. (If an SW transfer request is made while an HW transfer is in progress, the SW transfer starts after the HW transfer has ended.) 001: Sets the probability of the SW transfer acquiring the transfer right to 1/2. 010: Sets the probability of the SW transfer acquiring the transfer right to 1/3. 011: Sets the probability of the SW transfer acquiring the transfer right to 1/7. 100: Sets the probability of the SW transfer acquiring the transfer right to 1/15. (Initial value) 101: Sets the probability of the SW transfer acquiring the transfer right to 1/31. 110: Sets the probability of the SW transfer acquiring the transfer right to 1/63. 111: Sets the priority of the SW transfer to the lowest priority. (The SW transfer starts only when there is no HW transfer request.)

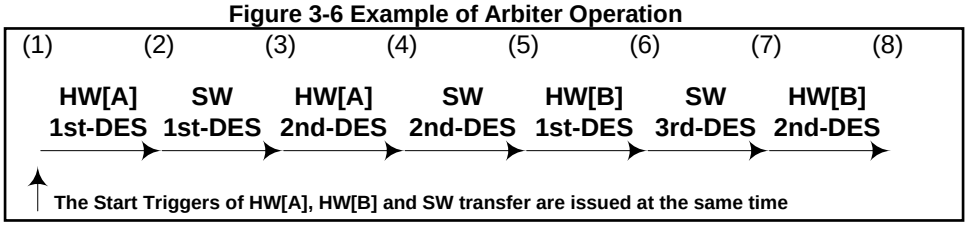
Example of Arbiter Operation

Figure 3-6 shows an operation example. The horizontal axis in the figure is the time axis. The figure illustrates the order of granting the transfer right to different transfer requests during the arbitration by the DSTC. There are three transfer sources: HW channel A transfer, HW channel B transfer and SW transfer. The HW[A] transfer and the HW[B] transfer are connected in a Chain transfer by two DES; SW transfers are connected in a Chain transfer by three DES. The SWPR bits in the CFG Register are set to 001 (probability of SW transfer: 1/2). No Chain lock is specified in any DES.

At timing (1), the respective transfer requests of HW[A] transfer, HW[B] transfer and SW transfer are made simultaneously. Arbiter 1 arbitrates the conflict between HW[A] transfer and HW[B] transfer. Arbiter

1 selects one from HW[A] transfer and HW[B] transfer according to the preceding rotation status. The following description assumes that Arbiter 1 has selected HW[A] transfer. Arbiter 2 arbitrates the conflict between HW[A] transfer and SW transfer. Arbiter 2 selects one from HW[A] transfer and SW transfer according to the preceding rotation status. The following description assumes that Arbiter 2 has selected HW[A] transfer. The transfer engine of the DSTC starts the transfer of 1st-DES of HW[A].

Figure 3-6 Example of Arbiter Operation



At timing (2), the transfer of HW[A] 1st-DES ends. A Chain Start request of HW[A] 2nd-DES is made. (The request of HW[B] is held until there is no more Chain Start of HW[A].) Arbiter 1 requests Arbiter 2 for 2nd-DESP of HW[A]. Arbiter 2 arbitrates the conflict between HW[A] transfer and SW transfer. As the probability for SW transfer is 1/2, and Arbiter 2 has selected HW[A] transfer at timing (1), Arbiter 2 selects SW 1st-DES transfer.

At timing (3), the transfer of SW 1st-DES ends and the Chain Start request of SW 2nd-DES is made. Arbiter 2 arbitrates the conflict between HW[A] 2nd-DES transfer and SW-2ndDES transfer. As the probability for SW transfer is 1/2, and Arbiter 2 has selected SW transfer at timing (2), Arbiter 2 selects HW[A] 2nd-DES transfer.

At timing (4), the transfer of HW[A]-2ndDES ends. Arbiter 1 makes a request for transferring HW[B] 1st-DES to Arbiter 2. Arbiter 2 arbitrates the conflict between HW[B] 1st-DES transfer and SW 2nd-DES transfer, and then selects SW 2nd-DES transfer.

At timings (5), (6) and (7), Arbiter 2 executes the same arbitration operations and selects HW[B] 2nd-DES transfer and SW 3rd-DES transfer.

As explained above, SW transfer may be executed during the Chain transfer of HW transfer, and HW transfer during the Chain transfer of SW transfer. During the Chain transfer of HW transfer, no HW transfer on any other channel is executed. If the Chain lock has been specified in the DES, regardless of the setting of the SWPR bits in the CFG Register, after the transfer of that DES has been executed, the transfers of the DES in the Chain Start are always executed successively.

In the above example, as the probability is set to 1/2 in the SWPR bits in the CFG Register, one SW transfer is executed in every two transfers. Taking account of the number of HW transfer channels of the DSTC to be used simultaneously, the number of Chains in the DES, the transfer data size in each transfer, etc., select an appropriate value for the SWPR bits in the CFG Register. The value of the SWPR bits in the CFG Register can be modified even when the DSTC is executing a transfer. After the value of the SWPR bits in the CFG Register has been modified, it is applied from the next SW Start Trigger.

3.2.6 Read Skip Buffer Function

The transfer engine of the DSTC refers to the transfer information of the DES on the memory while executing a transfer. If all transfers do not end in one Start Trigger, the DSTC writes back to each DES the number of executed transfers of a DES and transfer addresses. If a transfer address is fixed or does not need to be updated, the DSTC skips the write-back process. At the next Start Trigger, the DSTC continues executing transfers according to updated DES information.

Since it takes time for the DSTC to refer to the DES at every Start Trigger, the DSTC has the read skip buffer function as shown in Figure 3-5. The DSTC stores in its internal read skip buffer the transfer information in the DES that the DSTC has read. If the next Start Trigger refers to the same DESP as the current Start Trigger, the DSTC does not refer to the DES on the memory, but uses the values in the read skip buffer to execute the transfer to increase the processing speed.

The read skip buffer function can be enabled and disabled by using the RBDIS bit in the CFG Register. In practice, to increase the transfer speed, enable the read skip buffer function. If "1" is written to the RBDIS bit in the CFG Register to disable the read skip buffer function, always refer to the DES directly on the memory.

In the case of stopping the transfer operation of the DSTC (infinite loop out) by modifying the value of DV in DES0 via the CPU to "10", after modifying the value of DV in DES0, invalidate the DES information stored in the read skip buffer by writing "1" to the RBDIS bit in the CFG Register. As long as the read skip buffer function remains enabled, the DSTC may skip referring to the DES and not be able to recognize any change in the value of DV by the CPU. For its details, see "5.5 CFG Register".

3.2.7 Operation of the Transfer End

If the DSTC ends a transfer normally, according to details of the DES, it executes the Chain Start, sets the interrupt flag (SWTR:SWST or HWINT[n]) and the DES close process.

If a transfer error occurs, a transfer is interrupted immediately. This interruption of a transfer is called error end. In an error end, the DSTC does not execute the Chain Start. In addition, the DSTC does not set the interrupt flag (SWTR:SWST or HWINT[n]), but records details related to the occurrence of the error in the MONERS Register instead. Details of the error determine whether the DSTC executes the DES close process.

In an HW Transfer, if a transfer error occurs, the DSTC sets the DQMSK[n] Register corresponding to the channel on which that transfer error has occurred, and suppresses future HW transfer requests on that channel.

Since details of the DES in which an error has occurred remain the same as they were during the transfer, before starting a new transfer, rebuild the DES area with the CPU.

3.2.8 MONERS Register

If a transfer error occurs, details of that error are recorded in the MONERS Register. Table 3-12 shows details the MONERS Register displays.

Table 3-12 Details of MONERS

Area name	Name	Details
MONERS	EST[2:0]	Indicate details of an error that has occurred. 000: No error has occurred. 001: Source access error 010: Destination access error 011: Transfer compulsory stop error by standby transition command. 100: DES access error 101: DES open error Value other than the above: Undefined
	DER	This bit indicates whether a double error has occurred. 0: Indicates that no double error has occurred. 1: Indicates that a double error has occurred.
	ESTOP	Indicates whether the DSTC is in the error stop state. 0: Indicates that the DSTC is not in the error stop state. 1: Indicates that the DSTC is in the error stop state.
	EHS	Indicates whether the DES that has caused an error has been started by the HW Start or by the SW Start. 0: An error has occurred in a transfer started by the SW Start or by the Chain Start in that SW Start. 1: An error has occurred in a transfer started by the HW Start or by the Chain Start in that HW Start.
MONERS	ECH	Indicates the HW channel number if the DES that has caused an error has been started by the HW Start.
	EDESP	Indicates the DESP of the DES that has caused an error.

Details of a transfer error that has occurred can be checked by referring to the MONERS Register. With the EST[2:0] bits in the MONERS Register indicating that an error has occurred, the ERINT interrupt can be enabled by writing 1 to the ERINTE bit in the CFG Register. If the ERINTE bit is set to 1, the ERINT interrupt signal for the NVIC is asserted. The values of the MONERS Register and the ERINT interrupt can be cleared by issuing an ERCLR command to the CMD Register. Details of errors that may occur are explained below.

DES Access Error

If one of the following events occurs while the DSTC is referring to the DES area of DESTP+DESP, the DSTC ends a transfer in the form of error end (DES access error). The DSTC sets MONERS:EST to 100. At a DES access error, the DSTC does not execute the DES close process.

- The DES area address value calculated overflows (out of the range of 0x00000000-0xFFFFFFFF).
- The DSTC receives a bus error response from the system when accessing a DES area.

DES Open Error

After the DSTC has referred to the area of DESTP+DESP, if the value of DES0 or DES1 meets one of the following conditions, the DSTC regards that as an abnormal DES specified value, and ends a transfer in the form of error end (DES open error). The DSTC sets MONERS:EST to "101". At a DES open error, the DSTC does not execute the DES close process.

- DV[1:0]==00 (No DES ownership)
- PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4]) (DES0 parity error)
- One of the two bits of the reserved area of DES0 is 1. (abnormal specified value)

- TW[1:0]==11 (abnormal specified value)
- CHRS[5:4]==11 (abnormal specified value)
- (CHRS[5]==0) &&(CHRS[3]==0)&&(CHRS[1]==0) &&(CHLK ==1) (abnormal Chain setting)
- (MODE==0) && (CHRS[1:0] != 00) (abnormal setting)
- (MODE==0)&&(ORM==0x0000) && (IIN≥0x2000) (Out of allowed count value range in mode 0)
- (MODE==0)&&(ORM≥0x8000) && (IIN≥0x4000) (Out of allowed count value range in mode 0)
- (MODE==0)&&(ORM≥0x4000) && (IIN≥0x8000) (Out of allowed count value range in mode 0)
- (MODE==0)&&(ORM≥0x2000) &&(IIN==0x0000) (Out of allowed count value range in mode 0)
- (MODE==1)&&(IIN!=0x00)&&(IRM==0x00) (Out of allowed count value range in mode 1)
- (MODE==1)&&(IIN!=0x00)&&(IRM>IIN) (Out of allowed count value range in mode 1)
- (MODE==0)&&(DV[1]==1)&&(ORL[0]==0)&& (ORM != 0x0001) (abnormal counter reload setting)
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0)&& (ORM != 0x0001) (abnormal counter reload setting)
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0) && (IRM != 0x01) (abnormal counter reload setting)
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0)&& (IIN != 0x01) (abnormal counter reload setting)
- (DV[1]==1)&&(SAC[0]==0)&&(ORL[1]==0) (abnormal transfer source address reload setting)
- (DV[1]==1)&&(DAC[0]==0)&&(ORL[2]==0) (abnormal transfer destination address reload setting)

Source Access Error

If one of the following events occurs while the DSTC is accessing the transfer source address area, the DSTC ends a transfer in the form of error end (source access error). The DSTC sets MONERS:EST to 001. At the same time, the DSTC writes 01 to DES0:ST and executes the DES close process.

- The specified transfer source start address value (SA) is unaligned to TW.
- The transfer source address value having undergone increment calculation or decrement calculation overflows.
- The DSTC receives a bus error response from the system.

Destination Access Error

If one of the following events occurs while the DSTC is accessing the transfer destination address area, the DSTC ends a transfer in the form of error end (destination access error). The DSTC sets MONERS:EST to 010. At the same time, the DSTC writes 10 to DES0:ST and executes the DES close process.

- The specified transfer destination start address value (DA) is unaligned to TW.
- The transfer destination address value having undergone increment calculation or decrement calculation overflows.
- The DSTC receives a bus error response from the system.

Transfer Compulsory Stop Error

If the DSTC receives a standby transition command from the CPU while executing a transfer, it ends the transfer (transfer compulsory stop error). The DSTC sets MONERS:EST to 011. At the same time, the DSTC writes 11 to DES0:ST and executes the DES close process.

DER Function and ESTOP Function

If a transfer error occurs, the transfer of the DES that has caused the transfer error is interrupted and ended. After the transfer has been ended, if there is a transfer start request for another DES, the setting of the ESTE (error stop enable) bit in the CFG Register determines whether the DSTC starts the transfer requested in the transfer start request.

In the case of CFG:ESTE = 0, if there is a new transfer request after a transfer error has occurred, the DSTC starts the transfer for that new transfer request. The MONERS Register records error information and keeps it until the register is cleared by the CPU. While the MONERS Register is keeping error information (EST ≠ 000), if a transfer caused by another transfer request ends due to an error, the DSTC

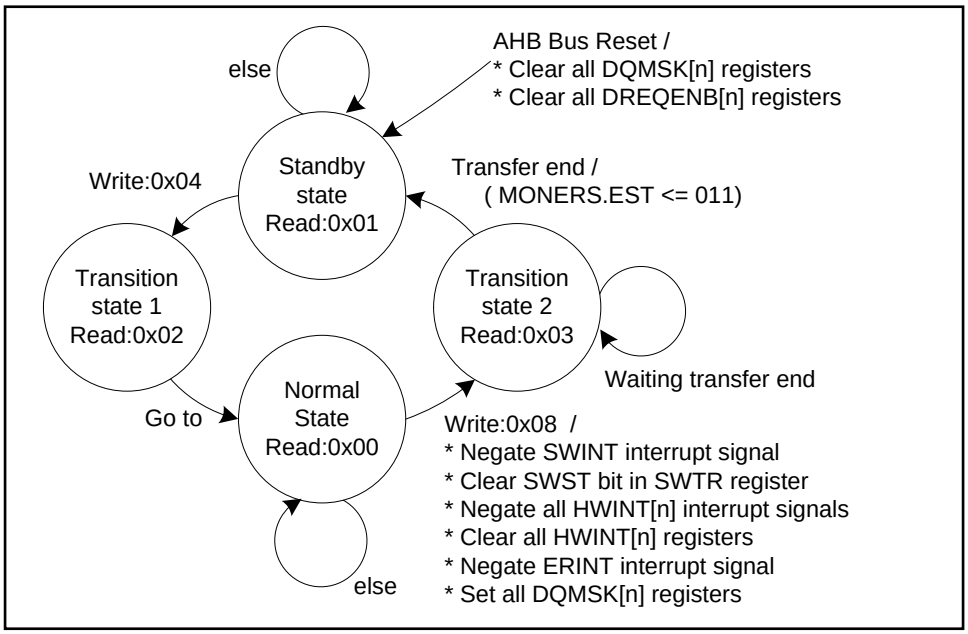
sets the DER (double error) bit to 1. The DER bit is a bit that indicates a double error has occurred. As for the second error, the DSTC notifies the CPU of only its occurrence. The MONERS Register keeps only the information of the first error, but does not keep details of the second error. Moreover, the MONERS Register does not record any error that occurs after the second error.

With CFG:ESTE set to 1, if a transfer error occurs, the DSTC transits to the error stop state. After transiting to the error stop state, the DSTC holds other transfer requests and no longer starts any transfer. That the ESTOP bit in the MONERS Register is set to 1 indicates that the DSTC is in the error stop state. If the CPU issues an ERCLR command to the CMD Register, the DSTC is released from the error stop state and starts transfers according to transfer requests it has been keeping.

3.2.9 Standby Function

To reduce power consumption, the DSTC has a function (standby function) for stopping the internal clocks of the DSTC to make the DSTC stop operating. The state of the DSTC can be switched by the standby transition command and standby release command issued to the CMD Register. Figure 3-7illustrates the operations executed in the issue of the standby transition command and in the state transition of the DSTC.

Figure 3-7 DSTC Standby State Transition Diagram



The DSTC has four states: standby state, transition state 1, normal state and transition state 2. The state of the DSTC can be checked by reading the value of the CMD Register via the CPU.

Upon a bus reset, the initial state of the DSTC is the standby state. If the CPU issues a standby release command (writing 0x04) to the CMD Register, the DSTC transits to the transition state 1 and then to the normal state.

In the normal state, if the CPU issues a standby transition command (writing 0x08) to the CMD Register, the DSTC transits to the transition state 2 to wait for a transfer to end.

In the transition state 2, if the DSTC does not execute any transfer, it immediately transits to the standby state. But, if the DSTC executes a transfer, it transits to the standby state after that transfer has been compulsorily ended.

If a transfer is ended by issuing a standby transition command, the DSTC writes 11 (compulsory end code) to ST of the DES for that transfer to execute the DES close process. In addition, the EST bits in the MONERS Register are set to 011. If the DSTC receives both HW transfer and SW transfer, it executes the DES close processes for both transfers.

In addition, if the CPU issues a standby transition command to the DSTC, the DSTC executes the following processes at the same time.

- The DSTC negates the SWINT interrupt signal and clears SWTR:SWST.
- The DSTC clears all HWINT[n] Registers and negates all HWINT[n] interrupt signals.
- The DSTC negates the ERINT interrupt signal.
- The DSTC sets all DQMSK[n] Registers to suppress the HW transfer request.

Though the issue of a standby transition command negates the ERINT interrupt signal, the values of the MONERS Register remain unchanged. Therefore, if a transfer has been compulsorily stopped by a standby transition command, the information of that transfer can still be checked by reading the MONERS Register. In addition, the error record in the MONERS Register can be cleared only when the DSTC is in the normal state. After a standby release command has made the DSTC return to the normal state, clear the MONERS Register with the ERCLR command.

The initial values of all bits in the DQMSK[n] Register after a bus reset are 0. If a standby transition command is issued, all bits in the DQMSK[n] are set to 1. To start an HW transfer after the DSTC has returned to the normal state, clear the DQMSK[n] to be used for that HW transfer after finishing the setup of a peripheral and rebuilding the DES.

Table 3-13 shows the accessibility of each control register in each state of the DSTC. "O" indicates that register is accessible. "-" indicates that the access to that register is ignored by the DSTC and does not function. "X" indicates that the process result becomes undefined depending on the change in the state of the DSTC. It is prohibited to execute an access marked with "X".

In the standby state, transition state 1 and transition state 2, writing a value to SWTR:SWDESP cannot start a new SW Transfer (the write access to the SWTR Register is ignored).

Table 3-13 Accessibility of Each Control Register in Each State of the DSTC

Register name	Register access	Stand-by State	Normal State	Transition State 1, 2
CMD Register	CMD Register read	O	O	O
	Standby release command (write)	O	-	-
	Standby transition command (write)	-	O	-
	SWCLR / ERCLR / MKCLR Command (write)	-	O	X
HWDESP[n] Register	Read access	-	O	X
	Write access	-	O	X
Other control registers	Read access	O	O	O
	Write access	-	O	X

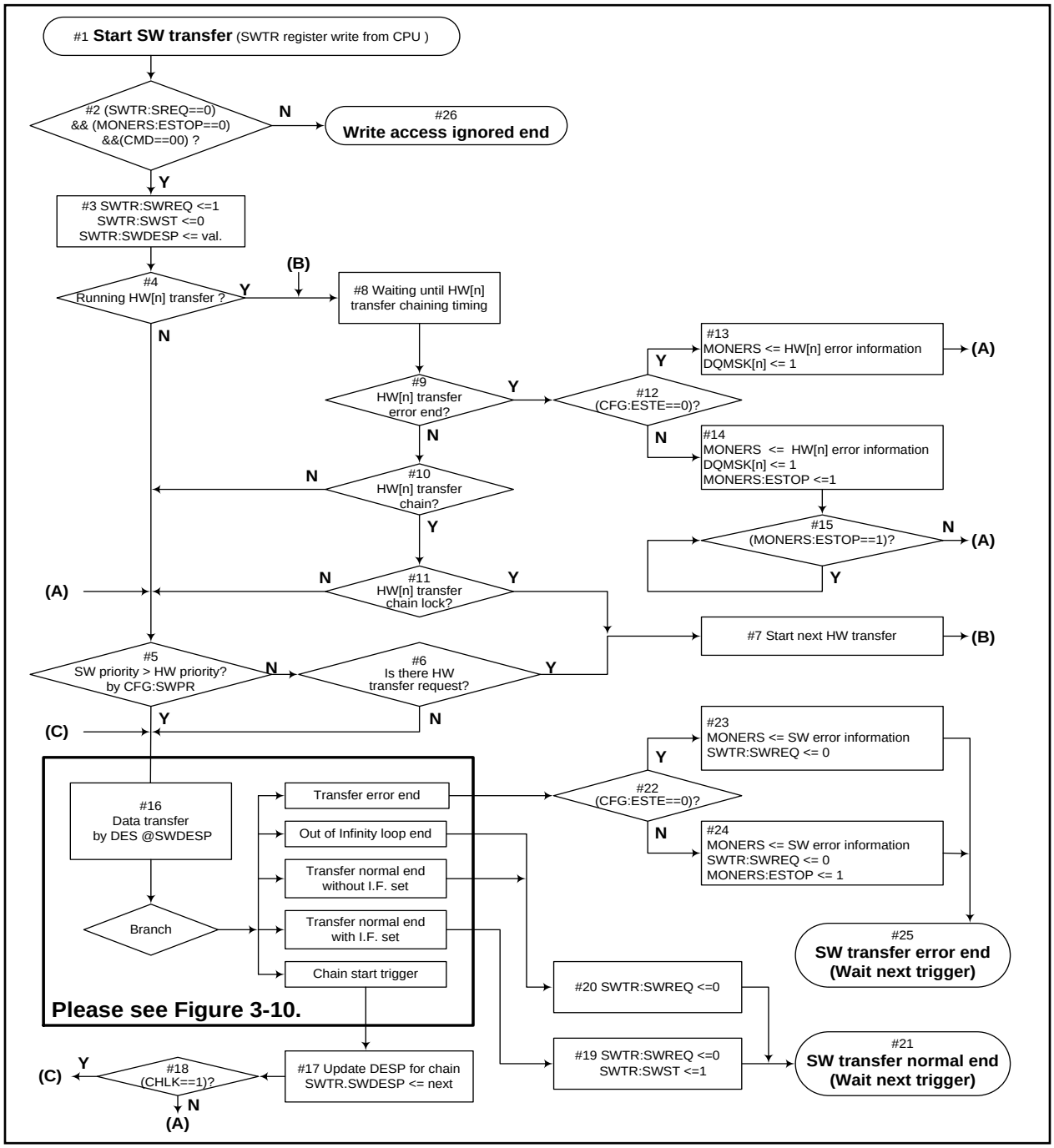
3.3 Operation Flows of DSTC

This section explains the operation of the DSTC with flow charts.

3.3.1 SW Transfer Flow

The operations the DSTC executes after receiving an SW Start Trigger from the CPU are explained below. Figure 3-8 shows a flow chart of the operations of the DSTC. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 3-8 DSTC Operation Flow (SW Transfer)



#1 Start the SW Start transfer from a write access to the SWTR Register from the CPU.

#2 If the SWTR Register, the MONERS Register and the CMD Register are (SWTR:SWREQ==0)&&(MONERS:ESTOP==0)&&(CMD==00), the DSTC proceeds to #3. Otherwise the DSTC proceeds to #26.

#3 Set "1" to SWTR:SWREQ and clear SWTR:SWST to 0. Store the specified value in SWTR:SWDESP.

Processes explained in #4 to #15 are details of operations of Arbiter 2 and processes in other HW Transfer.

#4 If other HW Transfer is being executed, the DSTC proceeds to #8. Otherwise the DSTC proceeds to #5.

#5 Determine whether the SW Transfer or the HW Transfer has higher priority according to the setting of the SWPR bit in the CFG Register. If the SW Transfer has higher priority, the DSTC proceeds to #16. Otherwise the DSTC proceeds to #6.

#6 If there is other HW Transfer request, the DSTC proceeds to #7. Otherwise the DSTC proceeds to #16.

#7 Start data transfer for that HW Transfer.

#8 Until that HW Transfer has completed the DES, the DSTC keeps waiting for the start of the execution of the SW Transfer which is issued in #1.

#9 If that HW Transfer has ended in the form of error, the DSTC proceeds to #12. Otherwise the DSTC proceeds to #10.

#10 If there is a Chain Start transfer in that HW Transfer, the DSTC proceeds to #11. Otherwise the DSTC proceeds to #5.

#11 If the Chain Start transfer in that HW Transfer is locked, the DSTC proceeds to #7. Otherwise the DSTC proceeds to #5.

#12 If CFG:ESTE is 0, the DSTC proceeds to #13. Otherwise the DSTC proceeds to #14.

#13 If there has been no error record (EST[2:0] = 000) in the MONERS Register, the MONERS Register records the error information of the DES of the HW Transfer that has caused an error. If there is an error record (EST[2:0] ≠ 000), the DSTC sets the DER bit to 1. The DSTC proceeds to #5.

#14 The same process as #13 is executed. At the same time, the ESTOP bit in the MONERS Register is set to 1. The DSTC proceeds to #15.

#15 The DSTC is holding the start of the execution of the SW Transfer which is issued in #1 while the ESTOP bit is 1. If an ERCLR command is issued by the CPU to the CMD Register and the ESTOP bit is cleared to 0, the DSTC proceeds to #5.

#16 The flow inside the bold box shows the transfer operations of the DSTC according to the DES specified in DESP. For details of the flow inside the bold box, see section Operation Flow after Specifying of DESP. In the case of the SW Start transfer, the DSTC executes the transfer according to the DES specified in SWDESP. After the transfer has been processed, the operation of the DSTC branches to one of the five operations shown in Figure 3-10.

#17 In the case of a Chain Start, the DSTC updates the value of SWTR:SWDESP.

#18 If CHLK is 1, the DSTC proceeds to #16 and successively executes the transfers started by the Chain Start. Otherwise the DSTC proceeds to #5.

#19 If the transfer ends normally and there is an interrupt flag set instruction, the DSTC executes the processes in #19. The DSTC clears SWTR:SWREQ to 0 and sets SWTR:SWST to 1.

#20 If the transfer ends normally and there is no interrupt flag set instruction, the DSTC executes the processes in #20. The DSTC clears SWTR:SWREQ to 0.

#21 The DSTC ends the transfer caused by the SW Start trigger in #1. The DSTC waits for either a new Start Trigger or a succeeding Start Trigger. The DESP of the DES whose transfer has ended is kept in SWTR:SWDESP.

#22 If that SW Start transfer has ended in the form of error and CFG:ESTE is 0, the DSTC proceeds to #23. Otherwise the DSTC proceeds to #24.

#23 If there has been no error record in the MONERS Register, the MONERS Register records the error information of the DES of the SW Start transfer that has caused an error. If there is an error record, the DSTC sets the DER bit to 1. The DSTC clears SWTR:SWREQ to 0.

#24 The same process as #23 is executed. At the same time, the ESTOP bit in the MONERS Register is set to 1, and the DSTC holds the start of the transfer of other HW Transfer.

#25 The transfer caused by the SW Start Trigger in #1 ends in the form of error. SWTR:SWST is not set to 1 regardless of the value of CHRS. The DSTC waits for a new Start Trigger.

#26 In a write access to the SWTR Register from the CPU as explained in #1, if the condition in #2 is not fulfilled, the DSTC ignores the write access to the SWTR Register. The DSTC does not accept the SW Start request.

Additional Information on Controlling DSTC in SW Transfer

If the condition in #2 is not fulfilled, meaning that the SW Start transfer instruction has been executed before #2, and that transfer has not ended (SWREQ ≠ 0) or the DSTC is not in the normal state (CMD ≠ 00) or the DSTC is in the error stop state (ESTOP ≠ 0), the DSTC ignores the new SW Start request from the CPU and does not accept it.

Pay attention to this behavior of the DSTC especially when using the DSTC with CFG:ESTE set to 1. If the DSTC has stopped for an error due to another HW Transfer, any new SW Start request (write access to a register) is ignored, and SWREQ is not set to 1. Therefore, if the DSTC reads 0 from the SWREQ bit in the SWTR Register after making a write access to the SWTR Register, it cannot determine whether an SW Start request has been ignored or a transfer has ended. Moreover, if using the DSTC with CFG:ESTE set to 1, in an SW Transfer, set DES0:CHRS to a value that when the DSTC does not execute the Chain Start, always sets SWTR:SWST to 1. With DES0:CHRS set in this way, after a write access has been made to the SWTR Register, that both SWREQ bit and SWST bit read 0 indicates that no transfer request has been accepted. If a transfer request has been accepted, since either SWREQ bit or SWST bit is 1, the DSTC can determine whether an SW Start transfer has been ignored or a transfer has ended.

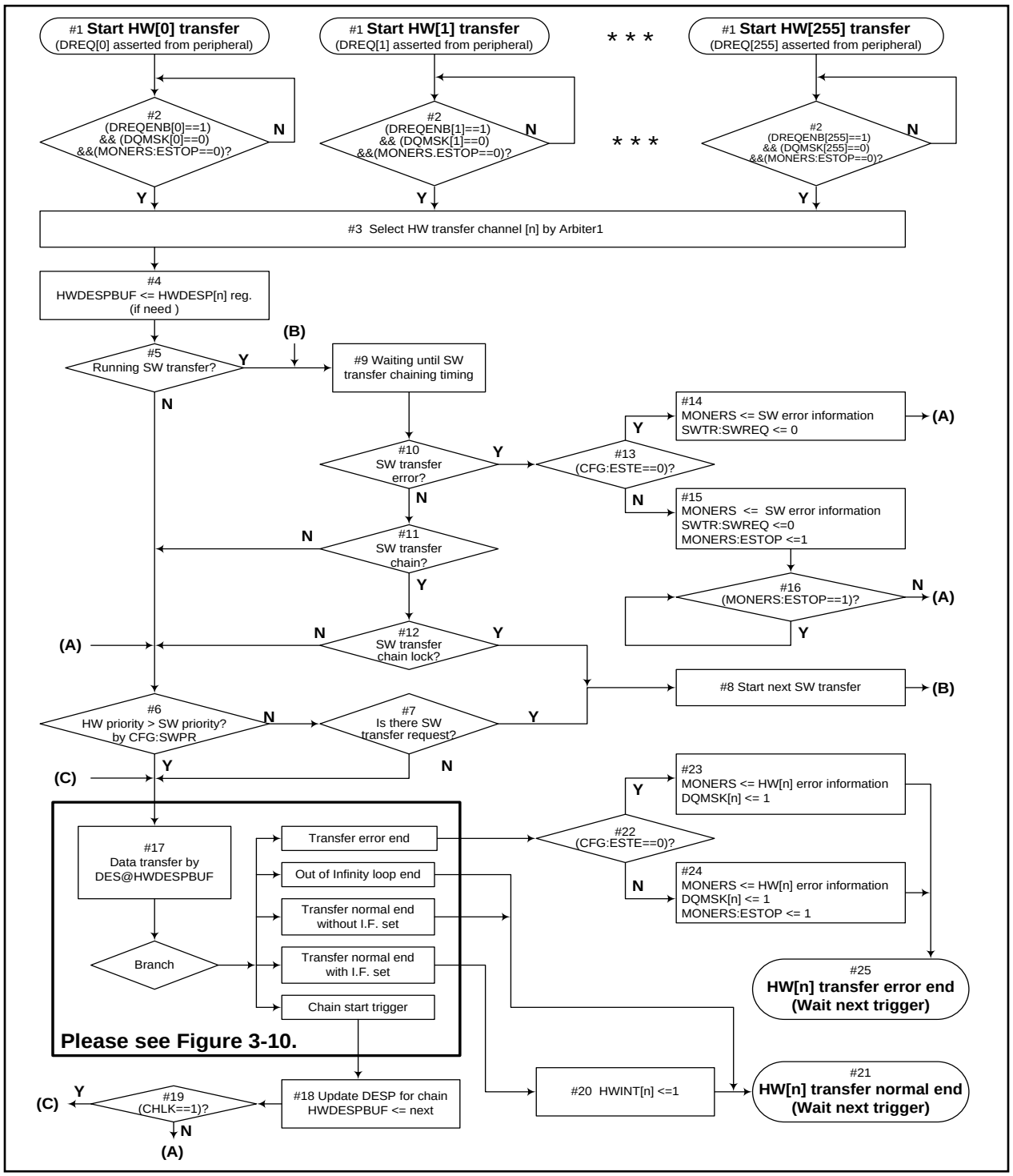
If the SWINT interrupt signal has been asserted by using the SWST bit, clear the SWST bit to 0 by issuing an SWCLR command during interrupt processing. Even if the SWST bit is not cleared to 0, a new SW Start request can be made by making a write access to the SWTR Register. However, in the process explained in #3, the SWST bit is always cleared to 0 and the SWINT interrupt signal is negated.

In #4, if other HW Transfer is being executed or the Chain Start in that HW Transfer has been locked, it may take time to start an SW Transfer even if it has a high priority.

3.3.2 HW Transfer Flow

The operations the DSTC executes after receiving an HW Start Trigger from a peripheral are explained below. Figure 3-9 shows a flow chart of the operations of the DSTC. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 3-9 DSTC Operation Flow (HW Transfer)



#1 The DSTC starts the HW Transfer from the assertion of the DREQ[n] signal from a peripheral.

#2 If the DREQENB[n] Register, the DQMSK[n] Register and the MONERS Register are (DREQENB[n]==1)&& (DQMSK[n]==0) &&(MONERS:ESTOP==0), the DSTC proceeds to #3. If the DQMSK[n] Register or ESTOP Register is set to 1, the DSTC ignores the DREQ[n] signal from a peripheral and holds the start of the HW Transfer.

#3 Processes in #3 are processes to be executed by Arbiter 1. In the case of transfer requests from multiple HW channels. the DSTC selects the number of the HW channel (n) on which it executes a transfer. The DSTC keeps the transfer requests from other channels until the transfer on the HW[n] selected ends normally or ends due to an error and the DSTC starts waiting for a Start Trigger.

#4 Based on the channel number (n) selected, store the DESP value of the HWDESP[n] Register in HWDESPBUF. If the channel number is the same as the one in the previous reference, and the value of HWDESPBUF is valid, it is skipped to refer to the HWDESP[n] Register.

Processes explained in #5 to #16 are details of operations of Arbiter 2 and processes in other SW Transfer. The DSTC executes the same processes as #4 to #15 in SW Transfer flow. If there is a SW Start request and that SW Transfer has a high priority, or if the Chain Start in that SW Transfer has been locked, the DSTC executes that SW Transfer first. In addition, if that SW Transfer ends in the form of error, the DSTC records the error information of the SW Transfer in the MONERS Register. If the ESTOP bit is set to 1 due to the error end of the SW Transfer, the DSTC holds the transfer start of HW[n].

#17 The flow inside the bold box shows the transfer operations of the DSTC according to the DES specified in DESP. For details of the flow inside the bold box, see section Operation flow after specifying of DESP. In the case of the HW Transfer, the DSTC executes the transfer according to the DES specified in HWDESP. After the transfer has been processed, the operation of the DSTC branches to one of the five operations shown in Figure 3-10.

#18 In the case of a Chain Start, the DSTC updates HWDESPBUF.

#19 If CHLK is 1, the DSTC proceeds to #17 and successively executes the transfers started by the Chain Start. Otherwise the DSTC proceeds to #6.

#20 If the transfer ends normally and there is an interrupt flag set instruction, the DSTC sets HWINT[n] to 1.

#21 The DSTC ends the transfer caused by the HW Start trigger in #1. The DSTC waits for either a new Start Trigger or a succeeding Start Trigger. If the DSTC keeps the HW Start Trigger for other channel in #3, Arbiter 1 selects the channel on which a transfer is to be executed and the DSTC proceeds to #4.

#22 If that HW Start transfer has ended in the form of error and CFG:ESTE is 0, the DSTC proceeds to #23. Otherwise the DSTC proceeds to #24.

#23 If there has been no error record in the MONERS Register, the MONERS Register records the error information of the DES of the HW Transfer being executed. If there is an error record, the DSTC sets the DER bit to 1. In addition, the DSTC sets the DQMSK[n] Register to 1 to suppress future transfer requests from HW channel n.

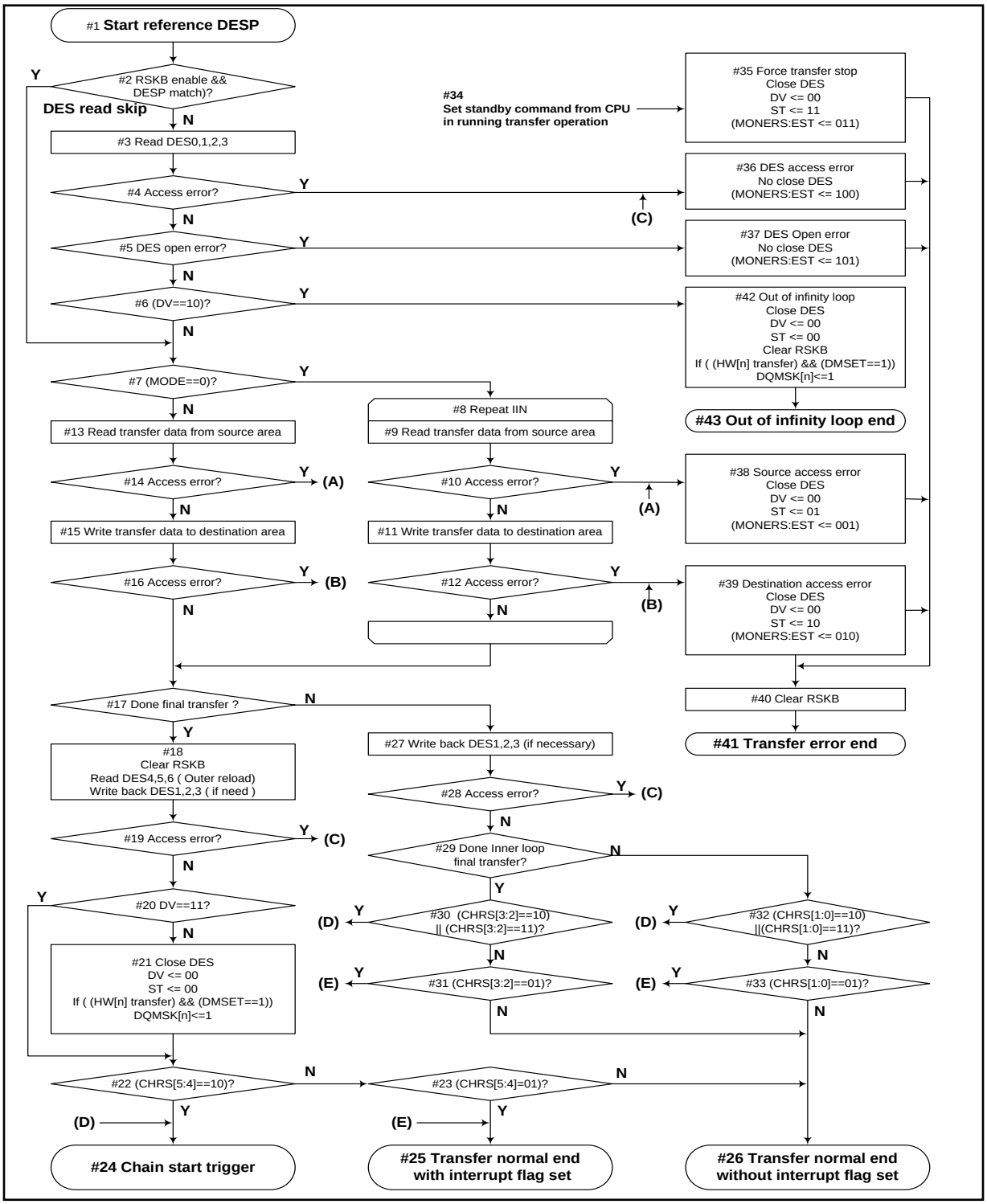
#24 The same process as #23 is executed. At the same time, the ESTOP bit in the MONERS Register is set to 1, and the DSTC holds the start of other Start transfer.

#25 The transfer caused by the HW Start Trigger in #1 ends in the form of error. The HWINT[n] Register is not set to 1 regardless of the value of CHRS. The DSTC waits for a new Start Trigger. If the DSTC keeps the HW Start Trigger for other channel, Arbiter 1 selects the channel on which a transfer is to be executed and the DSTC proceeds to #4.

3.3.3 Operation Flow after Specifying of DESP

The operations the DSTC executes after a DESP has been executed are explained below. Figure 3-10 shows a flow chart of the operations the DSTC executes after a DESP has been specified. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 3-10 DSTC Operation Flow after Specifying of DESP



#1 The DSTC starts its operation from referring to a DESP specified by Arbiter 2.

#2 If the read skip buffer function is enabled (CFG:RBDIS = 0) and the DESP to which the DSTC refers is the same as the one it referred to, the DSTC skips referring to the DES in the memory area and proceeds to #7. Otherwise the DSTC proceeds to #3.

#3, #4 The DSTC reads the DES in the area of DESTP+DESP specified. If an access error occurs in referring to the DES area, the DSTC proceeds to #36. Otherwise the DSTC proceeds to #5.

#5 The DSTC checks details of the DES. If a DES open error occurs, the DSTC proceeds to #37. Otherwise the DSTC proceeds to #6. For details of the DES open error, see "3.2.8 MONERS Register "

#6 If DES0:DV is 10, the DSTC proceeds to #42. Otherwise the DSTC proceeds to #7.

#7 If DES0:MODE is 0, the DSTC proceeds to #8. Otherwise the DSTC proceeds to #13.

#8 to #12 If mode 0 transfer has been specified, the DSTC executes transfers successively for the times specified in DES1:IIN. If an access error occurs in a transfer source access, the DSTC proceeds to #38. If an access error occurs in a transfer destination access, the DSTC proceeds to #39. If no access error occurs in a transfer source access or in a transfer destination access, the DSTC proceeds to #17.

#13 to #16 If mode 1 transfer has been specified, the DSTC executes one transfer. If an access error occurs in a transfer source access, the DSTC proceeds to #38. If an access error occurs in a transfer destination access, the DSTC proceeds to #39. If no access error occurs in a transfer source access or in a transfer destination access, the DSTC proceeds to #17.

#17 If IIN×ORM times of transfer have ended (ORM ==1 and IRM ==1), the DSTC proceeds to #18. Otherwise the DSTC proceeds to #27.

#18, #19 The DSTC clears the read skip buffer, reads required values from DES4 to DES6 areas according to the instruction specified in DES0:ORL, and writes the required values to DES1 to DES3 areas. If an access error occurs in updating the DES area reference, the DSTC proceeds to #36. Otherwise the DSTC proceeds to #20.

#20 If DES0:DV is 11, the DSTC proceeds to #22. If DES0:DV is 01, the DSTC proceeds to #21. (In #5, if DES0:DV is 00, the DSTC has proceeded to #37. In #6, if DES0.DV is 10, the DSTC has proceeded to #42)

#21 The DSTC executes the DES close process. The DSTC updates DES0:DV to 00 and DES0:ST to 00. In a transfer for an HW Start trigger (or a Chain transfer from an HW Start trigger), if DES0:DMSET is 1, the DSTC sets DQMSK[n] to 1 and suppresses future HW Start transfer requests of channel n.

#22, #23 If DES0:CHRS[5:4] are 10, the DSTC proceeds to #24. If DES0:CHRS[5:4] are 01, the DSTC proceeds to #25. If DES0:CHRS[5:4] are 00, the DSTC proceeds to #26. (In #5, if DES0:CHRS[5:4] are 11, the DSTC has proceeded to #37.)

#24 The DSTC executes a Chain Start transfer. The transfer of the DESP specified in #1 ends normally. For operations to be executed afterward, see the previous section.

#25 The transfer of the DESP specified in #1 ends normally with the interrupt flag set. For operations to be executed afterward, see the previous section.

#26 The transfer of the DESP specified in #1 ends normally without the interrupt flag being set. For operations to be executed afterward, see the previous section.

#27, #28 The DSTC writes back values required for the transfer for the next Start Trigger to DES1 to DES3 areas. If an access error occurs in updating the DES area, the DSTC proceeds to #36. Otherwise the DSTC proceeds to #29.

#29 If MODE is 0, the DSTC always proceeds to #30. If MODE is 1 and transfers for the times of the inner loop count have ended (ORM !=1 and IRM ==1), the DSTC proceeds to #30. Otherwise the DSTC proceeds to #32.

#30, #31 If DES0:CHRS[3:2] are 10 or 11, the DSTC proceeds to #24. If DES0:CHRS[3:2] are 01, the DSTC proceeds to #25. If DES0:CHRS[3:2] are 00, the DSTC proceeds to #26.

#32, #33 If DES0:CHRS[1:0] are 10 or 11, the DSTC proceeds to #24. If DES0:CHRS[1:0] are 01, the DSTC proceeds to #25. If DES0:CHRS[1:0] are 00, the DSTC proceeds to #26.

#34 If the CPU issues a standby transition command during a transfer, the DSTC interrupts the transfer and proceeds to #35.

#35 If the transfer is compulsorily stopped, the DSTC executes the DES error close process. The DSTC updates DES0:DV to 00 and DES0:ST to 11, and then proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to 011.

#36 If a DES access error occurs, the DSTC does not execute the DES error close process, and proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to 100.

#37 If a DES open error occurs, the DSTC does not execute the DES error close process, and proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to 101.

#38 If a transfer source access error occurs, the DSTC executes the DES error close process. The DSTC updates DES0:DV to 00 and DES0:ST to 01, and then proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to 001.

#39 If a transfer destination access error occurs, the DSTC executes the DES error close process. The DSTC updates DES0:DV to 00 and DES0:ST to 10, and then proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to 010.

#40, #41 The DSTC clears the read skip buffer. The transfer of the DESP specified in #1 ends in the form of error. For operations to be executed afterward, see the previous section.

#42, #43 The DSTC clears the read skip buffer. The DSTC executes the DES close process. The DSTC updates DES0:DV to 00 and DES0:ST to 00. In a transfer for an HW Start trigger (or a Chain transfer from an HW Start trigger), if DES0:DMSET is 1, the DSTC sets DQMSK[n] to 1 and suppresses future HW Start transfer requests. The transfer of the DESP specified in #1 ends as an infinite loop out. For operations to be executed afterward, see the previous section.

4. Examples of DSTC Operations and Control

This section describes examples of DSTC operations and control.

4.1. Transfer Operation Example 1
4.2. Transfer Operation Example 2
4.3. Transfer Operation Example 3
4.4. Transfer Operation Example 4
4.5. Transfer Operation Example 5
4.6. Examples of Controlling DSTC

4.1 Transfer Operation Example 1

This section describes transfer operation example 1. Transfer operation example 1 is an example on SW Transfer in mode 0.

DES Values at Transfer Start

Table 4-1 shows the settings of the DES in transfer operation example 1. As ORL[2:0] are set to 101, there is no DES5 area. The DES has 6-word configuration consisting of DES0 to DES4 and DES6. (The address of DES6 is DESP+0x0014.)

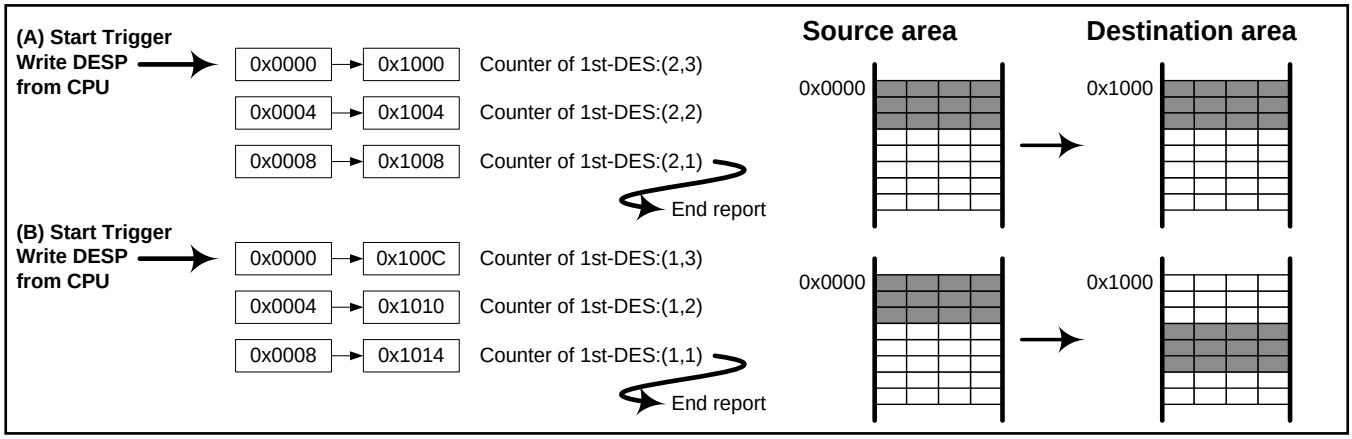
Table 4-1 DES Values at Transfer Start in Transfer Operation Example 1

Address	DES No.	Value
DESP+0x0000	DES0	DES0 = 0x901406A1 DV = 01 : DES close process to be executed at the end of transfer MODE = 0, TW = 10 : Mode 0, 32-bit (word) transfer ORL = 101 : OuterReload : DES1 <= DES4, DES3 <= DES6 SAC = 001 : Increment of TW×1 with InnerReload DAC = 000 : Increment of TW×1 without InnerReload CHRS = 010100 : There is no Chain Start; an interrupt flag has been set. DMSET = 0 : Set DMSET to 0 as the transfer is an SW Start transfer. CHLK = 0 : No Chain lock ACK = 00 : Set ACK to 00 as the transfer is an SW Start transfer. PCHK = 1001:Parity
DESP+0x0004	DES1	ORM = 0x0002, IIN = 0x0003
DESP+0x0008	DES2	SA = 0x00000000
DESP+0x000C	DES3	DA = 0x00001000
DESP+0x0010	DES4	ORM = 0x0002, IIN = 0x0003 (same as DES1)
DESP+0x0014	DES6	DA = 0x0000 1000 (same as DES3)

Transfer Operation Flow

Figure 4-1 Operation Flow in Transfer Operation Example 1 shows the transfer operation flow in transfer operation example 1. The Start Triggers of (A) and (B) in the figure show write accesses of the DESP to the SWTR Register from the CPU.

Figure 4-1 Operation Flow in Transfer Operation Example 1



The DSTC starts the transfer of the DES due to the Start Trigger of (A). Values inside rectangles in Figure 4-1 are transfer source addresses and transfer destination addresses. The DSTC starts from a 32-bit transfer to the area from address 0x0000 to address 0x1000. The DSTC executes three times (IIN = 3) of 32-bit transfer successively. As for the transfer number counter for the DES, at the start of the transfer, the outer loop counter remain (ORM) is 2, and the inner loop counter remain (IRM) is 3. In Figure 4-1, the remains of the transfer number counter are expressed as (2,3). After three times of transfer, the transfer number counter reads (2,1). As ORM is not 1 and IRM is 1, the DSTC uses the value of CHRS[3:2] for determining the next process. As CHRS[3:2] are 01, the DSTC sets SWTR:SWST to 1 and waits for the next Start Trigger.
The DSTC starts the transfer of the DES again due to the Start Trigger of (B). InnerReload of address is applied to SA. DA keeps increasing. The DSTC restarts from a 32-bit transfer to the area from address 0x0000 to address 0x100C. The DSTC executes three times (IIN = 3) of transfer successively. The transfer number counter starts counting from (1,3) and reads (1,1) after three times of transfer. As ORM is 1 and IRM is "1", the DSTC executes the DES close process as DV of the DES is set to 01. The DSTC uses the value of CHRS[5:4] for determining the next process. As CHRS[5:4] are 01, the DSTC sets SWTR:SWST to 1.

DES Values Stored after Transfer End

If the transfer in transfer operation example 1 ends normally, the values of DES are updated as shown in Table 4-2. Values that are different from what they were before the transfer start are in bold type in the table. According to the setting of ORL, values of DES4 and DES6 are copied to DES1 and DES3 respectively, making DES1 and DES3 have the same values as those before the start of transfer. Though OR[1] is 0, according to the settings of InnerReload, DES2 has the same value as that before the start of transfer. The DSTC updates the value of DV to 00, and returns the ownership of DES to the CPU. The DSTC updates the value of ST to 00, and notifies the CPU that the transfer has ended normally.
To execute a transfer with the updated DES mentioned above, update the value of DV via the CPU.

Table 4-2 DES Values after End of Transfer in Transfer Operation Example 1

DES No.	Value
DES0	**DV = 00, ST = 00,** other values is same as the start of transfer
DES1	It is same as the start of transfer.
DES2	It is same as the start of transfer.
DES3	It is same as the start of transfer.
DES4	It is same as the start of transfer.
DES6	It is same as the start of transfer.

The Operation for the DES from the DSTC

In this transfer operation example1, the operation for the DES from the DSTC is as follows.
After Start Trigger of (A):

The DSTC read the instruction from DES0.
The DSTC read (2,3) from DES1.
The DSTC read 0x0000 from DES2.
The DSTC read 0x1000 from DES3.

After 1st transfer:

The DSTC write back (1,3) to DES1,
The DSTC does not write back to DES2, so same value.
The DSTC write back 0x100C to DES3.

After Start Trigger of (B):

* The DSTC read the instruction from DES0.
* The DSTC read (1,3) from DES1.
* The DSTC read 0x0000 from DES2.
* The DSTC read 0x100C from DES3.

After 2nd transfer:

The DSTC copy (2,3) to DES1 from DES4 for OuterReload.
The DSTC does not write back to DES2, so same value.
The DSTC copy 0x1000 to DES3 from DES6 for OuterReload.
The DSTC write back DES0 for close DES.

If other transfer request is not issued between Start Trigger (A) and Start Trigger (B), the DSTC use the value of the DES0,1,2,3 in Read Skip Buffer when 2nd transfer. Therefore, above operations have *mark are skipped. For details, see 3.3.3 Operation Flow after Specifying of DESP.

Supplementary Information

The Start Trigger of (B) cannot be issued until the transfer triggered by the Start Trigger of (A) ends and the DSTC enters the Start Trigger wait state. When SWTR:SWREQ is 1, the write access (SW Start Trigger) to the SWTR Register is ignored.

If a transfer triggered by the Start Trigger of (A) ends and the DSTC enters the Start Trigger wait state, an SW Start Trigger other than that of (B) can be issued for other DES. After an SW Transfer for other DES has ended, if the DESP of that DES is written to the SWTR Register, the Start Trigger of (B) is issued, and the DSTC continues the data transfer from the location at which the transfer of (A) ended.

The Start Trigger of (B) is not always required after the transfer triggered by the Start Trigger of (A) has ended. If the Start Trigger of (B) is not issued and the DSTC does not continue transferring data, after the transfer triggered by the Start Trigger of (A) has ended (even no DES close process is executed), the CPU can modify the DES area of the transfer.

4.2 Transfer Operation Example 2

This section describes transfer operation example 2. Transfer operation example 2 is an example on HW Transfer in mode 1.

DES Values at Transfer Start

Table 4-3 shows the settings of the DES in transfer operation example 2. As ORL[2:0] are set to 000, there are no DES4 area, DES5 area or DES6 area. The DES has 4-word configuration consisting of DES0 to DES3.

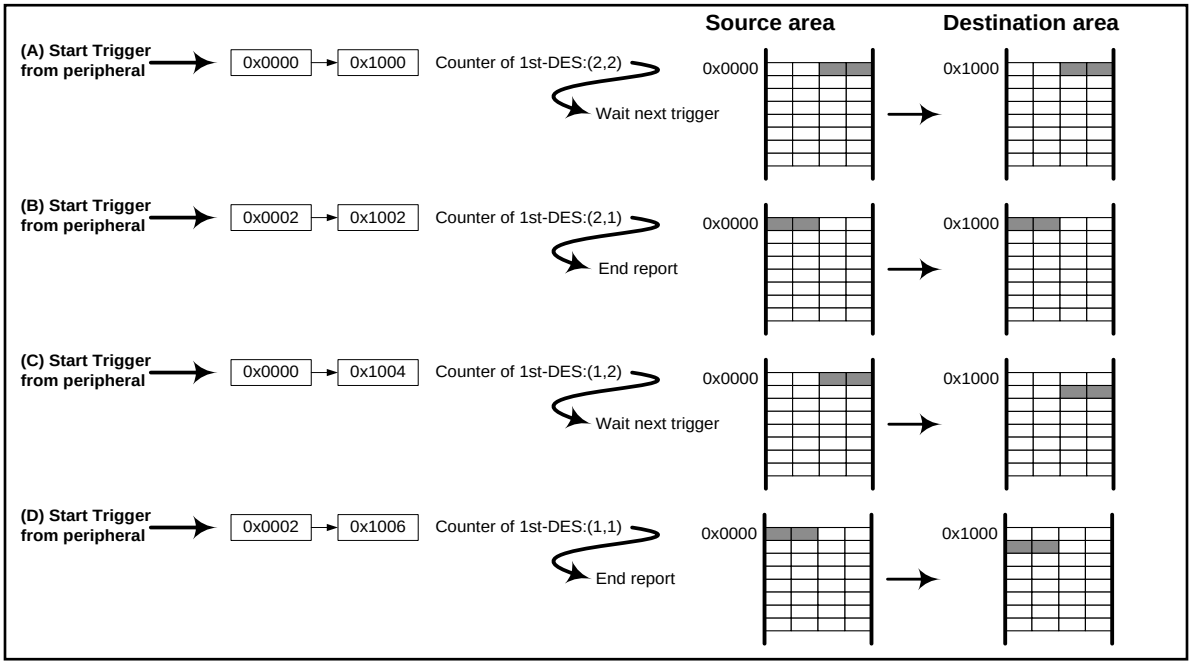
Table 4-3 DES Values at Transfer Start in Transfer Operation Example 2

Address	DES No.	Value
DESP+0x0000	DES0	DES0 = 0x01140511 DV = 01 : DES close process at the of transfer MODE = 1, TW = 01 : Mode 1, 16-bit (halfword) transfer ORL = 000 : No OuterReload SAC = 001 : Increment of TW×1 with InnerReload DAC = 000 : Increment of TW×1 without InnerReload CHRS = 010100 : There is no Chain Start; an interrupt flag has been set. DMSET = 0 : No DQMSK[n] is set in a DES close process. CHLK = 0 : No Chain lock ACK = 01 : Set ACK to 01 as the DES is directly started by the DREQ signal. PCHK = 0000 : Parity
DESP+0x0004	DES1	ORM = 0x0002, IIN = 0x02, IRM = 0x02
DESP+0x0008	DES2	SA = 0x00000000
DESP+0x000C	DES3	DA = 0x00001000

Transfer Operation Flow

Figure 4-2 shows the transfer operation flow in transfer operation example 2. The Start Trigger of (A) in Figure 4-2 corresponds to the assertion of the DREQ[n] signal from a peripheral. For DREQ[n], set the values of the DREQENB[n], DQMSK[n] and HWDESP[n] Registers before starting a transfer.

Figure 4-2 Operation Flow in Transfer Operation Example 2



The DSTC starts the transfer of the DES due to the Start Trigger of (A). The DSTC executes one 16-bit transfer to the area from address 0x0000 to address 0x1000. The transfer number counter reads (2,2). As IRM is not 1, according to the setting of CHRS[1:0] of the DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to 1. The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of the DES again due to the Start Trigger of (B). The DSTC executes one 16-bit transfer to the area from address 0x0002 to address 0x1002. The transfer number counter reads (2,1). As ORM is not 1 and IRM is 1, according to the setting of CHRS[3:2] of the DES (CHRS[3:2] = 01), the DSTC sets the HWINT[n] Register to 1. The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of the DES again due to the Start Trigger of (C). InnerReload of address is applied to SA. DA keeps increasing. The DSTC executes one 16-bit transfer to the area from address 0x0000 to address 0x1004. The transfer number counter reads (1,2). As IRM is not "1", according to the setting of CHRS[1:0] of the DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to "1". The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of the DES again due to the Start Trigger of (D). The DSTC executes one 16-bit transfer to the area from address 0x0002 to address 0x1006. The transfer number counter reads (1,1). As ORM is 1 and IRM is 1, the DSTC executes the DES close process as DV of the DES is 01. According to the setting of CHRS[5:4] of the DES (CHRS[5:4] = 01), the DSTC sets the HWINT[n] Register to 1.

DES Values Stored after Transfer End

If the transfer in transfer operation example 2 ends normally, the values of DES are updated as shown in Table 4-4. Values that are different from what they were before the transfer start are in bold type in the table. As ORL has been set to "000", if the values in the table are used in the next transfer, the DSTC cannot execute a transfer same as the transfer mentioned above. In this situation, rebuild the DES via the CPU.

Table 4-4 DES Values after End of Transfer in Transfer Operation Example 2

DES No.	Value
DES0	DV = 00, ST = 00, other values is same as the start of transfer.
DES1	ORM = 0x0001, IRM = 0x01, IIN is same as the start of transfer.
DES2	It is same as the start of transfer.
DES3	DA = 0x00001006

4.3 Transfer Operation Example 3

This section describes transfer operation example 3. Transfer operation example 3 is an example on using the Chain Start of the succeeding DES.

DES Values at Transfer Start

In transfer operation example 3, the DSTC uses the Chain Start to re-arrange data at 0x0000 to 0x00FF and transfer data to the area between 0x0100 and 0x01FF. Four DES are used in this example. The first DES is called 1st-DES, the second DES 2nd-DES, the third DES 3rd-DES and the fourth DES 4th-DES. Table 4-5 shows the respective details of the four DES. There is no DES4 in each DES. The respective addresses of the four DES are not shown in Table 4-5. However, allocate the four DES next to each other in sequence from 1st-DES to 4th DES on the memory.

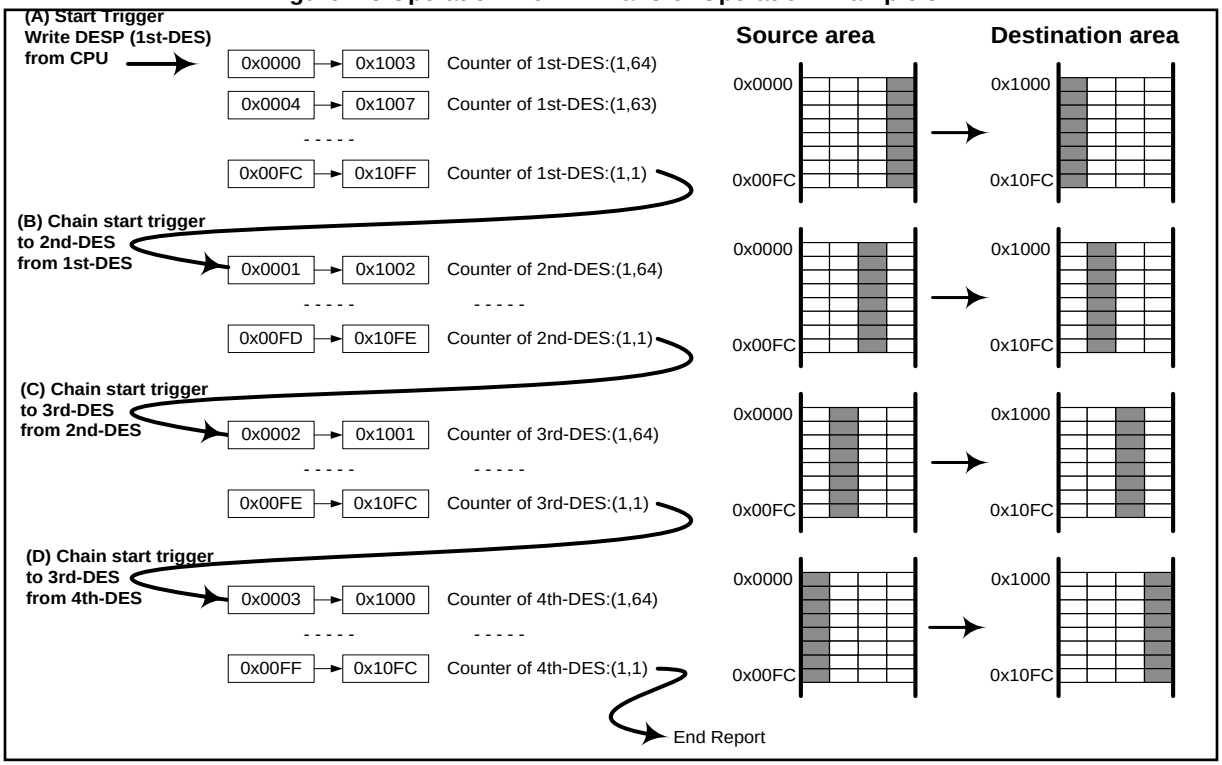
Table 4-5 DES Values at Transfer Start in Transfer Operation Example 3

Area	DES No.	Value
1st-DES	DES0	DES0 = 0x702090C3 DV = 11 : No DES close process to be executed at the end of transfer MODE = 0, TW = 00 : Mode 0, 8-bit (byte) transfer ORL = 110 : OuterReload of DES2 <= DES5, DES3 <= DES6 SAC = 100 : Increment of TW×4 without InnerReload DAC = 100 : Increment of TW×4 without InnerReload CHRS = 100000 : There is a Chain Start in the succeeding DES; no interrupt flag has been set. DMSET = 0 : Set DMSET to "0" as the transfer is an SW Transfer. CHLK = 0 : No Chain lock ACK = 00 : Set ACK to "00" as the transfer is an SW Transfer. PCHK = 0111 : Parity
	DES1	ORM = 0x0001, IIN = 0x0040
	DES2	SA = 0x00000000
	DES3	DA = 0x00001003
	DES5, DES6	DES5 has the same values as DES2 of 1st-DES, and DES6 as DES3 of 1st-DES.
2nd-DES	DES0	Same as DES0 of 1st-DES
	DES1	Same as DES1 of 1st-DES
	DES2	SA = 0x00000001
	DES3	DA = 0x00001002
	DES5, DES6	DES5 has the same values as DES2 of 2nd-DES, and DES6 as DES3 of 2nd-DES.
3rd-DES	DES0	Same as DES0 of 1st-DES
	DES1	Same as DES1 of 1st-DES
	DES2	SA = 0x00000002
	DES3	DA = 0x00001001
	DES5, DES6	DES5 has the same values as DES2 of 3rd-DES, and DES6 as DES3 of 3rd-DES.
4th-DES	DES0	DES0 = 0x401090C3 CHRS = 010000 : There is no Chain Start; an interrupt flag has been set. PCHK = 0100 : Parity Other values are the same as those of DES0 of 1st-DES.
	DES1	Same as DES1 of 1st-DES
	DES2	SA = 0x00000003
	DES3	DA = 0x00001000
	DES5, DES6	DES5 has the same values as DES2 of 4th-DES, and DES6 as DES3 of 4th-DES.

Transfer Operation Flow

Figure 4-3 shows the transfer operation flow in transfer operation example 3. The Start Trigger of (A) in the Figure 4-3 shows write accesses to the DESP of 1st-DES to the SWTR Register from the CPU. The Start Triggers of (B), (C) and (D) in the Figure 4-3 are Chain Start Triggers.

Figure 4-3 Operation Flow in Transfer Operation Example 3



The DSTC starts the transfer of 1st-DES due to the Start Trigger of (A). The DSTC executes one 8-bit transfer to the area from address 0x0000 to address 0x1003, and another 8-bit transfer to the area from address 0x0004 to address 0x1007. The DSTC executes 64 times of transfer (IIN = 64) successively according to the order above. The transfer number counter for 1st-DES starts counting from (1,64) and reads (1,1) after 64 times of transfer. As DV of 1st-DES is 11, the DSTC does not execute the DES close process for 1st-DES. As CHRS[5:4] of 1st-DES is 10, the Chain Start Trigger for the succeeding transfer of 2nd-DES is issued.

The DSTC starts the transfer of 2nd-DES due to the Chain Start Trigger of (B). After 64 times of 8-bit transfer, the counter value becomes (1,1). As DV of 2nd-DES is 11, the DSTC does not execute the DES close process for 2nd-DES. As CHRS[5:4] of 2nd-DES is 10, the Chain Start Trigger for the succeeding transfer of 3rd-DES is issued.

The DSTC starts the transfer of 3rd-DES due to the Start Trigger of (C). After 64 times of 8-bit transfer, the counter value becomes (1,1). As DV of 3rd-DES is 11, the DSTC does not execute the DES close process for 3rd-DES. As CHRS[5:4] of 3rd-DES is 10, the Chain Start Trigger for the succeeding transfer of 4th-DES is issued.

The DSTC starts the transfer of 4th-DES due to the Start Trigger of (D). After 64 times of 8-bit transfer, the counter value becomes (1,1). As DV of 4th-DES is 11, the DSTC does not execute the DES close process for 4th-DES. As CHRS[5:4] of 4th-DES is 01, the DSTC sets SWTR:SWST to 1 and ends the transfer.

As explained in transfer operation example 3, if the Chain Start function is used, transfers defined in multiple DES can be automatically executed one after the other by just issuing a Start Trigger to the first DES.

DES Values Stored after Transfer End

If the transfer in transfer operation example 3 ends normally, according to the settings of ORL and DV (ORL = 0 or 1, DV = 11), the values of DES in all DES areas are the same as what they were before the start of the transfer. Since the DSTC does not execute the DES close process, it does not update DV or ST either. In the next transfer, the transfer same as the previous transfer can be executed by just issuing the Start Trigger.

4.4 Transfer Operation Example 4

This section describes transfer operation example 4.

DES Values at Transfer Start

Below are details of transfer operation example 4. This example illustrates executing the Chain Start of the current DES again with CHRS set to 11. Table 4-6 shows settings of the DES.

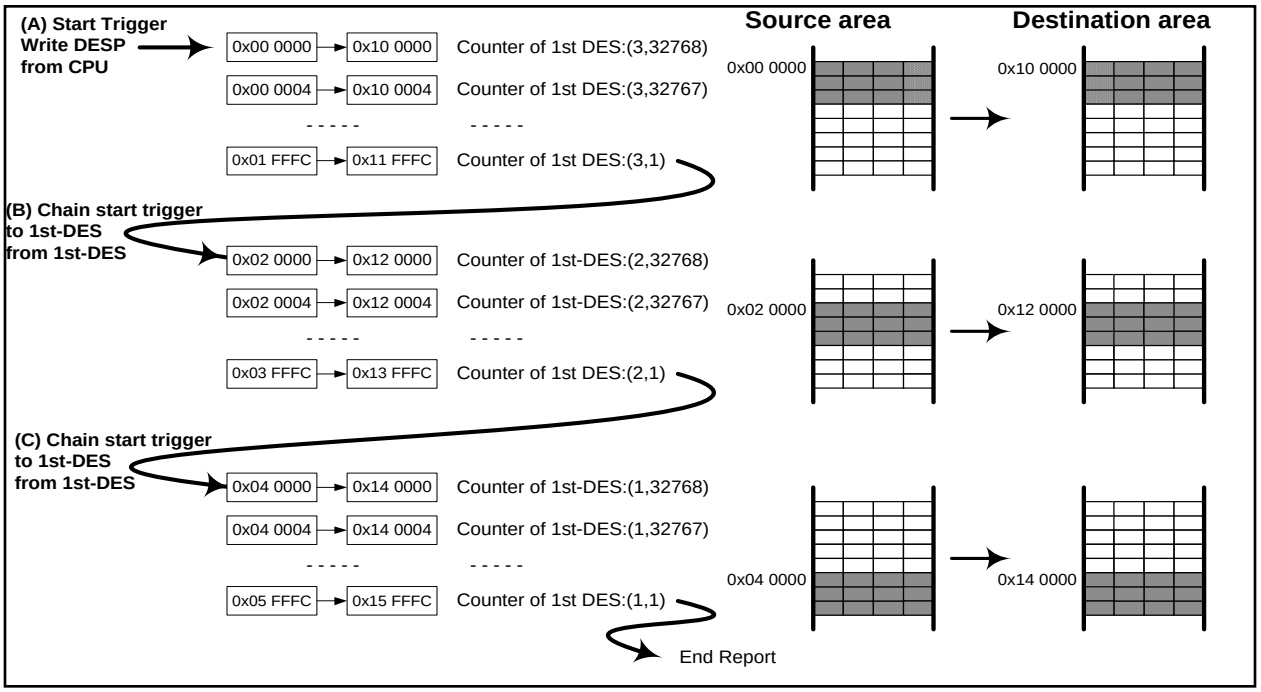
Table 4-6 DES Values at Transfer Start in Transfer Operation Example 4

DES No.	Value
DES0	DES0 = 0xF01C0201 DV = 01 : DES close process to be executed at the end of transfer MODE = 0, TW = 10 : Mode 0, 32-bit (word) transfer ORL = 000 : No OuterReload SAC = 000 : Increment of TW×1 without InnerReload DAC = 000 : Increment of TW×1 without InnerReload CHRS = 011100 : There is a Chain Start in the current DES; an interrupt flag has been set. DMSET = 0 : Set DMSET to "0" as the transfer is an SW Start transfer. CHLK = 0 : No Chain lock ACK = 00 : Set ACK to "00" as the transfer is an SW Start transfer. PCHK = 1111 : Parity
DES1	ORM = 0x0003, IIN = 0x8000
DES2	SA = 0x00000000
DES3	DA = 0x00100000

Transfer Operation Flow

Figure 4-4 shows the transfer operation flow in transfer operation example 4. The Start Trigger of (A) in the Figure 4-4 represents the write access of the DESP to the SWTR Register from the CPU. The Start Triggers of (B) and (C) in Figure 4-4 are Chain Start Triggers.

Figure 4-4 Operation Flow in Transfer Operation Example 4



The DSTC starts the transfer of the DES due to the Start Trigger of (A). The DSTC executes 32768 (IIN = 32768) times of 32-bit transfer successively with the address increasing during transfers. The transfer number counter for the DES starts counting from (3,32768) and reads (3,1) after 32768 times of transfer. As CHRS[3:2] of the DES is 11, the Chain Start Trigger for the transfer of the same DES is issued again.
The DSTC starts the transfer of the DES again due to the Chain Restart Trigger of (B). The DSTC executes 32768 times of transfer again. The transfer number counter for the DES reads (2,1) after 32768 times of transfer. As CHRS[3:2] of the DES is 11, the Chain Start Trigger for the transfer of the same DES is issued again.
The DSTC starts the transfer of the DES again due to the Chain Restart Trigger of (C). The DSTC executes 32768 times of transfer again. The transfer number counter for the DES reads (1,1) after 32768 times of transfer. As DV of the DES is 01, the DSTC executes the DES close process for the DES. As CHRS[5:4] of 4th-DES is 01, the DSTC sets SWTR:SWST to 1 and ends the transfer.

DES Values Stored after Transfer End

If the transfer in transfer operation example 4 ends, the values of DES are updated as shown in Table 4-7. Values that are different from what they were before the transfer start are in bold type in the table. In DES2 and DES3, the transfer start addresses in the third outer loop have been stored.

Table 4-7 DES Values after End of Transfer in Transfer Operation Example 4

DES No.	Value
DES0	DV = 00, ST = 00, other values is same as the start of transfer
DES1	ORM = 0x0001, IIN is same as the start of transfer
DES2	SA = 0x00040000
DES3	DA = 0x00140000

Supplementary Information

As explained above, since CHRS[3:2] are set to 11, the DSTC can trigger a Chain Start from the current DES to the current DES again. In general, in a transfer with MODE set to 0, one Start Trigger triggers the execution of IIN times of transfer. However, if CHRS[3:2] are set to 11, one Start Trigger can trigger the execution of ORM×IIN times of transfer.
In transfer operation example 4, the DSTC divides a total of 98304 times of transfer into three parts, and executes the three parts (ORM = 3) of transfers (IIN = 32768) separately. As long as the product of ORM and IIN is the same as the total number of transfers, the transfer result remains the same regardless of how many parts transfers are divided into. Once the DSTC starts a transfer, it can start processing another transfer request only when it enters the Start Trigger wait state or it meets the Chain Start time. Therefore, as explained in transfer operation example 4, when the DSTC transfers a large amount of data, the start of transfer for another HW transfer request may be delayed. To prevent such delay from occurring, adjust the value of ORM with the product of ORM and IIN the same as the total amount of transfers, so that a large amount of transfers can be divided into smaller parts. As a result, the DSTC can transfer much data without delaying other HW transfer requests.
Conversely, to prevent the DSTC from processing another HW transfer request at a Chain Start, use the Chain lock by setting CHLK to 1 so that the DSTC can execute transfers successively.

Note:

- *In the HW Ttransfer, for a DES whose transfer is triggered by a start request from a peripheral that has to have to a handshake with the DSTC at every data transfer, CHRS cannot be set to 11.*

4.5 Transfer Operation Example 5

This section describes transfer operation example 5.

DES Values at Transfer Start

Below are details of transfer operation example 5.

Table 4-8 DES Values at Transfer Start in Transfer Operation Example 5

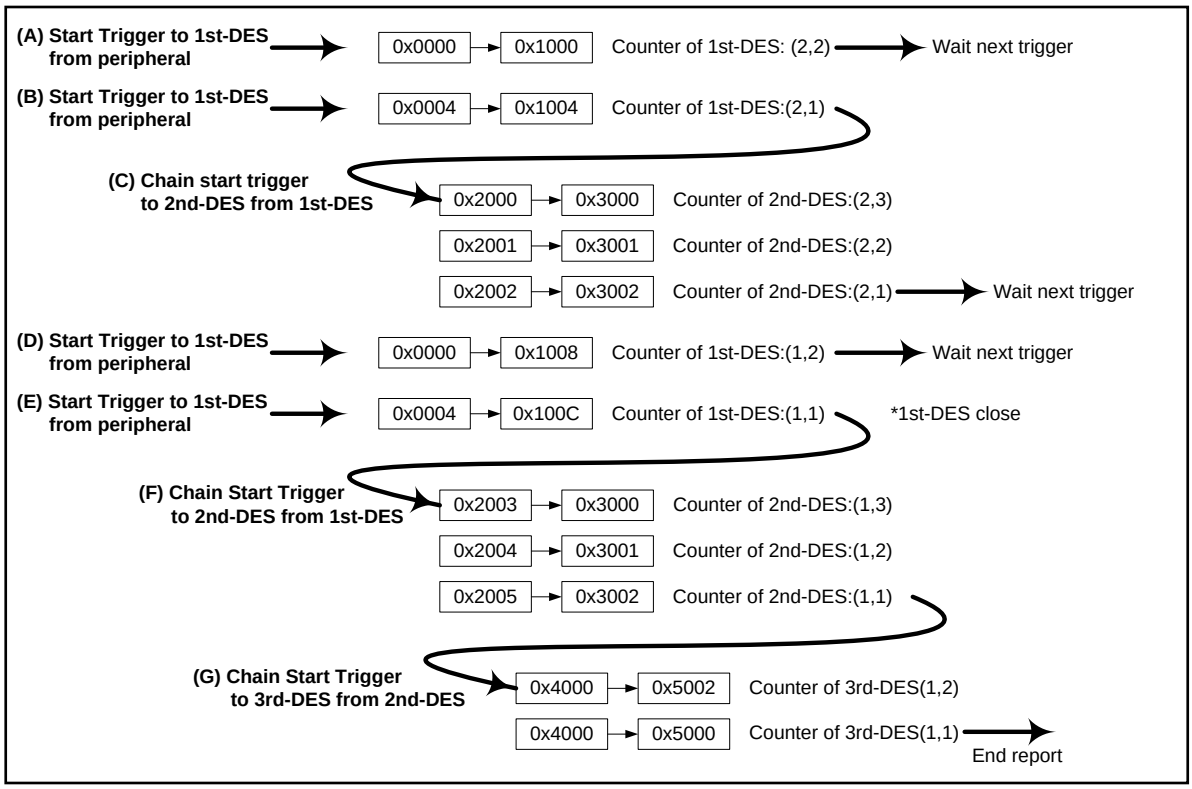
Area	DES No.	Value
1st-DES	DES0	DES0 = 0x612806B1 DV = 01 : DES close process to be executed at the end of transfer MODE = 1, TW = 10 : Mode 1, 32-bit (word) transfer ORL = 101 : OuterReload of DES1 <= DES4, DES3 <= DES6 SAC = 001 : Increment of TW×1 with InnerReload. DAC = 000 : Increment of TW×1 without InnerReload. CHRS = 101000 : There is a Chain Start in the succeeding DES; no interrupt flag has been set. DMSET = 0 : No DQMSK[n] is set in a DES close process. CHLK = 0 : No Chain lock ACK = 01 : Set ACK to "01" as the DES is directly started by the hardware. PCHK = 0110 : Parity
	DES1	ORM = 0x0002, IIN = 0x02, IRM = 0x02
	DES2	SA = 0x00000000
	DES3	DA = 0x00001000
	DES4, 6	Same as DES1 of 1st-DES, Same as DES3 of 1st-DES, respectively
2nd-DES	DES0	DES0 = 0x60202063 DV = 11 : No DES close process to be executed at the end of transfer MODE = 0, TW = 00 : Mode 0, 8-bit (byte) transfer ORL = 011 : OuterReload of DES1 <= DES4, DES2 <= DES5 SAC = 000 : Increment of TW×1 without InnerReload DAC = 001 : Increment of TW×1 with InnerReload CHRS = 100000 : There is a Chain Start in the succeeding DES; no interrupt flag has been set. DMSET = 0 : No DQMSK[n] is set in a DES close process. CHLK = 0 : No Chain lock ACK = 00 : Set ACK to "00" as the DES is directly started by a Chain Start. PCHK = 0110 : Parity
	DES1	ORM = 0x0002, IIN = 0x0003
	DES2	SA = 0x00002000
	DES3	DA = 0x00003000
	DES4,6	Same as DES1 of 2nd-DES, Same as DES2 of 2nd-DES, respectively
3rd-DES	DES0	DES0 = 0xB010F503 DV = 11 : No DES close process to be executed at the end of transfer MODE = 0, TW = 01 : Mode 0, 16-bit (halfword) transfer ORL = 000 : No OuterReload SAC = 101 : Fixed address DAC = 111 : Decrement of TW×1 with InnerReload. CHRS = 010000 : There is no Chain Start; an interrupt flag has been set. DMSET = 0 : No DQMSK[n] is set in a DES close process. CHLK = 0 : No Chain lock ACK = 00 : Set ACK to "00" as the DES is directly started by a Chain Start. PCHK = 1011 : Parity
	DES1	ORM = 0x0001, IIN = 0x0002
	DES2	SA = 0x00004000
	DES3	DA = 0x00005002

This example illustrates an operation in which relatively complicated Chain Start settings are done. Three DES are used in this example. Table 4-8 shows the respective values of 1st-DES, 2nd-DES and 3rd-DES.

Transfer Operation Flow

Figure 4-5 shows the transfer operation flow in transfer operation example 5. The Start Triggers of (A), (B), (D) and (E) in Figure 4-5 indicate HW Start transfers and correspond to the assertion of the transfer request signal from a peripheral. The Start Triggers of (C), (F) and (G) in Figure 4-5 are Chain Start Triggers.

Figure 4-5 Operation Flow in Transfer Operation Example 5



The DSTC starts the transfer of 1st-DES due to the Start Trigger of (A). The DSTC executes one 32-bit transfer to the area from address 0x0000 to address 0x1000. The transfer number counter for 1st-DES is (2,2). According to the setting of CHRS[1:0] of 1st-DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to 1. The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of 1st-DES again due to the Start Trigger of (B). The DSTC executes one 32-bit transfer to the area from address 0x0004 to address 0x1004. The transfer number counter for 1st-DES is (2,1). As CHRS[3:2] of 1st-DES is 10, the Chain Start Trigger for the succeeding transfer of 2nd-DES is issued.

The DSTC starts the transfer of 2nd-DES due to the Start Trigger of (C). The DSTC starts from an 8-bit transfer to the area from address 0x2000 to address 0x3000. The DSTC executes three times (IIN = 3) of 8-bit transfer successively. The transfer number counter for 2nd-DES starts counting from (2,3) and reads (2,1) after transfers. According to the setting of CHRS[3:2] of 2nd-DES (CHRS[3:2] = 00), the DSTC does not set the HWINT[n] Register to 1. The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of 1st-DES again due to the Start Trigger of (D). InnerReload of address is applied to SA. DA keeps increasing. The DSTC executes one 32-bit transfer to the area from address 0x0000 to address 0x1008. The transfer number counter reads (1,2). According to the setting of

CHRS[1:0] of 1st-DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to 1. The DSTC waits for the next Start Trigger.
The DSTC starts the transfer of 1st-DES again due to the Start Trigger of (E). The DSTC executes one 32-bit transfer to the area from address 0x0004 to address 0x100C. The transfer number counter for 1st-DES is (1,1). As DV of 1st-DES is 01, the DSTC executes the DES close process for 1st-DES. As CHRS[5:4] of 1st-DES is 10, the Chain Start Trigger for the succeeding transfer of 2nd-DES is issued. (Note that CHRS[5:0] in 1st-DES have been set to 101000.)
The DSTC starts the transfer of 2nd-DES again due to the Chain Start Trigger of (F). SA keeps increasing. InnerReload of address is applied to DA. The DSTC executes three times (IIN = 3) of 8-bit transfer successively. The transfer number counter for 2nd-DES starts counting from (1,3) and reads (1,1) after transfers. As DV of 2nd-DES is 11, the DSTC does not execute the DES close process for 2nd-DES. As CHRS[5:4] of 2nd-DES is 10, the Chain Start Trigger for the succeeding transfer of 3rd-DES is issued.
The DSTC starts the transfer of 3rd-DES due to the Start Trigger of (G). The DSTC executes two times (IIN = 2) of 16-bit transfer successively. The transfer number counter for 3rd-DES starts counting from (1,2) and reads (1,1) after transfers. As DV of 3rd-DES is 11, the DSTC does not execute the DES close process for 3rd-DES. According to the setting of CHRS[5:4] of 3rd-DES (CHRS[5:4] = 01), the DSTC sets the HWINT[n] Register to 1.
In transfer operation example 5, the transfer operation ends as above. The settings in this example cause the DSTC to execute the following operations. The Chain Start Trigger from 1st-DES to 2nd-DES is issued only after the transfer operation by Start Trigger (B) and (E). Also, it sets the HWINT[n] Register to 1 at the end of 3rd-DES transfer. After all transfers have ended, the DSTC notifies the CPU that all transfers ended.

DES Values Stored after Transfer End

If the transfer in transfer operation example 5 ends, the values of DES are updated as shown in Table 4-9. Values that are different from what they were before the transfer start are in bold type in the table. As the CPU has to have to a handshake with the DSTC, 1st-DES is set to be closed by the DSTC after the transfer has ended. 2nd-DES and 3rd-DES are set to not be closed by the DSTC.

Table 4-9 DES Values after End of Transfer in Transfer Operation example 5

Area	DES No.	Value
1st-DES	DES0	DV = 00, ST = 00, other values is same as the start of transfer
	DES1 to DES4, DES6	It is same as the start of transfer
2nd-DES	DES0 to DES5	It is same as the start of transfer
3rd-DES	DES0 to DES3	It is same as the start of transfer

4.6 Examples of Controlling DSTC

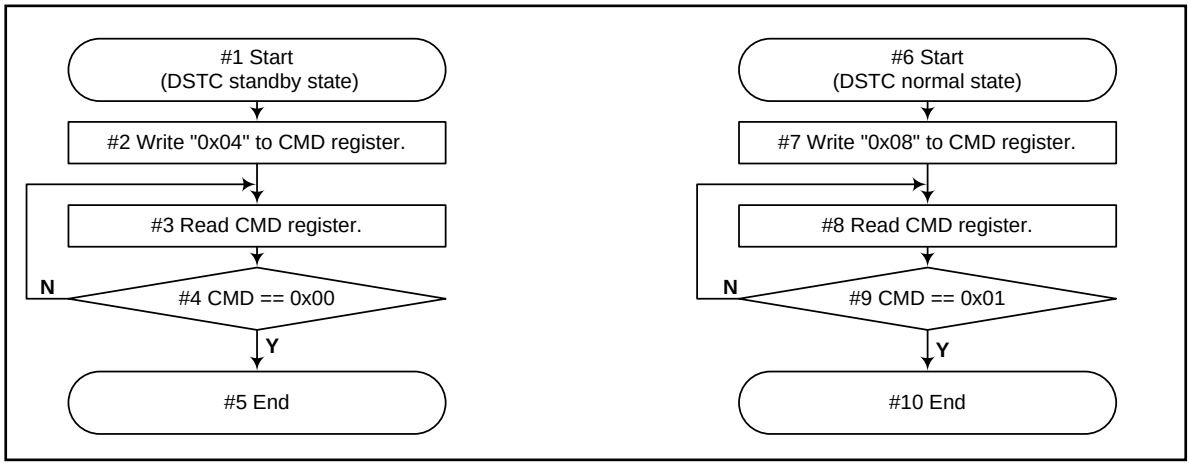
This section explains sample procedures for controlling the DSTC.

Sample Procedures for Transition to Standby State and for Transition to Normal State

The DSTC transits to the standby state upon a bus reset. To make the DSTC execute a transfer, it is necessary to make the DSTC first transit from the standby state to the normal state. In the case of not making the DSTC execute a transfer, the power consumption of the system can be reduced by keeping the DSTC in the standby state.

Figure 4-6 shows a sample procedure for making the DSTC transit from the standby state to the normal state and another sample procedure for making the DSTC transit from the normal state to the standby state. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 4-6 Examples of Transition to Standby State and Transition to Normal State



#1 to #5 show the procedure for making the DSTC transit from the standby state to the normal state. Issue a standby release command (write 0x04 to the CMD Register). If the DSTC transits to the normal state, the CMD Register reads 0x00. Check that the CMD Register reads 0x00.

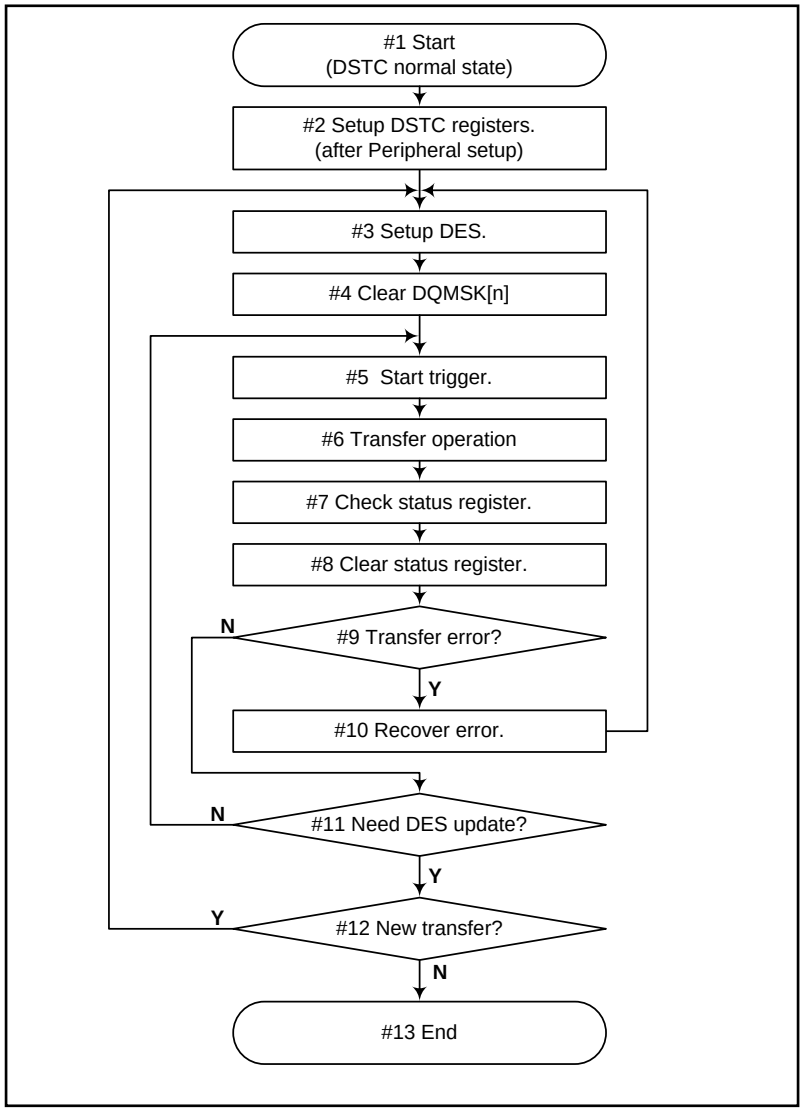
#6 to #10 show the procedure for making the DSTC transit from the normal state to the standby state. Issue a standby transition command (write 0x08 to the CMD Register). If the DSTC transits to the standby state, the CMD Register reads 0x01. Check that the CMD Register reads 0x01.

While the DSTC is executing a transfer, if a standby transition command is issued, the transfer in progress is compulsorily ended and a DES close process for that transfer is executed. After the DES close process has been completed, the DSTC transits to the standby state.

Sample Procedure for Controlling Transfer Operation

Figure 4-7 illustrates a sample procedure for controlling the transfer operation of the DSTC. Numbers in the Figure 4-7 correspond to those used in the explanation after the figure.

Figure 4-7 Sample Procedure for Controlling DSTC Transfer Operation



#1 This sample procedure starts from the point at which the DSTC is in the normal state. If the DSTC is not in the normal state, the following processes cannot be executed.

#2 Initialize the control registers of the DSTC. Set the DESTP Register, the CFG Register, the HWDESP[n] Register and the DREQENB[n] Register to their respective initial values. Write the initial value to the DREQENB[n] Register after completing the setup of the peripheral. The MONERS Register is cleared upon a bus reset. However, after the DSTC has been released from the standby state again, the error record of a previous compulsory end of transfer due to the transition to the standby state may be kept in the MONERS Register. Use the ERCLR Command (write 0x20 to the CMD Register) to clear the MONERS Register.

#3 Build in the CPU the DES area to be used by the DSTC.

#4 The DQMSK[n] Register has been cleared upon a bus reset. However, it may be set by an HW Transfer error, a standby transition command or a source specified in DMSET of the DES. Write 1 to the DQMSKCLR[n] Register to clear the DQMSK[n] Register. If a transfer request signal (DREQ[n]) from a peripheral has been asserted, the clearing of the DQMSK[n] Register by the CPU becomes the transfer start trigger in #5.

#5 Issues a Start Trigger. In an SW Transfer, write the DESP to the SWTR Register via the CPU. While an SW Transfer is in progress, no write access can be made to the SWTR Register. In an HW Transfer, the assertion of the DREQ[n] signal from a peripheral is the Start Trigger.

#6 The DSTC executes a transfer operation according to the DES. The DSTC updates the DES and returns the transfer status.

#7 Check the transfer status of the DSTC via the CPU. In an SW Transfer, read the SWREQ bit and SWST bit in the SWTR Register. In an HW Transfer, read the HWINT[n] Register. Read the MONERS Register to check whether there is a transfer error.

#8 Clear the transfer status of the DSTC via the CPU. In an SW Start transfer, issue an SWCLR command from the CPU (write "0x10" to the CMD Register) to clear the SWST bit in the SWTR Register. In an HW Start transfer, write "1" to the HWINTCLR[n] Register to clear the HWINT[n] Register.

#9, #10 After a transfer error has occurred, issue an ERCLR command (write 0x20 to the CMD Register) to clear the MONERS Register. If a transfer has been interrupted by an error, rebuild the DES. In an HW Transfer, peripheral setup may be necessary.

#11 to #13 If it is not necessary to rebuild the DES, issue the next Start Trigger to start a transfer. If it is necessary to rebuild the DES, rebuild the DES via the CPU, then issue the next Start Trigger to start a transfer.

Procedure to break off a Hardware Transfer of DSTC

This section explains the procedure to break off a hardware transfer of DSTC, by using HW transfer triggered by ADC as an example.

When performing the HW transfer triggered by ADC, setting DES0.DMSET=1 is required at the first build of DES before the transfer starts. In order to break off the HW transfer in the middle of the operation and close the DES, the following procedure can be used.

#1a Write DES0.DV=10 from the CPU

#2a Disable the read skip buffer by writing CFG.RBDIS=1 from the CPU

#3a DSTC waits for a transfer request signal from the ADC. When the transfer request signal is asserted, the DSTC closes the DES (write DES0.DV=00) and sets “1” to DQMSK[n].
This can avoid occurring subsequent DES open errors.

#4a The CPU waits until DES0.DV=00 can be read.

The transfer can be restarted by the following procedure.

#5a Build the DES again, and then write CFG.RBDIS=0 from the CPU

#6a Write DQMSKCLR=1 from the CPU

To close the DES without the step #3a above (Wait for a transfer request from the ADC), follow the procedure below.

#1b Disable the transfer request signal of the ADC
For ADC Scan Conversion Transfer: Write ADCR.SCIE=0
For ADC Priority Conversion Transfer: Write ADCR.PCIE=0

#2b Write DES0.DV=10 from the CPU

#3b Disable the read skip buffer by writing CFG.RBDIS=1 from the CPU

#4b Instruct the SW start from the CPU to the DESP performing the HW transfer
(Write DESP to SWTR register)

#5b DSTC closes the DES (Write DES0.DV=00)
In this case, DQMSK[n]=1 is not set.

#6b The CPU waits until DES0.DV=00 can be read.

The transfer can be restarted by the following procedure.

#7b Build the DES again, and tehn write CFG.RBDIS=0 from the CPU

#8b Write DQMSKCLR[n]=1 from the CPU

#9b Write ADCR.SCIE=1 or ADCR.PCIE=1 from the CPU

Note:

- *When performed the steps #1b-#6b above, it is possible to execute HW transfer request from the ADC and the DES close process at the step #3a, before the SW start instruction from the CPU at step #4b*
  In this case, the SW start instruction at #4b will be notified as a DES open error. This error can be ignored because it doesn’t cause any problem.
  Also, the step #8b is required for restart since DQMSK[n]=1 is set at the step #3a.

5. Registers and Descriptors of DSTC

This section explains the functions of registers of the DSTC and the functions of descriptors.

5.1. Lists of Control Registers and DES
5.2. DESTP Register
5.3. HWDESP[n] Register
5.4. CMD Register
5.5. CFG Register
5.6. SWTR Register
5.7. MONERS Register
5.8. DREQENB[n] Register
5.9. HWINT[n] Register
5.10. HWINTCLR[n] Register
5.11. DQMSK[n] Register
5.12. DQMSKCLR[n] Register
5.13. Descriptor 0 (DES0)
5.14. Descriptor 1 (DES1)
5.15. Descriptor 2 (DES2)
5.16. Descriptor 3 (DES3)
5.17. Descriptor 4 (DES4)
5.18. Descriptor 5 (DES5)
5.19. Descriptor 6 (DES6)

5.1 Lists of Control Registers and DES

This section shows the respective lists of control registers of the DSTC and DES.

Table 5-1 shows a list of the control registers of the DSTC and Table 5-2 a list of DES.

Table 5-1 List of Control Registers of DSTC

Address	Register name	Reference
+0x00	DESTP	5.2
+0x04	HWDESP	5.3
+0x08	CMD	5.4
+0x09	CFG	5.5
+0x0A	SWTR	5.6
+0x0C	MONERS	5.7
+0x10 to +0x2F	DREQENB	5.8
+0x30 to +0x4F	HWINT	5.9
+0x50 to +0x6F	HWINTCLR	5.10
+0x70 to +0x8F	DQMSK	5.11
+0x90 to +0xAF	DQMSKCLR	5.12

Table 5-2 List of DES

Address	Descriptor name	Reference
DESTP+DESP+0x00	DES0	5.13
DESTP+DESP+0x04	DES1	5.14
DESTP+DESP+0x08	DES2	5.15
DESTP+DESP+0x0C	DES3	5.16
DESTP+DESP+0x10 -	DES4	5.17
DESTP+DESP+0x10 -	DES5	5.18
DESTP+DESP+0x10 -	DES6	5.19

5.2 DESTP Register

The DESTP (Descriptor top address) Register sets the start address of the DES area.

Register configuration

Address: +0x00

bit	31	0
Field	DESTP[31:0]	
Attribute	R/W	
Initial value	0x00000000	

Register function

The DSTC refers to and updates the DES located at the address of "DESTP + DESP". Set the DES area in a memory area that is readable and writable. Align the DES area to the word boundary. Always write "00" to the lower 2 bits in the DESTP Register. The DES area must be located within a 4096-word (16 KB) area starting from the DESTP. The DESTP Register cannot be set to a value larger than "0xFFFFFFF0".

Set the DESTP Register when the DSTC is in the normal state and is doing initial settings. The value of this register cannot be modified when the DSTC is executing a transfer. After the value of this register has been modified, the DSTC cannot execute any normal transfer.

bit[31:0] DESTP (Descriptor top address)

Access	Function
Writing	A write access to these bits sets the start address of the DES area.
Reading	A read access to these bits reads the value of these bits.

5.3 HWDESP[n] Register

The HWDESP[n] (Hardware DES pointer) Register sets the DESP of the DES that the DSTC refers to at a transfer request of HW channel n.

Register configuration

Address: +0x04

bit	31	30	29 ... 16	15 ... 8	7 ... 0
Field	Reserved		HWDESP[13:0]	Reserved	CHANNEL[7:0]
Attribute	R	R	R/W	R	R/W
Initial value	0	0	XXXXXXXXXXXX00	00000000	00000000

Register function

Set the HWDESP[n] Register before making an HW transfer request. This register can be accessed only when the DSTC is in the normal state. Settings of an unused HW channel n are not necessary.
The number of the HWDESP[n] Registers corresponds to the number of HW channels. However, there is only one register window that can be seen from the CPU. Access this register as explained below.

- To read the value of the HWDESP[n] Register from the CPU, use an 8-bit (byte) access to write to CHANNEL[7:0] the channel number to be read first. Afterward, read the value of HWDESP[13:0] with a 16-bit (halfword) access.
- To write the value of the HWDESP[n] Register from the CPU, use an 8-bit (byte) access to write to CHANNEL[7:0] the channel number to be written first. Afterward, write a value to HWDESP[13:0] with a 16-bit (halfword) access. If the write access is a 32-bit (word) access, writing a value to CHANNEL[7:0] and writing a value to HWDESP[13:0] can be executed simultaneously.

The DSTC stores the DESP value of the HWDESP[n] Register in HWDESPBUF in Figure 3-5 before using it. If HW Start requests of channel n are made successively, the DSTC uses the DESP value stored in HWDESPBUF, but not the DESP value of the HWDESP[n] Register. Therefore, if the values of the HWDESP[n] Register are modified via the CPU, invalidate the value stored in HWDESPBUF. The DESP value of HWDESPBUF can be invalidated by modifying the value of the RBDIS bit in the CFG Register. For details of the CFG Register, see "5.5 CFG Register".

bit[7:0] CHANNEL[7:0]

Access	Function
Writing	A write access to these bits sets the channel number (n) for the HWDESP[n] to which a read access or a write access is made.
Reading	A read access to these bits reads the value of these bits.

If the DSTC with which a product equipped supports HW-128 channels, always write "0" to CHANNEL[7]. If the DSTC with which a product equipped supports HW-64 channels, always write "00" to CHANNEL[7:6].

bit[29:16] HWDESP[13:0]

Access	Function
Writing	A write access to these bits writes the channel number of HWDESP[n] specified in CHANNEL[7:0].
Reading	A read access to these bits reads the channel number of HWDESP[n] specified in CHANNEL[7:0].

Align DES to the word boundary. Always write "00" to the lower 2 bits in the HWDESP Register. HWDESP cannot be set to a value larger than "0x3FF0".

5.4 CMD Register

The CMD (Command) Register issues a command to the DSTC and reads the state of the DSTC.

Register configuration

Address: +0x08

bit	7	6	5	4	3	2	1	0
Field	CMD[7:0]							
Attribute	W	W	W	W	W	W	R	R
Initial value	0	0	0	0	0	0	0	1

Register function

A command can be issued to the DSTC by writing a value to the CMD (Command) Register. Use an 8-bit (byte) access to write a value to this register. The 16-bit write access and 32-bit write access to this register are ignored. In addition, the state (normal state, standby state, transition state 1 and transition state 2) of the DSTC can be checked by reading this register.

In each state of the DSTC, some commands can be issued and some cannot. For details, see Table 3-13.

bit[7:0] CMD[7:0]

Write value	Command	Process details
0x04	Standby release command	Instructs the DSTC to return from the standby state to the normal state.
0x08	Standby transition command	Instructs the DSTC to transit from the normal state to the standby state. Clears SWTR:SWST to "0". Negates the SWINT interrupt signal. Clears all HWINT[n] Registers. Negates the HWINT[n] interrupt signal. Negates the ERINT interrupt signal. Set all DQMSK[n] Registers to "1".
0x10	SWCLR command	Clears SWTR:SWST to "0". Negates the SWINT interrupt signal.
0x20	ERCLR command	Clears MONERS:EST to "0". Negates the ERINT interrupt signal. Clears MONERS:DER to "0". Clears MONERS:ESTOP to "0".
0x80	MKCLR command	Clears all DQMSK[n] Registers to "0". (This command is ignored in the transition state 2.)
Value other than those listed above	Writing a value other than those listed above to CMD[7:0] is prohibited. (If a value other than those listed above is written to CMD[7:0] via a write access such as an RMW access, such write access may be ignored.)	

Read value	State of DSTC
0x00	Indicates that the DSTC is in the normal state.
0x01	Indicates that the DSTC is in the standby state. (Initial value)
0x02	Indicates that the DSTC is in the transition state 1 (transiting from the standby state to the normal state).
0x03	Indicates that the DSTC is in the transition state 2 (transiting from the normal state to the standby state).
Value other than those listed above	The CMD Register never reads any value other than the above.

5.5 CFG Register

The CFG (configuration) Register sets operation functions of the DSTC.

Register configuration

Address: +0x09

bit	15	14	13	12	11	10	9	8
Field	Reserved	SWPR[2:0]			ESTE	RBDIS	ERINTE	SWINTE
Attribute	R	R/W			R/W	R/W	R/W	R/W
Initial value	0	100			0	0	0	0

Register function

The CFG (configuration) Register sets operation functions of the DSTC. Use an 8-bit (byte) access to write a value to this register. The 16-bit write access and 32-bit write access to this register are ignored. When the DSTC is in the normal state, no write access can be made to this register.

bit[8] SWINTE (Software interrupt enable)

Access	Function
Writing "0"	Disables the SWINT interrupt. (Initial value) If SWTR:SWST has been set to "1", the DSTC does not generate the SWINT interrupt.
Writing "1"	Enables the SWINT interrupt. If SWTR:SWST has been set to "1", the DSTC generates the SWINT interrupt.
Reading	A read access to this bit reads the value of this bit.

bit[9] ERINTE (Error interrupt enable)

Access	Function
Writing "0"	Disables the ERINT interrupt. (Initial value) If MONERS:EST has been set to "001", "010", "100" or "101", the DSTC does not generate the ERINT interrupt.
Writing "1"	Enables the ERINT interrupt. If MONERS:EST has been set to "001", "010", "100" or "101", the DSTC generates the ERINT interrupt.
Reading	A read access to this bit reads the value of this bit.

MONERS:EST=”011” is transfer compulsory stop error by standby transition command. In this case, ERINT interrupt is not asserted. For details, see "3.2.9 Standby Function"

bit[10] RBDIS (Read skip buffer disable)

Access	Function
Writing "0"	Enables the read skip buffer function of the DES. (Initial value) The HWDESP[n] reference skip function of HWDESPBUF is enabled.
Writing "1"	Disables the read skip buffer function. The HWDESP[n] reference skip function of HWDESPBUF is disabled.
Reading	A read access to this bit reads the value of this bit.

If the RBDIS bit is set to "0", the read skip buffer function and HWDESPBUF function shown in Figure 3-5 are enabled. Accordingly, the DSTC skips referring to the DES and HWDESP[n] Register on the memory and, in turn, the processing speed of the DSTC increases. Nonetheless, the DSTC may not be able to recognize a change in the value of DV of DES0 by the CPU or the modification of the value of the HWDESP[n] Register. Therefore, if the value of DV of DES0 has been changed or the value of the

HWDESP[n] Register has been modified, write "1" to the RBDIS bit. If the RBDIS bit is set to "1", the DSTC does not use the read skip buffer function or the HWDESPBUF function, but it operates referring to the DES value on the memory and the value of the HWDESP[n] Register. After the DSTC has executed processes with updated values of the DES and the HWDESP[n] Register, the buffer function can be enabled again by writing "0" to the RBDIS Register.

bit[11] ESTE (Error stop enable)

Access	Function
Writing "0"	The DSTC does not enter the error stop state even when a transfer error occurs. (Initial value) If there is another transfer request, the DSTC starts the transfer for that request.
Writing "1"	The DSTC enters the error stop state when a transfer error occurs. If there is another transfer request, the DSTC holds the start of the transfer for that request.
Reading	A read access to this bit reads the value of this bit.

bit[12:14] SWPR[2:0] (Software transfer priority)

In the arbitration of Arbiter 2, if the SW transfer request conflicts with the HW transfer request, Arbiter 2 specifies the probability of the SW transfer acquiring the transfer right. The value of the SWPR bits can be modified even when the DSTC is executing a transfer. After the value of the SWPR bits in the CFG Register has been modified, it is applied from the next SW Start Trigger.

Access	Function
Writing "000"	Sets the priority of the SW transfer to the highest priority. (If an SW transfer request is made while an HW transfer is in progress, the SW transfer starts after the HW transfer has ended.)
Writing "001"	Sets the probability of the SW transfer acquiring the transfer right to 1/2.
Writing "010"	Sets the probability of the SW transfer acquiring the transfer right to 1/3.
Writing "011"	Sets the probability of the SW transfer acquiring the transfer right to 1/7.
Writing "100"	Sets the probability of the SW transfer acquiring the transfer right to 1/15. (Initial value)
Writing "101"	Sets the probability of the SW transfer acquiring the transfer right to 1/31.
Writing "110"	Sets the probability of the SW transfer acquiring the transfer right to 1/63.
Writing "111"	Sets the priority of the SW transfer to the lowest priority. (The SW transfer starts only when there is no HW transfer request.)
Reading	A read access to these bits reads the value of these bits.

bit[15] Reserved

Write "0" in a write access to this bit. In a read access to this bit, "0" is read out.

5.6 SWTR Register

The SWTR (Software trigger) Register issues the Start Trigger of the SW Start transfer.

Register configuration

Address: +0x0A

bit	31	30	29 ... 16
Field	SWST	SWREQ	SWDESP[13:0]
Attribute	R	R	R/W
Initial value	0	0	00000000000000

Register function

The SWTR (Software trigger) Register issues the Start Trigger of the SW Transfer if a write access to this register is made. Use the 16-bit (halfword) access to write a value to this register. The 32-bit (word) write access is ignored. If the SW Start instruction has been executed, and that transfer has not ended (SWREQ ≠ 0) or the DSTC is not in the normal state (CMD ≠ 00) or the DSTC is in the error stop state (ESTOP ≠ 0), the DSTC ignores the write access from the CPU and also the new SW Start transfer request.

bit[29:16] SWDESP[13:0] (Software DES pointer)

Write the value of DESP of the DES to be started. The DSTC transfers data according to the DES area of DESTP+SWDESP. If a Chain Start is executed during the SW Transfer, SWDESP is updated by the DSTC to the value of DESP used in the Chain Start. Align DES to the word boundary. Always write "00" to the lower 2 bits in SWDESP. SWDESP cannot be set to a value larger than "0x3FF0".

Access	Function
Writing	A write access to these bits specifies the DESP for transfer to be started by the SW Start.
Reading	These bits indicate the DESP for SW Start transfer that is in progress or that has been ended.

bit[30] SWREQ (Software request)

The SWREQ bit is a read-only bit indicating whether the execution of the SW Transfer is pending, or the SW Transfer as well as the Chain Start transfer are being executed. The value written to this bit is ignored. A write access (Start Trigger) to the SWTR Register sets the SWREQ bit to "1". If the SW Transfer ends normally, abnormally, or is waiting for a Start Trigger, SWREQ is reset to "0".

Access	Function
Writing	The value written to this bit is meaningless.
Reading "0"	Indicate that either the SW Transfer is not requested or has been ended.
Reading "1"	Indicate that either the SW Transfer is pending or the DSTC is performing the SW Transfer.

bit[31] SWST (Software status)

The SWST bit is a read-only bit for sending the SW Transfer end notification to the CPU. The interrupt flag set is specified in the DES of the SW Transfer. If the SW Transfer ends normally, SWST is set to "1". SWST is cleared to "0" by the SWCLR command, the standby transition command or the write access to the SWTR Register. In the case of (CFG:SWINTE = 1)&&(SWTR:SWST == 1), the SWINT interrupt signal is asserted.

Access	Function
Writing	The value written to this bit is meaningless.
Reading "0"	Indicates that the SW Transfer has not ended normally.
Reading "1"	Indicates that the SW Transfer has ended normally.

5.7 MONERS Register

The MONERS Register shows details of a transfer error that has occurred.

Register configuration

Address: +0x0C

bit	31	30	29	28	27	26	25	24
Field	Reserved		EDESP[13:8]					
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	EDESP[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	0	0

bit	15	14	13	12	11	10	9	8
Field	ECH[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	Reserved	EHS	Reserved	ESTOP	DER	EST[2:0]		
Attribute	R	R	R	R	R	R		
Initial value	X	X	X	0	0	000		

Register function

The MONERS Register is a read-only register. The write access to this register is ignored. If a transfer error occurs, details of that error are recorded in the MONERS Register. Details of an error can be checked by referring to the MONERS Register. If MONER:EST indicates that an error has occurred, the DSTC can generate an ERINT interrupt. If the ERINTE bit is set to "1", the ERINT interrupt signal for the NVIC is asserted. The values of the MONERS Register and the ERINT interrupt can be cleared by issuing an ERCLR command to the CMD Register.
For details of the contents of MONERS register, see "3.2.8 MONERS Register "

bit[2:0] EST[2:0] (Error status)

EST indicates details of an error that has occurred. In the case of EST ≠ 000, even if a new transfer error occurs, DESP keeps details of the previous transfer error. ESTOP is cleared to "000" by the ERCLR command.

Access	Function
Writing	Writing a value to DESP causes no operation to be executed.
Reading "000"	No error has occurred. (initial value)
Reading "001"	Source access error
Reading "010"	Destination access error
Reading "011"	Transfer compulsory stop error by standby transition command.
Reading "100"	DES access error
Reading "101"	DES open error
Reading a value other than those listed above	Undefined

bit[3] DER (Double error)

The DER bit indicates whether a double error has occurred. With the EST[2:0] bits set to a value other than "000" and the DER bit set to "0", if a new transfer error occurs, the DER bit is set to "1". The DER bit is cleared to "0" by the ERCLR command.

Access	Function
Writing	Writing a value to DESP causes no operation to be executed.
Reading "0"	Indicates that no double error has occurred. (Initial value)
Reading "1"	Indicates that a double error has occurred.

bit[4] ESTOP (Error stop)

The ESTOP bit indicates that the DSTC is in the error stop state. With CFG:ESTE set to "1", if a transfer error occurs, the ESTOP bit is set to "1". In the error stop state, the transfer start of the DSTC is held. The ESTOP bit is cleared to "0" by the ERCLR command. If this bit is cleared to "0", the transfer held starts. If this bit is "1", a write access to the SWTR Register is ignored.

Access	Function
Writing	Writing a value to DESP causes no operation to be executed.
Reading "0"	Indicates that the DSTC is not in the error stop state. (Initial value)
Reading "1"	Indicates that the DSTC is in the error stop state.

bit[5] Reserved

The read value is indeterminate. The value written to this bit is ignored.

bit[6] EHS (Error hardware software)

The EHS bit indicates whether the DES that has caused an error has been started by the HW Start or by the SW Start. In the case of EST ≠ 000, even if a new transfer error occurs, DESP keeps details of the previous transfer error. In the case of EST = 000, the value of DESP is undefined.

Access	Function
Writing	Writing a value to this bit does not cause any operation to be executed.
Reading "0"	An error has occurred in a transfer started by the SW Start or by the Chain Start in that SW Start.
Reading "1"	An error has occurred in a transfer started by the HW Start or by the Chain Start in that HW Start.

bit[7] Reserved

The read value is indeterminate. The value written to this bit is ignored.

bit[15:8] ECH[7:0] (Error hardware channel)

If the EST[2:0] bits are not "000" and the EHS bit is "1", the ECH[7:0] bits indicate the number of the HW Start channel that has caused an error. In the case of EST ≠ 000, even if a new transfer error occurs, the ECH[7:0] bits keep the HW Start channel number of the previous transfer error. If the EST[2:0] bits are "000" or the EHS bit is "0", the value of the ECH[7:0] bits is indeterminate.

Access	Function
Writing	Writing a value to these bits does not cause any operation to be executed.
Reading	The ECH[7:0] bits indicate the HW channel number if the DES that has caused a transfer error was started by the HW Start.

bit[29:16] EDESP [13:0](Error DES pointer)

The EDESP[13:0] bits indicate the DESP of the DES that has caused a transfer error. In the case of EST ≠ 000, even if a new transfer error occurs, the EDESP[13:0] bits keep the DESP of the DES of the previous transfer error. In the case of EST = 000, the value of the EDESP[13:0] bits is indeterminate. The EDESP[1:0] bits always read "0".

Access	Function
Writing	Writing a value to these bits does not cause any operation to be executed.
Reading	The EDESP[13:0] bits indicate the DESP of the DES that has caused a transfer error.

bit[31:30] Reserved

The read value is "00". The value written to these bits is ignored.

5.8 DREQENB[n] Register

The DREQENB[n] Register determines whether HW channel n is used.

Register configuration

Address		
0x10	Field	DREQENB[31:0]
+0x14	Field	DREQENB[63:32]
+0x18	Field	DREQENB[95:64]
+0x1C	Field	DREQENB[127:96]
+0x20	Field	DREQENB[159:128]
+0x24	Field	DREQENB[191:160]
+0x28	Field	DREQENB[223:192]
+0x2C	Field	DREQENB[255:224]
Attribute (applicable to all areas)		R/W
Initial value (applicable to all areas)		0x00000000

Register function

The DREQENB[n] (DMA request enable) Register determines whether HW channel n is used in the initial settings. When the DSTC is in the normal state, write access can be made to this register.

Write "1" to the DREQENB[n] Register to use HW channel n. Write "0" to the DREQENB[n] Register to not use HW channel n. If the DREQENB[n] Register is "0", the interrupt signal or transfer request signal (DREQ[n]) of a peripheral connected to the DSTC is ignored. The value of the DREQENB[n] Register cannot be modified by the DSTC.

In case that the peripheral interrupts are shared with transfer request to DSTC, the value of the DREQENB[n] Register determines which of the interrupt signal from a peripheral and HWINT[n] from the DSTC is selected as an interrupt signal connected to the NVIC. For its details, see "2 DSTC Operations Overview and DSTC System Configuration".

bit[255:0] DREQENB[255:0] (DMA request enable)

Access	Function
Writing "0"	Disables the DREQ signal from the peripheral. (Initial value)
Writing "1"	Enables the DREQ signal from the peripheral.
Reading	A read access to these bits reads the value of these bits.

If the DSTC installed in a product supports HW-128 channels, the DREQENB[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the DREQENB[255:64] bits are a reserved area whose value is fixed at "0".

5.9 HWINT[n] Register

The HWINT[n] Register sends the HW Transfer end notification to the CPU.

Register configuration

Address		
+0x30	Field	HWINT[31:0]
+0x34	Field	HWINT[63:32]
+0x38	Field	HWINT[95:64]
+0x3C	Field	HWINT[127:96]
+0x40	Field	HWINT[159:128]
+0x44	Field	HWINT[191:160]
+0x48	Field	HWINT[223:192]
+0x4C	Field	HWINT[255:224]
Attribute (applicable to all areas)		R
Initial value (applicable to all areas)		0x0000000000000000

Register function

The HWINT[n] (Hardware transfer interrupt) Register is a read-only register for sending the HW Transfer end notification to the CPU. The write access to this register is ignored.

If the interrupt flag set is specified in the DES started by the HW Start, or CHRS in the DES started by the Chain Start after the DES started by the HW Start, the HWINT[n] Register is set to "1". The HWINT[n] Register can be cleared to "0" by writing "1" to the HWINTCLR[n] Register or issuing the standby transition command. If the HWINT[n] Register is set to "1", the HW transfer completion interrupt signal from the DSTC (HWINT[n]) for the NVIC is asserted.

bit[255:0] HWINT[255:0] (Hardware transfer interrupt)

Access	Function
Writing	Causes no operation to be executed.
Reading "0"	Indicates that the HW Transfer started has not ended normally.
Reading "1"	Indicates that the HW Transfer started has ended normally.

If the DSTC installed in a product supports HW-128 channels, the HWINT[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the HWINT[255:64] bits are a reserved area whose value is fixed at "0".

5.10 HWINTCLR[n] Register

The HWINTCLR[n] Register is a register for clearing the HWINT[n] Register.

Register configuration

Address		
+0x50	Field	HWINTCLR[31:0]
+0x54	Field	HWINTCLR[63:32]
+0x58	Field	HWINTCLR[95:64]
+0x5C	Field	HWINTCLR[127:96]
+0x60	Field	HWINTCLR[159:128]
+0x64	Field	HWINTCLR[191:160]
+0x68	Field	HWINTCLR[223:192]
+0x6C	Field	HWINTCLR[255:224]
Attribute (applicable to all areas)		W
Initial value (applicable to all areas)		0x0000000000000000

Register function

The HWINTCLR[n] Register is a write-only register for clearing the HWINT[n] Register from the CPU. When the DSTC is not in the normal state, no write access can be made to this register.

Writing "1" to this register can clear the HWINT[n] Register to "0". Writing "0" to this register is ignored. The read value is always "0".

bit[255:0] HWINTCLR[255:0] (Hardware transfer interrupt clear)

Access	Function
Writing "0"	Causes no operation to be executed.
Writing "1"	Clears the HWINT[n] Register to "0".
Reading	All bits in this register always read "0".

If the DSTC installed in a product supports HW-128 channels, the HWINTCLR[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the HWINTCLR[255:64] bits are a reserved area whose value is fixed at "0".

5.11 DQMSK[n] Register

The DQMSK[n] Register indicates whether the HW Start transfer request is being suppressed.

Register configuration

Address		
+0x70	Field	DQMSK[31:0]
+0x74	Field	DQMSK[63:32]
+0x78	Field	DQMSK[95:64]
+0x7C	Field	DQMSK[127:96]
+0x80	Field	DQMSK[159:128]
+0x84	Field	DQMSK[191:160]
+0x88	Field	DQMSK[223:192]
+0x8C	Field	DQMSK[255:224]
Attribute (applicable to all areas)		R
Initial value (applicable to all areas)		0x00000000

Register function

The DQMSK[n] Register is a read-only register. The write access to this register is ignored. That this register is "1" indicates the HW Start transfer request (DREQ[n]) to the DSTC is being suppressed. If one of the following conditions is met, the DSTC sets DQMSK[n] to "1" and suppresses transfer requests.

- A transfer error has occurred at a transfer on HW channel n.
- The CPU has issued a standby transition command to the CMD Register.
- DMSET in the DES for the transfer on HW channel n is "1" and the DSTC has executed a DES close process.

If one of the following conditions is met, the DSTC clears DQMSK[n] to "0" and releases the suppression of transfer requests.

- "1" has been written to the DQMSKCLR[n] Register.
- The CPU has issued a standby transition command to the CMD Register.

bit[255:0] DQMSK[255:0] (DMA request mask)

Access	Function
Writing	Causes no operation to be executed.
Reading "0"	Indicates that the DREQ[n] signal from the peripheral is not being suppressed.
Reading "1"	Indicates that the DREQ[n] signal from the peripheral is being suppressed.

If the DSTC installed in a product supports HW-128 channels, the DQMSK[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the DQMSK[255:64] bits are a reserved area whose value is fixed at "0".

5.12 DQMSKCLR[n] Register

The DQMSKCLR[n] Register is a register for clearing the DQMSK[n] Register.

Register configuration

Address		
+0x90	Field	DQMSKCLR[31:0]
+0x94	Field	DQMSKCLR[63:32]
+0x98	Field	DQMSKCLR[95:64]
+0x9C	Field	DQMSKCLR[127:96]
+0xA0	Field	DQMSKCLR[159:128]
+0xA4	Field	DQMSKCLR[191:160]
+0xA8	Field	DQMSKCLR[223:192]
+0xAC	Field	DQMSKCLR[255:224]
Attribute (applicable to all areas)		W
Initial value (applicable to all areas)		0x00000000

Register function

The DQMSKCLR[n] Register is a write-only register. When the DSTC is in the standby state, no write access can be made to this register.

If "1" is written to this register, the DQMSK[n] Register is cleared to "0". Clearing the DQMSK[n] Register makes a suppressed HW transfer start immediately. Therefore, complete the setup of the peripheral for that suppressed transfer and the setup of the DES before clearing the DQMSK[n] Register.

bit[255:0] DQMSKCLR[255:0] (DMA request mask clear)

Access	Function
Writing "0"	No operation is executed.
Writing "1"	Clears the DQMSK[n] Register to "0".
Reading	All bits in this register always read "0".

If the DSTC installed in a product supports HW-128 channels, the DQMSKCLR[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the DQMSKCLR[255:64] bits are a reserved area whose value is fixed at "0".

5.13 Descriptor 0 (DES0)

This section explains details of Descriptor 0 (DES0). DES0 sets the basic settings of a transfer.

Descriptor configuration

Address: DESTP + DESP + 0x00

bit	31	30	29	28	27	26	25	24
Field	PCHK[3:0]				Reserved		ACK[1:0]	
C attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D attribute	R	R	R	R	R	R	R	R

bit	23	22	21	20	19	18	17	16
Field	CHLK	DMSET	CHRS[5:0]					
C attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D attribute	R	R	R	R	R	R	R	R

bit	15	14	13	12	11	10	9	8
Field	DAC[1:0]			SAC[2:0]			TW	
C attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D attribute	R	R	R	R	R	R	R	R

bit	7	6	5	4	3	2	1	0
Field	ORL[2:0]			MODE	ST[1:0]		DV[1:0]	
C attribute	R/W	R/W	R/W	R/W	R	R	R/W	R/W
D attribute	R	R	R	R	W	W	R/W	R/W

*Symbols for DES area:
The row C attribute in the table above indicates the attribute of the R/W access from the CPU.
The row D attribute in the table above indicates the attribute of the R/W access from the DSTC.
The DES area is built on the memory. As all initial values are indeterminate, they are omitted.

Descriptor function

bit[1:0] DV[1:0] (Descriptor valid)

The DV[1:0] bits specify which of the CPU and the DSTC the ownership of the DES belongs to. The DV[1:0] bits also specify whether a transfer is executed after the DES open process and whether the DES close process is to be executed. In the DES close process, the DV[1:0] bits are updated to "00" by the DSTC.

If DV[1] is set to "1", specific restrictions on reloading the transfer counter and transfer address are added. If DV[1] is set to "1" and the settings of the DES make the values of DES1, DES2 and DES3 not return to their respective values, the DSTC notifies of the CPU a DES open error.

Value	Function
00	The owner is the CPU. No transfer is executed. No DES close process is executed. (If the DSTC read this value, the DSTC notifies a DES open error.)
01	The owner is the DSTC. A transfer is executed. The DES close process is executed.
10	The owner is the DSTC. No transfer is executed. The DES close process is executed.
11	The owner is the DSTC. A transfer is executed. No DES close process is executed.

bit[3:2] ST[1:0] (Transfer status)

After the transfer specified in a DES has ended, in a DES close process, the DSTC writes the end status value to the ST[1:0] bits. If the DSTC does not execute the DES close process, the value initially set by the CPU remains in the ST[1:0] bits. At the start of a transfer, the value in this area has no effect on the operation of the DSTC.

If a transfer ends abnormally due to a DES access error or a DES open error, the DSTC does not execute a DES close process and does not write data to the ST[1:0] bits either because the DSTC cannot access the DES area. Such error notifications are executed according to the MONERS Register of the DSTC but not ST of the DES.

Value	Function
00	The transfer has ended normally.
01	The transfer has ended abnormally because an error occurred at a transfer source access.
10	The transfer has ended abnormally because an error occurred at a transfer destination access.
11	The transfer has ended abnormally because a transfer compulsory stopped by standby transition command is issued from CPU.

bit[4] MODE (Transfer mode)

The MODE bit selects a transfer mode. In mode 0, the DSTC executes transfers for IIN times of data transfers for one Start Trigger. In mode 1, the DSTC executes transfers one transfer for one Start Trigger. To make the DSTC execute an HW Transfer at a Start Trigger from a peripheral that has to have to a handshake with the DSTC at every data transfer, use mode 1. The DSTC does not modify the value of this area.

Value	Function
0	The transfer is to be executed in mode 0.
1	The transfer is to be executed in mode 1.

bit[7:5] ORL[2:0] (Outer reload)

The ORL[2:0] specify whether OuterReload for the transfer number counter of DES1 (ORM/IRM/IIN), the transfer source address of DES2 (SA) and the transfer destination address (DA) of DES3 is executed after ORM×IIN times of transfer have ended. The DES size is determined by the value of ORL. The respective relative addresses from the DESP of DES4 to DES6 vary according to the value of the ORL[2:0] bits. The DSTC does not modify the value of this area.

bit	Value	Function
ORL[0]	0	OuterReload for DES1 is not to be executed. DES4 area is not required.
	1	OuterReload for DES1 is to be executed. DES4 area is required.
ORL[1]	0	OuterReload for DES2 is not to be executed. DES5 area is not required.
	1	OuterReload for DES2 is to be executed. DES5 area is required.
ORL[2]	0	OuterReload for DES3 is not to be executed. DES6 area is not required.
	1	OuterReload for DES3 is to be executed. DES6 area is required.

bit[9:8] TW[1:0] (Transfer width)

The TW[1:0] specify the data width in a single transfer. The DSTC does not modify the value of this area.

Value	Function
00	8 bits (byte)
01	16 bits (halfword)
10	32 bits (word)
11	Setting prohibited (If the DSTC reads "11" from TW, notifies a DES open error.)

bit[12:10] SAC[2:0] (Source Address Control)

The SAC[2:0] specify the method of updating the transfer source address during a transfer. The DSTC does not modify the value of this area. The setting is that DES0.DV[1]=1 and DES2 is need to rebuild (DES2 is not returned to the start value), caused to notify a DES open error from the DSTC.

Value	Function
000	The address is increased by TW×1 at every transfer without InnerReload.
001	The address is increased by TW×1 at every transfer with InnerReload.
010	The address is increased by TW×2 at every transfer without InnerReload.
011	The address is increased by TW×2 at every transfer with InnerReload.
100	The address is increased by TW×4 at every transfer without InnerReload.
101	The transfer address remains unchanged during a transfer.
110	The address is decreased by TW×1 at every transfer without InnerReload.
111	The address is decreased by TW×1 at every transfer with InnerReload.

bit[15:13] DAC[2:0] (Destination Address Control)

The DAC[2:0] specify the method of updating the transfer destination address during a transfer. The DSTC does not modify the value of this area. The setting is that DES0.DV[1]=1 and DES3 is need to rebuild (DES3 is not returned to the start value), caused to notify a DES open error from the DSTC.

Value	Function
000	The address is increased by TW×1 at every transfer without InnerReload.
001	The address is increased by TW×1 at every transfer with InnerReload.
010	The address is increased by TW×2 at every transfer without InnerReload.
011	The address is increased by TW×2 at every transfer with InnerReload.
100	The address is increased by TW×4 at every transfer without InnerReload.
101	The transfer address remains unchanged during a transfer.
110	The address is decreased by TW×1 at every transfer without InnerReload.
111	The address is decreased by TW×1 at every transfer with InnerReload.

bit[21:16] CHRS[5:0] (Chain & Return Status)

The CHRS[5:0] bits specify the process to be executed after specified times of transfer have been executed.

bit	Condition for selection	Value	Function
CHRS[1:0]	IRM ≠ 1 ORM: ignore	00	No interrupt flag is set. There is no Chain Start.
		01	An interrupt flag has been set. There is no Chain Start.
		10	No interrupt flag is set. There is a Chain Start in the succeeding DES.
		11	No interrupt flag is set. There is a Chain Start in the current DES.
CHRS[3:2]	IRM = 1 ORM ≠ 1	00	No interrupt flag is set. There is no Chain Start.
		01	An interrupt flag has been set. There is no Chain Start.
		10	No interrupt flag is set. There is a Chain Start in the succeeding DES.
		11	No interrupt flag is set. There is a Chain Start in the current DES.
CHRS[5:4]	IRM = 1 ORM = 1	00	No interrupt flag is set. There is no Chain Start.
		01	An interrupt flag has been set. There is no Chain Start.
		10	No interrupt flag is set. There is a Chain Start in the succeeding DES.
		11	11: Setting prohibited (A DES open error occurs.)

The status of the transfer number counter determines which of CHRS[5:4], CHRS[3:2] and CHRS[1:0] the DSTC follows when executing the next process after the current DES. If there is an interrupt flag set instruction, an SW Start Trigger, and a Chain Start Trigger after that SW Start Trigger set the SWST bit to the SWTR register to 1. An HW Start Trigger, and a Chain Start Trigger after that HW Start Trigger set the HWINT[n] register to 1. When MODE is 0, as CHRS[1:0] is meaningless, write 00 to it. If MODE is 0 and CHRS[1:0] is not 00, a DES open error occurs. The DSTC does not modify the value of this area.

bit[22] DMSET (DREQ Mask Set)

For a DES to be started by the HW Start directly from channel n of a peripheral, and a DES to be started by the Chain Start from the DES mentioned before, with the DMSET bit set to 1, if the DES close process is not executed, the bit corresponding to that DES in the DQMSK[n] Register is set to 1. The DSTC does not modify the value of this area.

Value	Function
0	The DQMSK[n] Register is not set to 1 when the DES close process for an HW Transfer is executed.
1	The DQMSK[n] Register is set to 1 when the DES close process for an HW Transfer is executed.

bit[23] CHLK (Chain Lock)

The CHLK bit specifies whether to execute the next transfer started by the Chain Start immediately after the current transfer (Chain Lock) or to enable other transfers to be executed before the next transfer started by the Chain Start. With the CHLK bit set to 1, if any of CHRS[5], CHRS[3] and CHRS[1] is not 1 (Chain Start selected), a DES open error occurs. The DSTC does not modify the value of this area.

Value	Function
0	After the current transfer, other transfers can be executed before the Chain Start transfer.
1	The Chain Start transfer is executed immediately after the current transfer.

bit[25:24] ACK[1:0] (Acknowledge)

The ACK[1:0] bits set the value for adjusting the timing of DSTC outputting the DMA transfer request acknowledge signal to a peripheral device when the HW Transfer is used. If the HW Transfer is used, set the ACK[1:0] bits to "01" for a DES to be directly started by the HW Start from a peripheral device. For the DES started by the Chain Start from the HW Transfer, the DES used in the SW Transfer, and the DES started by the Chain Start from the SW Transfer, set the ACK[1:0] bits to "00". The DSTC does not modify the value of this area.

Value	Function
00	The DSTC does not output the DMA transfer acknowledge signal to the peripheral connected to the DSTC.
01	The DSTC outputs the DMA transfer acknowledge signal to the peripheral connected to the DSTC.
10, 11	Reserved

bit[27:26] Reserved

Write 00 to this area. If 00 is not written to this area, the DSTC notifies the CPU of a DES open error. The DSTC does not modify the value of this area.

bit[31:28] PCHK[3:0] (Parity Check)

The PCHK[3:0] bits set the parity (to be called "equation" below) of the DES0 area.
PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4])
The CPU calculates the parity value while building the DES. The DSTC checks the consistency between PCHK[3:0] and the value of DES0 area. If a parity error occurs, the DSTC notifies the CPU of a DES open error. The DSTC does not modify the value of this area. The operation target of PCHK[3:0] is the area of DES0[27:4], which the DSTC does not modify. A change in the value of DES0[3:0] does not affect the value of PCHK[3:0].

5.14 Descriptor 1 (DES1)

This section explains details of Descriptor 1 (DES1). DES1 sets the number of transfers. The configuration and functions of DES1 area in mode 0 (DES0.MODE=0) are different from those in mode 1 (DES0.MODE=1).

Descriptor configuration (in mode 0)

Address: DESTP + DESP + 0x04

bit	31	16	15	0
Field	ORM[15:0]		IIN[15:0]	
C attribute	R/W		R/W	
D attribute	R/W		R/W	

Descriptor function (in mode 0)

bit[15:0] IIN[15:0] (Inner loop initial)

The IIN[15:0] bits specify the initial value of the inner loop counter in the transfer number counter. They can be set to a value in the range of "1" to "65536" inclusive. Setting the IIN[15:0] bits to "0x0000" is equivalent to setting them to "65536". The DSTC imports the value of IIN and uses it as the internal loop counter remain value (IRM). Therefore, in mode 0, it is not necessary to specify the IRM.

The DSTC does not modify the value of this area during a transfer. If OuterReload of DES1 is enabled (ORL[0] = 1), the DSTC copies the value of DES[15:0] to IIN.

bit[31:16] ORM[15:0] (Outer loop remain)

The ORM[15:0] bits specify the remain value of the outer loop counter in the transfer number counter. They can be set to a value in the range of "1" to "65536" inclusive. Setting the ORM[15:0] bits to "0x0000" is equivalent to setting them to "65536".

The DSTC decreases the value of ORM before writing back it to the DES. The DSTC stores "0x0001" in ORM at the end of a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES4[31:16] to ORM after the final transfer. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the ORM[15:0] bits via the CPU.

In mode 0, set the number of transfers to a value within the following range. If the value is out of the range, the DSTC notifies a DES open error to the CPU.

If ORM = 65536, IIN < 0x2000
If ORM ≥ 0x8000, IIN < 0x4000
If ORM ≥ 0x4000, IIN < 0x8000
If ORM ≥ 0x2000, IIN < 65536

(If ORM is smaller than 0x2000, there is no limit on the value of IIN.)

The setting is that DES0.DV[1]=1 and DES1 is need to rebuild (DES1 is not returned to the start value), caused to notify a DES open error from the DSTC.

Descriptor configuration (in mode 1)

Address: DESTP + DESP + 0x04

bit	31	16	15	8	7	0
Field	ORM[15:0]		IRM[7:0]		IIN[7:0]	
C attribute	R/W		R/W		R/W	
D attribute	R/W		R/W		R/W	

Descriptor function (in mode 1)

bit[7:0] IIN[7:0] (Inner loop initial)

The IIN[15:0] bits specify the initial value of the inner loop counter in the transfer number counter. They can be set to a value in the range of "1" to "256" inclusive. Setting the IIN[7:0] bits to "0x00" is equivalent to setting them to "256".

The DSTC does not modify the value of this area during a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES[7:0] to the IIN[7:0] bits after the final transfer.

bit[15:8] IRM[7:0] (Inner loop remain)

The IRM[7:0] bits specify the remain value of the inner loop counter in the transfer number counter. Set the IRM[7:0] bits to the same value as the IIN[7:0] bits.

The DSTC decreases the value of the IRM[7:0] bits before writing back it to the DES. The DSTC stores "0x01" in IRM at the end of a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES4[15:8] to the IRM[7:0] bits after the final transfer. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the IRM[7:0] bits via the CPU. If the DSTC detects that the value of the IRM[7:0] bits is larger than the value of the IIN[7:0] bits, it notifies the system of a DES open error.

bit[31:16] ORM[15:0] (Outer loop remain)

The ORM[15:0] bits specify the remain value of the outer loop counter in the transfer number counter. They can be set to a value in the range of "1" to "65536" inclusive. Setting the ORM[15:0] bits to "0x0000" is equivalent to setting them to "65536".

The DSTC decreases the value of the ORM[15:0] bits before writing back it to the DES. The DSTC stores "0x0001" in the ORM[15:0] bits at the end of a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES4[31:16] to the IRM[7:0] bits after the final transfer. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the ORM[15:0] bits via the CPU.

The setting is that DES0.DV[1]=1 and DES1 is need to rebuild (DES1 is not returned to the start value), caused to notify a DES open error from the DSTC.

5.15 Descriptor 2 (DES2)

This section explains details of Descriptor 2 (DES2).

DES2 Descriptor configuration

Address: DESTP + DESP + 0x08

bit	31	0
Field	SA[31:0]	
C attribute	R/W	
D attribute	R/W	

DES2 Descriptor function

bit[31:0] SA[31:0] (Source address)

The SA[31:0] bits set the transfer source address. The SA[31:0] bits cannot be set to a value unaligned to the data size specified in DES0:TW. The DSTC updates the value of DES3 when writing back values to the DES. If OuterReload of DES2 is enabled (ORL[1] = 1), the DSTC copies the value of DES5 to DES2. Pay attention to the value at the end of a transfer if InnerReload/OuterReload of the transfer source address is not enabled. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the SA[31:0] bits via the CPU.

5.16 Descriptor 3 (DES3)

This section explains details of Descriptor 3 (DES3).

DES3 Descriptor configuration

Address: DESTP + DESP + 0x0C

bit	31	0
Field	DA[31:0]	
C attribute	R/W	
D attribute	R/W	

DES3 Descriptor function

bit[31:0] DA[31:0] (Destination Address)

The DA[31:0] bits set the transfer destination address. The DA[31:0] bits cannot be set to a value unaligned to the data size specified in DES0:TW. The DSTC updates the value of DES3 when writing back values to the DES. If OuterReload of DES3 is enabled (ORL[2] = 1), the DSTC copies the value of DES6 to DES3. Pay attention to the value at the end of a transfer if InnerReload/OuterReload of the transfer destination address is not enabled. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the DA[31:0] bits via the CPU.

5.17 Descriptor 4 (DES4)

This section explains details of Descriptor 4 (DES4).

DES4 Descriptor configuration

Address: DESTP + DESP + 0x10 (ORL[2:0] = xx1)

bit	31	0
Field	DES4[31:0]	
C attribute	R/W	
D attribute	R	

DES4 Descriptor function

bit[31:0] DES4[31:0] (Descriptor 4)

DES4 sets the value to be loaded to DES1 (number of transfers) in OuterReload. Set DES4 to the same value as DES1 under the same configuration. The DSTC does not modify the value of this area.

5.18 Descriptor 5 (DES5)

This section explains details of Descriptor 5 (DES5).

DES5 Descriptor configuration

Address: DESTP + DESP + 0x10 (ORL[2:0] = x10)

Address: DESTP + DESP + 0x14 (ORL[2:0] = x11)

bit	31	0
Field	DES5[31:0]	
C attribute	R/W	
D attribute	R	

DES5 Descriptor function

bit[31:0] DES5[31:0] (Descriptor 5)

DES5 sets the setting to be loaded to DES2 (transfer source start address) in OuterReload. Set DES5 to the same value as DES2. The DSTC does not modify the value of this area.

5.19 Descriptor 6 (DES6)

This section explains details of Descriptor 6 (DES6).

DES6 Descriptor configuration

Address: DESTP + DESP + 0x10 (ORL[2:0] = 100)

Address: DESTP + DESP + 0x14 (ORL[2:0] = 110, 101)

Address: DESTP + DESP + 0x18 (ORL[2:0] = 111)

bit	31	0
Field	DES6[31:0]	
C attribute	R/W	
D attribute	R	

DES6 Descriptor function

bit[31:0] DES6[31:0] (Descriptor 6)

DES6 sets the setting to be loaded to DES3 (transfer destination start address) in OuterReload. Set DES6 to the same value as DES3. The DSTC does not modify the value of this area.

APPENDIXES

This chapter shows the register map and list of notes.

A. Product Type

B. Register Map (TYPE1-M0+)

C. Register Map (TYPE2-M0+)

D. Register Map (TYPE3-M0+)

E. List of Notes

CODE: 9AFAPPENDIXES-E03.0

A. Product Type

This section describes the product TYPE.

1. Product TYPE List

CODE: xxxx

1. Product TYPE List

In this manual, the products are classified into the following groups and are described as follows.

For the descriptions such as "TYPE1-M0+","TYPE2-M0+" and “TYPE3-M0+”, see the relevant items of the target FM0+ product in the lists below.

Table 1-1 FM0+ family TYPE1-M0+ Product list

TYPE	Flash memory size	
	56 Kbytes	**88 Kbytes**
TYPE1-M0+	S6E1A11B S6E1A11C	S6E1A12B S6E1A12C

Table 1-2 FM0+ family TYPE2-M0+ Product list

TYPE	Flash memory size	
	304K bytes	560K bytes
TYPE2-M0+	S6E1B84E S6E1B84F S6E1B84G S6E1B34E S6E1B34F S6E1B34G	S6E1B86E S6E1B86F S6E1B86G S6E1B36E S6E1B36F S6E1B36G

Table 1-3 FM0+ family TYPE3 Product list

TYPE	Flash memory size	
	64K bytes	128K bytes
TYPE3-M0+	S6E1C31B S6E1C31C S6E1C31D S6E1C11B S6E1C11C S6E1C11D	S6E1C32B S6E1C32C S6E1C32D S6E1C12B S6E1C12C S6E1C12D

B. Register Map (TYPE1-M0+)

This chapter shows the register map.

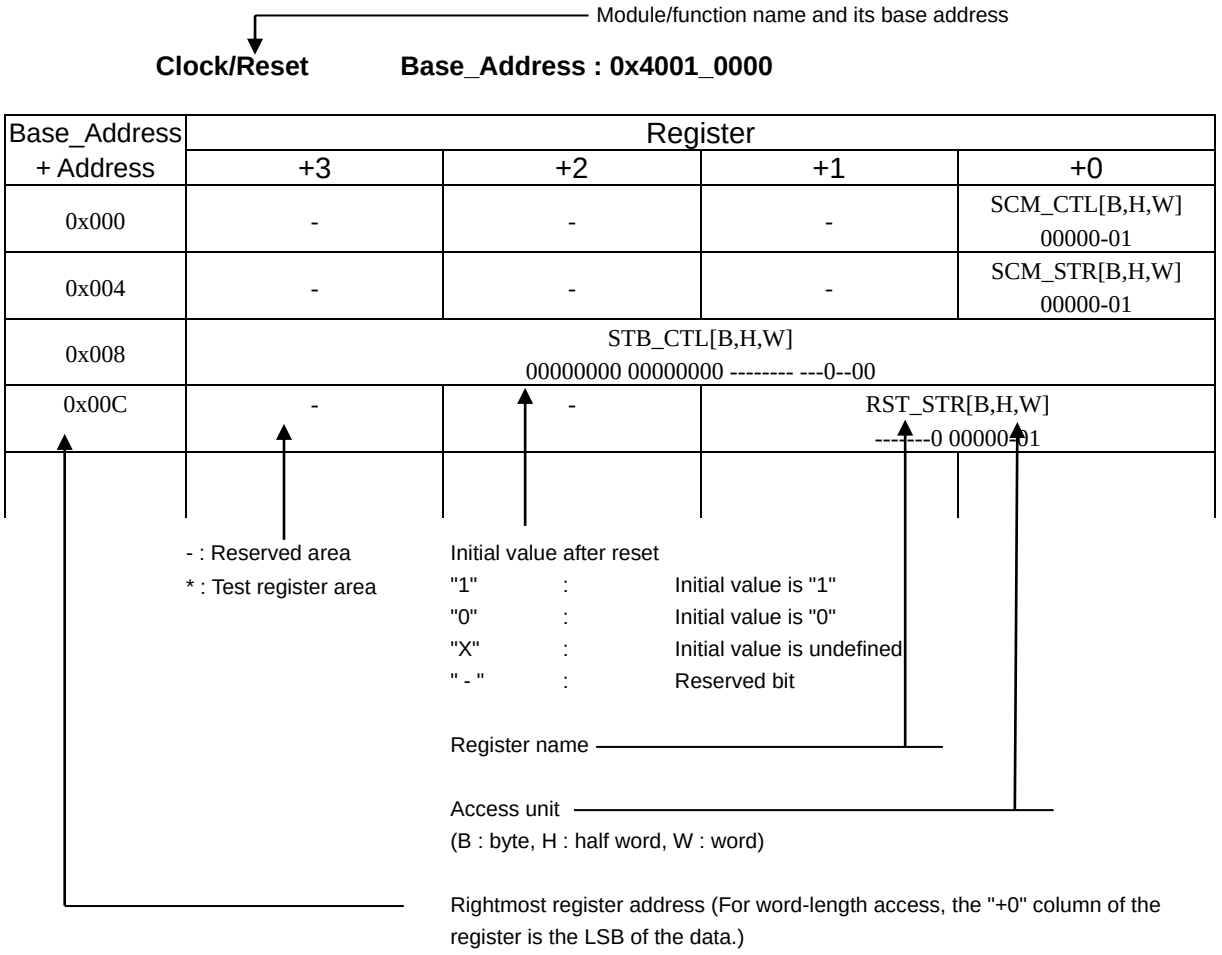
1. Register Map

CODE: 9AFREGMAP-E01.0

1. Register Map

Register map is shown on the table every module/function.

[How to Read the Each Table]



Notes:

- The register table is represented in the little-endian.
- When performing a data access, the addresses should be as below according to the access size.
  - Word access : Address should be multiples of 4 (least significant 2 bits should be "0x00")
  - Half word access : Address should be multiples of 2 (least significant bit should be "0x0")
  - Byte access : -
- Do not access the test register area.
- Do not access the area that is not written in the register table.
- When the register is accessed by larger unit than register size, for the reserved area to access at the same time, the read value is undefined, and writing is invalid.

1.1 Flash I/F

Flash I/F Base_Address : 0x4000_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C				
0x010	FSYNDN[B,H,W]			
0x014 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
0x024	FISR[B,H,W]			
0x028	FICLR[B,H,W]			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
0x104 - 0x1FC	-	-	-	-

Note:

- *For details of Flash I/F registers, refer to "FLASH PROGRAMMING MANUAL" of the product used.*

1.2 Unique ID

Unique ID Base_Address : 0x4000_0200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 Clock/Reset

Clock/Reset Base_Address : 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 0000--01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
0x030	-	-	-	CSW_TMR[W]
				00000000
0x034	-	-	-	PSW_TMR[W]
				---0-000
0x038	-	-	-	PLL_CTL1[W]
				00000000
0x03C	-	-	-	PLL_CTL2[W]
				--000000
0x040	-	-	CSV_CTL[W]	
			-111--00 ------11	
0x044	-	-	-	CSV_STR[W]
				------00
0x048	-	-	FCSWH_CTL[W]	
			11111111 11111111	
0x04C	-	-	FCSWL_CTL[W]	
			00000000 00000000	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----
0x058	-	-	-	*

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT Base_Address : 0x4001_1000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	WDG_LDR[W]			
	00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	-	-	-	WDG_CTL[W]
				------11
0x00C	-	-	-	WDG_ICL[W]
				XXXXXXXX
0x010	-	-	-	WDG_RIS[W]
				-------0
0x014 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW WDT

SW WDT Base_Address : 0x4001_2000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	WdogLoad[W]			
	11111111 11111111 11111111 11111111			
0x004	WdogValue[W]			
	11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W]
				---00000
0x00C	WdogIntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	-	-	-	WdogRIS[W]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C - 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			
0xC04 - 0xDFC	-	-	-	-
0xF00	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 Dual Timer

Dual Timer Base_Address : 0x4001_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	Timer1Load[W]			
	00000000 00000000 00000000 00000000			
0x004	Timer1Value[W]			
	11111111 11111111 11111111 11111111			
0x008	Timer1Control[W]			
	-------- -------- -------- 00100000			
0x00C	Timer1IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	Timer1RIS[W]			
	-------- -------- -------- -------0			
0x014	Timer1MIS[W]			
	-------- -------- -------- -------0			
0x018	Timer1BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x020	Timer2Load[W]			
	00000000 00000000 00000000 00000000			
0x024	Timer2Value[W]			
	11111111 11111111 11111111 11111111			
0x028	Timer2Control[W]			
	-------- -------- -------- 00100000			
0x02C	Timer2IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	Timer2RIS[W]			
	-------- -------- -------- -------0			
0x034	Timer2MIS[W]			
	-------- -------- -------- -------0			
0x038	Timer2BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

1.7 MFT

MFT unit0 Base_Address : 0x4002_0000

MFT unit1 Base_Address : 0x4002_1000

MFT unit2 Base_Address : 0x4002_2000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	OCCP0[H,W]		-	-
	00000000 00000000			
0x104	OCCP1[H,W]		-	-
	00000000 00000000			
0x108	OCCP2[H,W]		-	-
	00000000 00000000			
0x10C	OCCP3[H,W]		-	-
	00000000 00000000			
0x110	OCCP4[H,W]		-	-
	00000000 00000000			
0x114	OCCP5[H,W]		-	-
	00000000 00000000			
0x118	-	OCSD10[B,H,W]	OCSB10[B,H,W]	OCSA10[B,H,W]
		00000000	00000000	00000000
0x11C	-	OCSD32[B,H,W]	OCSB32[B,H,W]	OCSA32[B,H,W]
		00000000	00000000	00000000
0x120	-	OCSD54[B,H,W]	OCSB54[B,H,W]	OCSA54[B,H,W]
		00000000	00000000	00000000
0x124	-	-	OCSC[B,H,W]	-
			--000000	
0x128	-	-	OCSE0[H,W]	
			00000000 00000000	
0x12C	OCSE1[H,W]			
	00000000 00000000 00000000 00000000			
0x130	-	-	OCSE2[H,W]	
			00000000 00000000	
0x134	OCSE3[H,W]			
	00000000 00000000 00000000 00000000			
0x138	-	-	OCSE4[H,W]	
			00000000 00000000	
0x13C	OCSE5[H,W]			
	00000000 00000000 00000000 00000000			
0x140	TCCP0[H,W]		-	-
	11111111 11111111			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x144	TCDT0[H,W]		-	-
	00000000 00000000			
0x148	TCSC0[B,H,W]		TCSA0[B,H,W]	
	00000000 00000000		000---00 01000000	
0x14C	TCCP1[H,W]		-	-
	11111111 11111111			
0x150	TCDT1[H,W]			
	00000000 00000000			
0x154	TCSC1[B,H,W]		TCSA1[B,H,W]	
	00000000 00000000		000---00 01000000	
0x158	TCCP2[H,W]		-	-
	11111111 11111111			
0x15C	TCDT2[H,W]		-	-
	00000000 00000000			
0x160	TCSC2[B,H,W]		TCSA2[B,H,W]	
	00000000 00000000		000---00 01000000	
0x164	TCAL[B,H,W] (only in unit 0)			
	00000000 00000000 11111111 11111111			
0x168	-	OCFS54[B,H,W]	OCFS32[B,H,W]	OCFS10[B,H,W]
		00000000	00000000	00000000
0x16C	-	-	ICFS32[B,H,W]	ICFS10[B,H,W]
			00000000	00000000
0x170	-	ACFS54[B,H,W]	ACFS32[B,H,W]	ACFS10[B,H,W]
		00000000	00000000	00000000
0x174	ICCP0[H,W]		-	-
	00000000 00000000			
0x178	ICCP1[H,W]		-	-
	00000000 00000000			
0x17C	ICCP2[H,W]		-	-
	00000000 00000000			
0x180	ICCP3[H,W]		-	-
	00000000 00000000			
0x184	-	-	ICSB10[B,H,W]	ICSA10[B,H,W]
			------00	00000000
0x188	-	-	ICSB32[B,H,W]	ICSA32[B,H,W]
			------00	00000000
0x18C	WFTF10[H,W]		-	-
	00000000 00000000			
0x190	WFTB10[H,W]		WFTA10[H,W]	
	00000000 00000000		00000000 00000000	
0x194	WFTF32[H,W]		-	-
	00000000 00000000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x198	WFTB32[H,W]		WFTA32[H,W]	
	00000000 00000000		00000000 00000000	
0x19C	WFTF54[H,W]		-	-
	00000000 00000000			
0x1A0	WFTB54[H,W]		WFTA54[H,W]	
	00000000 00000000		00000000 00000000	
0x1A4	-	-	WFSA10[H,W]	
			---00000 000000	
0x1A8	-	-	WFSA32[H,W]	
			---00000 000000	
0x1AC	-	-	WFSA54[H,W]	
			---00000 000000	
0x1B0	-	-	WFIR[H,W]	
			00000000 00000000	
0x1B4	-	-	NZCL[H,W]	
			-000--00 ---00000	
0x1B8	ACMP0		-	-
	00000000 00000000			
0x1BC	ACMP1		-	-
	00000000 00000000			
0x1C0	ACMP2		-	-
	00000000 00000000			
0x1C4	ACMP3		-	-
	00000000 00000000			
0x1C8	ACMP4		-	-
	00000000 00000000			
0x1CC	ACMP5		-	-
	00000000 00000000			
0x1D0	-	-	ACSA[B,H,W]	
			--000000 --000000	
0x1D4	-	-	ACSD0[B,H,W]	ACSC0[B,H,W]
			00000000	00000000
0x1D8	-	-	ACSD1[B,H,W]	ACSC1[B,H,W]
			00000000	00000000
0x1DC	-	-	ACSD2[B,H,W]	ACSC2[B,H,W]
			00000000	00000000
0x1E0	-	-	ACSD3[B,H,W]	ACSC3[B,H,W]
			00000000	00000000
0x1E4	-	-	ACSD4[B,H,W]	ACSC4[B,H,W]
			00000000	00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x1E8	-	-	ACSD5[B,H,W]	ACSC5[B,H,W]
			00000000	00000000
0x1EC - 0xFFC	-	-	-	-

1.8 PPG

PPG Base_Address : 0x4002_4000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	TTCR0[B,H,W] 11110000	-
0x004	-	-	-	*
0x008	-	-	COMP0[B,H,W] 00000000	-
0x00C	-	-	-	COMP2[B,H,W] 00000000
0x010	-	-	COMP4[B,H,W] 00000000	-
0x014	-	-	-	COMP6[B,H,W] 00000000
0x018 - 0x01C	-	-	-	-
0x020	-	-	TTCR1[B,H,W] 11110000	-
0x024	-	-	-	*
0x028	-	-	COMP1[B,H,W] 00000000	-
0x02C	-	-	-	COMP3[B,H,W] 00000000
0x030	-	-	COMP5[B,H,W] 00000000	-
0x034	-	-	-	COMP7[B,H,W] 00000000
0x038 - 0x03C	-	-	-	-
0x040	-	-	TTCR2[B,H,W] 11110000	-
0x044	-	-	-	*
0x048	-	-	COMP8[B,H,W] 00000000	-
0x04C	-	-	-	COMP10[B,H,W] 00000000
0x050	-	-	COMP12[B,H,W] 00000000	-
0x054	-	-	-	COMP14[B,H,W] 00000000
0x058 - 0x0FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	-	-	TRG0[B,H,W]	
			00000000 00000000	
0x104	-	-	REVC0[B,H,W]	
			00000000 00000000	
0x108 - 0x13C	-	-	-	-
0x140	-	-	TRG1[B,H,W]	
			-------- 00000000	
0x144	-	-	REVC1[B,H,W]	
			-------- 00000000	
0x148 - 0x1FC	-	-	-	-
0x200	-	-	PPGC0[B,H,W]	PPGC1[B,H,W]
			00000000	00000000
0x204	-	-	PPGC2[B,H,W]	PPGC3[B,H,W]
			00000000	00000000
0x208	-	-	PRLH0[B,H,W]	PRLL0[B,H,W]
			XXXXXXXX	XXXXXXXX
0x20C	-	-	PRLH1[B,H,W]	PRLL1[B,H,W]
			XXXXXXXX	XXXXXXXX
0x210	-	-	PRLH2[B,H,W]	PRLL2[B,H,W]
			XXXXXXXX	XXXXXXXX
0x214	-	-	PRLH3[B,H,W]	PRLL3[B,H,W]
			XXXXXXXX	XXXXXXXX
0x218	-	-	-	GATEC0[B,H,W]
				--00---00
0x21C - 0x23C	-	-	-	-
0x240	-	-	PPGC4[B,H,W]	PPGC5[B,H,W]
			00000000	00000000
0x244	-	-	PPGC6[B,H,W]	PPGC7[B,H,W]
			00000000	00000000
0x248	-	-	PRLH4[B,H,W]	PRLL4[B.H.W]
			XXXXXXXX	XXXXXXXX
0x24C	-	-	PRLH5[B,H,W]	PRLL5[B,H,W]
			XXXXXXXX	XXXXXXXX
0x250	-	-	PRLH6[B,H,W]	PRLL6[B,H,W]
			XXXXXXXX	XXXXXXXX
0x254	-	-	PRLH7[B,H,W]	PRLL7[B,H,W]
			XXXXXXXX	XXXXXXXX
0x258	-	-	-	GATEC4[B,H,W]
				--00--00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8[B,H,W]	PPGC9[B,H,W]
			00000000	00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x284	-	-	PPGC10[B,H,W]	PPGC11[B,H,W]
			00000000	00000000
0x288	-	-	PRLH8[B,H,W]	PRLL8[B,H,W]
			XXXXXXXX	XXXXXXXX
0x28C	-	-	PRLH9[B,H,W]	PRLL9[B,H,W]
			XXXXXXXX	XXXXXXXX
0x290	-	-	PRLH10[B,H,W]	PRLL10[B,H,W]
			XXXXXXXX	XXXXXXXX
0x294	-	-	PRLH11[B,H,W]	PRLL11[B,H,W]
			XXXXXXXX	XXXXXXXX
0x298	-	-	-	GATEC8[B,H,W]
				--00--00
0x29C - 0x2BC	-	-	-	-
0x2C0	-	-	PPGC12[B,H,W]	PPGC13[B,H,W]
			00000000	00000000
0x2C4	-	-	PPGC14[B,H,W]	PPGC15[B,H,W]
			00000000	00000000
0x2C8	-	-	PRLH12[B,H,W]	PRLL12[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2CC	-	-	PRLH13[B,H,W]	PRLL13[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D0	-	-	PRLH14[B,H,W]	PRLL14[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D4	-	-	PRLH15[B,H,W]	PRLL15[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D8	-	-	-	GATEC12[B,H,W]
				--00--00
0x2DC - 0x2FC	-	-	-	-
0x300	-	-	PPGC16[B,H,W]	PPGC17[B,H,W]
			00000000	00000000
0x304	-	-	PPGC18[B,H,W]	PPGC19[B,H,W]
			00000000	00000000
0x308	-	-	PRLH16[B,H,W]	PRLL16[B,H,W]
			XXXXXXXX	XXXXXXXX
0x30C	-	-	PRLH17[B,H,W]	PRLL17[B,H,W]
			XXXXXXXX	XXXXXXXX
0x310	-	-	PRLH18[B,H,W]	PRLL18[B,H,W]
			XXXXXXXX	XXXXXXXX
0x314	-	-	PRLH19[B,H,W]	PRLL19[B,H,W]
			XXXXXXXX	XXXXXXXX
0x318	-	-	-	GATEC16[B,H,W]
				--00---00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x31C - 0x33C	-	-	-	-
0x340			PPGC20[B,H,W]	PPGC21[B,H,W]
			00000000	00000000
0x344	-	-	PPGC22[B,H,W]	PPGC23[B,H,W]
			00000000	00000000
0x348	-	-	PRLH20[B,H,W]	PRLL20[B,H,W]
			XXXXXXXX	XXXXXXXX
0x34C	-	-	PRLH21[B,H,W]	PRLL21[B,H,W]
			XXXXXXXX	XXXXXXXX
0x350	-	-	PRLH22[B,H,W]	PRLL22[B,H,W]
			XXXXXXXX	XXXXXXXX
0x354	-	-	PRLH23[B,H,W]	PRLL23[B,H,W]
			XXXXXXXX	XXXXXXXX
0x358	-	-	-	GATEC20[B,H,W]
				--00--00
0x35C - 0x37C	-	-	-	-
0x380	-	-	-	IGBTC[B,H,W]
				00000000
0x384 - 0xFFC	-	-	-	-

1.9 Base Timer

Base Timer ch.0 Base Address : 0x4002_5000

Base Timer ch.1 Base Address : 0x4002_5040

Base Timer ch.2 Base Address : 0x4002_5080

Base Timer ch.3 Base Address : 0x4002_50C0

Base Timer ch.4 Base Address : 0x4002_5200

Base Timer ch.5 Base Address : 0x4002_5240

Base Timer ch.6 Base Address : 0x4002_5280

Base Timer ch.7 Base Address : 0x4002_52C0

Base Timer ch.8 Base Address : 0x4002_5400

Base Timer ch.9 Base Address : 0x4002_5440

Base Timer ch.10 Base Address : 0x4002_5480

Base Timer ch.11 Base Address : 0x4002_54C0

Base Timer ch.12 Base Address : 0x4002_5600

Base Timer ch.13 Base Address : 0x4002_5640

Base Timer ch.14 Base Address : 0x4002_5680

Base Timer ch.15 Base Address : 0x4002_56C0

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL[H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF[H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	TMR[H,W]	
			00000000 00000000	
0x00C	-	-	TMCR[B,H,W]	
			-0000000 00000000	
0x010	-	-	TMCR2[B,H,W]	STC[B,H,W]
			-------0	0000-000
0x014 - 0x03C	-	-	-	-

1.10 IO Selector for Base Timer

IO Selector for ch.0-ch.3 (Base Timer) **Base Address : 0x4002_5100**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL0123[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.4-ch.7 (Base Timer) **Base Address : 0x4002_5300**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL4567[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.8-ch.11 (Base Timer) **Base Address : 0x4002_5500**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL89AB[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.12-ch.15 (Base Timer) **Base Address : 0x4002_5700**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSELCDEF[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simultaneous Startup (Base Timer) **Base Address : 0x4002_5F00**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR[B,H,W] XXXXXXXX XXXXXXXX	

1.11 QPRC

QPRC ch.0 **Base Address : 0x4002_6000**

QPRC ch.1 **Base Address : 0x4002_6040**

QPRC ch.2 **Base Address : 0x4002_6080**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	-	-	QPCR[H,W]	
			00000000 00000000	
0x0004	-	-	QRCR[H,W]	
			00000000 00000000	
0x0008	-	-	QPCCR[H,W]	
			00000000 00000000	
0x000C	-	-	QPRCR[H,W]	
			00000000 00000000	
0x0010	-	-	QMPR[H,W]	
			11111111 11111111	
0x0014	-	-	QICRH[B,H,W]	QICRL[B,H,W]
			--000000	00000000
0x0018	-	-	QCRH[B,H,W]	QCRL[B,H,W]
			00000000	00000000
0x001C	-	-	QECR[B,H,W]	
			-------- -----000	
0x0020 - 0x0038	-	-	-	-
0x003C	QPCRR[B,H,W]		QRCRR[B,H,W]	
	00000000 00000000		00000000 00000000	

1.12 QPRC NF

QPRC ch.0 NF **Base Address : 0x4002_6100**

QPRC ch.1 NF **Base Address : 0x4002_6110**

QPRC ch.2 NF **Base Address : 0x4002_6120**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	NFCTLA[B,H,W]			
	-------- -------- -------- --00-000			
0x0004	NFCTLB[B,H,W]			
	-------- -------- -------- --00-000			
0x0008	NFCTLC[B,H,W]			
	-------- -------- -------- --00-000			
0x000C	-	-	-	-

1.13 A/DC

12-bit A/DC unit0 Base_Address : 0x4002_7000

12-bit A/DC unit1 Base_Address : 0x4002_7100

12-bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W]	ADSR[B,H,W]
			000-0000	00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W]	SFNS[B,H,W]
			1000-000	----0000
0x00C	SCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W]	SCIS2[B,H,W]
			00000000	00000000
0x014	-	-	SCIS1[B,H,W]	SCIS0[B,H,W]
			00000000	00000000
0x018	-	-	PCCR[B,H,W]	PFNS[B,H,W]
			1000-000	--XX--00
0x01C	PCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X-XXX ---XXXXX			
0x020	-	-	-	PCIS[B,H,W]
				00000000
0x024	CMPD[B,H,W]		-	CMPCR[B,H,W]
	00000000 00------			00000000
0x028	-	-	ADSS3[B,H,W]	ADSS2[B,H,W]
			00000000	00000000
0x02C	-	-	ADSS1[B,H,W]	ADSS0[B,H,W]
			00000000	00000000
0x030	-	-	ADST0[B,H,W]	ADST1[B,H,W]
			00010000	00010000
0x034	-	-	-	ADCT[B,H,W]
				00000111
0x038	-	-	SCTSL[B,H,W]	PRTSL[B,H,W]
			----0000	----0000
0x03C	-	-	ADCEN[B,H,W]	
			11111111 ------00	
0x040	*			
0x044	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			
0x04C	-	-	WCMPSR[B,H,W]	WCMPCR[B,H,W]
			--000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.14 D/AC

10-bit D/AC Base_Address : 0x4002_8000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	DACR0[B,H,W]	DADR0[B,H,W]	
		-------0	------XX XXXXXXXX	
0x04	-	DACR1[B,H,W]	DADR1[B,H,W]	
		-------0	------XX XXXXXXXX	
0x08 - 0xFC	-	-	-	-

1.15 CR Trim

CR Trim Base_Address : 0x4002_E000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W]
				------001
0x004	-	-	MCR_FTRM[B,H,W]	
			------01 11101111	
0x008	-	-	-	MCR_TTRM[B,H,W]
				---10000
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.16 EXTI

EXTI Base_Address : 0x4003_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	ENIR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	EICL[B,H,W]			
	11111111 11111111 11111111 11111111			
0x00C	ELVR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C - 0x0FC	-	-	-	-

1.17 INT-Req. READ

INT-Req. READ Base_Address : 0x4003_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	DRQSEL[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	*			
0x008 - 0x00B	-	-	-	-
0x00C	-	-	-	IRQCMODE[B,H,W]
				-------0
0x010	EXC02MON[B,H,W]			
	-------- -------- -------- ------00			
0x014	IRQ00MON[B,H,W]			
	-------- -------- -------- -------0			
0x018	IRQ01MON[B,H,W]			
	-------- -------- -------- -------0			
0x01C	IRQ02MON[B,H,W]			
	-------- -------- -------- -------0			
0x020	IRQ03MON[B,H,W]			
	-------- -------- ----0000 00000000			
0x024	IRQ04MON[B,H,W]			
	-------- -------- -------- 00000000			
0x028	IRQ05MON[B,H,W]			
	-------- 00000000 00000000 00000000			
0x02C	IRQ06MON[B,H,W]			
	-------- ----0000 00000000 00000000			
0x030	IRQ07MON[B,H,W]			
	-------- -------- -------- ------00			
0x034	IRQ08MON[B,H,W]			
	-------- -------- -------- ----0000			
0x038	IRQ09MON[B,H,W]			
	-------- -------- -------- ------00			
0x03C	IRQ10MON[B,H,W]			
	-------- -------- -------- ----0000			
0x040	IRQ11MON[B,H,W]			
	-------- -------- -------- ------00			
0x044	IRQ12MON[B,H,W]			
	-------- -------- -------- ----0000			
0x048	IRQ13MON[B,H,W]			
	-------- -------- -------- ------00			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x04C	IRQ14MON[B,H,W]			
	-------- -------- -------- ----0000			
0x050	IRQ15MON[B,H,W]			
	-------- -------- -------- ------00			
0x054	IRQ16MON[B,H,W]			
	-------- -------- -------- ----0000			
0x058	IRQ17MON[B,H,W]			
	-------- -------- -------- ------00			
0x05C	IRQ18MON[B,H,W]			
	-------- -------- -------- ----0000			
0x060	IRQ19MON[B,H,W]			
	-------- -------- -------- ---0--00			
0x064	IRQ20MON[B,H,W]			
	-------- -------- -------- ---00000			
0x068	IRQ21MON[B,H,W]			
	-------- -------- -------- ---0--00			
0x06C	IRQ22MON[B,H,W]			
	-------- -------- -------- ---00000			
0x070	IRQ23MON[B,H,W]			
	-------- -------- -------0 00000000			
0x074	IRQ24MON[B,H,W]			
	-------- -------- -------- --00-000			
0x078	IRQ25MON[B,H,W]			
	-------- -------- -------- ---00000			
0x07C	IRQ26MON[B,H,W]			
	-------- -------- -------- ---00000			
0x080	IRQ27MON[B,H,W]			
	-------- -------- -------- --000000			
0x084	IRQ28MON[B,H,W]			
	-------- ------00 00000000 00000000			
0x088	IRQ29MON[B,H,W]			
	-------- -------- ----0000 00000000			
0x08C	IRQ30MON[B,H,W]			
	-------- ------00 00000000 00000000			
0x090	IRQ31MON[B,H,W]			
	----0--- -------- 00000000 00000000			
0x094 - 0x20C	-	-	-	-
0x210	RCINTSEL0[B,H,W] 00000000 00000000 00000000 00000000			
0x214	RCINTSEL1[B,H,W] 00000000 00000000 00000000 00000000			
0x218 - 0xFFC	-	-	-	-

1.18 GPIO

GPIO Base_Address : 0x4003_3000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x004	PFR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	PFR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024	PFR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x028	PFRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x02C	PFRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x030	PFRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x034	PFRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C	PFRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x040 - 0x0FC	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x108	PCR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x11C	PCR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x120	-	-	-	-
0x124	PCR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x128	PCRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x12C	PCRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x130	PCRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x134	PCRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x13C	PCRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x140 - 0x1FC	-	-	-	-
0x200	DDR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x21C	DDR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224	DDR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x228	DDRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x22C	DDRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x230	DDRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x234	DDRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C	DDRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x240 - 0x2FC	-	-	-	-
0x300	PDIR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	PDIR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324	PDIR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x328	PDIRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x32C	PDIRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x330	PDIRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x334	PDIRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C	PDIRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x340 - 0x3FC	-	-	-	-
0x400	PDOR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x408	PDOR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x414	PDOR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	PDOR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424	PDOR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x428	PDORA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x42C	PDORB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x430	PDORC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x434	PDORD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x43C	PDORF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x440 - 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	1111 1111 1111 1111 1111 1111 1111 1111			
0x504 - 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0101			
0x584 - 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- ---- ---- ---1 ---- ---- 0000 -000			
0x604	EPFR01[B,H,W]			
	0000 0000 0000 0000 --00 0000 0000 0000			
0x608	EPFR02[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x60C	EPFR03[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x610	EPFR04[B,H,W]			
	--00 0000 --00 00-- --00 0000 -000 00--			
0x614	EPFR05[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x618	EPFR06[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	---- ---- 0000 0000 0000 0000 0000 0000			
0x628 - 0x62C	-	-	-	-
0x630	EPFR12[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x634	EPFR13[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x638	EPFR14[B,H,W]			
	---- ---- ---- ---- ---- ---- --00 0000			
0x63C	EPFR15[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x640	EPFR16[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x644	EPFR17[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x648	EPFR18[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0000			
0x64C - 0x650	-	-	-	-
0x654	EPFR21[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- -000			
0x658	EPFR22[B,H,W]			
	---- ---- ---- ---- 0000 ---- 0000 ----			
0x65C - 0x6FC	-	-	-	-
0x700	PZR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	PZR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710	PZR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x714	PZR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x71C	PZR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x720	PZR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x724	PZR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x728	PZRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x72C	PZRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x730	PZRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x734	PZRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x738	PZRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x73C	PZRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x740 - 0x7FC	-	-	-	-
0x800	*			
0x804	*			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x808 - 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	FPOER7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924	FPOER9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x928	FPOERA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x92C	FPOERB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x930	FPOERC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x934	FPOERD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C	FPOERF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x940 - 0xFFC	-	-	-	-

1.19 HDMI-CEC

HDMI-CEC/Remote Control Receiver ch.0 Base_Address : 0x4003_4000

HDMI-CEC/Remote Control Receiver ch.1 Base_Address : 0x4003_4100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				--0000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				--00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 - 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 - 0xFC	-	-	-	-

1.20 LVD

LVD Base_Address : 0x4003_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	LVD_CTL[B,H,W]	
			100000-- 000011--	
0x004	-	-	-	LVD_STR[B,H,W]
				0-------
0x008	-	-	-	LVD_CLR[B,H,W]
				1-------
0x00C	LVD_RLR[W]			
	00000000 00000000 00000000 00000001			
0x010	-	-	-	LVD_STR2
				01------
0x014 - 0x0FC	-	-	-	-

1.21 DS Mode

DS Mode Base_Address : 0x4003_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	REG_CTL[B,H,W]
				-------0
0x004	-	-	-	RCK_CTL[B,H,W]
				------01
0x008 - 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL[B,H,W]
				-------0
0x704	-	-	-	WRFSR[B,H,W]
				------00
0x708	-	-	WIFSR[B,H,W]	
			------00 00000000	
0x70C	-	-	WIER[B,H,W]	
			------00 00000-00	
0x710	-	-	-	WILVR[B,H,W]
				-----000
0x714	-	-	-	DSRAMR[B,H,W]
				------00
0x718 - 0x7FC	-	-	-	-
0x800	BUR04[B,H,W]	BUR03[B,H,W]	BUR02[B,H,W]	BUR01[B,H,W]
	00000000	00000000	00000000	00000000
0x804	BUR08[B,H,W]	BUR07[B,H,W]	BUR06[B,H,W]	BUR05[B,H,W]
	00000000	00000000	00000000	00000000
0x808	BUR012[B,H,W]	BUR11[B,H,W]	BUR10[B,H,W]	BUR09[B,H,W]
	00000000	00000000	00000000	00000000
0x80C	BUR16[B,H,W]	BUR15[B,H,W]	BUR14[B,H,W]	BUR13[B,H,W]
	00000000	00000000	00000000	00000000
0x810 - 0xEFC	-	-	-	-

1.22 MFS

MFS ch.0 Base_Address : 0x4003_8000

MFS ch.1 Base_Address : 0x4003_8100

MFS ch.2 Base_Address : 0x4003_8200

MFS ch.3 Base_Address : 0x4003_8300

MFS ch.4 Base_Address : 0x4003_8400

MFS ch.5 Base_Address : 0x4003_8500

MFS ch.6 Base_Address : 0x4003_8600

MFS ch.7 Base_Address : 0x4003_8700

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0--00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	-	-	RDR/TDR[H,W]	
			00000000 00000000	
0x00C	-	-	BGR1[B,H,W]	BGR0[B,H,W]
			00000000	00000000
0x010	-	-	ISMK[B,H,W]	ISBA[B,H,W]
			--------	--------
0x014	-	-	FCR1[B,H,W]	FCR0[B,H,W]
			---00100	-0000000
0x018	-	-	FBYTE2[B,H,W]	FBYTE1[B,H,W]
			00000000	00000000
0x01C	-	-	SCSTR1/ EIBCR[B,H,W]	SCSTR0 [B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000--0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x044 - 0x0FC	-	-	-	-

1.23 CRC

CRC Base_Address : 0x4003_9000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W]
				-0000000
0x004	CRCINIT[B,H,W]			
	11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W]			
	11111111 11111111 11111111 11111111			

1.24 Watch Counter

Watch Counter Base_Address : 0x4003_A000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	WCCR[B,H,W]	WCRL[B,H,W]	WCRD[B,H,W]
		00--0000	--000000	--000000
0x004 - 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W]	
			-----000 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 - 0xFFC	-	-	-	-

1.25 RTC

RTC Base_Address : 0x4003_B000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	WTCR1[B,H,W]			
	00000000 00000000 ---00000 -00000-0			
0x004	WTCR2[B,H,W]			
	-------- -------- -----000 -------0			
0x008	WTBR[B,H,W]			
	-------- 00000000 00000000 00000000			
0x00C	WTDR[B,H,W]	WTHR[B,H,W]	WTMIR[B,H,W]	WTSR[B,H,W]
	--000000	--000000	-0000000	-0000000
0x010	-	WTYR[B,H,W]	WTMOR[B,H,W]	WTDW[B,H,W]
		00000000	---00000	-----000
0x014	ALDR[B,H,W]	ALHR[B,H,W]	ALMIR[B,H,W]	-
	--000000	--000000	-0000000	
0x018	-	ALYR[B,H,W]	ALMOR[B,H,W]	-
		00000000	---00000	
0x01C	WTTR[B,H,W]			
	-------- ------00 00000000 00000000			
0x020	-	-	WTCLKM[B,H,W]	WTCLKS[B,H,W]
			------00	-------0
0x024	-	WTCALEN[B,H,W]	WTCAL[B,H,W]	
		-------0	------00 00000000	
0x028	-	-	WTDIVEN[B,H,W]	WTDIV[B,H,W]
			------00	----0000
0x02C	-	-	-	WTCALPRD[B,H,W]
				--010011
0x030	-	-	-	WTCOSEL[B,H,W]
				-------0
0x034 - 0xFFC	-	-	-	-

1.26 Low-speed CR Prescaler

Low-speed CR Prescaler Base_Address : 0x4003_C000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W]
				--000000
0x000 - 0x0FC	-	-	-	-

1.27 Peripheral Clock Gating

Peripheral Clock Gating Base_Address : 0x4003_C100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1---1 ----1111 11111111 11111111			
0x004	MRST0[B,H,W]			
	-------0 ----0000 00000000 00000000			
0x008 - 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W]			
	-------- ----0000 ----0000 ----0000			
0x018 - 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- -------- -------- --00----			
0x024	MRST2[B,H,W]			
	-------- -------- -------- --00----			
0x028 - 0x0FC	-	-	-	-

1.28 DMAC

DMAC Base_Address : 0x4006_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000 - 0x00FC	-	-	-	-

1.29 MTB_DWT

MTB_DWT Base_Address : 0xF000_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 - 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD4	PID5[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD8	PID6[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFDC	PID7[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE0	PID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE4	PID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE8	PID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFEC	PID3[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF0	CID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF4	CID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF8	CID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

1.30 Fast GPIO

Fast GPIO Base_Address : 0xF800_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	FPDIR7[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024	-	-	FPDIR9[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x028	-	-	FPDIRA[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x02C	-	-	FPDIRB[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x030	-	-	FPDIRC[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x034	-	-	FPDIRD[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	FPDIRF[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	FPDOR7[B,H,W]	
			00000000 00000000	
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064	-	-	FPDOR9[B,H,W]	
			00000000 00000000	
0x068	-	-	FPDORA[B,H,W]	
			00000000 00000000	
0x06C	-	-	FPDORB[B,H,W]	
			00000000 00000000	
0x070	-	-	FPDORC[B,H,W]	
			00000000 00000000	
0x074	-	-	FPDORD[B,H,W]	
			00000000 00000000	
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	FPDORF[B,H,W]	
			00000000 00000000	
0x080	-	-	-	M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088	-	-	-	M_FPDIR2[B,H,W]
				XXXXXXXX
0x08C	-	-	-	M_FPDIR3[B,H,W]
				XXXXXXXX
0x090	-	-	-	M_FPDIR4[B,H,W]
				XXXXXXXX
0x094	-	-	-	M_FPDIR5[B,H,W]
				XXXXXXXX

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x098	-	-	-	M_FPDIR6[B,H,W]
				XXXXXXXX
0x09C	-	-	-	M_FPDIR7[B,H,W]
				XXXXXXXX
0x0A0	-	-	-	M_FPDIR8[B,H,W]
				XXXXXXXX
0x0A4	-	-	-	M_FPDIR9[B,H,W]
				XXXXXXXX
0x0A8	-	-	-	M_FPDIRA[B,H,W]
				XXXXXXXX
0x0AC	-	-	-	M_FPDIRB[B,H,W]
				XXXXXXXX
0x0B0	-	-	-	M_FPDIRC[B,H,W]
				XXXXXXXX
0x0B4	-	-	-	M_FPDIRD[B,H,W]
				XXXXXXXX
0x0B8	-	-	-	M_FPDIRE[B,H,W]
				XXXXXXXX
0x0BC	-	-	-	M_FPDIRF[B,H,W]
				XXXXXXXX
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8	-	-	-	M_FPDOR2[B,H,W]
				00000000
0x0CC	-	-	-	M_FPDOR3[B,H,W]
				00000000
0x0D0	-	-	-	M_FPDOR4[B,H,W]
				00000000
0x0D4	-	-	-	M_FPDOR5[B,H,W]
				00000000
0x0D8	-	-	-	M_FPDOR6[B,H,W]
				00000000
0x0DC	-	-	-	M_FPDOR7[B,H,W]
				00000000
0x0E0	-	-	-	M_FPDOR8[B,H,W]
				00000000
0x0E4	-	-	-	M_FPDOR9[B,H,W]
				00000000
0x0E8	-	-	-	M_FPDORA[B,H,W]
				00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0EC	-	-	-	M_FPDORB[B,H,W]
				00000000
0x0F0	-	-	-	M_FPDORC[B,H,W]
				00000000
0x0F4	-	-	-	M_FPDORD[B,H,W]
				00000000
0x0F8	-	-	-	M_FPDORE[B,H,W]
				00000000
0x0FC	-	-	-	M_FPDORF[B,H,W]
				00000000
0x100 - 0xFFC	-	-	-	-

C. Register Map (TYPE2-M0+)

This chapter shows the register map.

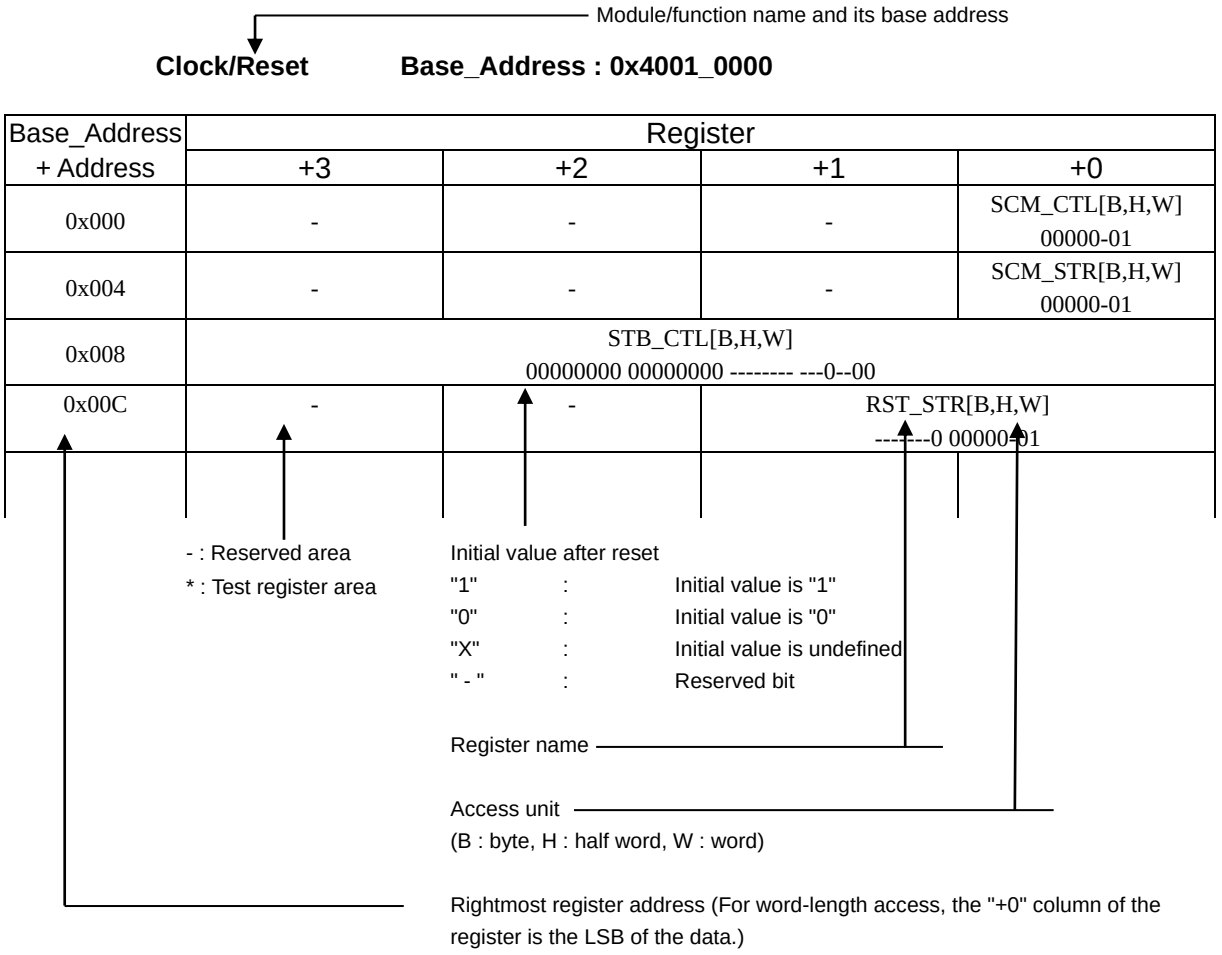
1. Register Map

CODE: 9AFREGMAP-E01.0

1. Register Map

Register map is shown on the table every module/function.

[How to Read the Each Table]



Notes:

- The register table is represented in the little-endian.
- When performing a data access, the addresses should be as below according to the access size.
  - Word access : Address should be multiples of 4 (least significant 2 bits should be "0x00")
  - Half word access : Address should be multiples of 2 (least significant bit should be "0x0")
  - Byte access : -
- Do not access the test register area.
- Do not access the area that is not written in the register table.
- When the register is accessed by larger unit than register size, for the reserved area to access at the same time, the read value is undefined, and writing is invalid.

1.1 Flash I/F

Flash I/F Base_Address : 0x4000_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C	FRVRC[B,H,W]			
0x010	FSYNDN[B,H,W]			
0x014 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
0x024	FISR[B,H,W]			
0x028	FICLR[B,H,W]			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
0x104 - 0x1FC	-	-	-	-

Note:

- *For details of Flash I/F registers, refer to "FLASH PROGRAMMING MANUAL" of the product used.*

1.2 Unique ID

Unique ID Base_Address : 0x4000_0200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 Clock/Reset

Clock/Reset Base_Address : 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 0000--01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
0x030	-	-	-	CSW_TMR[W]
				00000000
0x034	-	-	-	PSW_TMR[W]
				---0-000
0x038	-	-	-	PLL_CTL1[W]
				00000000
0x03C	-	-	-	PLL_CTL2[W]
				--000000
0x040	-	-	CSV_CTL[W]	
			-111--00 ------11	
0x044	-	-	-	CSV_STR[W]
				------00
0x048	-	-	FCSWH_CTL[W]	
			11111111 11111111	
0x04C	-	-	FCSWL_CTL[W]	
			00000000 00000000	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----
0x058	-	-	-	*
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT Base_Address : 0x4001_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	WDG_LDR[W]			
	00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	-	-	-	WDG_CTL[W]
				------11
0x00C	-	-	-	WDG_ICL[W]
				XXXXXXXX
0x010	-	-	-	WDG_RIS[R]
				-------0
0x014 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW_WDT

SW WDT Base_Address : 0x4001_2000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	WdogLoad[W]			
	11111111 11111111 11111111 11111111			
0x004	WdogValue[W]			
	11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W]
				---00000
0x00C	WdogIntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	-	-	-	WdogRIS[R]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C - 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			
0xC04 - 0xDFC	-	-	-	-
0xF00	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 Dual Timer

Dual Timer Base_Address : 0x4001_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	Timer1Load[W]			
	00000000 00000000 00000000 00000000			
0x004	Timer1Value[W]			
	11111111 11111111 11111111 11111111			
0x008	Timer1Control[W]			
	-------- -------- -------- 00100000			
0x00C	Timer1IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	Timer1RIS[W]			
	-------- -------- -------- -------0			
0x014	Timer1MIS[W]			
	-------- -------- -------- -------0			
0x018	Timer1BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x020	Timer2Load[W]			
	00000000 00000000 00000000 00000000			
0x024	Timer2Value[W]			
	11111111 11111111 11111111 11111111			
0x028	Timer2Control[W]			
	-------- -------- -------- 00100000			
0x02C	Timer2IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	Timer2RIS[W]			
	-------- -------- -------- -------0			
0x034	Timer2MIS[W]			
	-------- -------- -------- -------0			
0x038	Timer2BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

1.7 MFT

MFT unit0 Base_Address : 0x4002_0000

MFT unit1 Base_Address : 0x4002_1000

MFT unit2 Base_Address : 0x4002_2000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x100	OCCP0[H,W] 00000000 00000000		-	-
0x104	OCCP1[H,W] 00000000 00000000		-	-
0x108	OCCP2[H,W] 00000000 00000000		-	-
0x10C	OCCP3[H,W] 00000000 00000000		-	-
0x110	OCCP4[H,W] 00000000 00000000		-	-
0x114	OCCP5[H,W] 00000000 00000000		-	-
0x118	OCSD10[B,H,W] --000000 00000000		OCSB10[B,H,W] 00000000	OCSA10[B,H,W] 00000000
0x11C	OCSD32[B,H,W] --000000 00000000		OCSB32[B,H,W] 00000000	OCSA32[B,H,W] 00000000
0x120	OCSD54[B,H,W] --000000 00000000		OCSB54[B,H,W] 00000000	OCSA54[B,H,W] 00000000
0x124	-	-	OCSC[B,H,W] --000000	-
0x128	-	-	OCSE0[B,H,W] 00000000 00000000	
0x12C	OCSE1[B,H,W] 00000000 00000000 00000000 00000000			
0x130	-	-	OCSE2[B,H,W] 00000000 00000000	
0x134	OCSE3[B,H,W] 00000000 00000000 00000000 00000000			
0x138	-	-	OCSE4[B,H,W] 00000000 00000000	
0x13C	OCSE5[B,H,W] 00000000 00000000 00000000 00000000			
0x140	TCCP0[H,W] 11111111 11111111		-	-

Base_Address + Address	Register			
	+3	+2	+1	+0
0x144	TCDT0[H,W] 00000000 00000000		-	-
0x148	TCSC0[H,W] 00000000 00000000		TCSA0[B,H,W] 00000000 01000000	
0x14C	TCCP1[H,W] 11111111 11111111		-	-
0x150	TCDT1[H,W] 00000000 00000000		-	-
0x154	TCSC1[H,W] 00000000 00000000		TCSA1[B,H,W] 00000000 01000000	
0x158	TCCP2[H,W] 11111111 11111111		-	-
0x15C	TCDT2[H,W] 00000000 00000000		-	-
0x160	TCSC2[H,W] 00000000 00000000		TCSA2[B,H,W] 00000000 01000000	
0x164	TCAL[W] 00000000 00000000 11111111 11111111 *1			
	-	-	-	- *2
	*1 MFT unit0 *2 MFT unit1,unit2			
0x168	-	OCFS54[B,H,W] 00000000	OCFS32[B,H,W] 00000000	OCFS10[B,H,W] 00000000
0x16C	-	-	ICFS32[B,H,W] 00000000	ICFS10[B,H,W] 00000000
0x170	-	ACFS54[B,H,W] 00000000	ACFS32[B,H,W] 00000000	ACFS10[B,H,W] 00000000
0x174	ICCP0[H,W] 00000000 00000000		-	-
0x178	ICCP1[H,W] 00000000 00000000		-	-
0x17C	ICCP2[H,W] 00000000 00000000		-	-
0x180	ICCP3[H,W] 00000000 00000000		-	-
0x184	-	-	ICSB10[B,H,W] ------00	ICSA10[B,H,W] 00000000
0x188			ICSB32[B,H,W] ------00	ICSA32[B,H,W] 00000000
0x18C	WFTF10[H,W] 00000000 00000000		-	-
0x190	WFTB10[H,W] 00000000 00000000		WFTA10[H,W] 00000000 00000000	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x194	WFTF32[H,W] 00000000 00000000		-	-
0x198	WFTB32[H,W] 00000000 00000000		WFTA32[H,W] 00000000 00000000	
0x19C	WFTF54[H,W] 00000000 00000000		-	-
0x1A0	WFTB54[H,W] 00000000 00000000		WFTA54[H,W] 00000000 00000000	
0x1A4	-	-	WFSA10[B,H,W] --000000 000000	
0x1A8	-	-	WFSA32[B,H,W] --000000 000000	
0x1AC	-	-	WFSA54[B,H,W] --000000 000000	
0x1B0	-	-	WFIR[H,W] 00000000 00000000	
0x1B4	-	-	NZCL[H,W] 00000000 00000000	
0x1B8	ACMP0[H,W] 00000000 00000000		-	-
0x1BC	ACMP1[H,W] 00000000 00000000		-	-
0x1C0	ACMP2[H,W] 00000000 00000000		-	-
0x1C4	ACMP3[H,W] 00000000 00000000		-	-
0x1C8	ACMP4[H,W] 00000000 00000000		-	-
0x1CC	ACMP5[H,W] 00000000 00000000		-	-
0x1D0	-	-	ACSA[B,H,W] 00000000 00000000	
0x1D4	-	ACMC0[B,H,W] 00--0000	ACSD0[B,H,W] 00000000	ACSC0[B,H,W] 00000000
0x1D8	-	ACMC1[B,H,W] 00--0000	ACSD1[B,H,W] 00000000	ACSC1[B,H,W] 00000000
0x1DC	-	ACMC2[B,H,W] 00--0000	ACSD2[B,H,W] 00000000	ACSC2[B,H,W] 00000000
0x1E0	-	ACMC3[B,H,W] 00--0000	ACSD3[B,H,W] 00000000	ACSC3[B,H,W] 00000000
0x1E4	-	ACMC4[B,H,W] 00--0000	ACSD4[B,H,W] 00000000	ACSC4[B,H,W] 00000000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x1E8	-	ACMC5[B,H,W] 00--0000	ACSD5[B,H,W] 00000000	ACSC5[B,H,W] 00000000
0x1EC	-	-	-	TCSD[B,H,W] ------00
0x1F0-0xFFC	-	-	-	-

1.8 PPG

PPG Base_Address : 0x4002_4000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	TTCR0[B,H,W] 11110000	-
0x004	-	-	-	*
0x008	-	-	COMP0[B,H,W] 00000000	-
0x00C	-	-	-	COMP2[B,H,W] 00000000
0x010	-	-	COMP4[B,H,W] 00000000	-
0x014	-	-	-	COMP6[B,H,W] 00000000
0x018 - 0x01C	-	-	-	-
0x020	-	-	TTCR1[B,H,W] 11110000	-
0x024	-	-	-	*
0x028	-	-	COMP1[B,H,W] 00000000	-
0x02C	-	-	-	COMP3[B,H,W] 00000000
0x030	-	-	COMP5[B,H,W] 00000000	-
0x034	-	-	-	COMP7[B,H,W] 00000000
0x038 - 0x03C	-	-	-	-
0x040	-	-	TTCR2[B,H,W] 11110000	-
0x044	-	-	-	*
0x048	-	-	COMP8[B,H,W] 00000000	-
0x04C	-	-	-	COMP10[B,H,W] 00000000
0x050	-	-	COMP12[B,H,W] 00000000	-
0x054	-	-	-	COMP14[B,H,W] 00000000
0x058 - 0x0FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	-	-	TRG0[B,H,W]	
			00000000 00000000	
0x104	-	-	REVC0[B,H,W]	
			00000000 00000000	
0x108 - 0x13C	-	-	-	-
0x140	-	-	TRG1[B,H,W]	
			-------- 00000000	
0x144	-	-	REVC1[B,H,W]	
			-------- 00000000	
0x148 - 0x1FC	-	-	-	-
0x200	-	-	PPGC0[B,H,W]	PPGC1[B,H,W]
			00000000	00000000
0x204	-	-	PPGC2[B,H,W]	PPGC3[B,H,W]
			00000000	00000000
0x208	-	-	PRLH0[B,H,W]	PRLL0[B,H,W]
			XXXXXXXX	XXXXXXXX
0x20C	-	-	PRLH1[B,H,W]	PRLL1[B,H,W]
			XXXXXXXX	XXXXXXXX
0x210	-	-	PRLH2[B,H,W]	PRLL2[B,H,W]
			XXXXXXXX	XXXXXXXX
0x214	-	-	PRLH3[B,H,W]	PRLL3[B,H,W]
			XXXXXXXX	XXXXXXXX
0x218	-	-	-	GATEC0[B,H,W]
				--00---00
0x21C - 0x23C	-	-	-	-
0x240	-	-	PPGC4[B,H,W]	PPGC5[B,H,W]
			00000000	00000000
0x244	-	-	PPGC6[B,H,W]	PPGC7[B,H,W]
			00000000	00000000
0x248	-	-	PRLH4[B,H,W]	PRLL4[B.H.W]
			XXXXXXXX	XXXXXXXX
0x24C	-	-	PRLH5[B,H,W]	PRLL5[B,H,W]
			XXXXXXXX	XXXXXXXX
0x250	-	-	PRLH6[B,H,W]	PRLL6[B,H,W]
			XXXXXXXX	XXXXXXXX
0x254	-	-	PRLH7[B,H,W]	PRLL7[B,H,W]
			XXXXXXXX	XXXXXXXX
0x258	-	-	-	GATEC4[B,H,W]
				--00--00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8[B,H,W]	PPGC9[B,H,W]
			00000000	00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x284	-	-	PPGC10[B,H,W]	PPGC11[B,H,W]
			00000000	00000000
0x288	-	-	PRLH8[B,H,W]	PRLL8[B,H,W]
			XXXXXXXX	XXXXXXXX
0x28C	-	-	PRLH9[B,H,W]	PRLL9[B,H,W]
			XXXXXXXX	XXXXXXXX
0x290	-	-	PRLH10[B,H,W]	PRLL10[B,H,W]
			XXXXXXXX	XXXXXXXX
0x294	-	-	PRLH11[B,H,W]	PRLL11[B,H,W]
			XXXXXXXX	XXXXXXXX
0x298	-	-	-	GATEC8[B,H,W]
				--00--00
0x29C - 0x2BC	-	-	-	-
0x2C0	-	-	PPGC12[B,H,W]	PPGC13[B,H,W]
			00000000	00000000
0x2C4	-	-	PPGC14[B,H,W]	PPGC15[B,H,W]
			00000000	00000000
0x2C8	-	-	PRLH12[B,H,W]	PRLL12[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2CC	-	-	PRLH13[B,H,W]	PRLL13[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D0	-	-	PRLH14[B,H,W]	PRLL14[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D4	-	-	PRLH15[B,H,W]	PRLL15[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D8	-	-	-	GATEC12[B,H,W]
				--00--00
0x2DC - 0x2FC	-	-	-	-
0x300	-	-	PPGC16[B,H,W]	PPGC17[B,H,W]
			00000000	00000000
0x304	-	-	PPGC18[B,H,W]	PPGC19[B,H,W]
			00000000	00000000
0x308	-	-	PRLH16[B,H,W]	PRLL16[B,H,W]
			XXXXXXXX	XXXXXXXX
0x30C	-	-	PRLH17[B,H,W]	PRLL17[B,H,W]
			XXXXXXXX	XXXXXXXX
0x310	-	-	PRLH18[B,H,W]	PRLL18[B,H,W]
			XXXXXXXX	XXXXXXXX
0x314	-	-	PRLH19[B,H,W]	PRLL19[B,H,W]
			XXXXXXXX	XXXXXXXX
0x318	-	-	-	GATEC16[B,H,W]
				--00---00
0x31C - 0x33C	-	-	-	-
0x340	-	-	PPGC20[B,H,W]	PPGC21[B,H,W]

Base_Address	Register			
+ Address	+3	+2	+1	+0
			00000000	00000000
0x344	-	-	PPGC22[B,H,W]	PPGC23[B,H,W]
			00000000	00000000
0x348	-	-	PRLH20[B,H,W]	PRLL20[B,H,W]
			XXXXXXXX	XXXXXXXX
0x34C	-	-	PRLH21[B,H,W]	PRLL21[B,H,W]
			XXXXXXXX	XXXXXXXX
0x350	-	-	PRLH22[B,H,W]	PRLL22[B,H,W]
			XXXXXXXX	XXXXXXXX
0x354	-	-	PRLH23[B,H,W]	PRLL23[B,H,W]
			XXXXXXXX	XXXXXXXX
0x358	-	-	-	GATEC20[B,H,W]
				--00--00
0x35C - 0x37C	-	-	-	-
0x380	-	-	-	IGBTC[B,H,W]
				00000000
0x384 - 0xFFC	-	-	-	-

1.9 Base Timer

Base Timer ch.0 Base Address : 0x4002_5000

Base Timer ch.1 Base Address : 0x4002_5040

Base Timer ch.2 Base Address : 0x4002_5080

Base Timer ch.3 Base Address : 0x4002_50C0

Base Timer ch.4 Base Address : 0x4002_5200

Base Timer ch.5 Base Address : 0x4002_5240

Base Timer ch.6 Base Address : 0x4002_5280

Base Timer ch.7 Base Address : 0x4002_52C0

Base Timer ch.8 Base Address : 0x4002_5400

Base Timer ch.9 Base Address : 0x4002_5440

Base Timer ch.10 Base Address : 0x4002_5480

Base Timer ch.11 Base Address : 0x4002_54C0

Base Timer ch.12 Base Address : 0x4002_5600

Base Timer ch.13 Base Address : 0x4002_5640

Base Timer ch.14 Base Address : 0x4002_5680

Base Timer ch.15 Base Address : 0x4002_56C0

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL[H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF[H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	TMR[H,W]	
			00000000 00000000	
0x00C	-	-	TMCR[B,H,W]	
			-0000000 00000000	
0x010	-	-	TMCR2[B,H,W]	STC[B,H,W]
			-------0	0000-000
0x014 - 0x03C	-	-	-	-

1.10 IO Selector for Base Timer

IO Selector for ch.0-ch.3 (Base Timer) Base Address : 0x4002_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL0123[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.4-ch.7 (Base Timer) Base Address : 0x4002_5300

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL4567[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.8-ch.11 (Base Timer) Base Address : 0x4002_5500

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL89AB[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.12-ch.15 (Base Timer) Base Address : 0x4002_5700

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSELCDEF[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simultaneous Startup (Base Timer) Base Address : 0x4002_5F00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR[B,H,W] XXXXXXXX XXXXXXXX	

1.11 QPRC

QPRC ch.0 **Base Address : 0x4002_6000**

QPRC ch.1 **Base Address : 0x4002_6040**

QPRC ch.2 **Base Address : 0x4002_6080**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	-	-	QPCR[H,W]	
			00000000 00000000	
0x0004	-	-	QRCR[H,W]	
			00000000 00000000	
0x0008	-	-	QPCCR[H,W]	
			00000000 00000000	
0x000C	-	-	QPRCR[H,W]	
			00000000 00000000	
0x0010	-	-	QMPR[H,W]	
			11111111 11111111	
0x0014	-	-	QICRH[B,H,W]	QICRL[B,H,W]
			--000000	00000000
0x0018	-	-	QCRH[B,H,W]	QCRL[B,H,W]
			00000000	00000000
0x001C	-	-	QECR[B,H,W]	
			-------- -----000	
0x0020 - 0x0038	-	-	-	-
0x003C	QPCRR[B,H,W]		QRCRR[B,H,W]	
	00000000 00000000		00000000 00000000	

1.12 QPRC NF

QPRC ch.0 NF **Base Address : 0x4002_6100**

QPRC ch.1 NF **Base Address : 0x4002_6110**

QPRC ch.2 NF **Base Address : 0x4002_6120**

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	NFCTLA[B,H,W]			
	-------- -------- -------- --00-000			
0x0004	NFCTLB[B,H,W]			
	-------- -------- -------- --00-000			
0x0008	NFCTLC[B,H,W]			
	-------- -------- -------- --00-000			
0x000C	-	-	-	-

1.13 A/DC

12-bit A/DC unit0 Base_Address : 0x4002_7000

12-bit A/DC unit1 Base_Address : 0x4002_7100

12-bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W]	ADSR[B,H,W]
			000-0000	00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W]	SFNS[B,H,W]
			1000-000	----0000
0x00C	SCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W]	SCIS2[B,H,W]
			00000000	00000000
0x014	-	-	SCIS1[B,H,W]	SCIS0[B,H,W]
			00000000	00000000
0x018	-	-	PCCR[B,H,W]	PFNS[B,H,W]
			1000-000	--XX--00
0x01C	PCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X-XXX ---XXXXX			
0x020	-	-	-	PCIS[B,H,W]
				00000000
0x024	CMPD[B,H,W]		-	CMPCR[B,H,W]
	00000000 00------			00000000
0x028	-	-	ADSS3[B,H,W]	ADSS2[B,H,W]
			00000000	00000000
0x02C	-	-	ADSS1[B,H,W]	ADSS0[B,H,W]
			00000000	00000000
0x030	-	-	ADST0[B,H,W]	ADST1[B,H,W]
			00010000	00010000
0x034	-	-	-	ADCT[B,H,W]
				00000111
0x038	-	-	SCTSL[B,H,W]	PRTSL[B,H,W]
			----0000	----0000
0x03C	-	-	ADCEN[B,H,W]	
			11111111 ------00	
0x040	*			
0x044	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			
0x04C	-	-	WCMPSR[B,H,W]	WCMPCR[B,H,W]
			--000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.14 D/AC

10-bit D/AC Base_Address : 0x4002_8000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	DACR0[B,H,W]	DADR0[B,H,W]	
		-------0	------XX XXXXXXXX	
0x04	-	DACR1[B,H,W]	DADR1[B,H,W]	
		-------0	------XX XXXXXXXX	
0x08 - 0xFC	-	-	-	-

1.15 CR Trim

CR Trim Base_Address : 0x4002_E000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W]
				------001
0x004	-	-	MCR_FTRM[B,H,W]	
			------01 11101111	
0x008	-	-	-	MCR_TTRM[B,H,W]
				-0111111
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.16 EXTI

EXTI Base_Address : 0x4003_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	ENIR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	EICL[B,H,W]			
	11111111 11111111 11111111 11111111			
0x00C	ELVR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C – 0x0FC	-	-	-	-

1.17 INT-Req. READ

INT-Req. READ Base_Address : 0x4003_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	-
0x004	*			
0x008 – 0x00B	-	-	-	-
0x00C	-	-	-	IRQCMODE[B,H,W]
				-------0
0x010	EXC02MON[B,H,W]			
	-------- -------- -------- ------00			
0x014	IRQ00MON[B,H,W]			
	-------- -------- -------- -------0			
0x018	IRQ01MON[B,H,W]			
	-------- -------- -------- -------0			
0x01C	IRQ02MON[B,H,W]			
	-------- -------- -------- -------0			
0x020	IRQ03MON[B,H,W]			
	-------- -------- -------- ----0000			
0x024	IRQ04MON[B,H,W]			
	-------- -------- -------- 00000000			
0x028	IRQ05MON[B,H,W]			
	-------- -------- 00000000 00000000			
0x02C	IRQ06MON[B,H,W]			
	-------- -------- -------- ------00			
0x030	IRQ07MON[B,H,W]			
	-------- -------- -------- -------0			
0x034	IRQ08MON[B,H,W]			
	-------- -------- -------- ------00			
0x038	IRQ09MON[B,H,W]			
	-------- -------- -------- -------0			
0x03C	IRQ10MON[B,H,W]			
	-------- -------- -------- ------00			
0x040	IRQ11MON[B,H,W]			
	-------- -------- -------- -------0			
0x044	IRQ12MON[B,H,W]			
	-------- -------- -------- ------00			
0x048	IRQ13MON[B,H,W]			
	-------- -------- -------- -------0			
0x04C	IRQ14MON[B,H,W]			
	-------- -------- -------- ------00			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x050	IRQ15MON[B,H,W]			
	-------- -------- -------- -------0			
0x054	IRQ16MON[B,H,W]			
	-------- -------- -------- ------00			
0x058	IRQ17MON[B,H,W]			
	-------- -------- -------- -------0			
0x05C	IRQ18MON[B,H,W]			
	-------- -------- -------- ------00			
0x060	IRQ19MON[B,H,W]			
	-------- -------- -------- -------0			
0x064	IRQ20MON[B,H,W]			
	-------- -------- -------- ------00			
0x068	IRQ21MON[B,H,W]			
	-------- -------- -------- -------0			
0x06C	IRQ22MON[B,H,W]			
	-------- -------- -------- ------00			
0x070	IRQ23MON[B,H,W]			
	-------- -------- ---0000- -----000			
0x074	IRQ24MON[B,H,W]			
	-------- -------- -----00- --000000			
0x078	IRQ25MON[B,H,W]			
	-------- -------- -------- ---00000			
0x07C	IRQ26MON[B,H,W]			
	-------- -------- -------- --000000			
0x080	IRQ27MON[B,H,W]			
	-------- -------- -------- --0-----			
0x084	IRQ28MON[B,H,W]			
	-------- -------- -------- --000000			
0x088	IRQ29MON[B,H,W]			
	-------- -------- -------0 00000000			
0x08C	IRQ30MON[B,H,W]			
	-------- -------- -------- --000000			
0x090	IRQ31MON[B,H,W]			
	----0--- -------- 00000000 00000000			
0x094 – 0x20C	-	-	-	-
0x210	RCINTSEL0[B,H,W]			
	00000000 00000000 00000000 00000000			
0x214	RCINTSEL1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x218 – 0xFFC	-	-	-	-

1.18 LCDC

LCDC Base_Address : 0x4003_2000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	LCDCC3[B,H,W]	LCDCC2[B,H,W]	LCDCC1[B,H,W]
		0011111-	--010100	-00000--
0x04	LCDC_PSR[B,H,W]			
	-------- -0000000 00000000 00000000			
0x08	LCDC_COMEN[B,H,W]			
	-------- -------- -------- 00000000			
0x0C	LCDC_SEGEN1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x10	LCDC_SEGEN2[B,H,W]			
	-------- -------- -------- 00000000			
0x14	-	-	LCDC_BLINK[B,H,W]	
			00000000 00000000	
0x18	-	-	LCDC_BOOSTER[B,H,W]	
			--001110	----0011
0x1C	LCDRAM03[B,H,W]	LCDRAM02[B,H,W]	LCDRAM01[B,H,W]	LCDRAM00[B,H,W]
	00000000	00000000	00000000	00000000
0x20	LCDRAM07[B,H,W]	LCDRAM06[B,H,W]	LCDRAM05[B,H,W]	LCDRAM04[B,H,W]
	00000000	00000000	00000000	00000000
0x24	LCDRAM11[B,H,W]	LCDRAM10[B,H,W]	LCDRAM09[B,H,W]	LCDRAM08[B,H,W]
	00000000	00000000	00000000	00000000
0x28	LCDRAM15[B,H,W]	LCDRAM14[B,H,W]	LCDRAM13[B,H,W]	LCDRAM12[B,H,W]
	00000000	00000000	00000000	00000000
0x2C	LCDRAM19[B,H,W]	LCDRAM18[B,H,W]	LCDRAM17[B,H,W]	LCDRAM16[B,H,W]
	00000000	00000000	00000000	00000000
0x30	LCDRAM23[B,H,W]	LCDRAM22[B,H,W]	LCDRAM21[B,H,W]	LCDRAM20[B,H,W]
	00000000	00000000	00000000	00000000
0x34	LCDRAM27[B,H,W]	LCDRAM26[B,H,W]	LCDRAM25[B,H,W]	LCDRAM24[B,H,W]
	00000000	00000000	00000000	00000000
0x38	LCDRAM31[B,H,W]	LCDRAM30[B,H,W]	LCDRAM29[B,H,W]	LCDRAM28[B,H,W]
	00000000	00000000	00000000	00000000
0x3C	LCDRAM35[B,H,W]	LCDRAM34[B,H,W]	LCDRAM33[B,H,W]	LCDRAM32[B,H,W]
	00000000	00000000	00000000	00000000
0x40	LCDRAM39[B,H,W]	LCDRAM38[B,H,W]	LCDRAM37[B,H,W]	LCDRAM36[B,H,W]
	00000000	00000000	00000000	00000000
0x44 – 0xFC	-	-	-	-

1.19 GPIO

GPIO Base_Address : 0x4003_3000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0001 1111			
0x004	PFR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	PFR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024	PFR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x028	PFRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x02C	PFRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x030	PFRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x034	PFRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C	PFRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x040 – 0x0FC	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x108	PCR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x11C	PCR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x120	-	-	-	-
0x124	PCR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x128	PCRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x12C	PCRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x130	PCRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x134	PCRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x13C	PCRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x140 – 0x1FC	-	-	-	-
0x200	DDR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x21C	DDR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224	DDR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x228	DDRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x22C	DDRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x230	DDRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x234	DDRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C	DDRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x240 – 0x2FC	-	-	-	-
0x300	PDIR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	PDIR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324	PDIR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x328	PDIRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x32C	PDIRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x330	PDIRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x334	PDIRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C	PDIRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x340 – 0x3FC	-	-	-	-
0x400	PDOR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x408	PDOR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x414	PDOR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	PDOR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424	PDOR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x428	PDORA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x42C	PDORB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x430	PDORC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x434	PDORD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x43C	PDORF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x440 – 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	---- ---- 1111 1111 1111 1111 1111 1111			
0x504 – 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0101			
0x584 – 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- ---- ---- ---1 ---- ---- 0000 -000			
0x604	EPFR01[B,H,W]			
	0000 0000 0000 0000 –00 0000 0000 0000			
0x608	EPFR02[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x60C	EPFR03[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x610	EPFR04[B,H,W]			
	--00 0000 –00 00-- --00 0000 -000 00--			
0x614	EPFR05[B,H,W]			
	--00 0000 –00 00-- --00 0000 –00 00--			
0x618	EPFR06[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	---- ---- ---- ---- 0000 ---- ---- ----			
0x628 – 0x638	-	-	-	-
0x63C	EPFR15[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x640	EPFR16[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x644	-			
	-	-	-	-
0x648	EPFR18[B,H,W]			
	--00 0000 0000 0000 0000 0000 0000 0000			
0x64C – 0x658	-	-	-	-
0x65C	EPFR23[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x660 – 0x680	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x684	EPFR33[B,H,W]			
	---- 0000 0000 0000 ---- 0000 0000 0000			
0x688	EPFR34[B,H,W]			
	---- ---- ---- ---- ---- ---- 0000 ----			
0x68C – 0x690	-	-	-	-
0x694	EPFR37[B,H,W]			
	---- 0000 0000 0000 ---- ---- ---- ----			
0x698	EPFR38[B,H,W]			
	---- ---- ---- ---- ---- 0000 0000 0000			
0x69C – 0x6FC	-	-	-	-
0x700	PZR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	PZR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710	PZR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x714	PZR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x71C	PZR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x720	PZR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x724	PZR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x728	PZRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x72C	PZRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x730	PZRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x734	PZRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x738	PZRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x73C	PZRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x740	LVDIE[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- ---0			
0x744 – 0x7FC	-	-	-	-
0x800	*			
0x804	*			
0x808 – 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	FPOER7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924	FPOER9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x928	FPOERA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x92C	FPOERB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x930	FPOERC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x934	FPOERD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C	FPOERF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x940 – 0xFFC	-	-	-	-

1.20 HDMI-CEC

HDMI-CEC/Remote Control Receiver ch.0 Base_Address : 0x4003_4000

HDMI-CEC/Remote Control Receiver ch.1 Base_Address : 0x4003_4100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				000000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				0-00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 – 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 – 0xFC	-	-	-	-

1.21 LVD

LVD Base_Address : 0x4003_5000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	-	-	LVD_CTL [B,H,W]	
			100000–000011--	
0x004	-	-	LVD_STR [B,H,W]	
			0------1 0------1	
0x008	-	-	LVD_CLR [B,H,W]	
			1------- 1-------	
0x00C	LVD_RLR [W]			
	---------- 00000000 00000001			
0x010	-	-	LVD_STR2 [B,H,W]	
			0------- 01------	
0x014	-	-	LVD_CTL2 [B,H,W]	
			-------0 000011--	
0x018	-	-	-	LVD2_CTL [B,H,W]
				000011--
0x01C	-	-	LVD2_CTL2 [B,H,W]	
			0------0 000011--	
0x020 – 0x0FC	-	-	-	-

1.22 DS Mode

DS Mode Base_Address : 0x4003_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	REG_CTL [B,H,W]
				---- -10-
0x004	-	-	-	RCK_CTL [B,H,W]
				---- --01
0x008	-	-	-	REG_CTL2 [B,H,W]
				---- -011
0x00C	-	-	-	-
0x010	-	-	-	MOSC_CTL [B,H,W]
				---- -10-
0x014 – 0x0FC	-	-	-	-
0x100	-	-	-	CAL_CTL [B,H,W]
				---- 0000
0x104	-	-	-	CAL_SET [B,H,W]
				---1 0001
0x108	CAL_KEY [W]			
	00000000 00000000 00000000 00000001			
0x10C – 0x1FC	-	-	-	-
0x200 – 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL [B,H,W]
				---- ---0
0x704	-	-	-	WRFSR [B,H,W]
				---- --00
0x708	-	-	WIFSR [B,H,W]	
			00000000 00000000	
0x70C	-	-	WIER [B,H,W]	
			00000000 00000-00	
0x710	-	-	WILVR [B,H,W]	
			-----000 00000000	
0x714	-	-	-	DSRAMR [B,H,W]
				---- --00
0x718 – 0x71C	-	-	-	-
0x720	-	-	-	STBFLASHPDX [B,H,W]
				---- ---0
0x724	RST_MSK [W]			
	00000000 00000000 -------- -------0			
0x728 – 0x7FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x800	BUR04 [B,H,W]	BUR03 [B,H,W]	BUR02 [B,H,W]	BUR01 [B,H,W]
	00000000	00000000	00000000	00000000
0x804	BUR08 [B,H,W]	BUR07 [B,H,W]	BUR06 [B,H,W]	BUR05 [B,H,W]
	00000000	00000000	00000000	00000000
0x808	BUR012 [B,H,W]	BUR11 [B,H,W]	BUR10 [B,H,W]	BUR09 [B,H,W]
	00000000	00000000	00000000	00000000
0x80C	BUR16 [B,H,W]	BUR15 [B,H,W]	BUR14 [B,H,W]	BUR13 [B,H,W]
	00000000	00000000	00000000	00000000
0x810 – 0xEFC	-	-	-	-

1.23 USB Clock

USB Clock Base_Address : 0x4003_6000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	UCCR [B,H,W]
				-000 0000
0x004	-	-	-	UPCR [B,H,W]
				---- --00
0x008	-	-	-	UPCR2 [B,H,W]
				---- -000
0x00C	-	-	-	UPCR3 [B,H,W]
				---0 0000
0x010	-	-	-	UPCR4 [B,H,W]
				-011 1011
0x014	-	-	-	UP_STR [B,H,W]
				---- ---0
0x018	-	-	-	UPINT_ENR [B,H,W]
				---- ---0
0x01C	-	-	-	UPINT_CLR [B,H,W]
				---- ---0
0x020	-	-	-	UPINT_STR [B,H,W]
				---- ---0
0x024	-	-	-	UPCR5 [B,H,W]
				---- 0001
0x028	-	-	-	UPCR6 [B,H,W]
				---- 0010
0x02C	-	-	-	UP_CR7 [B,H,W]
				---- ---0
0x030	-	-	-	USBEN0 [B,H,W]
				---- -100
0x034	-	-	-	USBEN1 [B,H,W]
				---- -100
0x038 – 0xFFC	-	-	-	-

1.24 MFS

MFS ch.0 Base_Address : 0x4003_8000

MFS ch.1 Base_Address : 0x4003_8100

MFS ch.2 Base_Address : 0x4003_8200

MFS ch.3 Base_Address : 0x4003_8300

MFS ch.4 Base_Address : 0x4003_8400

MFS ch.5 Base_Address : 0x4003_8500

MFS ch.6 Base_Address : 0x4003_8600

MFS ch.7 Base_Address : 0x4003_8700

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0–00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	RDR/TDR[H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	BGR1[B,H,W]	BGR0[B,H,W]
			00000000	00000000
0x010	-	-	ISMK[B,H,W]	ISBA[B,H,W]
			--------	--------
0x014	-	-	FCR1[B,H,W]	FCR0[B,H,W]
			---00100	-0000000
0x018	-	-	FBYTE2[B,H,W]	FBYTE1[B,H,W]
			00000000	00000000
0x01C	-	-	SCSTR1/ EIBCR[B,H,W]	SCSTR0[B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000–0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x044	-	-	FTICR2[B,H,W] 00000000	FTICR1[B,H,W] 00000000
0x048 – 0x0FC	-	-	-	-

Note:

- *RDR/TDR register's higher 16 bits can be accessed by word operation in MFS-I²S mode.*

1.25 CRC

CRC Base_Address : 0x4003_9000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W]
				-0000000
0x004	CRCINIT[B,H,W]			
	11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W]			
	11111111 11111111 11111111 11111111			

1.26 Watch Counter

Watch Counter Base_Address : 0x4003_A000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	-	WCCR[B,H,W]	WCRL[B,H,W]	WCRD[B,H,W]
		00–0000	--000000	--000000
0x004 – 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W]	
			-----000 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 – 0xFFC	-	-	-	-

1.27 RTC

RTC Base_Address : 0x4003_B000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x100	-	-	-	WTCR10[B,H,W] 00000000
0x104	-	-	-	WTCR11[B,H,W] ---00000
0x108	-	-	-	WTCR12[B,H,W] 00000000
0x10C	-	-	-	WTCR13[B,H,W] 00000000
0x110	-	-	-	WTCR20[B,H,W] --000000
0x114	-	-	-	WTCR21[B,H,W] -----000
0x118	-	-	-	*
0x11C	-	-	-	WTSR[B,H,W] -0000000
0x120	-	-	-	WTMIR[B,H,W] -0000000
0x124	-	-	-	WTHR[B,H,W] --000000
0x128	-	-	-	WTDR[B,H,W] --000000
0x12C	-	-	-	WTDW[B,H,W] -----000
0x130	-	-	-	WTMOR[B,H,W] ---00000
0x134	-	-	-	WTYR[B,H,W] 00000000
0x138	-	-	-	ALMIR[B,H,W] -0000000
0x13C	-	-	-	ALHR[B,H,W] --000000
0x140	-	-	-	ALDR[B,H,W] --000000
0x144	-	-	-	ALMOR[B,H,W] ---00000
0x148	-	-	-	ALYR[B,H,W] 00000000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x14C	-	-	-	WTTR0[B,H,W] 00000000
0x150	-	-	-	WTTR1[B,H,W] 00000000
0x154	-	-	-	WTTR2[B,H,W] ------00
0x158	-	-	-	WTCAL0[B,H,W] 00000000
0x15C	-	-	-	WTCAL1[B,H,W] ------00
0x160	-	-	-	WTCALEN[B,H,W] -------0
0x164	-	-	-	WTDIV[B,H,W] ----0000
0x168	-	-	-	WTDIVEN[B,H,W] ------00
0x16C	-	-	-	WTCALPRD[B,H,W] --010011
0x170	-	-	-	WTCOSEL[B,H,W] -------0
0x174	-	-	-	VB_CLKDIV[B,H,W] 00000111
0x178	-	-	-	WTOSCCNT[B,H,W] ------01
0x17C	-	-	-	CCS[B,H,W] 11001110
0x180	-	-	-	CCB[B,H,W] 11001110
0x184	-	-	-	*
0x188	-	-	-	BOOST[B,H,W] ------11
0x18C	-	-	-	EWKUP[B,H,W] -------0
0x190	-	-	-	VDET[B,H,W] 00------
0x194	-	-	-	*
0x198	-	-	-	HIBRST[B,H,W] -------0
0x19C	-	-	-	VBPFR[B,H,W] --011100
0x1A0	-	-	-	VBPCR[B,H,W] ----0000
0x1A4	-	-	-	VBDDR[B,H,W] ----XXXX

Base_Address + Address	Register			
	+3	+2	+1	+0
0x1A8	-	-	-	VBDIR[B,H,W] ----0000
0x1AC	-	-	-	VBDOR[B,H,W] ----1111
0x0B0	-	-	-	VBPZR[B,H,W] ------11
0x1B4-1FF	-	-	-	-
0x200	BREG03[B,H,W] 00000000	BREG02[B,H,W] 00000000	BREG01[B,H,W] 00000000	BREG00[B,H,W] 00000000
0x204	BREG07[B,H,W] 00000000	BREG06[B,H,W] 00000000	BREG05[B,H,W] 00000000	BREG04[B,H,W] 00000000
0x208	BREG0B[B,H,W] 00000000	BREG0A[B,H,W] 00000000	BREG09[B,H,W] 00000000	BREG08[B,H,W] 00000000
0x20C	BREG0F[B,H,W] 00000000	BREG0E[B,H,W] 00000000	BREG0D[B,H,W] 00000000	BREG0C[B,H,W] 00000000
0x210	BREG13[B,H,W] 00000000	BREG12[B,H,W] 00000000	BREG11[B,H,W] 00000000	BREG10[B,H,W] 00000000
0x214	BREG17[B,H,W] 00000000	BREG16[B,H,W] 00000000	BREG15[B,H,W] 00000000	BREG14[B,H,W] 00000000
0x218	BREG1B[B,H,W] 00000000	BREG1A[B,H,W] 00000000	BREG19[B,H,W] 00000000	BREG18[B,H,W] 00000000
0x21C	BREG1F[B,H,W] 00000000	BREG1E[B,H,W] 00000000	BREG1D[B,H,W] 00000000	BREG1C[B,H,W] 00000000
0x220	BREG23[B,H,W] 00000000	BREG22[B,H,W] 00000000	BREG21[B,H,W] 00000000	BREG20[B,H,W] 00000000
0x224	BREG27[B,H,W] 00000000	BREG26[B,H,W] 00000000	BREG25[B,H,W] 00000000	BREG24[B,H,W] 00000000
0x228	BREG2B[B,H,W] 00000000	BREG2A[B,H,W] 00000000	BREG29[B,H,W] 00000000	BREG28[B,H,W] 00000000
0x22C	BREG2F[B,H,W] 00000000	BREG2E[B,H,W] 00000000	BREG2D[B,H,W] 00000000	BREG2C[B,H,W] 00000000
0x230	BREG33[B,H,W] 00000000	BREG32[B,H,W] 00000000	BREG31[B,H,W] 00000000	BREG30[B,H,W] 00000000
0x234	BREG37[B,H,W] 00000000	BREG36[B,H,W] 00000000	BREG35[B,H,W] 00000000	BREG34[B,H,W] 00000000
0x238	BREG3B[B,H,W] 00000000	BREG3A[B,H,W] 00000000	BREG39[B,H,W] 00000000	BREG38[B,H,W] 00000000
0x23C	BREG3F[B,H,W] 00000000	BREG3E[B,H,W] 00000000	BREG3D[B,H,W] 00000000	BREG3C[B,H,W] 00000000
0x240	BREG43[B,H,W] 00000000	BREG42[B,H,W] 00000000	BREG41[B,H,W] 00000000	BREG40[B,H,W] 00000000
0x244	BREG47[B,H,W] 00000000	BREG46[B,H,W] 00000000	BREG45[B,H,W] 00000000	BREG44[B,H,W] 00000000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x248	BREG4B[B,H,W] 00000000	BREG4A[B,H,W] 00000000	BREG49[B,H,W] 00000000	BREG48[B,H,W] 00000000
0x24C	BREG4F[B,H,W] 00000000	BREG4E[B,H,W] 00000000	BREG4D[B,H,W] 00000000	BREG4C[B,H,W] 00000000
0x250	BREG53[B,H,W] 00000000	BREG52[B,H,W] 00000000	BREG51[B,H,W] 00000000	BREG50[B,H,W] 00000000
0x254	BREG57[B,H,W] 00000000	BREG56[B,H,W] 00000000	BREG55[B,H,W] 00000000	BREG54[B,H,W] 00000000
0x258	BREG5B[B,H,W] 00000000	BREG5A[B,H,W] 00000000	BREG59[B,H,W] 00000000	BREG58[B,H,W] 00000000
0x25C	BREG5F[B,H,W] 00000000	BREG5E[B,H,W] 00000000	BREG5D[B,H,W] 00000000	BREG5C[B,H,W] 00000000
0x260	BREG63[B,H,W] 00000000	BREG62[B,H,W] 00000000	BREG61[B,H,W] 00000000	BREG60[B,H,W] 00000000
0x264	BREG67[B,H,W] 00000000	BREG66[B,H,W] 00000000	BREG65[B,H,W] 00000000	BREG64[B,H,W] 00000000
0x268	BREG6B[B,H,W] 00000000	BREG6A[B,H,W] 00000000	BREG69[B,H,W] 00000000	BREG68[B,H,W] 00000000
0x26C	BREG6F[B,H,W] 00000000	BREG6E[B,H,W] 00000000	BREG6D[B,H,W] 00000000	BREG6C[B,H,W] 00000000
0x270	BREG73[B,H,W] 00000000	BREG72[B,H,W] 00000000	BREG71[B,H,W] 00000000	BREG70[B,H,W] 00000000
0x274	BREG77[B,H,W] 00000000	BREG76[B,H,W] 00000000	BREG75[B,H,W] 00000000	BREG74[B,H,W] 00000000
0x278	BREG7B[B,H,W] 00000000	BREG7A[B,H,W] 00000000	BREG79[B,H,W] 00000000	BREG78[B,H,W] 00000000
0x27C	BREG7F[B,H,W] 00000000	BREG7E[B,H,W] 00000000	BREG7D[B,H,W] 00000000	BREG7C[B,H,W] 00000000
0x280-0xFFC	-	-	-	-

1.28 Low-speed CR Prescaler

Low-speed CR Prescaler Base_Address : 0x4003_C000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W]
				--000000
0x000 – 0x0FC	-	-	-	-

1.29 Peripheral Clock Gating

Peripheral Clock Gating Base_Address : 0x4003_C100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1–11 ----1111 11111111 11111111			
0x004	MRST0[B,H,W]			
	------00 ----0000 00000000 00000000			
0x008 – 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W]			
	-------- ----0000 ----0000 ----0000			
0x018 – 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- ---1-1-1 1111-1-- --00–00			
0x024	MRST2[B,H,W]			
	-------- ---0-0-0 0000-0-- --00–00			
0x028 – 0x0FC	-	-	-	-

1.30 Smart Card I/F

Smart Card I/F ch.0Base_Address : 0x4003_C900

Smart Card I/F ch.1Base_Address : 0x4003_C980

Base_Address + Address	Register			
	+3	+2	+1	+0
0x00	-	-	GLOBALCONTROL1[H,W] -0001000 00000000	
0x04	-	-	STATUS[H,W] --000000 00000001	
0x08	-	-	PORTCONTROL[H,W] 0000–00 00-0-0-0	
0x0C	-	-	DATA[H,W] -------0 00000000	
0x10	-	-	CARDCLOCK [H,W] 00000000 00101000	
0x14	-	-	BAUDRATE[H,W] 0000001 01110100	
0x18	-	-	GUARDTIMER[H,W] -------- 00000000	
0x1C	-	-	IDLETIMER[H,W] 00000000 00000000	
0x20	-	-	GLOBALCONTROL2[H,W] -------- ----1-00	
0x24	-	-	DATA_FIFO[H,W] -------0 00000000	
0x28	-	-	FIFO_LEVEL_READ[H,W] 00000000 00000000	
0x2C	-	-	FIFO_LEVEL_WRITE[H,W] 00000000 00000000	
0x30	-	-	FIFO_MODE[H,W] 00000000 ----0000	
0x34	-	-	FIFO_CLEAR_MSB_WRITE[H,W] -------- -------0	
0x38	-	-	FIFO_CLEAR_MSB_ READ[H,W] -------- -------0	
0x3C	-	-	-	-
0x40	-	-	IRQ_STATUS[H,W] -------- 00000000	
0x44- 0x7C	-	-	-	-

1.31 MFSI2S

MFSI2S ch.5 Base_Address : 0x4003_CA00

MFSI2S ch.6 Base_Address : 0x4003_CA80

Base_Address	Register			
+Address	+3	+2	+1	+0
0x00	-		CNTLREG [H,W] -------- -------- -----000 00000001	
0x04	-		I2SCLK [H,W] -------- -------- 000----- 00000000	
0x08	-		I2SST [B] ------00	I2SRST[B] 00000000

1.32 High Resilience

High Resilience Base_Address : 0x4003_E000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	RTR_CTL3 [B,H,W]	RTR_CTL2 [B,H,W]	RTR_CTL1 [B,H,W]	RTR_CTL0 [B,H,W]
	000- 000-	000- ----	---- ----	1111 1111
0x004	RTR_RTS3 [B,H,W]	RTR_RTS2 [B,H,W]	RTR_RTS1 [B,H,W]	RTR_RTS0 [B,H,W]
	1111 1111	1111 1111	1111 1111	1111 1111
0x008	RTR_TGS3 [B,H,W]	RTR_TGS2 [B,H,W]	RTR_TGS1 [B,H,W]	RTR_TGS0 [B,H,W]
	1111 1111	1111 1111	1111 1111	1111 1111
0x00C	RTR_STR3 [B,H,W]	RTR_STR2 [B,H,W]	RTR_STR1 [B,H,W]	RTR_STR0 [B,H,W]
	00-- ----	---- ----	00-- ----	---- ----
0x010	RTR_RLR [W]			
	00000000 00000000 00000000 00000000			
0x014	RTR_CT23 [B,H,W]	RTR_CT22 [B,H,W]	RTR_CT21 [B,H,W]	RTR_CT20 [B,H,W]
	0000 0000	0000 0000	0000 0000	---0 ---0
0x018	RTR_REV [B,H,W]			
	00000000 00010101 00000001 00000000			
0x01C – 0xFFC	-	-	-	-

1.33 USB

USB ch.0 Base_Address : 0x4004_0000

USB ch.1 Base_Address : 0x4005_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 00000000 00000000	
0x2110	-	-	HADR[B,H,W] -0000000	HRTIMER(2)[B,H,W] ------00
0x2114	-	-	HEOF(1/0)[B,H,W] --000000 00000000	
0x2118	-	-	HFRAME(1/0)[B,H,W] -----000 00000000	
0x211C	-	-	-	HTOKEN[B,H,W] 00000000
0x2120	-	-	UDCC[B,H,W] -------- 10100-00	
0x2124	-	-	EP0C[H,W] ------0- -1000000	
0x2128	-	-	EP1C[H,W] 01100001 00000000	
0x212C	-	-	EP2C[H,W] 0110000- -1000000	
0x2130	-	-	EP3C[H,W] 0110000- -1000000	
0x2134	-	-	EP4C[H,W] 0110000- -1000000	
0x2138	-	-	EP5C[H,W] 0110000- -1000000	
0x213C	-	-	TMSP[H,W] -----000 00000000	
0x2140	-	-	UDCIE[B,H,W] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EP0IS[H,W] 10---1-- --------	
0x2148	-	-	EP0OS[H,W] 100–00- -XXXXXXX	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXX	
0x2150	-	-	EP2S[H,W] 100-000- -XXXXXXX	
0x2154	-	-	EP3S[H,W] 100-000- -XXXXXXX	
0x2158	-	-	EP4S[H,W] 100-000- -XXXXXXX	
0x215C	-	-	EP5S[H,W] 100-000- -XXXXXXX	
0x2160	-	-	EP0DTH[B,H,W] XXXXXXXX	EP0DTL[B,H,W] XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXXX	EP1DTL[B,H,W] XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXXX	EP2DTL[B,H,W] XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXXX	EP3DTL[B,H,W] XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXXX	EP4DTL[B,H,W] XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXXX	EP5DTL[B,H,W] XXXXXXXX
0x2178 – 0x217C	-	-	-	-

1.34 DSTC

DSTC Base_Address : 0x4006_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	DESTP[B,H,W] 00000000 00000000 00000000 00000000			
0x004	HWDESP[B,H,W] 00XXXXXX XXXXXX00 00000000 00000000			
0x008	SWTR[H] 00000000 00000000		CFG[B] 01000000	CMD[B] 00000001
0x00C	MONERS[B,H,W] 00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x010	DREQENB[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x014	DREQENB[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x018-0x02C	-			
0x030	HWINT[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x034	HWINT[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x038-0x04C	-			
0x050	HWINTCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x054	HWINTCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x058-0x06C	-			
0x070	DQMSK[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x078-0x08C	-			
0x090	DQMSKCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x098-0xFFC	-			

1.35 MTB_DWT

MTB_DWT Base_Address : 0xF000_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 – 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD4	PID5[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD8	PID6[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFDC	PID7[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE0	PID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE4	PID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE8	PID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFEC	PID3[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF0	CID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF4	CID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF8	CID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

1.36 Fast GPIO

Fast GPIO Base_Address : 0xF800_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	FPDIR7[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024	-	-	FPDIR9[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x028	-	-	FPDIRA[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x02C	-	-	FPDIRB[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x030	-	-	FPDIRC[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x034	-	-	FPDIRD[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	FPDIRF[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	FPDOR7[B,H,W]	
			00000000 00000000	
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064	-	-	FPDOR9[B,H,W]	
			00000000 00000000	
0x068	-	-	FPDORA[B,H,W]	
			00000000 00000000	
0x06C	-	-	FPDORB[B,H,W]	
			00000000 00000000	
0x070	-	-	FPDORC[B,H,W]	
			00000000 00000000	
0x074	-	-	FPDORD[B,H,W]	
			00000000 00000000	
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	FPDORF[B,H,W]	
			00000000 00000000	
0x080	-	-	-	M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088	-	-	-	M_FPDIR2[B,H,W]
				XXXXXXXX
0x08C	-	-	-	M_FPDIR3[B,H,W]
				XXXXXXXX
0x090	-	-	-	M_FPDIR4[B,H,W]
				XXXXXXXX
0x094	-	-	-	M_FPDIR5[B,H,W]
				XXXXXXXX

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x098	-	-	-	M_FPDIR6[B,H,W]
				XXXXXXXX
0x09C	-	-	-	M_FPDIR7[B,H,W]
				XXXXXXXX
0x0A0	-	-	-	M_FPDIR8[B,H,W]
				XXXXXXXX
0x0A4	-	-	-	M_FPDIR9[B,H,W]
				XXXXXXXX
0x0A8	-	-	-	M_FPDIRA[B,H,W]
				XXXXXXXX
0x0AC	-	-	-	M_FPDIRB[B,H,W]
				XXXXXXXX
0x0B0	-	-	-	M_FPDIRC[B,H,W]
				XXXXXXXX
0x0B4	-	-	-	M_FPDIRD[B,H,W]
				XXXXXXXX
0x0B8	-	-	-	M_FPDIRE[B,H,W]
				XXXXXXXX
0x0BC	-	-	-	M_FPDIRF[B,H,W]
				XXXXXXXX
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8	-	-	-	M_FPDOR2[B,H,W]
				00000000
0x0CC	-	-	-	M_FPDOR3[B,H,W]
				00000000
0x0D0	-	-	-	M_FPDOR4[B,H,W]
				00000000
0x0D4	-	-	-	M_FPDOR5[B,H,W]
				00000000
0x0D8	-	-	-	M_FPDOR6[B,H,W]
				00000000
0x0DC	-	-	-	M_FPDOR7[B,H,W]
				00000000
0x0E0	-	-	-	M_FPDOR8[B,H,W]
				00000000
0x0E4	-	-	-	M_FPDOR9[B,H,W]
				00000000
0x0E8	-	-	-	M_FPDORA[B,H,W]
				00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0EC	-	-	-	M_FPDORB[B,H,W]
				00000000
0x0F0	-	-	-	M_FPDORC[B,H,W]
				00000000
0x0F4	-	-	-	M_FPDORD[B,H,W]
				00000000
0x0F8	-	-	-	M_FPDORE[B,H,W]
				00000000
0x0FC	-	-	-	M_FPDORF[B,H,W]
				00000000
0x100 – 0xFFC	-	-	-	-

D. Register Map (TYPE3-M0+)

This chapter shows the register map.

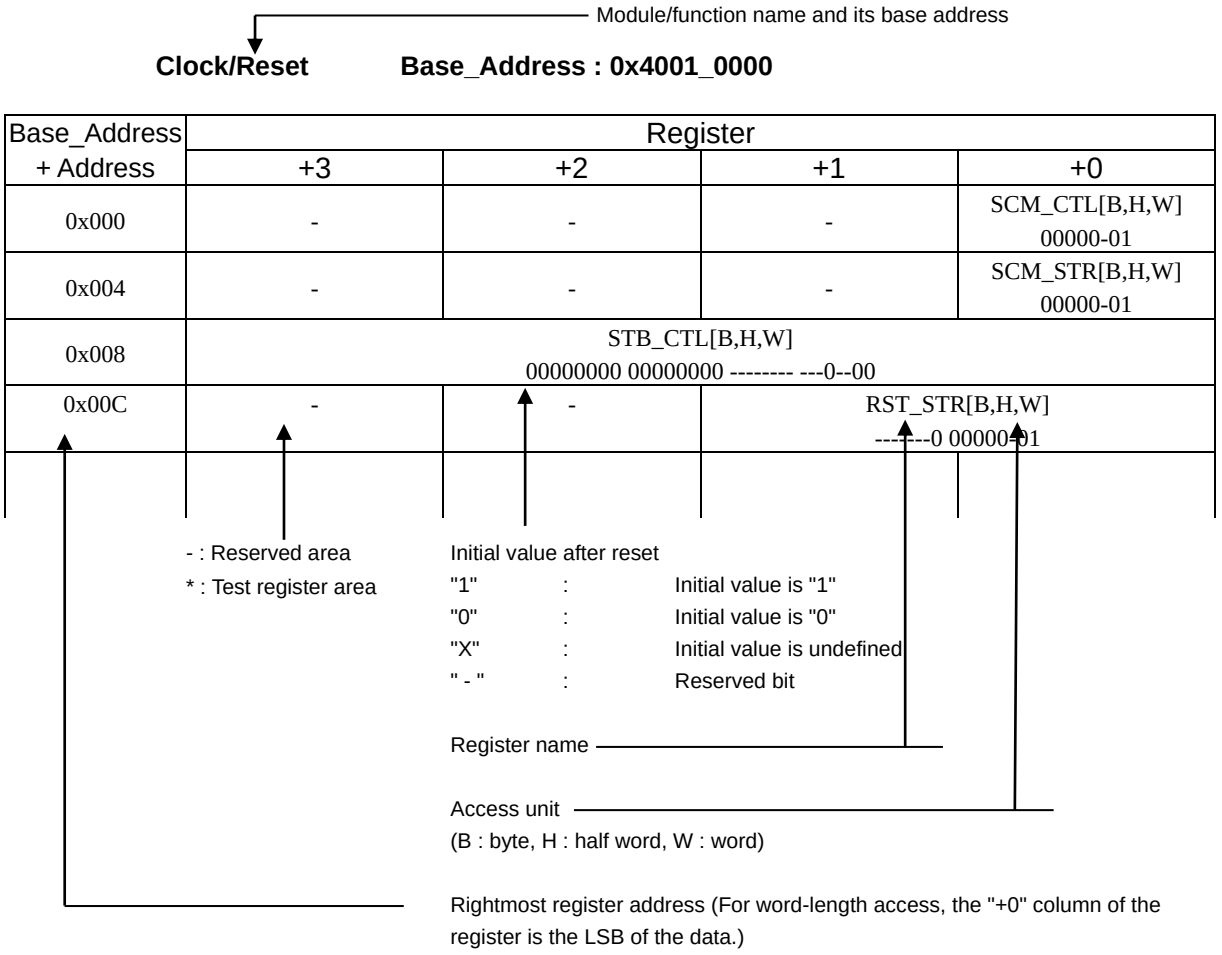
1. Register Map

CODE: 9AFREGMAP-E03.0

1. Register Map

Register map is shown on the table every module/function.

[How to Read the Each Table]



Notes:

- The register table is represented in the little-endian.
- When performing a data access, the addresses should be as below according to the access size.
  - Word access : Address should be multiples of 4 (least significant 2 bits should be "0x00")
  - Half word access : Address should be multiples of 2 (least significant bit should be "0x0")
  - Byte access : -
- Do not access the test register area.
- Do not access the area that is not written in the register table.
- When the register is accessed by larger unit than register size, for the reserved area to access at the same time, the read value is undefined, and writing is invalid.

1.1 Flash I/F

Flash I/F Base_Address : 0x4000_0000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
	-------- -------- -------- -----011			
0x008	FSTR[B,H,W]			
	-------- -------- -------- --00000X			
0x00C	-	-	-	-
0x010	FSYNDN[B,H,W]			
	-------- -------- -------- ----0001			
0x014 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
	-------- -------- -------- ------00			
0x024	FISR[B,H,W]			
	-------- -------- -------- ------00			
0x028	FICLR[B,H,W]			
	-------- -------- -------- ------00			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x104 - 0x1FC	-	-	-	-

Note:

- *For details of Flash I/F registers, refer to "FLASH PROGRAMMING MANUAL" of the product used.*

1.2 Unique ID

Unique ID Base_Address : 0x4000_0200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 Clock/Reset

Clock/Reset Base_Address : 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 00000-01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
0x030	-	-	-	CSW_TMR[W]
				00000000
0x034	-	-	-	PSW_TMR[W]
				---0-000
0x038	-	-	-	PLL_CTL1[W]
				00000000
0x03C	-	-	-	PLL_CTL2[W]
				--000000
0x040	-	-	CSV_CTL[W]	
			-111--00 ------11	
0x044	-	-	-	CSV_STR[W]
				------00
0x048	-	-	FCSWH_CTL[W]	
			11111111 11111111	
0x04C	-	-	FCSWL_CTL[W]	
			00000000 00000000	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x058	-	-	-	*
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT Base_Address : 0x4001_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	WDG_LDR[W]			
	00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	-	-	-	WDG_CTL[W]
				------11
0x00C	-	-	-	WDG_ICL[W]
				XXXXXXXX
0x010	-	-	-	WDG_RIS[W]
				-------0
0x014 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW WDT

SW WDT Base_Address : 0x4001_2000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	WdogLoad[W]			
	11111111 11111111 11111111 11111111			
0x004	WdogValue[W]			
	11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W]
				---00000
0x00C	WdogIntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	-	-	-	WdogRIS[W]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C - 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			
0xC04 - 0xDFC	-	-	-	-
0xF00	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 Dual Timer

Dual Timer Base_Address : 0x4001_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	Timer1Load[W]			
	00000000 00000000 00000000 00000000			
0x004	Timer1Value[W]			
	11111111 11111111 11111111 11111111			
0x008	Timer1Control[W]			
	-------- -------- -------- 00100000			
0x00C	Timer1IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	Timer1RIS[W]			
	-------- -------- -------- -------0			
0x014	Timer1MIS[W]			
	-------- -------- -------- -------0			
0x018	Timer1BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x020	Timer2Load[W]			
	00000000 00000000 00000000 00000000			
0x024	Timer2Value[W]			
	11111111 11111111 11111111 11111111			
0x028	Timer2Control[W]			
	-------- -------- -------- 00100000			
0x02C	Timer2IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	Timer2RIS[W]			
	-------- -------- -------- -------0			
0x034	Timer2MIS[W]			
	-------- -------- -------- -------0			
0x038	Timer2BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

1.7 Base Timer

Base Timer ch.0 Base Address : 0x4002_5000

Base Timer ch.1 Base Address : 0x4002_5040

Base Timer ch.2 Base Address : 0x4002_5080

Base Timer ch.3 Base Address : 0x4002_50C0

Base Timer ch.4 Base Address : 0x4002_5200

Base Timer ch.5 Base Address : 0x4002_5240

Base Timer ch.6 Base Address : 0x4002_5280

Base Timer ch.7 Base Address : 0x4002_52C0

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL[H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF[H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	TMR[H,W]	
			00000000 00000000	
0x00C	-	-	TMCR[B,H,W]	
			-0000000 00000000	
0x010	-	-	TMCR2[B,H,W]	STC[B,H,W]
			0-------0	0000-000
0x014 - 0x03C	-	-	-	-

1.8 IO Selector for Base Timer

IO Selector for ch.0-ch.3 (Base Timer) Base Address : 0x4002_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL0123[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.4-ch.7 (Base Timer) Base Address : 0x4002_5300

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL4567[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simultaneous Startup (Base Timer) Base Address : 0x4002_5F00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR [B,H,W] -------- XXXXXXXX	

1.9 A/DC

12-bit A/DC unit0 Base_Address : 0x4002_7000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W]	ADSR[B,H,W]
			000-0000	00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W]	SFNS[B,H,W]
			1000-000	----0000
0x00C	SCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W]	SCIS2[B,H,W]
			00000000	00000000
0x014	-	-	SCIS1[B,H,W]	SCIS0[B,H,W]
			00000000	00000000
0x018	-	-	PCCR[B,H,W]	PFNS[B,H,W]
			10000000	--XX--00
0x01C	PCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X-XXX ---XXXXX			
0x020	-	-	-	PCIS[B,H,W]
				00000000
0x024	CMPD[B,H,W]		-	CMPCR[B,H,W]
	00000000 00------			00000000
0x028	-	-	ADSS3[B,H,W]	ADSS2[B,H,W]
			00000000	00000000
0x02C	-	-	ADSS1[B,H,W]	ADSS0[B,H,W]
			00000000	00000000
0x030	-	-	ADST0[B,H,W]	ADST1[B,H,W]
			00010000	00010000
0x034	-	-	-	ADCT[B,H,W]
				00000111
0x038	-	-	SCTSL[B,H,W]	PRTSL[B,H,W]
			----0000	----0000
0x03C	-	-	ADCEN[B,H,W]	
			11111111 ------00	
0x040	*			
0x044	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x04C	-	-	WCMPSR[B,H,W]	WCMPCR[B,H,W]
			00000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.10 CR Trim

CR Trim Base_Address : 0x4002_E000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	-	-	-	MCR_PSR[B,H,W]
				-----001
0x004	-	-	MCR_FTRM[B,H,W]	
			------10 00000110	
0x008	-	-	-	MCR_TTRM[B,H,W]
				-1111111
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.11 EXTI

EXTI Base_Address : 0x4003_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	ENIR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	EICL[B,H,W]			
	11111111 11111111 11111111 11111111			
0x00C	ELVR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C	ELVR2[B,H,W]			
	00000000 00000000 00000000 00000000			
0x020	-	-	-	NMIENR[B,H,W]
	-	-	-	-------0
0x024 – 0x0FC	-	-	-	-

1.12 INT-Req. READ

INT-Req. READ Base_Address : 0x4003_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 – 0x004	-	-	-	-
0x008	VIR_OFFSET[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	-	-	-	ODDPKS[B,H,W]
				---00000
0x014 – 0x1FC	-	-	-	-
0x200	EXC02MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x204	IRQ00MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x208	IRQ01MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x20C	IRQ02MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x210	IRQ03MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x214	IRQ04MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x218	IRQ05MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x21C	IRQ06MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x220	IRQ07MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x224	IRQ08MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x228	IRQ09MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x22C	IRQ10MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x230	IRQ11MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x234	IRQ12MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x238	IRQ13MON[B,H,W]			
	00000000 00000000 00000000 00000000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x23C	IRQ14MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x240	IRQ15MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x244	IRQ16MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x248	IRQ17MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x24C	IRQ18MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x250	IRQ19MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x254	IRQ20MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x258	IRQ21MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x25C	IRQ22MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x260	IRQ23MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x264	IRQ24MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x268	IRQ25MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x26C	IRQ26MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x270	IRQ27MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x274	IRQ28MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x278	IRQ29MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x27C	IRQ30MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x280	IRQ31MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x284 – 0xFFC	-	-	-	-

1.13 GPIO

GPIO Base_Address : 0x4003_3000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x004	PFR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	-	-	-	-
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024 – 0x034	-	-	-	-
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C – 0x0FC	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x108	PCR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x11C – 0x134	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x13C – 0x1FC	-	-	-	-
0x200	DDR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x21C	-	-	-	-
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224 – 0x234	-	-	-	-
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C – 0x2FC	-	-	-	-
0x300	PDIR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	-	-	-	-
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324 – 0x334	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C – 0x3FC	-	-	-	-
0x400	PDOR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x408	PDOR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x414	PDOR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	-	-	-	-
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424 – 0x434	-	-	-	-
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x43C – 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	1111 1111 1111 1111 1111 1111 1111 1111			
0x504 – 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---0 0101			
0x584 – 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- --00 ---- --01 ---- --0- ---- --00			
0x604 – 0x60C	-	-	-	-
0x610	EPFR04[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x614	EPFR05[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x618	EPFR06[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x620	EPFR08[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	---- ---- 0000 0000 0000 0000 0000 0000			
0x628 – 0x654	-	-	-	-
0x658	EPFR22[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x65C	EPFR23[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x660 – 0x678	-	-	-	-
0x67C	EPFR31[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x680	-	-	-	-
0x684	EPFR33[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x688 – 0x690	-	-	-	-
0x694	EPFR37[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x698	EPFR38[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x69C – 0x6FC	-	-	-	-
0x700	-	-	-	-
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	-	-	-	-
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710 – 0x714	-	-	-	-
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x71C – 0x7FC	-	-	-	-
0x800 – 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	-	-	-	-
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924 – 0x934	-	-	-	-
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C – 0xFFC	-	-	-	-

1.14 HDMI-CEC

HDMI-CEC/Remote Control Receiver ch.0 Base_Address : 0x4003_4000

HDMI-CEC/Remote Control Receiver ch.1 Base_Address : 0x4003_4100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				000000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				0-00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 – 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 – 0xFC	-	-	-	-

1.15 LVD

LVD Base_Address : 0x4003_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	LVD_CTL [B,H,W]	
			10000000 00001100	
0x004	-	-	LVD_STR [B,H,W]	
			00000000 0000000-	
0x008	-	-	LVD_CLR [B,H,W]	
			00000000 10000000	
0x00C	LVD_RLR [W]			
	00000000 00000000 00000000 00000001			
0x010	-	-	LVD_STR2 [B,H,W]	
			00000000 01000000	
0x014 – 0x0FC	-	-	-	-

1.16 DS Mode

DS Mode Base_Address : 0x4003_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	REG_CTL [B,H,W]
				---- -10-
0x004	-	-	-	RCK_CTL [B,H,W]
				---- --01
0x008 – 0x00C	-	-	-	-
0x010	-	-	-	MOSC_CTL [B,H,W]
				---- -10-
0x014 – 0x0FC	-	-	-	-
0x100	-	-	-	CAL_CTL [B,H,W]
				---- 0000
0x104	-	-	-	-
0x108	CAL_KEY [W]			
	00000000 00000000 00000000 00000001			
0x10C – 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL [B,H,W]
				---- ---0
0x704	-	-	-	WRFSR [B,H,W]
				---- --00
0x708	-	-	WIFSR [B,H,W]	
			00000000 00000000	
0x70C	-	-	WIER [B,H,W]	
			00000000 00000-00	
0x710	-	-	WILVR [B,H,W]	
			-----000 00000000	
0x714	-	-	-	DSRAMR [B,H,W]
				---- --00
0x718 – 0x7FC	-	-	-	-
0x800	BUR04 [B,H,W]	BUR03 [B,H,W]	BUR02 [B,H,W]	BUR01 [B,H,W]
	00000000	00000000	00000000	00000000
0x804	BUR08 [B,H,W]	BUR07 [B,H,W]	BUR06 [B,H,W]	BUR05 [B,H,W]
	00000000	00000000	00000000	00000000
0x808	BUR012 [B,H,W]	BUR11 [B,H,W]	BUR10 [B,H,W]	BUR09 [B,H,W]
	00000000	00000000	00000000	00000000
0x80C	BUR16 [B,H,W]	BUR15 [B,H,W]	BUR14 [B,H,W]	BUR13 [B,H,W]
	00000000	00000000	00000000	00000000
0x810 – 0x8FC	-	-	-	-

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x900	WIOLC_CTL [B,H,W]			
	-------0 -------1 -------0 -------0			
0x904	-	-	-	SUBOSC_CTL[B,H,W]
				------01
0x908	-	-	-	CEC_CTL [B,H,W]
				----0000
0x90C	-	-	-	DEBUG_SW_CTL[B,H,W]
				-------1
0x910 – 0xEFC	-	-	-	-

1.17 USB Clock

USB Clock Base_Address : 0x4003_6000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	UCCR [B,H,W]
				-----000
0x004 – 0x024	-	-	-	-
0x028	-	-	-	UPCR6[B,H,W]
				----0010
0x02C	-	-	-	-
0x030	-	-	-	USBEN0[B,H,W]
				-------0
0x038 – 0x0FC	-	-	-	-

1.18 I2CSLAVE

I2CSLAVE ch.6 Base_Address : 0x4003_7980

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	IBSCR[B,H,W]		IBSSR[B,H,W]	
	------00 0-000000		-----001 00000000	
0x04	-	IBSDSTUPR[B,H,W]	IBSMSKR[B,H,W]	IBSADR[B,H,W]
	-	11111111	01111111	00000000
0x08	-	-	-	IBSTDR[B,H,W]
	-	-	-	11111111
0x0C	-	-	-	IBSRDR[B,H,W]
	-	-	-	11111111
0x10	-	-	IBSSCR[B,H,W]	
	-	-	-----0-- -----00-	
0x14	-	-	IBSSSR[B,H,W]	
	-	-	-------0 --------	
0x18 – 0x3F	-	-	-	-

1.19 MFS

MFS ch.0 Base_Address : 0x4003_8000

MFS ch.1 Base_Address : 0x4003_8100

MFS ch.3 Base_Address : 0x4003_8300

MFS ch.4 Base_Address : 0x4003_8400

MFS ch.6 Base_Address : 0x4003_8600

MFS ch.7 Base_Address : 0x4003_8700

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0–00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	RDR/TDR[H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	BGR1[B,H,W]	BGR0[B,H,W]
			00000000	00000000
0x010	-	-	ISMK[B,H,W]	ISBA[B,H,W]
			--------	--------
0x014	-	-	FCR1[B,H,W]	FCR0[B,H,W]
			---00100	-0000000
0x018	-	-	FBYTE2[B,H,W]	FBYTE1[B,H,W]
			00000000	00000000
0x01C	-	-	SCSTR1/ EIBCR[B,H,W]	SCSTR0[B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000–0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x038	-	-	-	SCSFR2[B,H,W] 10000000
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x044 – 0x0FC	-	-	-	-

Note:

- *RDR/TDR register's higher 16 bits can be accessed by word operation in I^2S mode.*

1.20 CRC

CRC Base_Address : 0x4003_9000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W]
				-0000000
0x004	CRCINIT[B,H,W]			
	11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W]			
	11111111 11111111 11111111 11111111			

1.21 Watch Counter

Watch Counter Base_Address : 0x4003_A000

Base_Address	Register			
+ Address	**+3**	**+2**	**+1**	**+0**
0x000	-	WCCR[B,H,W]	WCRL[B,H,W]	WCRD[B,H,W]
		00–0000	--000000	--000000
0x004 – 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W]	
			-----000 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 – 0xFFC	-	-	-	-

1.22 RTC

RTC Base_Address : 0x4003_B000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	WTCR1[B,H,W]			
	00000000 00000000 ---00000 -00000-0			
0x004	WTCR2[B,H,W]			
	-------- -------- -----000 -------0			
0x008	WTBR[B,H,W]			
	-------- 00000000 00000000 00000000			
0x00C	WTDR[B,H,W]	WTHR[B,H,W]	WTMIR[B,H,W]	WTSR[B,H,W]
	--000000	--000000	-0000000	-0000000
0x010	-	WTYR[B,H,W]	WTMOR[B,H,W]	WTDW[B,H,W]
		00000000	---00000	-----000
0x014	ALDR[B,H,W]	ALHR[B,H,W]	ALMIR[B,H,W]	-
	--000000	--000000	-0000000	
0x018	-	ALYR[B,H,W]	ALMOR[B,H,W]	-
		00000000	---00000	
0x01C	WTTR[B,H,W]			
	-------- ------00 00000000 00000000			
0x020	-	-	WTCLKM[B,H,W]	WTCLKS[B,H,W]
			------00	-------0
0x024	-	WTCALEN[B,H,W]	WTCAL[B,H,W]	
		-------0	------00 00000000	
0x028	-	-	WTDIVEN[B,H,W]	WTDIV[B,H,W]
			------00	----0000
0x02C	-	-	-	WTCALPRD[B,H,W]
				--010011
0x030	-	-	-	WTCOSEL[B,H,W]
				-------0
0x034 – 0xFFC	-	-	-	-

1.23 Low-speed CR Prescaler

Low-speed CR Prescaler Base_Address : 0x4003_C000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W]
				--000000
0x000 – 0x0FC	-	-	-	-

1.24 Peripheral Clock Gating

Peripheral Clock Gating Base_Address : 0x4003_C100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1---- -------1 -------- 11-11-11			
0x004	MRST0[B,H,W]			
	-------- -------0 -------- 00-00-00			
0x008 – 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- -------- -------- ------11			
0x014	MRST1[B,H,W]			
	-------- -------- -------- ------00			
0x018 – 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- -----1–111----- -------0			
0x024	MRST2[B,H,W]			
	-------- -----0–000----- -------0			
0x028 – 0x0FC	-	-	-	-

1.25 Smart Card I/F

Smart Card I/F ch.1 Base_Address : 0x4003_C980

Base_Address + Address	Register			
	+3	+2	+1	+0
0x00	-	-	GLOBALCONTROL1[H,W] -0001000 00000000	
0x04	-	-	STATUS[H,W] --000000 00000001	
0x08	-	-	PORTCONTROL[H,W] 0000–00 00-0-0-0	
0x0C	-	-	DATA[H,W] -------0 00000000	
0x10	-	-	CARDCLOCK [H,W] 00000000 00101000	
0x14	-	-	BAUDRATE[H,W] 0000001 01110100	
0x18	-	-	GUARDTIMER[H,W] -------- 00000000	
0x1C	-	-	IDLETIMER[H,W] 00000000 00000000	
0x20	-	-	GLOBALCONTROL2[H,W] -------- ----1-00	
0x24	-	-	DATA_FIFO[H,W] -------0 00000000	
0x28	-	-	FIFO_LEVEL_READ[H,W] 00000000 00000000	
0x2C	-	-	FIFO_LEVEL_WRITE[H,W] 00000000 00000000	
0x30	-	-	FIFO_MODE[H,W] 00000000 ----0000	
0x34	-	-	FIFO_CLEAR_MSB_WRITE[H,W] -------- -------0	
0x38	-	-	FIFO_CLEAR_MSB_ READ[H,W] -------- -------0	
0x3C	-	-	-	-
0x40	-	-	IRQ_STATUS[H,W] -------- 00000000	
0x44- 0x7C	-	-	-	-

1.26 MFSI2S

MFSI2S ch.4 Base_Address : 0x4003_CA00

MFSI2S ch.6 Base_Address : 0x4003_CA80

Base_Address + Address	Register			
	+3	+2	+1	+0
0x00	-	-	CNTLREG[B, H,W] -----0-0 -0000-01	
0x04	-	-	I2SCLK[B, H,W] 00------ 00000000	
0x08	-	-	I2SST[B,H,W] ------00	I2SRST[B,H,W] 00000000
0x0C- 0x3C	-	-	-	-

1.27 USB

USB ch.0 Base_Address : 0x4004_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 00000000 00000000	
0x2110	-	-	HADR[B,H,W] -0000000	HRTIMER(2)[B,H,W] ------00
0x2114	-	-	HEOF(1/0)[B,H,W] --000000 00000000	
0x2118	-	-	HFRAME(1/0)[B,H,W] -----000 00000000	
0x211C	-	-	-	HTOKEN[B,H,W] 00000000
0x2120	-	-	UDCC[B,H,W] -------- 10100-00	
0x2124	-	-	EP0C[H,W] ------0- -1000000	
0x2128	-	-	EP1C[H,W] 01100001 00000000	
0x212C	-	-	EP2C[H,W] 0110000- -1000000	
0x2130	-	-	EP3C[H,W] 0110000- -1000000	
0x2134	-	-	EP4C[H,W] 0110000- -1000000	
0x2138	-	-	EP5C[H,W] 0110000- -1000000	
0x213C	-	-	TMSP[H,W] -----000 00000000	
0x2140	-	-	UDCIE[B,H,W] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EP0IS[H,W] 10---1-- --------	
0x2148	-	-	EP0OS[H,W] 100–00- -XXXXXXX	

Base_Address + Address	Register			
	+3	+2	+1	+0
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXX	
0x2150	-	-	EP2S[H,W] 100-000- -XXXXXXX	
0x2154	-	-	EP3S[H,W] 100-000- -XXXXXXX	
0x2158	-	-	EP4S[H,W] 100-000- -XXXXXXX	
0x215C	-	-	EP5S[H,W] 100-000- -XXXXXXX	
0x2160	-	-	EP0DTH[B,H,W] XXXXXXXX	EP0DTL[B,H,W] XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXXX	EP1DTL[B,H,W] XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXXX	EP2DTL[B,H,W] XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXXX	EP3DTL[B,H,W] XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXXX	EP4DTL[B,H,W] XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXXX	EP5DTL[B,H,W] XXXXXXXX
0x2178 – 0x217C	-	-	-	-

1.28 DSTC

DSTC Base_Address : 0x4006_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	DESTP[B,H,W] 00000000 00000000 00000000 00000000			
0x004	HWDESP[B,H,W] 00XXXXXX XXXXXX00 00000000 00000000			
0x008	SWTR[H] 00000000 00000000		CFG[B] 01000000	CMD[B] 00000001
0x00C	MONERS[B,H,W] 00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x010	DREQENB[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x014	DREQENB[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x018-0x02C				
0x030	HWINT[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x034	HWINT[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x038-0x04C				
0x050	HWINTCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x054	HWINTCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x058-0x06C				
0x070	DQMSK[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x078-0x08C				
0x090	DQMSKCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x098 – 0xFFC	-	-	-	-

1.29 MTB_DWT

MTB_DWT Base_Address : 0xF000_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 – 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD4	PID5[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD8	PID6[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFDC	PID7[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE0	PID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE4	PID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE8	PID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFEC	PID3[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF0	CID0[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF4	CID1[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF8	CID2[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

1.30 Fast GPIO

Fast GPIO Base_Address : 0xF800_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	-	-
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024 – 0x034	-	-	-	-
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	-	-
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064 – 0x074	-	-	-	-
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	-	-
0x080				M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088 – 0x0BF	-			
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8 – 0x0FC	-	-	-	-

1.31 VIR

VIR Base_Address : 0xF800_0100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	VIR00[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x004	VIR01[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	VIR02[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x00C	VIR03[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	VIR04[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x014	VIR05[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x018	VIR06[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x01C	VIR07[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x020	VIR08[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x024	VIR09[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x028	VIR10[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x02C	VIR11[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	VIR12[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x034	VIR13[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x038	VIR14[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x03C	VIR15[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x040	VIR16[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x044	VIR17[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x048	VIR18[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x04C	VIR19[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x050	VIR20[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x054	VIR21[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x058	VIR22[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x05C	VIR23[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x060	VIR24[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x064	VIR25[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x068	VIR26[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x06C	VIR27[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x070	VIR28[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x074	VIR29[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x078	VIR30[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x07C	VIR31[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

E. List of Notes

This section explains notes for each function.

1. Notes when High-speed CR Is Used for Master Clock

CODE: 9APRECAUTION-FM0-E03.0

1. Notes when High-speed CR Is Used for Master Clock

This section explains notes when the high-speed CR is used for the master clock.

The frequency of the high-speed CR varies depending on the temperature and/or the power supply voltage. The following table shows notes on each function macro when the high-speed CR is used for the master clock.

Furthermore, pay attention to notes when the high-speed CR is used as an input clock of the PLL and the master clock is selected for PLL.

Notes on each macro

Macro	Function/Mode	Notes
Internal Bus Clock	HCLK/FCLK/PCLK0/PCLK1	The maximum frequency of the high-speed CR shall not exceed the upper limit of the internal operation clock frequency specified in the "Data Sheet" of the product used.
Timer	Multi-function Timer Base Timer Watch Timer Dual Timer Watch Dog Timer Quadrature	The frequency variation of the high-speed CR should be considered for the timer count value of each macro.
A/D Converter	Sampling Time Compare Tim	Considering the frequency variation of the high-speed CR, the sampling time and the compare time of the A/D converter shall satisfy the specification specified in the "Data Sheet" of the product used.
USB	-	As the frequency accuracy does not meet the required specification, these macros cannot be used when the high-speed CR is used for the master clock.
Multi Function Serial Interface	UART	Because of the frequency change of the High-Speed CR, the Board rate set may turn worse more. In case of over range of board rate error, this Function/Mode cannot be used.
	CSIO, I2C, MFS-I2S	The frequency variation of the high-speed CR should be considered for the communication of each macro.
	LIN	As the required frequency accuracy cannot be met, this function cannot be used as master. As slave, this function can be used. As a slave, the specified baud rate has more error at the maximum/minimum frequency of high-speed clock. So, if the error limit of the baud rate is exceeded, this function cannot be used.
Smart-card interface	-	Because of the frequency change of the High-Speed CR, the Board rate set may turn worse more. In case of over range of board rate error, this Function/Mode cannot be used.

F. Major Changes

Spansion Publication Number: MN710-00001

Page	Section	Changes
Revision 1.0		
-	-	Initial release
Revision 2.0		
7	The target products in this manual	Added TYPE2-M0+ products.
22	CHAPTER1: System Overview 1.1 Bus Block Diagram	Revised Figure 1-1 Bus Block Diagram
24	CHAPTER1: System Overview 1.3 Memory Map	Revised Figure 1-2 Memory Map Added Note
26	CHAPTER1: System Overview 1.4 Peripheral Address Map	Revised Table 1-1 Peripheral Address Map
37	CHAPTER2-1:Clock 2. Configration	Revised Figure 2 1 Block Diagram of Clock Generation Unit
153	CHAPTER4:Resets 3.1 Reset Factors	Corrected Software Watchdog Reset (SWDGR) Corrected Anomalous Frequency Detection Reset (FCSR)
167	CHAPTER5: Low-voltage Detection 5.1 LVD_CTL	Corrected SVHI table
329	CHAPTER7:Interrupts 1. Overview	Added the following description: Each bit of IRQxxxMON register in the case of non-equipped in each product, is a reserved bit
751	CHAPTER16:DSTC 2.2 DSTC and system configuration	Revised DREQ → DREQENB
763	CHAPTER16:DSTC 3.1.5 Other DES settings	Revised the following description: 3.3.2 HW Transfer flow → 3.2.4 Control of HW Transfer
571	CHAPTER10:I/O Port	Revised Table 2-4 Fixed Priority of EPFR
574	CHAPTER10:I/O Port	Revised Table 4-1 Register List of the I/O Port
588	CHAPTER10:I/O Port 4.7 Extended Pin Function Setting Register	Added EPFR27, EPFR28, EPFR29, EPFR30
592	CHAPTER10:I/O Port 4.9 EPFR01	Revised the following description: Uses the internal macro pin CRTRIM for input of the input capture IC03. →Setting is prohibited.
Revision 3.0		
7	The target products in this manual	Added TYPE3-M0+ products.
21	CHAPTER 1:System Overview	Updated Figure1-2, Table1-1

Page	Section	Changes
217	CHAPTER 5-3:Low-voltage Detection (TYPE3)	Added this chapter newly.
237	CHAPTER6-1:Low Power Consumption Mode	Updated function of TYPE3-M0+product
299	CHAPTER6-2:Vbat domain	Updated function of TYPE2-M0+product
515	CHAPTER 7-8:Interrupts Top (TYPE3)	Added this chapter newly.
533	CHAPTER8 :External Interrupt NMI	Updated function of TYPE3-M0+product
601	CHAPTER10-1:I/O port	Updated function of TYPE3-M0+product
725	CHAPTER10-2:Fast GPIO	Updated function of TYPE3-M0+product
961	D. Register Map (TYPE3-M0+)	Added this chapter newly.

NOTE: Please see “Revision History” about later revised information.

Revision History

Document Revision History

Document Title:	32-Bit Microcontroller FM0+ Family Peripheral Manual
Document Number:	002-04969

Revision	ECN No.	Origin of Change	Description of Change
**	-	TOYO	New Specification
*A	5337586	KEMU	Updated to Cypress format. Updated description of CHAPTER2-1:Clock on page 39. Added description in Section 5. Peripheral Clock Gating Function Usage Precautions of CHAPTER 2-2: Peripheral Clock Gating on page 117. Updated description of CHAPTER2-3: High-Speed CR Trimming 5.2 High-Speed CR Oscillation Frequency Trimming Register (MCR_FTRM) on page 138. Updated description of CHAPTER5-1 Low-voltage Detection Overview on page 193. Updated description of CHAPTER5-2 Low-voltage Detection(TYPE1) on page 197. Updated description of CHAPTER5-3 Low-voltage Detection(TYPE2) on page 215. Updated description of CHAPTER5-4 Low-voltage Detection(TYPE3) on page 245. Detailed register function names of CHAPTER 6-2: Vbat Domain on page 333. Updated description of CHAPTER6-2: Vbat Domain 7.6 HIBRST Register on page 378. Updated description of CHAPTER7-5 Interrupts Top(TYPE2) on page 473. Updated description of CHAPTER7-6 Interrupts (TYPE2-A) on page 479. Updated description of CHAPTER7-7 Interrupts (TYPE2-B) on page 491. Updated Table 3-1 of CHA`TER 7-8: Interrupts (TYPE3) on page 507. Fixed Typo. In Register section of CHAPTER 13: Micro Trace Buffer Data Watchpoint and Trace on page 804. Updated description of CHAPTER 16: DSTC Section 2.2 on page 821. Updated Table 1-2 of APPENDIX Product Type List on page 900. Updated description of APPENDIX B Register Map (TYPE1-M0+) on page 901. Updated description of APPENDIX C Register Map (TYPE2-M0+) on page 955. Updated description of APPENDIX D Register Map (TYPE3-M0+) on page 1019.
*B	5764951	AESATM P8	Updated logo and Copyright.

*C	6045772	NOSU	Preface Added Microcontroller support information How to Use This Manual The target products in this manual Modified the part numbers in Table 1, 2, 3 from 10 digits to 8 digits discription CHAPTER 1: System Overview 1.3 Memory Map Removed CAN Prescaler, CAN ch.0 and ch.1 from Figure 1-2 1.4 Peripheral Address Map Removed CAN and CAN Prescaler from Table 1-1 CHAPTER 2-1: Clock 1 Overview Removed description of CAN Prescaler clock 2 Configuration Removed description of CAN Prescaler clock CHAPTER 2-2: Peripheral Clock Gating 4.5 Peripheral Clock Control Register 2 (CKEN2) Added a description of LCDCCK bit for products which doesn't have LCDC function. Removed CANCK[1:0] bits. 4.6 Peripheral Function Reset Control Register 2 (MRST2) Removed CANRST[1:0] bits CHAPTER 6-1: Low Power Consumption Mode 4 Standby Mode Setting Procedure Examples Removed CAN related descriptions from Figure 4-1, 4-2 and 4-3. 5.1 Operation of Deep Standby Mode Added a note of data retention area in SRAM to Table 5-3 5.2 Operation of Deep Standby RTC Mode Fixed incorrect descriptions about powered-off areas during deep standby modes. 5.3 Operation of Deep Standby Stop Mode Fixed incorrect descriptions about powered-off areas during deep standby modes. CHAPTER 7-2: Interrupts Top (TYPE1) Removed DMA related descriptions. CHAPTER 7-3: Interrupts (TYPE1-A) Removed DMA related descriptions. CHAPTER 7-4: Interrupts (TYPE1-B) Removed DMA related descriptions. CHAPTER 7-6: Interrupts (TYPE2-A) 3. Usage Precautions Removed a word "CAN-FD". CHAPTER 7-7: Interrupts (TYPE2-B) 3. Usage Precautions Removed a word "CAN-FD". CHAPTER 7-8: Interrupts (TYPE3) 4.5 Vector Indicate Register xx (VIRxx) Added a note of procedure to use VIRxx register for MFS interrupts.

			CHAPTER 10-1: I/O Port 2.6 Operation in Deep Standby Mode (TYPE3-M0+) Added a note of how to configure clock oscillation pins after power-on. 4.7 Extended Pin Function Setting Register (EPFRx) Removed description of CAN function from EPFR09 4.33 Special Port Setting Register (SPSR) Added a note of how to configure clock oscillation pins after power-on. CHAPTER 11: CRC 2.1 CRC Calculation Sequence Modified a word “DMAC” to “DMA” in Figure 2-1. CHAPTER 16: DSTC Fixed typo. InnerRelaod → InnerReload 4.6 Examples of Controlling DSTC Added a description of procedure to break off a Hardware Transfer of DSTC. Appendix A: Product Type 1. Product TYPE List Modified the part numbers in Table 1, 2, 3 from 10 digits to 8 digits discription Appendix B: Register Map (TYPE1-M0+) 1.18 GPIO Modified initial values of Bit[31:24] of EPFR09 from “0000 0000” to “---- ----“ Appendix D: Register Map (TYPE3-M0+) 1.13 GPIO Modified initial values of Bit[31:24] of EPFR09 from “0000 0000” to “---- ----“ 1.28 DMAC Removed DMAC related registers. Appendix E: List of Notes 1. Notes when High-speed CR Is Used for Master Clock Removed CAN from Notes of Macros