The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

MB91610 Series

32-Bit Microcontroller FR80 Hardware Manual

Spec. # 002-04478 Rev. *A

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intnl): 408.943.2600
http://www.cypress.com

Copyrights

© Cypress Semiconductor Corporation, 2008-2017. 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. 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-M3 are the trademarks of ARM Limited in the EU and other countries.

Preface

Thank you for your continued use of Cypress semiconductor products.

Read this manual and "Datasheet" thoroughly before using products in the MB91610 series.

Purpose of this manual and intended readers

This manual explains the functions and operations of the MB91610 series and describes how it is used. The manual is intended for engineers engaged in the actual development of products using the MB91610 series.

Note: FR is a product of Cypress.

Trademark

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

Sample programs and development environment

Cypress offers sample programs free of charge for using the peripheral functions of the FR80 family. Cypress also makes available descriptions of the development environment required for the MB91610 series. Feel free to use them to verify the operational specifications and usage of this Cypress microcontroller.

Microcontroller support information:

http://www.cypress.com/cypress-microcontrollers

Note: 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.

Related Manuals

The manuals used for this series 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.

Hardware manual

- FR80 Family MB91610 Series Hardware Manual (CM71-10148) (this manual)

Datasheet

- Microcontroller 32-bit Original FR80 Family MB91610 Series Datasheet (DS07-16907)

Programming manual

- FR80 Family Programming Manual (CM71-00104)

This manual explains a programming model and instructions for the FR80 family CPUs.

Hardware tool-related manual

- DSU-FR Emulator MB2198-01 Hardware Manual (CM71-00413)

This manual explains emulator handling and specifications, and it explains how to connect and operate the emulator.

Software tool-related manuals

- SOFTUNE™ Workbench Operation Manual for V6 (CM71-00328)

  This manual explains how to operate the integrated development environment called SOFTUNE and the development procedures.
- SOFTUNE™ Workbench User's Manual for V6 (CM71-00329)

  This manual explains the basic functions and dependent functions of SOFTUNE Workbench.
- SOFTUNE™ Workbench Command Reference Manual for V6 (CM71-00330)

  This manual explains the commands and built-in variables/functions of SOFTUNE Workbench.
- FR Family 32-Bit Microcontroller Embedded C Programming Manual For Application (CM71-00324)

  This manual describes the know-how for creating built-in systems using the C compiler fcc911 for the FR family. The manual explains how to create efficient C programs using the architecture of the FR family and provides the notes.
- FR Family SOFTUNE C/C++ Compiler Manual for V6 (CM81-00206)

  Refer to this manual when using SOFTUNE C/C++ compiler to create/develop application programs in C and C++.
- FR Family SOFTUNE™ Assembler Manual for V6 (CM71-00203)

  This manual explains the functions of SOFTUNE™ Assembler operating in Windows 98, Windows Me, Windows 2000, or Windows XP and how to use it.

- SOFTUNE™ Linkage Kit Manual for V6 (CM71-00327)

  This manual explains the functions of SOFTUNE™ Linkage Kit operating in Windows 98, Windows Me, Windows 2000, or Windows XP and how to use it.

  See the manual when developing an application program.
- FR Family Absolute Assembly List Generator Tool Manual (CM71-00305)

  This manual explains absolute assemble lists.
- FR-V/FR Family SOFTUNE C/C++ Analyzer Manual for V5 (CM81-00309)

  This manual explains the functions of C/C++ Analyzer and how to use it.
- FR-V/FR Family SOFTUNE C/C++ Checker Manual for V5 (CM81-00310)

  This manual explains the functions of C/C++ Checker and how to use it.

REALOS-Related Manuals

- REALOS μITRON3.0-related manuals
  - FR/F^{2}MC Family in Conformance with μITRON Specifications SOFTUNE™ REALOS™/FR/907/896 Configurator Manual (CM71-00322)

    This manual explains the functions and operations of SOFTUNE REALOS Configurator.
  - FR-V/FR/F^{2}MC Family in Conformance with μITRON Specifications SOFTUNE™ REALOS™/Analyzer Manual (CM81-00315)

    This manual explains the functions provided by SOFTUNE REALOS Analyzer and how to utilize the functions.
  - FR Family in Conformance with μITRON 3.0 Specifications SOFTUNE REALOS/FR User's Guide (CM71-00320)

    This manual explains the configuration/activation of REALOS/FR application systems.

    See the manual when performing comprehensive work for an entire system.
  - FR Family in Conformance with μITRON 3.0 Specifications SOFTUNE REALOS/FR Kernel Manual (CM71-00321)

    This manual explains the functions provided by SOFTUNE REALOS/FR and how to utilize the functions.

    See the manual when creating an application system or user program.
- REALOS μITRON4.0-related manuals
  - FR Family in Conformance with μITRON 4.0 Specifications SOFTUNE™ REALOS™/FR Spec.4 Programming Manual (CM81-00316)

    This manual explains the functions provided by SOFTUNE REALOS/FR Spec.4 and how to utilize the functions.
  - FR-V/FR Family in Conformance with μITRON 4.0 Specifications SOFTUNE™ REALOS™ Kernel Manual (CM81-00312)

    This manual explains the functions provided by SOFTUNE REALOS/FRV/FR Spec.4 and how to utilize the functions.
  - FR-V/FR Family in Conformance with μITRON 4.0 Specifications SOFTUNE™ REALOS™ Configurator Manual (CM81-00311)

    This manual explains the functions provided by SOFTUNE REALOS Configurator (GUI) and how to utilize the functions.
  - FR-V/FR /F^{2}MC Family in Conformance with μITRON Specifications SOFTUNE REALOS™ Analyzer Manual (CM81-00315)

    This manual explains the functions provided by SOFTUNE REALOS Analyzer and how to utilize the functions.

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 list

  The register list lists all the registers of this device. You can look up the name of a desired register on the list to find the address of its location or the page that explains it.

  The address where each register is located is not described in the text. To verify the address of a register, see "A.1 I/O Map", and "A.2 List of Registers".
- Search from the index

  You can look up the keyword such as the name of a peripheral function in the index to find the explanation of the function.

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.

Contents

1. Overview **23**

1.1 MB91610 Series Overview 24
1.2 MB91610 Series Product Configuration 29
1.3 MB91610 Series Block Diagram 30
1.4 Package Dimensions 31

2. Pins of the MB91610 Series **33**

2.1 Pin Assignment Diagram 34
2.2 Pin Functions 35
2.3 I/O Circuit Types 43
2.4 Setting Method for Pins 49

3. CPU **59**

3.1 Memory Space 60
3.2 Features of the Internal Architecture 62
3.3 Operation Modes 63
3.4 Pipeline 64
3.5 Overview of Instructions 66
3.5.1 Arithmetic Operation 66
3.5.2 Load and Store 66
3.5.3 Branch 66
3.5.4 Logical Operation and Bit Operation 67
3.5.5 Direct Addressing 67
3.5.6 Bit Search 67
3.5.7 Other 67
3.6 Basic Programming Model 68
3.7 Registers 69
3.7.1 General-purpose Registers (R0 to R15) 69
3.7.2 Program Status Register (PS) 70
3.7.3 Program Counter (PC) 74
3.7.4 Table Base Register (TBR) 74
3.7.5 Return Pointer (RP) 75
3.7.6 System Stack Pointer (SSP) 75
3.7.7 User Stack Pointer (USP) 76
3.7.8 Multiply & Divide Registers 76
3.8 Data Configuration 77
3.8.1 Bit Ordering 77
3.8.2 Byte Ordering 77
3.8.3 Word Alignment 78
3.9 Addressing 79
3.9.1 Direct Addressing Areas 79
3.9.2 20-bit Addressing Area 80

3.9.3 32-bit Addressing Area 80
3.9.4 Vector Table Initial Area 80
3.10 Branch Instructions 81
3.10.1 Operation with Delay Slots 81
3.10.2 Operation without Delay Slots 84
3.11 EIT (Exception, Interrupt, Trap) 85
3.11.1 EIT Sources 85
3.11.2 Return from EIT 85
3.11.3 Interrupt Level 86
3.11.4 I Flag 87
3.11.5 Interrupt Level Mask Register (ILM) 88
3.11.6 Level Mask for Interrupts 89
3.11.7 Interrupt Control Register (ICR) 89
3.11.8 System Stack Pointer (SSP) 89
3.11.9 Interrupt Stack 89
3.11.10 Table Base Register (TBR) 90
3.11.11 EIT Vector Table 90
3.11.12 Multi-EIT Processing 91
3.11.13 Operation 93
3.11.14 INT Instruction Operation 94
3.11.15 INTE Instruction Operation 94
3.11.16 Step Trace Trap Operation 95
3.11.17 Undefined Instruction Exception Operation 96
3.11.18 RETI Instruction Operation 96
3.11.19 Delay Slots and EIT 96

4. Clock Generating Parts 97

4.1 Overview 98
4.2 Configuration 99
4.2.1 Clock Generating Parts 99
4.2.2 Source Clock (SRCCLK) Selection Block 102
4.3 Pins 103
4.4 Registers 104
4.4.1 Clock Source Select Register (CSELR) 104
4.4.2 Clock Source Monitor Register (CMONR) 107
4.4.3 Clock Stabilization Time Select Register (CSTBR) 109
4.4.4 PLL Configuration Register (PLLCR) 112
4.5 Explanation of Operations 115
4.5.1 Explanation of Clock Source Operations 115
4.5.2 Switching the Source Clock (SRCCLK) 118
4.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK) 120

5. Clock Division Control Part 121

5.1 Overview 122
5.2 Internal Clocks 123
5.3 Configuration 125
5.4 Registers 126
5.4.1 Divide Clock Configuration Register 0 (DIVR0) 127
5.4.2 Divide Clock Configuration Register 2 (DIVR2) 129
5.5 Division Rate 130

6. Main Timer 133

6.1 Overview 134
6.2 Configuration 135
6.3 Registers 136
6.3.1 Main Timer Control Register (MTMCR) 136
6.4 Interrupts 140
6.5 An Explanation of Operations and Setting Procedure Examples 141
6.5.1 Main Timer Operation 141
6.5.2 Transition to Stop Mode 142

7. Sub Timer 143

7.1 Overview 144
7.2 Configuration 145
7.3 Registers 146
7.3.1 Sub Timer Control Register (STMCR) 147
7.4 Interrupts 150
7.5 An Explanation of Operations and Setting Procedure Examples 151
7.5.1 Sub timer operation 151
7.5.2 Transition to Stop Mode, and Watch Mode 153

8. Low-power Dissipation Mode 155

8.1 Overview 156
8.2 Configuration 157
8.3 Registers 159
8.3.1 Standby Mode Control Register (STBCR) 160
8.3.2 Sleep Rate Configuration Register (SLPRR) 162
8.4 An Explanation of Operations and Setting Procedure Examples 163
8.4.1 Operation When Clock Control Is Set 164
8.4.2 Operation in Doze Mode 165
8.4.3 Operation in Sleep Mode 167
8.4.4 Operation in Main Timer Mode 169
8.4.5 Operation in Watch Mode 171
8.4.6 Operation in Stop Mode 173
8.5 Notes on Use 175

9. Reset 177

9.1 Overview 178
9.2 Configuration 179
9.3 Pins 181
9.4 Registers 182
9.4.1 Reset Result Register (RSTRR) 183
9.4.2 Reset Control Register (RSTCR) 185
9.5 Explanation of Operations 187
9.5.1 Reset Types 187
9.5.2 Reset Resource 188
9.5.3 Operation of Reset 190
9.5.4 Irregular Reset 195
9.6 Operating State and Transition 196

10. Interrupt Controller 201

10.1 Overview 202

10.2 Configuration 203
10.3 Registers 204
10.3.1 Interrupt Control Register (ICR00 to ICR47) 205
10.4 An Explanation of Operations and Setting Procedure Examples 207
10.4.1 Explanation of Operations of Interrupt Controller 207
10.5 Notes on Use 208

11. Interrupt Request Batch-Read Function 209

11.1 Overview 210
11.2 Configuration 211
11.3 Registers 212
11.3.1 Interrupt Request Batch-Read Register 0 Upper (IRPR0H) 213
11.3.2 Interrupt Request Batch-Read Register 1 Upper/Lower (IRPR1H/ IRPR1L) 214
11.3.3 Interrupt Request Batch-Read Register 2 Lower (IRPR2L) 216
11.3.4 Interrupt Request Batch-Read Register 3 Upper (IRPR3H) 217
11.3.5 Interrupt Request Batch-Read Register 4 Upper (IRPR4H) 218
11.3.6 Interrupt Request Batch-Read Register 5 Upper (IRPR5H) 219
11.3.7 Interrupt Request Batch-Read Register 5 Lower (IRPR5L) 220
11.3.8 Interrupt Request Batch-Read Register 7 Lower (IRPR7L) 221
11.4 Notes on Use 222

12. Delay Interrupt 223

12.1 Overview 224
12.2 Configuration 225
12.3 Registers 226
12.3.1 Delayed Interrupt Control Register (DICR) 227
12.4 An Explanation of Operations and Setting Procedure Examples 228
12.4.1 Explanation of Delay Interrupt Operations 228
12.5 Notes on Use 229

13. I/O Ports 231

13.1 Overview 232
13.2 Configuration 233
13.3 Pins 236
13.4 Registers 237
13.4.1 Port Data Direction Registers (DDR0 to DDRK) 238
13.4.2 Port Function Registers (PFR0 to PFR7) 240
13.4.3 Extended Port Function Registers (EPFR0 to EPFR34) 242
13.4.4 Port Data Registers (PDR0 to PDRK) 249
13.4.5 Pull-up Resistor Control Registers (PCR0 to PCR7) 250
13.4.6 A/D Channel Enable Register (ADCHE) 251
13.5 Notes on Use 252

14. External Interrupt Controllers 253

14.1 Overview 254
14.2 Configuration 255
14.3 Pins 257
14.4 Registers 258
14.4.1 External Interrupt Request Level Registers (ELVR0, ELVR1) 259
14.4.2 External Interrupt Request Registers (EIRR0, EIRR1) 261
14.4.3 Enable Interrupt Request Registers (ENIR0, ENIR1) 262
14.5 Explanation of Operations and Setting Procedure Examples 263

14.5.1 Operations of the External Interrupt Controllers....263
14.5.2 Return from Standby Mode....266
14.5.3 Return from Sleep Mode....268

15. Watchdog Timer 269

15.1 Overview....270
15.2 Configuration....271
15.3 Registers....273
15.3.1 Watchdog Timer Control Register 0 (WDTCR0)....274
15.3.2 Watchdog Timer Clear Pattern Register 0 (WDTCPR0)....276
15.3.3 Watchdog Timer Control Register 1 (WDTCR1)....277
15.3.4 Watchdog Timer Clear Pattern Register 1 (WDTCPR1)....278
15.4 Explanation of Operations and Setting Procedure Examples....279
15.4.1 Operations of the Watchdog Timer 0....279
15.4.2 Operations of the Watchdog Timer 1....281

16. Watch Counter 283

16.1 Overview....284
16.2 Configuration....285
16.3 Registers....287
16.3.1 Watch Counter Reload Register (WCRL)....288
16.3.2 Watch Counter Control Register (WCCR)....289
16.3.3 Watch Counter Read Register (WCRD)....291
16.4 Interrupts....292
16.5 Explanation of Operations and Setting Procedure Examples....293
16.5.1 Operations of the Watch Counter....293
16.6 Notes on Use....295

17. 32-bit Free-Run Timer 297

17.1 Overview....298
17.2 Configuration....299
17.3 Pins....302
17.4 Registers....303
17.4.1 Compare Clear Register (CPCLR0)....304
17.4.2 Timer Data Register (TCDT0)....305
17.4.3 Timer Status Control Register Upper/Lower (TCCSH0/TCCSL0)....306
17.5 Interrupts....309
17.6 An Explanation of Operations and Setting Procedure Examples....310
17.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected....311
17.6.2 Operation When an External Clock Is Selected....312

18. 32-bit Input Capture 313

18.1 Overview....314
18.2 Configuration....315
18.3 Pins....316
18.4 Registers....317
18.4.1 Input Capture Status Control Registers (ICS01, ICS23)....318
18.4.2 Input Capture Data Register (IPCP0 to IPCP3)....321
18.5 Interrupts....322
18.6 An Explanation of Operations and Setting Procedure Examples....323
18.6.1 Explanation of 32-bit Input Capture Operation....323

19. 32-bit Output Compare 325
19.1 Overview....326
19.2 Configuration....327
19.3 Pins....328
19.4 Registers....329
19.4.1 Output Compare Register (OCCP0 to OCCP3)....330
19.4.2 Compare Control Register Upper (OCSH1, OCSH3)....331
19.4.3 Compare Control Register Lower (OCSL0, OCSL2)....333
19.5 Interrupts....335
19.6 An Explanation of Operations and Setting Procedure Examples....336
19.6.1 When the 2 Channels Are Used Independently of Each Other....336
19.6.2 When the 2 Channels Are Used as a Pair....338

20. 16-bit Reload Timer 341
20.1 Overview....342
20.2 Configuration....343
20.3 Pins....345
20.4 Registers....346
20.4.1 Timer Control Status Register (TMCSR0 to TMCSR2)....347
20.4.2 16-bit Timer Reload Register A (TMRLRA0 to TMRLRA2)....352
20.4.3 16-bit Timer Register (TMR0 to TMR2)....352
20.5 Interrupts....353
20.6 An Explanation of Operations and Setting Procedure Examples....354
20.6.1 Operation in Interval Timer Mode....356
20.6.2 Operations in Event Counter Mode....368
20.6.3 Operation in Cascade Mode....374
20.7 Notes on Use....376

21. Base Timer I/O Select Function 377
21.1 Overview....378
21.2 Configuration....379
21.3 Pins....380
21.4 Registers....382
21.4.1 Base Timer IO Select Register for Ch.0/1/2/3 (BTSEL0123)....383
21.4.2 Base Timer IO Select Register for Ch.4/5/6/7 (BTSEL4567)....385
21.4.3 Base Timer Same Time Soft Start Register (BTSSSR)....387
21.5 I/O Mode....390
21.5.1 I/O Mode 0 (16-bit Timer Standard Mode)....390
21.5.2 I/O Mode 1 (Timer Full Mode)....391
21.5.3 I/O Mode 2 (External Trigger Shared Mode)....392
21.5.4 I/O Mode 3 (Other Channel Trigger Shared Mode)....394
21.5.5 Operations in I/O Mode 4 (Timer Activation/Stop Mode)....396
21.5.6 Operations in I/O Mode 5 (Same Time Software Activation Mode)....398
21.5.7 Operations in I/O Mode 6 (Software Activation Timer Activation/Stop Mode)....400
21.5.8 Operations in I/O Mode 7 (Timer Activation Mode)....402
21.5.9 Operations in I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode) 404

22. Base Timer 407
22.1 Overview of the Base Timer....408
22.2 Block Diagrams of the Base Timer....410
22.3 Base Timer's Registers....416

22.4 Operations of the Base Timer 421
22.5 32-bit Mode Operations 423
22.6 Notes of Using the Base Timer 425
22.7 Base Timer Interrupts 426
22.8 Base Timer Description by Function Mode 427
22.8.1 PWM Function 427
22.8.2 PPG Function 441
22.8.3 Reload Timer Function 455
22.8.4 PWC Function 466

23. 10-Bit A/D Converter 481

23.1 Overview 482
23.2 Configuration 483
23.3 Pins 485
23.4 Registers 486
23.4.1 A/DC Control Registers (ADCR0) 487
23.4.2 A/DC Status Registers (ADSR0) 490
23.4.3 Scan Conversion Control Registers (SCCR0) 493
23.4.4 Scan Conversion FIFO Number Setting Register (SFNS0) 496
23.4.5 Scan Conversion FIFO Data Registers (SCFD0) 498
23.4.6 Scan Conversion Input Select Registers (SCIS00) 501
23.4.7 Priority Conversion Control Registers (PCCR0) 502
23.4.8 Priority Conversion FIFO Number Setting Registers (PFNS0) 504
23.4.9 Priority Conversion FIFO Data Registers (PCFD0) 505
23.4.10 Priority Conversion Input Select Registers (PCIS0) 508
23.4.11 A/D Comparison Data Setting Registers (CMPD0) 510
23.4.12 A/D Comparison Control Registers (CMPCR0) 511
23.4.13 Sampling Time Setting Registers (ADST00, ADST10) 513
23.4.14 Sampling Time Select Registers (ADSS00) 516
23.4.15 Compare Time Setting Registers (ADCT0) 517
23.5 Interrupts 518
23.6 Explanation of Operations and Setting Procedure Examples 519
23.6.1 Operation of A/D Scan Conversion 526
23.6.2 Operation of A/D Priority Conversion 529
23.6.3 FIFO Operations 531
23.6.4 Activating the DMA Controller (DMAC) 536

24. Multi-function Serial Interface 539

24.1 Characteristics of Multi-function Serial Interface 540
24.2 UART (Asynchronous Serial Interface) 541
24.3 Overview of UART (Asynchronous Serial Interface) 542
24.4 Registers of UART (Asynchronous Serial Interface) 543
24.4.1 Serial Control Register (SCR) 546
24.4.2 Serial Mode Register (SMR) 548
24.4.3 Serial Status Register (SSR) 550
24.4.4 Extended Serial Control Register (ESCR) 552
24.4.5 Reception Data Register / Transmission Data Register (RDR/TDR) 554
24.4.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 556
24.4.7 FIFO Control Register 1 (FCR1) 557
24.4.8 FIFO Control Register 0 (FCR0) 559
24.4.9 FIFO Byte Register (FBYTE1/FBYTE2) 561
24.5 Interrupts of UART 563

24.5.1 Occurrence of Reception Interrupts and Flag Set Timing 564
24.5.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing 565
24.5.3 Occurrence of Transmission Interrupts and Flag Set Timing 566
24.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing 567
24.6 Operation of UART 568
24.7 Dedicated Baud Rate Generator 572
24.7.1 Setting Baud Rate 573
24.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode) 577
24.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)579
24.10 Notes on UART Mode 582
24.11 CSIO (Clock Synchronous Serial Interface) 583
24.12 Overview of CSIO (Clock Synchronous Serial Interface) 584
24.13 Registers of CSIO (Clock Synchronous Serial Interface) 585
24.13.1Serial Control Register (SCR) 588
24.13.2Serial Mode Register (SMR) 590
24.13.3Serial Status Register (SSR) 592
24.13.4Extended Serial Control Register (ESCR) 594
24.13.5Reception Data Register / Transmission Data Register (RDR/TDR) 596
24.13.6Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 598
24.13.7FIFO Control Register 1 (FCR1) 599
24.13.8FIFO Control Register 0 (FCR0) 601
24.13.9FIFO Byte Register (FBYTE1/FBYTE2) 603
24.13.10Serial Mode Select Registers (SSEL0123) 605
24.13.11Received Data Mirror Registers/Transmitted Data Mirror Registers (RDRM/TDRM) ... 606
24.14 Interrupts of CSIO (Clock Synchronous Serial Interface) 607
24.14.1Occurrence of Reception Interrupts and Flag Set Timing 608
24.14.2Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing 609
24.14.3Occurrence of Transmission Interrupts and Flag Set Timing 611
24.14.4Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing 612
24.15 Operation of CSIO (Clock Synchronous Serial Interface) 613
24.16 Dedicated Baud Rate Generator 631
24.16.1Setting Baud Rate 631
24.17 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface) 634
24.18 Notes on CSIO Mode 636
24.19 I^2C Interface 637
24.20 Overview of I^2C Interface 638
24.21 Registers of I^2C Interface 639
24.21.1I^2C Bus Control Register (IBCR) 642
24.21.2Serial Mode Register (SMR) 646
24.21.3I^2C Bus Status Register (IBSR) 648
24.21.4Serial Status Register (SSR) 651
24.21.5Reception Data Register / Transmission Data Register (RDR/TDR) 653
24.21.67-bit Slave Address Mask Register (ISMK) 655
24.21.77-bit Slave Address Register (ISBA) 656
24.21.8Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 657
24.21.9FIFO Control Register 1 (FCR1) 658
24.21.10FIFO Control Register 0 (FCR0) 660
24.21.11FIFO Byte Register (FBYTE1/FBYTE2) 663
24.22 Interrupts of I^2C Interface 665
24.22.1Operation of I^2C Interface Communication 666
24.22.2Master Mode 667

24.22.3Slave Mode 684
24.22.4Bus Error 688
24.23 Dedicated Baud Rate Generator 689
24.23.1Example of I2C Flowcharts 691
24.24 Notes on I2C Mode 703

25. USB Clock Generating Part 705

25.1 Overview 706
25.2 Configuration 707
25.3 Registers 708
25.3.1 USB Clock Configuration Register (UCCR) 709
25.4 Explanation of Operations and Setting Procedure Examples 711

26. DMA Transfer Request Selector 713

26.1 Overview 714
26.2 Configuration 715
26.3 Registers 716
26.3.1 DREQ Select Register (DREQSEL) 717

27. USB Function 719

27.1 ..... Overview720
27.2 Configuration 721
27.3 Registers 722
27.3.1 USB Selection Register (USBSEL) 724
27.3.2 USB Enable Register (USBEN) 725
27.3.3 UDC Control Register (UDCC) 726
27.3.4 EP0 Control Register (EP0C) 728
27.3.5 EP1 to 5 Control Registers (EP1C to EP5C) 730
27.3.6 Time Stamp Register (TMSP) 733
27.3.7 UDC Status Register (UDCS) 734
27.3.8 UDC Enable Interrupt Request Register (UDCIE) 736
27.3.9 EP0I Status Register (EP0IS) 738
27.3.10EP0O Status Register (EP0OS) 740
27.3.11EP1 to 5 Status Registers (EP1S to EP5S) 743
27.3.12EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) 746
27.4 Explanation of Operations and Setting Procedure Examples 748
27.4.1 Detection of Connections and Disconnections 750
27.4.2 Register Operation for a Command Response 753
27.4.3 STALL Response and Release 755
27.4.4 Suspend Function 759
27.4.5 Wakeup Function 760
27.4.6 DMA Transfer Function 762
27.4.7 NULL Transfer Function 767
27.4.8 Software Control Examples 768

28. USB Host 775

28.1 Overview 776
28.2 Configuration 777
28.3 Registers 778
28.3.1 Host Control Register 0, 1 (HCNT) 780
28.3.2 Host Interrupt Register (HIRQ) 785
28.3.3 Host Error Status Register (HERR) 788

28.3.4 Host Status Register (HSTATE) 791
28.3.5 SOF Interrupt FRAME Comparison Register (HFCOMP) 794
28.3.6 Retry Timer Setting Register (HRTIMER) 795
28.3.7 Host Address Register (HADR) 797
28.3.8 EOF Setting Register (HEOF) 798
28.3.9 FRAME Setting Register (HFRAME) 800
28.3.10 Host Token Endpoint Register (HTOKEN) 802
28.4 Explanation of Operations and Setting Procedure Examples 804
28.4.1 Connecting a USB Device 804
28.4.2 Resetting the USB Bus 806
28.4.3 Token Packets 807
28.4.4 Data Packets 809
28.4.5 Operation of a Handshake Packet 809
28.4.6 Retry Function 810
28.4.7 SOF Interrupts 811
28.4.8 Error Status 813
28.4.9 Packet End 814
28.4.10 J-ST: J State 815
28.4.11 Suspend and Resume 815
28.4.12 Disconnecting a USB Device 818
28.4.13 Flowcharts for tokens in operation with USB HOST 819

29. Remote Control Reception 825

29.1 Overview of Remote Control Reception 826
29.2 Remote Control Reception Registers 827
29.2.1 Remote Control Reception Control Register (RCCR) 828
29.2.2 Remote Control Reception Interrupt Control Register (RCST) 830
29.2.3 Device Address Configuration Register 1, 2 (RCADR1, RCADR2) 832
29.2.4 Start Bit High Width Configuration Register (RCSHW) 833
29.2.5 High Width Configuration Register A (RCDAHW) 834
29.2.6 High Width Configuration Register B (RCDBHW) 835
29.2.7 Data Store Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL) 836
29.2.8 Clock Division Configuration Register (RCCKD) 837
29.3 Description of Remote Control Reception Operation and Example Configuration Procedures .. 838

30. OSDC 841

30.1 OSDC Specifications 842
30.1.1 Features 843
30.1.2 Block Diagram 845
30.2 Display Functions 846
30.2.1 Screen Configuration 847
30.2.2 Screen Display Modes 850
30.2.3 Screen Output Control 853
30.2.4 Screen Display Position Control 854
30.2.5 Font Memory Configuration 861
30.2.6 Display Memory (VRAM) Configuration 862
30.2.7 Writing to Display Memory (VRAM) 863
30.2.8 Palette Configuration 865
30.2.9 Character Display 866
30.2.10 Character Background Display 906
30.2.11 Line Background Display 917

30.2.12Screen Background Display 925
30.2.13Sprite Character Display 933
30.3 Control Functions 936
30.3.1 Dot Clock Control 937
30.3.2 Sync Signal Input 940
30.3.3 Display Signal Output 945
30.3.4 Display Period Control 947
30.3.5 Synchronization Control 949
30.3.6 Interrupt Control 952
30.3.7 OSDC Operation Control 955
30.4 Display Control Commands (Main/OSDC Operation) 959
30.4.1 List of Main Screen Display/OSDC Operation and Control Commands 960
30.4.2 VRAM Write Address Set (Command 0) 962
30.4.3 Character Data Set (Command 1, Command 2) 963
30.4.4 Line Control Data Set (Command 3, Command 4) 966
30.4.5 Screen Output Control (Command 5-0) 969
30.4.6 Screen Display Position Control (Command 5-1) 970
30.4.7 Transparent Color Control (Command 6-0) 971
30.4.8 Graphics Color Control (Command 6-1) 972
30.4.9 Screen Background Character Control (Command 7) 973
30.4.10Screen Background Control (Command 8-0) 974
30.4.11Window Period Control (Command 8-1, Command 8-2) 975
30.4.12Sprite Character Control (Command 9-0, Command 9-1) 977
30.4.13Synchronization Control (Command 11) 979
30.4.14Character Background Character Code Set (Command 12-0 to Command 12-7) 980
30.4.15I/O Pin Control (Command 13) 981
30.4.16Display Period Control (Command 14-0, Command 14-1) 982
30.4.17Interrupt Control (Command 15) 983
30.4.18Shaded Background Frame Color Control (Command 15-0 to Command 15-3) 984
30.4.19Palette Control (Command 16-0 to Command 16-127) 985
30.4.20Operation Control (Command 17) 986
30.4.21PLL Clock Control (Command 18) 988
30.5 Display Control Commands (Sub Operation) 989
30.5.1 List of Sub Screen Display and Control Commands 990
30.5.2 VRAM Write Address Set (Command 0) 991
30.5.3 Character Data Set (Command 1, Command 2) 992
30.5.4 Line Control Data Set (Command 3, Command 4) 995
30.5.5 Screen Output Control (Command 5-0) 998
30.5.6 Screen Display Position Control (Command 5-1) 999
30.5.7 Transparent Color Control (Command 6-0) 1000
30.5.8 Graphics Color Control (Command 6-1) 1001
30.5.9 Screen Background Control (Command 8-0) 1002
30.5.10Window Period Control (Command 8-1, Command 8-2) 1003
30.5.11Sprite Character Control (Command 9-0, Command 9-1) 1004
30.5.12Interrupt Control (Command 15) 1006
30.5.13Shaded Background Frame Color Control (Command 15-0 to Command 15-3) 1007
30.6 Display Control Command Write Conditions and Update Timing 1008
30.6.1 List of Display Control Command Write Conditions and Update Timing 1009
30.7 Font Memory • CPU Read Access 1016

31. DMA Controller (DMAC) 1017

31.1 Overview 1018
31.2 Configuration 1021

31.3 Registers 1023
31.3.1 DMA Cont2rol Register (DMACR) 1025
31.3.2 DMA Source Address Registers (DSAR0 to DSAR7) 1027
31.3.3 DMA Destination Address Registers (DDAR0 to DDAR7) 1028
31.3.4 DMA Transfer Count Registers (DTCR0 to DTCR7) 1029
31.3.5 DMA Channel Control Registers (DCCR0 to DCCR7) 1030
31.3.6 DMA Channel Status Registers (DCSR0 to DCSR7) 1038
31.3.7 DMA-Halt by Interrupt Level Register (DILVR) 1040
31.4 Interrupts 1041
31.5 An Explanation of Operations and Setting Procedure Examples 1042
31.5.1 Transfer Settings 1042
31.5.2 Transfer Operations 1045
31.5.3 Transfer Suspension 1051
31.5.4 Operation at the End of Transfer 1053
31.5.5 Post-transfer Operation 1054
31.5.6 DMA Transfer Halt 1057

32. Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function 1059

32.1 Overview 1060
32.2 Configuration 1061
32.3 Registers 1063
32.3.1 IO-Data Request Registers (IORR0 to IORR7) 1064
32.3.2 Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0). 1067
32.3.3 Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1). 1068
32.3.4 Select Register 4 for DMA Transfer Request Clear by a Peripheral Function (ICSEL4). 1069
32.3.5 Select Register 6 for DMA Transfer Request Clear by a Peripheral Function (ICSEL6). 1070
32.3.6 Select Register 7 for DMA Transfer Request Clear by a Peripheral Function (ICSEL7). 1071
32.3.7 Select Register 8 for DMA Transfer Request Clear by a Peripheral Function (ICSEL8). 1072
32.3.8 Select Register 10 for DMA Transfer Request Clear by a Peripheral Function (ICSEL10) 1073
32.3.9 Select Register 11 for DMA Transfer Request Clear by a Peripheral Function (ICSEL11) 1076
32.4 An Explanation of Operations and Setting Procedure Examples 1079
32.4.1 Operations upon a DMA Transfer 1079

33. Control of Built-in Program Memory 1081

33.1 Overview of Built-in Program Memory Controller 1082
33.2 Register for Built-in Program Memory Controller 1083
33.2.1 FLASH Control Register (FCTLR) 1084

34. Flash Memory 1087

34.1 Overview 1088
34.2 Configuration 1089
34.3 Registers 1091
34.3.1 FLASH Status Register (FSTR) 1092

34.3.2 FLASH Control Register (FCTLR) 1092
34.4 Flash Memory Access Mode 1093
34.5 Automatic Programming Algorithm 1094
34.5.1 Command Sequence 1094
34.5.2 Execution State of Automatic Programming Algorithm 1097
34.6 Explanation of Flash Memory Operation 1102
34.6.1 Read/Reset Operation 1102
34.6.2 Write Operation 1102
34.6.3 Chip Erase 1105
34.6.4 Sector Erase 1105
34.6.5 Sector Erase Suspending 1108
34.6.6 Sector Erase Restarting 1109
34.7 Notes on Using Flash Memory 1110

35. Wild Register 1111

35.1 Overview 1112
35.2 Configuration 1113
35.3 Registers 1114
35.3.1 Wild Register Address Register (WRAR00 to WRAR15) 1115
35.3.2 Wild Register Data Register (WRDR00 to WRDR15) 1116
35.3.3 Wild Register Enable Register (WREN) 1117
35.4 Explanation of Operations and Setting Procedure Examples 1118
35.4.1 Wild Register Operation 1118
35.5 Notes on Using the Wild Register 1119

36. Serial Programming Connection 1121

36.1 Cypress Serial Programmer 1122
36.1.1 Pins Used 1129

37. Handling the Device 1131

37.1 Notes on Handling the Device 1132

A. Appendix 1141

A.1 I/O Map 1142
A.2 List of Registers 1161
A.3 Interrupt Vectors 1175
A.4 Pin State in Each CPU State 1178
A.5 PLL for Dot Clock Generation 1185
A.6 Lists of Instructions 1186
A.6.1 Instruction List 1186
A.6.2 Instruction Tables 1189
A.6.3 List of Instructions That Can Be Specified for Delay Slots 1199

Main changes in this edition 1201

Revision History 1205

1. Overview

This chapter explains the features and basic specifications of the MB91610 series.

1.1 MB91610 Series Overview

1.2 MB91610 Series Product Configuration

1.3 MB91610 Series Block Diagram

1.4 Package Dimensions

1.1 MB91610 Series Overview

The MB91610 series, a microcontroller that uses 32-bit RISC CPUs, has built-in peripheral control functions for embedded control which requires high-performance/high-speed CPU processing.

This series is based on the FR80 family CPUs and is implemented in a single-chip.

FR80 family CPUs

- 32-bit RISC, load/store architecture, 5-stage pipeline
- 16 general-purpose 32-bit registers
- 16-bit fixed-length instructions (basic instructions), 1 instruction per cycle
- Instructions suitable for embedded applications
  - Instructions for memory-to-memory transfer, bit processing, barrel shift, etc.
  - High-level language support instructions
    Function entry/exit instructions and multi-load/store instructions for register contents
  - Bit search instruction
    1 detection, 0 detection, and transition point detection
  - Branch instruction with delay slot(s)
    Reduced overhead time in branch executions
  - Register interlock function
    Efficient assembly language coding
  - Support for multipliers at the built-in function/instruction level
    Signed 32-bit multiplication - 5 cycles
    Signed 16-bit multiplication - 3 cycles
  - Interrupt (Save PC and PS)
    High-speed response at a minimum of 6 cycles, 16 levels of priority
  - Simultaneous access to a program and data enabled by Harvard architecture
  - The prefetch function for instructions using the 4-word instruction queue in the CPU
- Basic instruction compatibility with the FR family CPUs
  - Addition of the bit search instruction
  - No resource instruction and coprocessor instruction provided

Maximum operating frequency

CPU	33 MHz
Peripheral	33 MHz

DMA controller (DMAC)

- Number of channels: 8
- Address space: 32 bits (4 GB)
- Transfer mode: Block transfer/Burst transfer/Demand transfer
- Address update: Increment/Decrement/Fix (increment/decrement value fixed to 1, 2, or 4)
- Transfer size: 8 bits, 16 bits, and 32 bits
- Block size: 1 to 16
- Transfer count: 1 to 65,535 times
- Transfer request:
  - Request by software
  - Interrupt request of a built-in peripheral function (a shared interrupt request or external interrupt request)
- Reload function: Reloading of all channels can be specified.
- Level of priority: Fixed (ch.0 > ch.1 > ch.2 > ch.3 > ...), or round robin
- Interrupt request: Occurrence of a normal end interrupt request, abnormal end interrupt request, or transfer suspension interrupt request

Multifunction serial interface

- 4 channels with 16-byte FIFO, 4 channels without FIFO
- Any of the following uses can be selected for each channel (For ch.0, I^2C is not available.):
  - UART
  - CSIO
  - I^2C

[Features of UART]

- Full-duplex double buffer
- Selection with or without parity supported
- Built-in dedicated baud rate generator
- External clock available as a serial clock
- Various error detection functions available (parity errors, framing errors, and overrun errors)

[Features of CSIO]

- Full-duplex double buffer
- Built-in dedicated baud rate generator
- Overrun error detection function available

[Features of I^2C]

- Standard mode (maximum: 100 kbps)/High-speed mode (maximum: 400 kbps) supported
- 5V tolerance supported for some channels

Interrupts

- Total of 16 external interrupts (5V tolerance supported for some pins)
- Interrupt from an internal peripheral function
- Programmable setting of interrupt levels (16 levels)
- Return from stop mode or sleep mode supported

A/D converter

- 8 channels, 1 unit
- 10-bit resolution
- Successive comparison type Conversion time: Approximately 1.2 μs (PCLK=33 MHz)
- Priority A/D conversion available (2 levels)
- Conversion mode (one shot conversion mode, scanning conversion mode)
- Activation trigger (software/external trigger/base timer)
- FIFO for storing conversion data available (scanning conversion: 16 levels; priority conversion: 4 levels)

Base timer

- Number of channels: 8 built-in channels
- Any of the following uses can be selected for each channel:
  - 16/32-bit reload timer
  - 16-bit PWM timer
  - 16/32-bit PWC timer
  - 16-bit PPG timer
- 32-bit timer available by connecting 2 channels in cascade
- Function for activating multiple channels simultaneously available
- I/O select function available

16-bit reload timer

- Number of channels: 3 (including a channel for REALOS)
- Interval timer function
- Function for selecting count clock (Peripheral clock (PCLK) divided by a value ranging from 2 to 64)

Compare timer

- 32-bit input capture: 4 built-in channels
- 32-bit output compare: 4 built-in channels
- 32-bit free-run timer: 1 built-in channels

Other interval timers

- Watch counter: 1 built-in channel
- Watchdog timer: 2 built-in channel

[Watchdog timer 0]

- After resetting this device, the watchdog timer becomes active when an arbitrary value is written to the WDTCPR0 register.
- The cycle of the watchdog timer 0 can be selected from the peripheral clock (PCLK) × (2^9 to 2^{24}).

[Watchdog timer 1]

- After releasing the reset of this device, it counts with the CPU clock (CCLK).
- Disable/ enable of the counter operation can be controlled by HWDE pin.
- The cycle of the watchdog timer 1 is CCLK × 2^{23} cycle fixed.

USB function/HOST

- Number of channels: 1
- Supports Full-Speed only
- The USB function and USB HOST are the switch types (USB I/O multiplexed)
- Support of DMA transfer

[USB Function]

- Support of up to six endpoints
  - Endpoint 0 is provided for the fixed use of control transfers
  - Bulk or interrupt transfer can be selected for endpoint 1 to 5
- Double buffer structure for endpoint 1 to 5

[USB HOST]

- Support control transfer, bulk transfer, interrupt transfer, and isochronous transfer
- Automatic detection of connection/disconnection of USB devices
- Automatic processing of a handshake packet for IN/OUT token processing
- Support of a maximum packet length of up to 256 bytes
- Support for a wakeup function

HDMI-CEC/ Remote Control Reception

- 1 channel
- HDMI-CEC reception function (with automatic ACK response function)
- Remote control reception function (built-in 4-byte receive buffer)

OSDC function

- 16 bits RGB (256 colors available among 65536 colors)
- Analog RGB output: Max 50 MHz
  Digital RGB output: Max 75 MHz
- A font in 32 × 32 dots can be displayed up to 60 × 32
- Two-layered display of MAIN/SUB
- 16384 characters at the maximum
- Equipped with one PLL for dot clock generation

Main timer

- Number of channels: 1
- Count of the oscillation stabilization wait time of the main clock (MCLK).
- Count of the oscillation stabilization wait time of the PLL clock (PLLCLK).
- Interval timer when the oscillation of the main clock (MCLK) is stable.

Sub timer

- Number of channels: 1
- Count of the oscillation stabilization wait time of the sub clock (SBCLK).
- Interval timer when the oscillation of the sub clock (SBCLK) is stable.

Clock generation

- Main clock (MCLK) oscillation
- Sub clock (SBCLK) oscillation
- PLL clock (PLLCLK) oscillation

Low-power dissipation mode

- Stop mode
- Watch mode
- Sleep mode
- Doze mode
- Clock division function

Other features

- I/O port
- $\overline{\text{INIT}}$ pin available as a reset pin.
- Watchdog timer reset and software reset available
- Delay interrupt
- Power supply
  - Single power supply (3.0 V to 3.6 V)

1.2 MB91610 Series Product Configuration

This section explains the products in the MB91610 series.

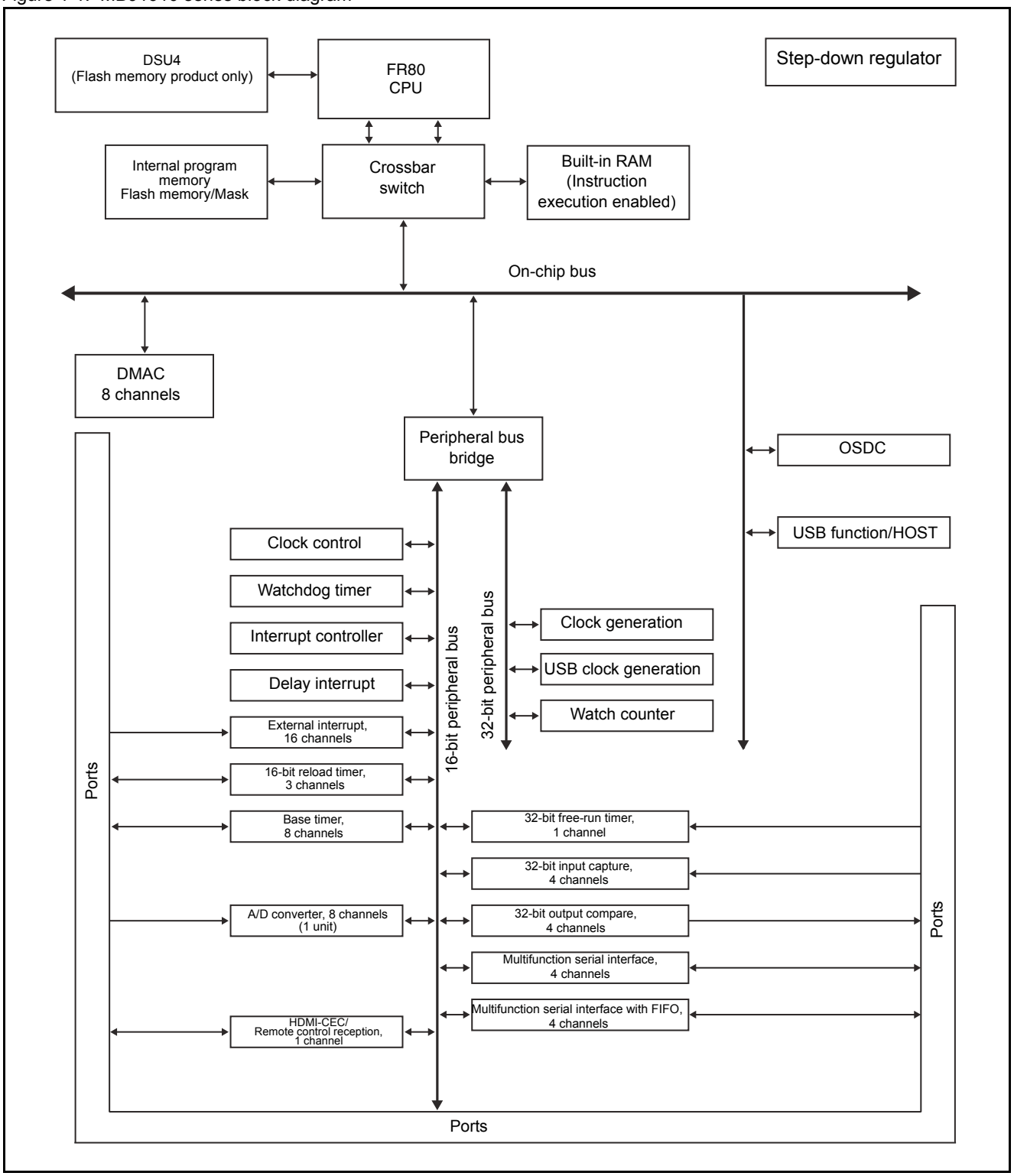
Table 1-1. MB91610 series product configuration

Items \ Product Name	MB91F610A	MB91613
Product type	Flash memory product	MASK ROM product
Built-in program memory capacity	512 Kbytes (Flash)	512 Kbytes (ROM)
Built-in RAM capacity (Instruction execution enabled)	32Kbytes	
DMA controller (DMAC)	8 channels	
Base timer	8 channels	
Multifunction serial interface	Without FIFO : 4 channels (ch.0 to ch.3) With FIFO : 4 channels (ch.8 to ch.11)	
External interrupt	16 channels	
10-bit A/D converter	8 channels (1 unit)	
16-bit reload timer	3 channels	
Compare timer	32-bit input capture : 4 channels 32-bit output compare : 4 channels 32-bit free-run timer : 1 channel	
Watch counter	1 channel	
I/O port	50 (Max)	
USB function/HOST	1 channel	
HDMI-CEC/ Remote control reception	1 channel	
OSDC	Font FLASH : 16384 characters	Font ROM : 7168 characters
Main timer	1 channel	
Sub timer	1 channel	
Wild register	16 channels	
Debug function	DSU4	-
Package	Type: LQFP-120 Package code: FPT-120P-M21 Pin pitch: 0.50mm Size:16.0mm × 16.0mm	

1.3 MB91610 Series Block Diagram

Figure 1-1 is the block diagrams of the MB91610 series.

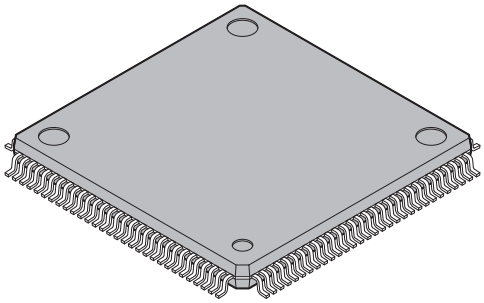
Figure 1-1. MB91610 series block diagram



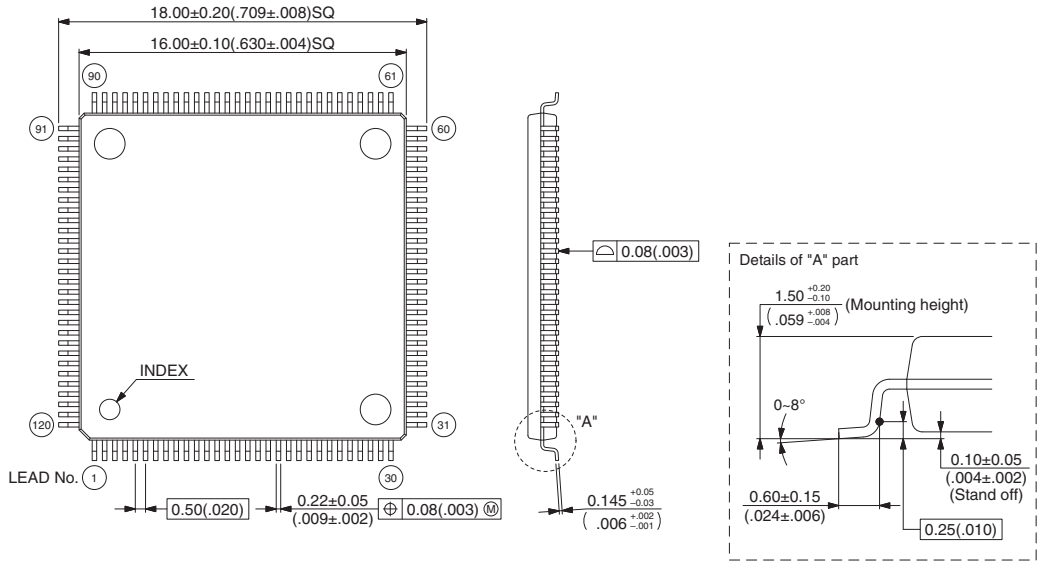
1.4 Package Dimensions

The dimensions of the packages used for the MB91610 series are shown below.

Figure 1-2. Package dimensions (FPT-120P-M21)

120-pin plastic LQFP		
	Lead pitch	0.50 mm
	Package width × package length	16.0 × 16.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.88 g
(FPT-120P-M21)		

120-pin plastic LQFP
(FPT-120P-M21)



© 2001-2008 FUJITSU MICROELECTRONICS LIMITED F120033S-c-3-4

Dimensions in mm (inches).
Note: The values in parentheses are reference values.

2. Pins of the MB91610 Series

This chapter explains the pins and multiplexed pin settings of the MB91610 series.

2.1 Pin Assignment Diagram

2.2 Pin Functions

2.3 I/O Circuit Types

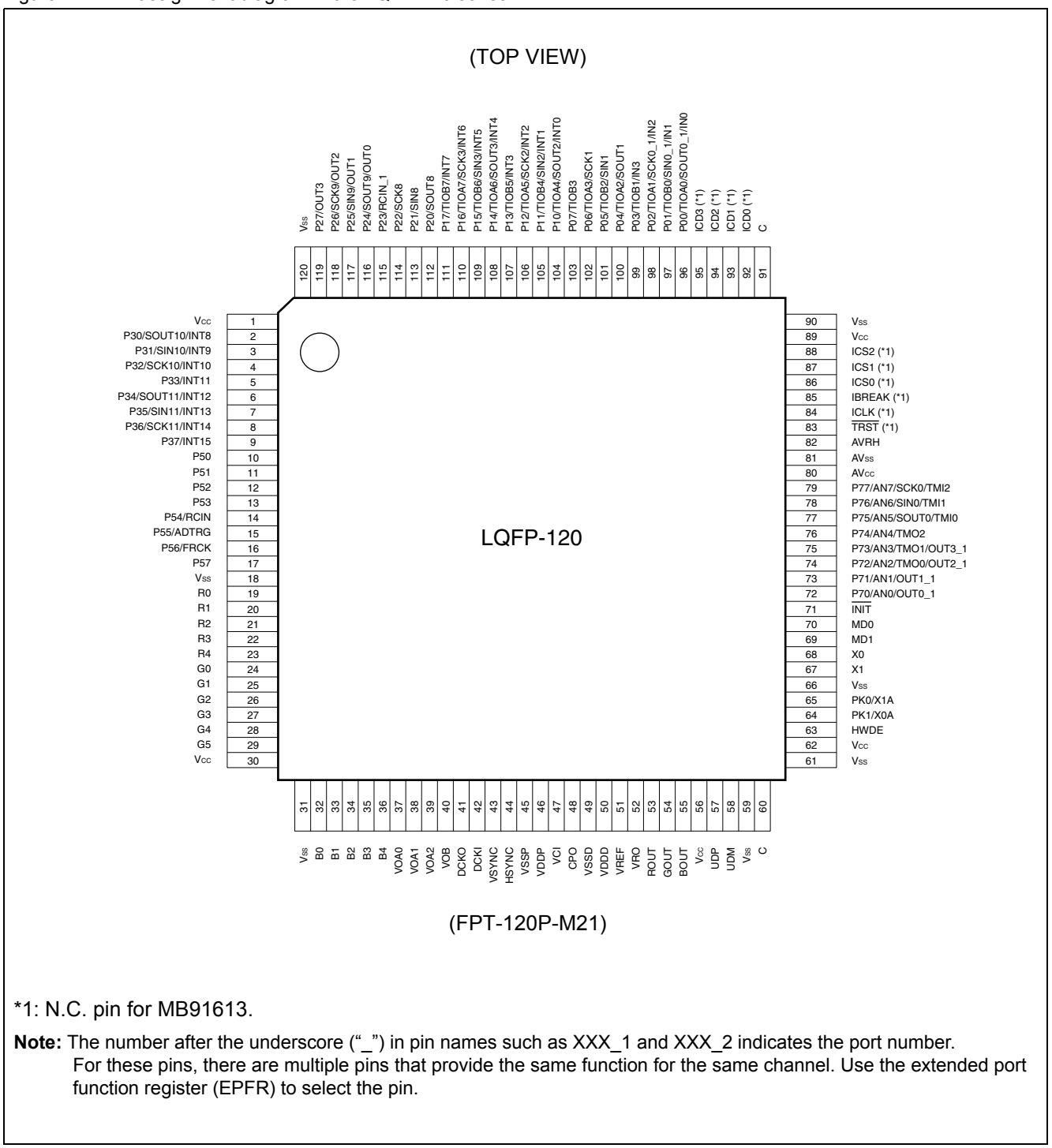
2.4 Setting Method for Pins

2.1 Pin Assignment Diagram

1 type of package is available for the MB91610 series.

LQFP-120

Figure 2-1. Pin assignment diagram in the LQFP-120 series



*1: N.C. pin for MB91613.

Note: The number after the underscore ("_") in pin names such as XXX_1 and XXX_2 indicates the port number. For these pins, there are multiple pins that provide the same function for the same channel. Use the extended port function register (EPFR) to select the pin.

2.2 Pin Functions

Table 2-1 lists the pin functions of the MB91610 series.

In a pin that includes an underscore (_), such as XXX_1 and XXX_2, the number following the underscore represents a port number. For details of the port numbers, see "2.4 Setting Method for Pins".

Pin Function List

Table 2-1. Pin functions (Sheet 1 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
1	V_{CC}	-	3.3 V power supply
2	P30	C	General-purpose I/O port
	SOUT10		Multifunction serial ch.10 output [operation modes 0 to 2]
	(SDA10)		I^2C ch.10 serial data line [operation mode 4]
	INT8		External interrupt 8 input
3	P31	C	General-purpose I/O port
	SIN10		Multifunction serial ch.10 input
	INT9		External interrupt 9 input
4	P32	C	General-purpose I/O port
	SCK10		Multifunction serial ch.10 clock [operation modes 0 to 2]
	(SCL10)		I^2C ch.10 serial clock line [operation mode 4]
	INT10		External interrupt 10 input
5	P33	C	General-purpose I/O port
	INT11		External interrupt 11 input
6	P34	C	General-purpose I/O port
	SOUT11		Multifunction serial ch.11 output [operation modes 0 to 2]
	(SDA11)		I^2C ch.11 serial data line [operation mode 4]
	INT12		External interrupt 12 input
7	P35	C	General-purpose I/O port
	SIN11		Multifunction serial ch.11 input
	INT13		External interrupt 13 input
8	P36	C	General-purpose I/O port
	SCK11		Multifunction serial ch.11 clock [operation modes 0 to 2]
	(SCL11)		I^2C ch.11 serial clock line [operation mode 4]
	INT14		External interrupt 14 input
9	P37	C	General-purpose I/O port
	INT15		External interrupt 15 input

Table 2-1. Pin functions (Sheet 2 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
10	P50	B	General-purpose I/O port
11	P51	B	General-purpose I/O port
12	P52	B	General-purpose I/O port
13	P53	B	General-purpose I/O port
14	P54	B	General-purpose I/O port
	RCIN		Remote control I/O
15	P55	B	General-purpose I/O port
	ADTRG		A/D converter external trigger input
16	P56	B	General-purpose I/O port
	FRCK		Free-run timer clock input
17	P57	B	General-purpose I/O port
18	V_{SS}	-	GND
19	R0	H	RGB digital output
20	R1	H	RGB digital output
21	R2	H	RGB digital output
22	R3	H	RGB digital output
23	R4	H	RGB digital output
24	G0	H	RGB digital output
25	G1	H	RGB digital output
26	G2	H	RGB digital output
27	G3	H	RGB digital output
28	G4	H	RGB digital output
29	G5	H	RGB digital output
30	V_{CC}	-	3.3V power supply
31	V_{SS}	-	GND
32	B0	H	RGB digital output
33	B1	H	RGB digital output
34	B2	H	RGB digital output
35	B3	H	RGB digital output
36	B4	H	RGB digital output
37	VOA0	H	Alpha blend output
38	VOA1	H	Alpha blend output
39	VOA2	H	Alpha blend output

Table 2-1. Pin functions (Sheet 3 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
40	VOB	H	OSD display period output
41	DCKO	H	Dot clock output
42	DCKI	F	Dot clock input
43	VSYNC	F	Vertical synchronous input
44	HSYNC	F	Horizontal synchronous input
45	VSSP	-	Dot clock PLL ground
46	VDDP	-	Dot clock PLL power supply
47	VCI	-	VCO control voltage input
48	CPO	M	Charge pump output
49	VSSD	-	RGB analog output GND
50	VDDD	-	RGB analog output power supply
51	VREF	M	RGB analog output reference power supply
52	VRO	M	RGB analog output resistance connected pin
53	ROUT	M	R output (analog)
54	GOUT	M	G output (analog)
55	BOUT	M	B output (analog)
56	V_{CC}	-	3.3 V power supply
57	UDP	USB	USB pin
58	UDM	USB	USB pin
59	V_{SS}	-	GND
60	C	-	C pin for a regulator
61	V_{SS}	-	GND
62	V_{CC}	-	3.3 V power supply
63	HWDE	F	Hardware watchdog enable input
64	PK1	G	General-purpose I/O port
	X0A		32kHz oscillation pin
65	PK0	G	General-purpose I/O port
	X1A		32 kHz oscillation pin
66	V_{SS}	-	GND
67	X1	A	Main oscillation pin
68	X0	A	Main oscillation pin
69	MD1	F,L	Mode pin
70	MD0	F,L	Mode pin

Table 2-1. Pin functions (Sheet 4 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
71	INIT	F,L	Initial (reset) pin
72	P70	D	General-purpose I/O port
	AN0		A/D converter ch.0 analog input
	OUT0_1		Output compare ch.0 output (Port 1)
73	P71	D	General-purpose I/O port
	AN1		A/D converter ch.1 analog input
	OUT1_1		Output compare ch.1 output (Port 1)
74	P72	D	General-purpose I/O port
	AN2		A/D converter ch.2 analog input
	TMO0		Reload timer ch.0 output
	OUT2_1		Output compare ch.2 output (Port 1)
75	P73	D	General-purpose I/O port
	AN3		A/D converter ch.3 analog input
	TMO1		Reload timer ch.1 output
	OUT3_1		Output compare ch.3 output (Port 1)
76	P74	D	General-purpose I/O port
	AN4		A/D converter ch.4 analog input
	TMO2		Reload timer ch.2 output
77	P75	D	General-purpose I/O port
	AN5		A/D converter ch.5 analog input
	SOUT0		Multifunction serial ch.0 output [operation modes 0 to 2]
	TMI0		Reload timer ch.0 input
78	P76	D	General-purpose I/O port
	AN6		A/D converter ch.6 analog input
	SIN0		Multifunction serial ch.0 input
	TMI1		Reload timer ch.1 input
79	P77	D	General-purpose I/O port
	AN7		A/D converter ch.7 analog input
	SCK0		Multifunction serial ch.0 clock [operation modes 0 to 2]
	TMI2		Reload timer ch.2 input
80	AV_{CC}	-	A/D converter analog power supply
81	AV_{SS}	-	A/D converter GND
82	AVRH	-	A/D converter analog reference power supply

Table 2-1. Pin functions (Sheet 5 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
83	TRST	E	Tool reset input for DSU4 *N.C. pin for MASK products.
84	ICLK	K	Clock pin for DSU4 *N.C. pin for MASK products.
85	IBREAK	I	Break pin for DSU4 *N.C. pin for MASK products.
86	ICS0	H	DSU4 status *N.C. pin for MASK products.
87	ICS1	H	DSU4 status *N.C. pin for MASK products.
88	ICS2	H	DSU4 status *N.C. pin for MASK products.
89	V_{CC}	-	3.3 V power supply
90	V_{SS}	-	GND
91	C	-	C pin for a regulator
92	ICD0	J	DSU4 data *N.C. pin for MASK products.
93	ICD1	J	DSU4 data *N.C. pin for MASK products.
94	ICD2	J	DSU4 data *N.C. pin for MASK products.
95	ICD3	J	DSU4 data *N.C. pin for MASK products.
96	P00	B	General-purpose I/O port
	TIOA0		Base timer ch.0 TIOA
	SOUT0_1		Multifunction serial ch.0 output (Port 1) [operation modes 0 to 2]
	IN0		Input capture ch.0 input
97	P01	B	General-purpose I/O port
	TIOB0		Base timer ch.0 TIOB
	SIN0_1		Multifunction serial ch.0 input (Port 1)
	IN1		Input capture ch.1 input
98	P02	B	General-purpose I/O port
	TIOA1		Base timer ch.1 TIOA
	SCK0_1		Multifunction serial ch.0 clock (Port 1) [operation modes 0 to 2]
	IN2		Input capture ch.2 input
99	P03	B	General-purpose I/O port
	TIOB1		Base timer ch.1 TIOB
	IN3		Input capture ch.3 input
100	P04	B	General-purpose I/O port
	TIOA2		Base timer ch.2 TIOA
	SOUT1		Multifunction serial ch.1 output [operation modes 0 to 2]
	(SDA1)		I^2C ch.1 serial data line [operation mode 4]

Table 2-1. Pin functions (Sheet 6 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
101	P05	B	General-purpose I/O port
	TIOB2		Base timer ch.2 TIOB
	SIN1		Multifunction serial ch.1 input
102	P06	B	General-purpose I/O port
	TIOA3		Base timer ch.3 TIOA
	SCK1		Multifunction serial ch.1 clock [operation modes 0 to 2]
	(SCL1)		I^2C ch.1 serial clock line [operation mode 4]
103	P07	B	General-purpose I/O port
	TIOB3		Base timer ch.3 TIOB
104	P10	B	General-purpose I/O port
	TIOA4		Base timer ch.4 TIOA
	SOUT2		Multifunction serial ch.2 output [operation modes 0 to 2]
	(SDA2)		I^2C ch.2 serial data line [operation mode 4]
	INT0		External interrupt 0 input
105	P11	B	General-purpose I/O port
	TIOB4		Base timer ch.4 TIOB
	SIN2		Multifunction serial ch.2 input
	INT1		External interrupt 1 input
106	P12	B	General-purpose I/O port
	TIOA5		Base timer ch.5 TIOA
	SCK2		Multifunction serial ch.2 clock [operation modes 0 to 2]
	(SCL2)		I^2C ch.2 serial clock line [operation mode 4]
	INT2		External interrupt 2 input
107	P13	B	General-purpose I/O port
	TIOB5		Base timer ch.5 TIOB
	INT3		External interrupt 3 input
108	P14	B	General-purpose I/O port
	TIOA6		Base timer ch.6 TIOA
	SOUT3		Multifunction serial ch.3 output [operation modes 0 to 2]
	(SDA3)		I^2C ch.3 serial data line [operation mode 4]
	INT4		External interrupt 4 input

Table 2-1. Pin functions (Sheet 7 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
109	P15	B	General-purpose I/O port
	TIOB6		Base timer ch.6 TIOB
	SIN3		Multifunction serial ch.3 input
	INT5		External interrupt 5 input
110	P16	B	General-purpose I/O port
	TIOA7		Base timer ch.7 TIOA
	SCK3		Multifunction serial ch.3 clock [operation modes 0 to 2]
	(SCL3)		I^2C ch.3 serial clock line [operation mode 4]
	INT6		External interrupt 6 input
111	P17	B	General-purpose I/O port
	TIOB7		Base timer ch.7 TIOB
	INT7		External interrupt 7 input
112	P20	C	General-purpose I/O port
	SOUT8		Multifunction serial ch.8 output [operation modes 0 to 2]
	(SDA8)		I^2C ch.8 serial data line [operation mode 4]
113	P21	C	General-purpose I/O port
	SIN8		Multifunction serial ch.8 input
114	P22	C	General-purpose I/O port
	SCK8		Multifunction serial ch.8 clock [operation modes 0 to 2]
	(SCL8)		I^2C ch.8 serial clock line [operation mode 4]
115	P23	C	General-purpose I/O port
	RCIN_1		Remote control I/O (1)
116	P24	C	General-purpose I/O port
	SOUT9		Multifunction serial ch.9 output [operation modes 0 to 2]
	(SDA9)		I^2C ch.9 serial data line [operation mode 4]
	OUT0		Output compare ch.0 output
117	P25	C	General-purpose I/O port
	SIN9		Multifunction serial ch.9 input
	OUT1		Output compare ch.1 output
118	P26	C	General-purpose I/O port
	SCK9		Multifunction serial ch.9 clock [operation modes 0 to 2]
	(SCL9)		I^2C ch.9 serial clock line [operation mode 4]
	OUT2		Output compare ch.2 output

Table 2-1. Pin functions (Sheet 8 of 8)

Pin no.	Pin name	I/O circuit type[1]	Function
119	P27	C	General-purpose I/O port
	OUT3		Output compare ch.3 output
120	V_{SS}	-	GND

[1]: Refer to "2.3 I/O Circuit Types" for details on the I/O circuit types.

2.3 I/O Circuit Types

Table 2-2 lists the I/O circuit types for the MB91610 series.

I/O circuit types

Table 2-2. I/O circuit types (Sheet 1 of 6)

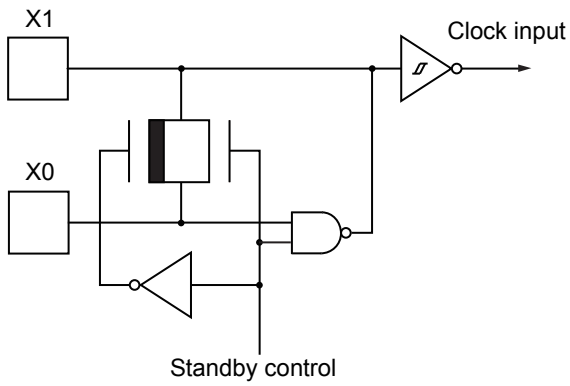
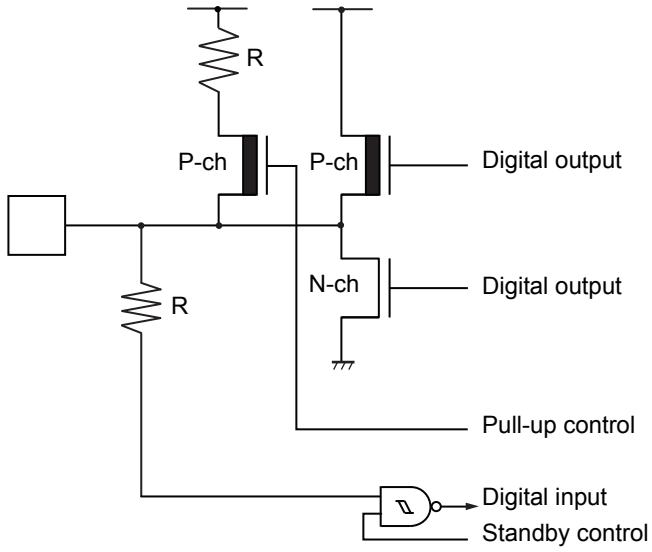
Type	Circuit	Remarks
A		■ Oscillation feedback resistance approx.1 MΩ ■ With standby control
B		■ CMOS level output ■ CMOS level hysteresis input ■ With pull-up control ■ With standby control **Note:** When this pin is used as an I^2C pin, the digital output P-ch transistor is always off.

Table 2-2. I/O circuit types (Sheet 2 of 6)

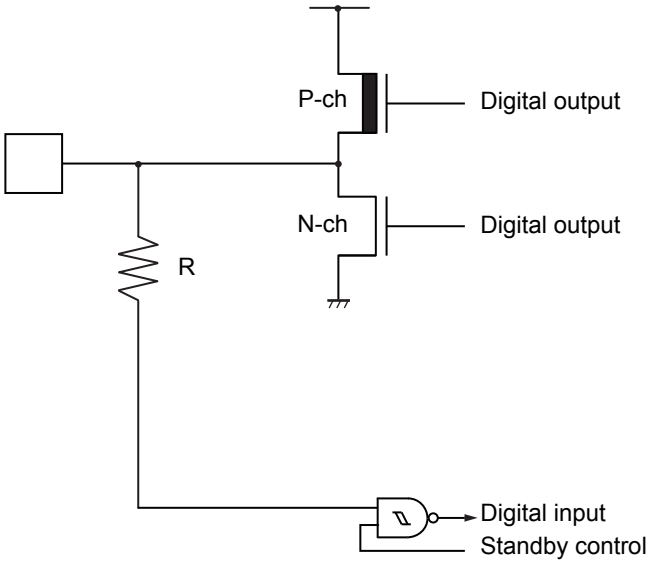
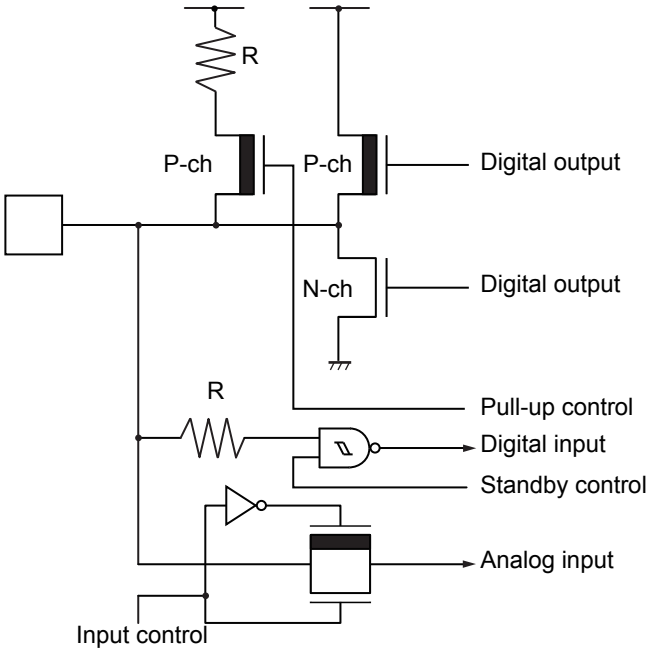
Type	Circuit	Remarks
C		■ CMOS level output ■ CMOS level hysteresis input ■ 5 V tolerant input ■ With standby control **Note:** When this pin is used as an I^2C pin, the digital output P-ch transistor is always off.
D		■ CMOS level output ■ CMOS level hysteresis input ■ With input control ■ Analog input ■ With pull-up control ■ With standby control **Note:** When this pin is used as an I^2C pin, the digital output P-ch transistor is always off.

Table 2-2. I/O circuit types (Sheet 3 of 6)

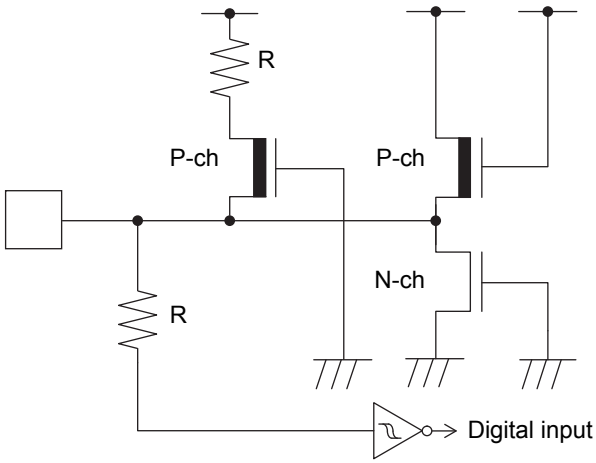
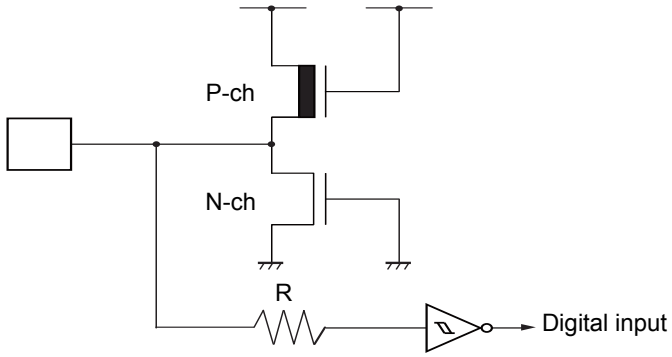
Type	Circuit	Remarks
E		■ CMOS level hysteresis input ■ With pull-up
F		CMOS level hysteresis input

Table 2-2. I/O circuit types (Sheet 4 of 6)

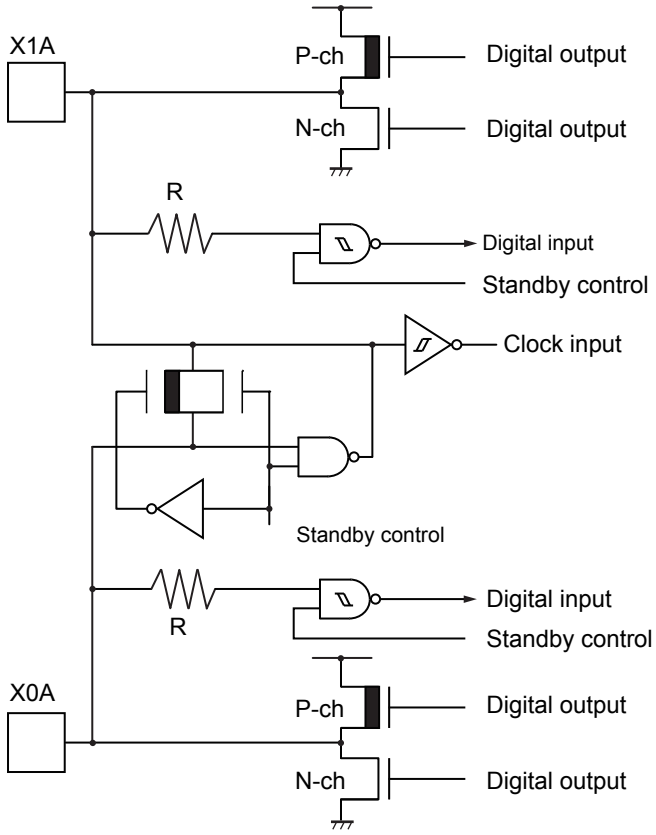
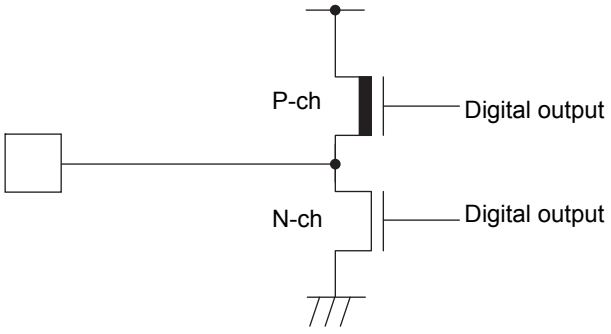
Type	Circuit	Remarks
G		■ Oscillation feedback resistance approx.10MΩ ■ CMOS level output ■ CMOS level hysteresis input ■ With standby control
H		CMOS level output

Table 2-2. I/O circuit types (Sheet 5 of 6)

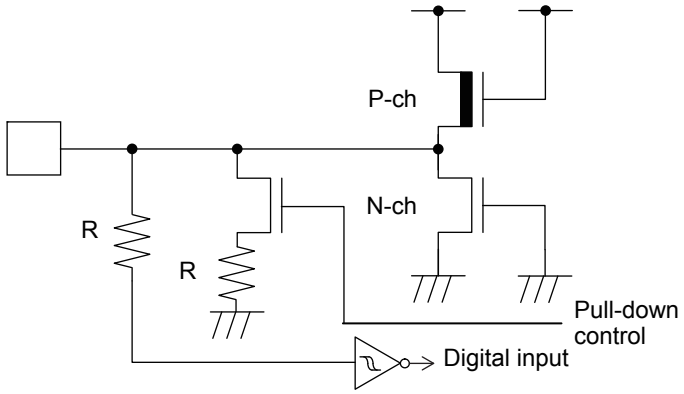
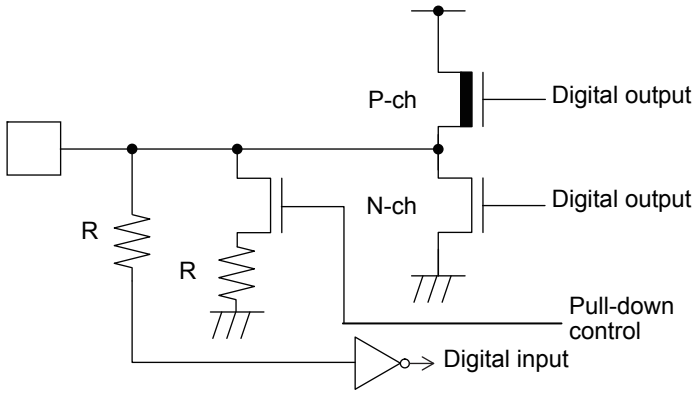
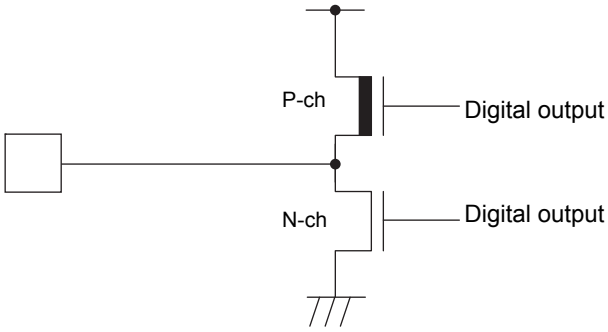
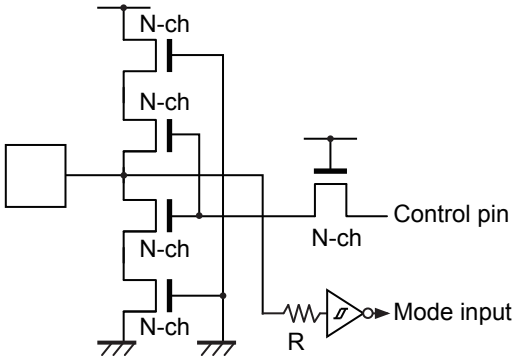
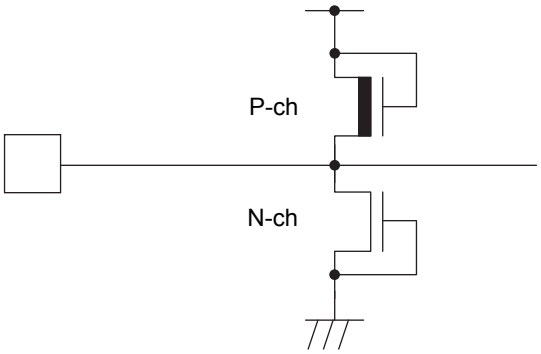
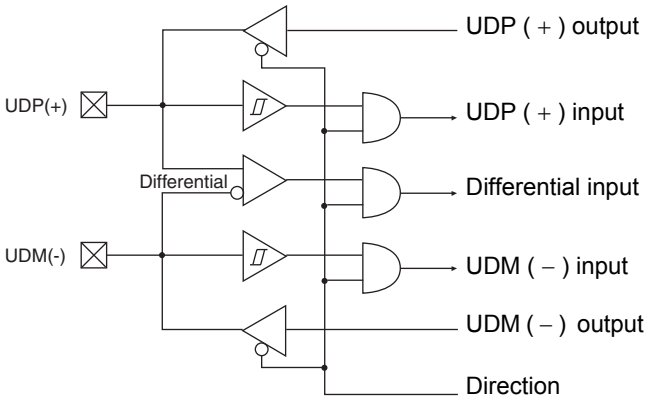
Type	Circuit	Remarks
I		■ CMOS level hysteresis input ■ With Pull-down control
J		■ CMOS level output ■ CMOS level input ■ With Pull-down control
K		CMOS level output (8 mA)

Table 2-2. I/O circuit types (Sheet 6 of 6)

Type	Circuit	Remarks
L		■ Flash memory product only ■ CMOS level hysteresis input ■ High voltage control for testing Flash memory
M		Analog pin
USB		USB I/O pin

2.4 Setting Method for Pins

This section explains how to set registers for the multiplexed pins.

More than one function has been assigned to the multiplexed pins. The tables below list the register setting values used to assign each of these functions to the pins, as categorized by peripheral function.

The register names appearing in these tables are abbreviated names.

- EPFR: Extended port function register
- PFR: Port function register
- DDR: Port data direction register

For details of these registers, see "13. I/O Ports".

Other abbreviated register names are explained in the notes under each table. For details, see the respective chapters.

Ports

Pin Name	Register Name	Bit Name	Written Value
P00 to P07	PFR0	PFR00 to PFR07	0
P10 to P17	PFR1	PFR10 to PFR17	0
P20 to P27	PFR2	PFR20 to PFR27	0
P30 to P37	PFR3	PFR30 to PFR37	0
P50 to P57	PFR5	PFR50 to PFR57	0
P70 to P77	PFR7	PFR70 to PFR77	0

Note: For details of the settings of the port data direction register (DDR) see "13. I/O Ports".

Clocks

Pin Name	Register Name	Bit Name	Written Value
X0A, X1A	DDRK	DDRK1, DDRK0	00
	EPFR19	XAE	1
	CSELR	SCEN	1

CSELR: Clock source select register

External interrupt controllers

To use the INT pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Enable the operation of the external interrupt controller (For details, see "14. External Interrupt Controllers").

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0 to 3	Port 0	INT0 to INT3	DDR1	DDR10 to DDR13	0
			PFR1	PFR10 to PFR13	0
4 to 7	Port 0	INT4 to INT7	DDR1	DDR14 to DDR17	0
			PFR1	PFR14 to PFR17	0
8 to 15	Port 0	INT8 to INT15	DDR3	DDR30 to DDR37	0
			PFR3	PFR30 to PFR37	0

32-bit free-run timer

To use the FRCK pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Enable the operation of the 32-bit free-run timer (For details, see "17. 32-bit Free-Run Timer").

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	FRCK	DDR5	DDR56	0
			PFR5	PFR56	0

32-bit input capture

To use the IN pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Enable the operation of the 32-bit input capture (For details, see "18. 32-bit Input Capture").

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	IN0	DDR0	DDR00	0
			PFR0	PFR00	0
1	Port 0	IN1	DDR0	DDR01	0
			PFR0	PFR01	0
2	Port 0	IN2	DDR0	DDR02	0
			PFR0	PFR02	0
3	Port 0	IN3	DDR0	DDR03	0
			PFR0	PFR03	0

32-bit output compare

The 32-bit output compare provides 2 OUT pins for use with each channel.

One of each of the pins can be selected for use with each channel.

To use the OUT pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	OUT0	PFR2	PFR24	1
			EPFR0	OUT0E1, OUT0E0	01
			EPFR15	SOUT9E0	0
	Port 1	OUT0_1	PFR7	PFR70	1
			EPFR0	OUT0E1, OUT0E0	10
			ADCHE	ADE0	0
1	Port 0	OUT1	PFR2	PFR25	1
			EPFR0	OUT1E1, OUT1E0	01
	Port 1	OUT1_1	PFR7	PFR71	1
			EPFR0	OUT1E1, OUT1E0	10
			ADCHE	ADE1	0
2	Port 0	OUT2	PFR2	PFR26	1
			EPFR1	OUT2E1, OUT2E0	01
			EPFR15	SCK9E0	0
	Port 1	OUT2_1	PFR7	PFR72	1
			EPFR1	OUT2E1, OUT2E0	10
			EPFR33	TMO0E0	0
			ADCHE	ADE2	0
3	Port 0	OUT3	PFR2	PFR27	1
			EPFR1	OUT3E1, OUT3E0	01
	Port 1	OUT3_1	PFR7	PFR73	1
			EPFR1	OUT3E1, OUT3E0	10
			EPFR33	TMO1E0	0
			ADCHE	ADE3	0

16-bit reload timer

The 16-bit reload timer provides 2 of each of the TMI/TMO pins for use with each channel.

To use the TMI pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the 16-bit reload timer (For details, see "20. 16-bit Reload Timer").

To use the TMO pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	TMI0	DDR7	DDR75	0
			PFR7	PFR75	0
			ADCHE	ADE5	0
		TMO0	PFR7	PFR72	1
			EPFR33	TMO0E0	1
			ADCHE	ADE2	0
1	Port 0	TMI1	DDR7	DDR76	0
			PFR7	PFR76	0
			ADCHE	ADE6	0
		TMO1	PFR7	PFR73	1
			EPFR33	TMO1E0	1
			ADCHE	ADE3	0
2	Port 0	TMI2	DDR7	DDR77	0
			PFR7	PFR77	0
			ADCHE	ADE7	0
		TMO2	PFR7	PFR74	1
			EPFR34	TMO2E0	1
			ADCHE	ADE4	0

Base timer

To use the TIOA/TIOB pins for input, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the base timer (For details, see "22. Base Timer").

To use the TIOA pin for output, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	TIOA0	PFR0	PFR00	1
			EPFR20	TIOA0E0	1
			EPFR6	SOUT0E1, SOUT0E0	Other than 10 [1]
		TIOB0	PFR0	PFR01	0
			DDR0	DDR01	0
1	Port 0	TIOA1	PFR0	PFR02	At input: 0 At output: 1
			DDR0	DDR02	0 (only at input)
			EPFR20	TIOA1E0	1
			EPFR6	SCK0E1, SCK0E0	Other than 10 [1]
		TIOB1	PFR0	PFR03	0
			DDR0	DDR03	0
2	Port 0	TIOA2	PFR0	PFR04	1
			EPFR21	TIOA2E0	1
			EPFR7	SOUT1E0	0
		TIOB2	PFR0	PFR05	0
			DDR0	DDR05	0
3	Port 0	TIOA3	PFR0	PFR06	At input: 0 At output: 1
			DDR0	DDR06	0 (only at input)
			EPFR21	TIOA3E0	1
			EPFR7	SCK1E0	0
		TIOB3	PFR0	PFR07	0
			DDR0	DDR07	0
4	Port 0	TIOA4	PFR1	PFR10	1
			EPFR22	TIOA4E0	1
			EPFR8	SOUT2E0	0
		TIOB4	PFR1	PFR11	0
			DDR1	DDR11	0

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
5	Port 0	TIOA5	PFR1	PFR12	At input: 0 At output: 1
			DDR1	DDR12	0 (only at input)
			EPFR22	TIOA5E0	1
			EPFR8	SCK2E0	0
		TIOB5	PFR1	PFR13	0
			DDR1	DDR13	0
6	Port 0	TIOA6	PFR1	PFR14	1
			EPFR23	TIOA6E0	1
			EPFR9	SOUT3E0	0
		TIOB6	PFR1	PFR15	0
			DDR1	DDR15	0
7	Port 0	TIOA7	PFR1	PFR16	At input: 0 At output: 1
			DDR1	DDR16	0 (only at input)
			EPFR23	TIOA7E0	1
			EPFR9	SCK3E0	0
		TIOB7	PFR1	PFR17	0
			DDR1	DDR17	0

[1]: Do not write a setting prohibited value. For details, see "13. I/O Ports".

10-bit A/D converter

- AN pins

Pin Name	Register Name	Bit Name	Written Value
AN0 to AN7	ADCHE	ADE0 to ADE7	1

ADCHE: A/D channel enable register

- ADTRG pins

  To use the ADTRG pin, the following settings are required.

  1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
  2. Enable the operation of the 10-bit A/D converter (For details, see "23. 10-Bit A/D Converter").

  For details of the basic settings, see the following table.

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	ADTRG	DDR5	DDR55	0
		PFR5	PFR55	0

Multifunction serial interface

The multifunction serial interface provides multiple SCK pins, SIN pins, and SOUT pins for use with one channel.

One of each of the SCK/SIN/SOUT pins can be selected for use with each channel. However, to use pins for the same channel, the pins must be assigned to the same port number.

To use the SIN/SCK pins for input, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the multifunction serial interface (For details, see "CHAPTER 24 Multi-function Serial Interface").

To use the SOUT/SCK pins for output, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	SCK0	PFR7	PFR77	At SCK input: 0 At SCK output: 1
			DDR7	DDR77	0 (only at SCK input)
			EPFR6	SCK0E1, SCK0E0	01
			SMR0	SCKE	Input enable: 0 Output enable: 1
			ADCHE	ADE7	0
		SIN0	DDR7	DDR76	0
			PFR7	PFR76	0
			EPFR6	SIN0E0	0
			ADCHE	ADE6	0
		SOUT0	PFR7	PFR75	1
			EPFR6	SOUT0E1, SOUT0E0	01
			SMR0	SOE	1
			ADCHE	ADE5	0
	Port 1	SCK0_1	PFR0	PFR02	At SCK input: 0 At SCK output: 1
			DDR0	DDR02	0 (only at SCK input)
			EPFR6	SCK0E1, SCK0E0	10
			SMR0	SCKE	Input enable: 0 Output enable: 1
		SIN0_1	DDR0	DDR01	0
			PFR0	PFR01	0
			EPFR6	SIN0E0	1
		SOUT0_1	PFR0	PFR00	1
			EPFR6	SOUT0E1, SOUT0E0	10
			SMR0	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
1	Port 0	SCK1 (SCL1)	PFR0	PFR06	At SCK input:0 At SCK output or SCL:1
			DDR0	DDR06	0 (only at SCK input)
			EPFR7	SCK1E0	1
			SMR1	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN1	DDR0	DDR05	0
			PFR0	PFR05	0
		SOUT1 (SDA1)	PFR0	PFR04	1
			EPFR7	SOUT1E0	1
			SMR1	SOE	1
2	Port 0	SCK2 (SCL2)	PFR1	PFR12	At SCK input:0 At SCK output or SCL:1
			DDR1	DDR12	0 (only at SCK input)
			EPFR8	SCK2E0	1
			SMR2	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN2	DDR1	DDR11	0
			PFR1	PFR11	0
		SOUT2 (SDA2)	PFR1	PFR10	1
			EPFR8	SOUT2E0	1
			SMR2	SOE	1
3	Port 0	SCK3 (SCL3)	PFR1	PFR16	At SCK input:0 At SCK output or SCL:1
			DDR1	DDR16	0 (only at SCK input)
			EPFR9	SCK3E0	1
			SMR3	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN3	DDR1	DDR15	0
			PFR1	PFR15	0
		SOUT3 (SDA3)	PFR1	PFR14	1
			EPFR9	SOUT3E0	1
			SMR3	SOE	1
8	Port 0	SCK8 (SCL8)	PFR2	PFR22	At SCK input:0 At SCK output or SCL:1
			DDR2	DDR22	0 (only at SCK input)
			EPFR14	SCK8E0	1
			SMR8	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN8	DDR2	DDR21	0
			PFR2	PFR21	0
		SOUT8 (SDA8)	PFR2	PFR20	1
			EPFR14	SOUT8E0	1
			SMR8	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
9	Port 0	SCK9 (SCL9)	PFR2	PFR26	At SCK input:0 At SCK output or SCL:1
			DDR2	DDR26	0 (only at SCK input)
			EPFR15	SCK9E0	1
			SMR9	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN9	DDR2	DDR25	0
			PFR2	PFR25	0
		SOUT9 (SDA9)	PFR2	PFR24	1
			EPFR15	SOUT9E0	1
			SMR9	SOE	1
10	Port 0	SCK10 (SCL10)	PFR3	PFR32	At SCK input:0 At SCK output or SCL:1
			DDR3	DDR32	0 (only at SCK input)
			EPFR16	SCK10E0	1
			SMR10	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN10	DDR3	DDR31	0
			PFR3	PFR31	0
			EPFR16	SIN10E	0
		SOUT10 (SDA10)	PFR3	PFR30	1
			EPFR16	SOUT10E0	1
			SMR10	SOE	1
11	Port 0	SCK11 (SCL11)	PFR3	PFR36	At SCK input:0 At SCK output or SCL:1
			DDR3	DDR36	0 (only at SCK input)
			EPFR17	SCK11E0	1
			SMR11	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN11	DDR3	DDR35	0
			PFR3	PFR35	0
		SOUT11 (SDA11)	PFR3	PFR34	1
			EPFR17	SOUT11E0	1
			SMR11	SOE	1

SMR: Serial mode register

Note: Different pins are enabled depending on the operation mode. For details, see "24. Multi-function Serial Interface".

Remote control reception

To use the RCIN/RCIN_1 pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the Remote Control Reception (For details, see "29. Remote Control Reception").
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Pin Name	Register Name	Bit Name	Written Value
RCIN	DDR5	DDR54	0
	PFR5	PFR54	1
	EPFR10	RCIN	1
RCIN_1	DDR2	DDR23	0
	PFR2	PFR23	1
	EPFR10	RCIN	1

Note: To use the RCIN/RCIN_1 pin, select either one of the pins.
Do not set "1" to PFR54 and PFR23 at the same time.

3. CPU

This chapter explains the basics of the FR80 family CPUs, including its architecture, specifications, and instructions, to provide a better understanding of the CPU functions.

3.1 Memory Space

3.2 Features of the Internal Architecture

3.3 Operation Modes

3.4 Pipeline

3.5 Overview of Instructions

3.6 Basic Programming Model

3.7 Registers

3.8 Data Configuration

3.9 Addressing

3.10 Branch Instructions

3.11 EIT (Exception, Interrupt, Trap)

3.1 Memory Space

The logical address space of the FR80 family CPUs is 4 GB (2^{32} locations), and the CPUs can linearly access it.

Direct Addressing Areas

The address spaces 0000 0000_H to 0000 $03FF_H$ are called the direct addressing areas.

These areas allow operands to be specified directly in instructions.

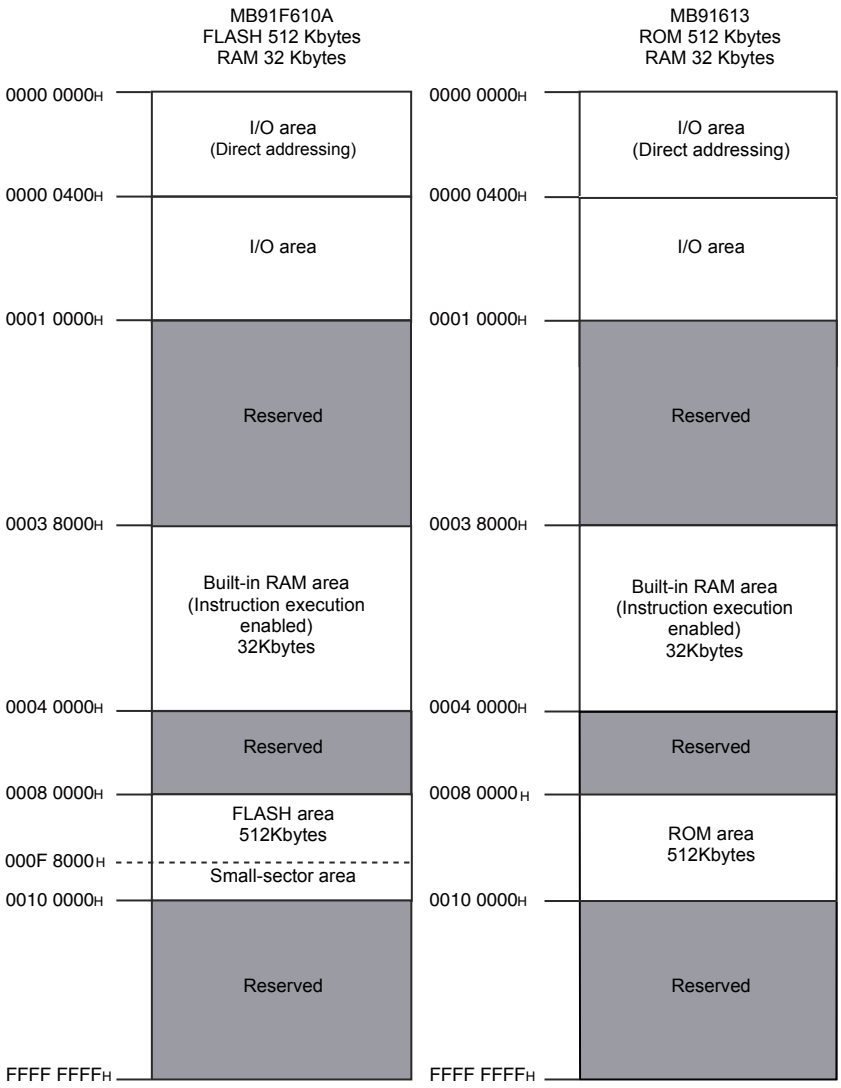
The direct addressing areas vary as follows depending on the size of the data accessed:

- Byte data access: 0000 0000_H to 0000 $00FF_H$
- Half word data access: 0000 0000_H to 0000 $01FF_H$
- Word data access: 0000 0000_H to 0000 $03FF_H$

Memory map

Figure 3-1 shows a memory map of the MB91610 series.

Figure 3-1. Memory map



Notes:

- For details of the small-sector area in flash memory, see "34. Flash Memory".
  The small-sector area concerns only the flash memory products.
- Do not access the reserved areas.

3.2 Features of the Internal Architecture

The FR80 family CPUs have a high-performance core based on the RISC architecture with high-level functions and instructions included for embedded applications.

- Adoption of the RISC architecture
- Basic instructions: 1 instruction/1 cycle
- 32-bit architecture
- 16 general-purpose 32-bit registers
- Linearly accessed 4-GB memory space
- Built-in multipliers
  - 32-bit × 32-bit multiplication: 5 cycles
  - 16-bit × 16-bit multiplication: 3 cycles
- Enhanced interrupt processing functions
  - High-speed response (6 cycles)
  - Multi-interrupt support
  - Level mask function (16 levels)
- Enhanced instructions for I/O operations
  - Memory-to-memory transfer instruction
  - Bit processing instruction
- High code efficiency
  - Basic instruction word length: 16 bits
- Compatibility of basic instructions with the FR60 family
- Addition of the following instructions to the instructions of the FR60 family:
  - Bit search instructions (SRCH0, SRCH1, and SRCHC)
- Deletion of the following instructions from the instructions of the FR60 family:
  - Coprocessor instructions (COPOP, COPLD, COPST, and COPSV)
  - Resource instructions (LDRES and STRES)
- Non-blocking load

  Up to 4 load instructions can be issued in advance.

3.3 Operation Modes

This section explains the operation modes of this series.

This series provides the operation modes below. At an activation of the device, one of these operation modes can be selected.

- User single-chip mode
- Serial programming mode

Table 3-1 lists the operation modes of this series.

Table 3-1. Operation modes

MD Pin		Control Pin	Operation Mode
MD1	MD0	P75	
0	0	X	User single-chip mode
	1	1	Serial programming mode

3.4 Pipeline

The FR architecture of the FR80 family CPUs is a compact 32-bit RISC architecture.

It has not only the normal instruction execution pipeline but also an additional pipeline for loading memory, which can reduce pipeline hazards during load instruction execution.

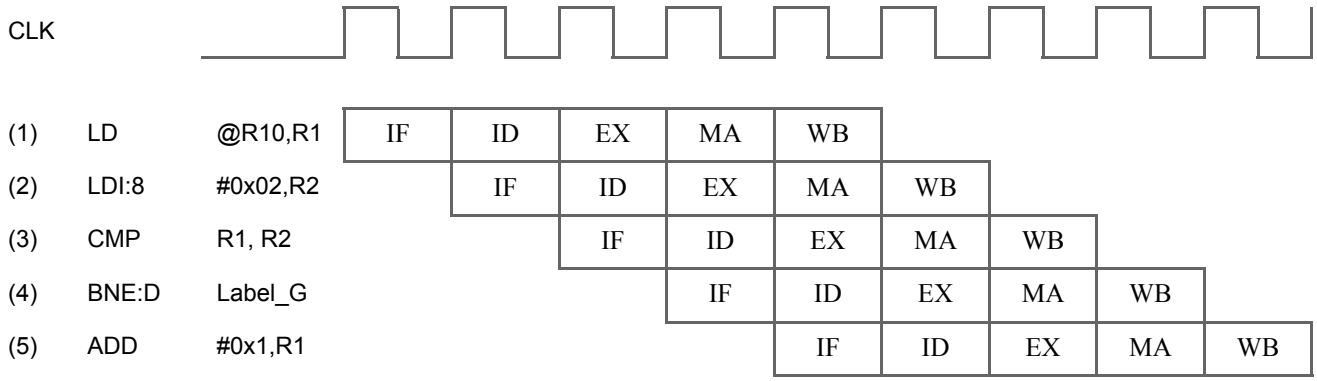
A five-stage instruction pipeline method is used in executing 1 instruction per cycle. The pipeline consists of the following stages:

- Instruction fetch (IF) stage: Fetches the instruction at the output address.
- Instruction decode (ID) stage: Decodes the fetched instruction. It also reads a register.
- Execution (EX) stage: Executes the decoded instruction.
- Memory access (MA) stage: Accesses the target memory.
- Register writing (WB) stage: Writes the operation results (or loaded memory data) to a register.

The pipeline for loading memory has been added so that the MA and WB stages of the instruction, which does not access memory, can overlap the MA and WB stages of an LD instruction.

As a rule, 1 instruction is executed per cycle. However, more than one cycle is required for execution of a load/store instruction with memory wait, a branch instruction without a delay slot, or a multi-cycle instruction. In addition, the instruction execution speed is slower when there is a delay in supplying an instruction.

Example 1:

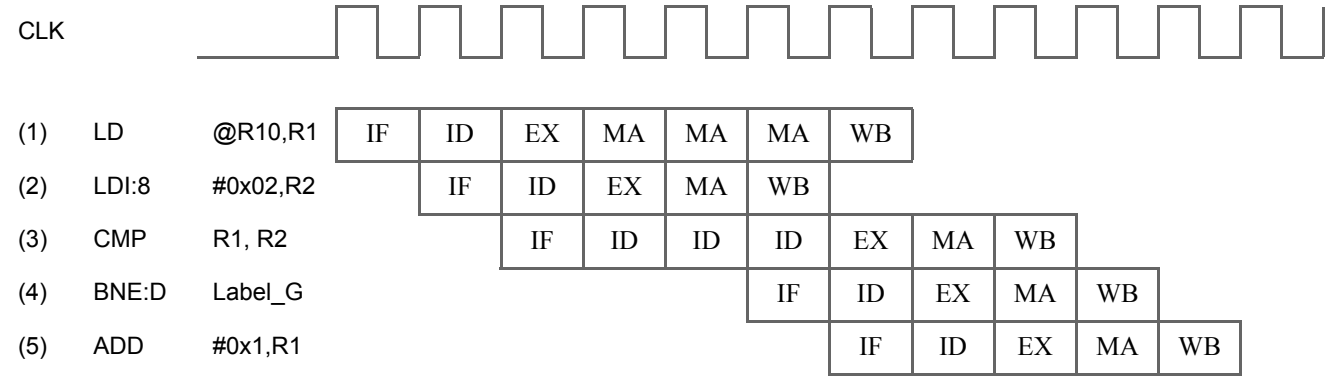


Example 1:The instructions are executed in sequence because the data that uses R1 to write the (1) LD instruction is returned in the (3) CMP instruction within 1 cycle.

In the load operation, the MA stage is extended until reading of the loaded data is completed.

However, if the register used for loading will not be used for the subsequent instructions, the instruction is executed as is.

Example 2:



Example 2:The data that uses R1 to write the (1) LD instruction is not returned within 1 cycle in the (3) CMP instruction, resulting in execution only up to the (2) LDI:8 instruction and keeping the CMP instruction waiting in the ID stage because of a register conflict.

3.5 Overview of Instructions

In addition to the general RISC instruction set, the FR80 family CPUs support the logical operations optimized for embedded applications, bit operation instructions, and direct addressing instructions.

Each instruction has a length of 16 bits (some instructions have a length of 32 or 48 bits) and provides superior performance in memory usage efficiency.

The instruction sets can be divided into the following function groups:

- Arithmetic operation
- Load and store
- Branch
- Logical operation and bit operation
- Direct addressing
- Bit search
- Other

3.5.1 Arithmetic Operation

These instructions are standard arithmetic instructions (addition, subtraction, and comparison) and shift instructions (logical shift and arithmetic operation shift). The arithmetic operations of addition and subtraction can include operations with a carry used in individual operations with a multi-word length (operation for 32 or more bits of data) and operations suitable for address calculation in which flag values are not changed.

Also included in these instructions are the 32-bit × 32-bit multiplication instruction, 16-bit × 16-bit multiplication instruction, and 32-bit / 32-bit step division instruction.

The immediate transfer instruction that sets immediate data in a register and the register-to-register transfer instruction are also included.

All the operations of arithmetic operation instructions use the general-purpose registers and Multiply & Divide registers in the CPUs.

3.5.2 Load and Store

Load and store are instructions for reading and writing external memory. They are also used for reading and writing by the internal peripheral functions of the chip.

The access lengths of load and store are in any of 3 units: byte, half word, and word. In addition to general-purpose register indirect memory addressing, some load and store instructions can use register indirect memory addressing with either displacement or register increment/decrement operations.

3.5.3 Branch

Branch instructions include branch, call, interrupt, and return instructions. The branch instructions consist of instructions with delay slots and instructions without delay slots, and they can be optimized as required. For details of the branch instructions, see "3.10 Branch Instructions".

3.5.4 Logical Operation and Bit Operation

Logical operation instructions can perform the AND, OR, and EOR logical operations between general-purpose registers or between a general-purpose register and memory (and I/O). Also, bit operation instructions can directly manipulate data on memory (and of I/O).
Memory addressing is general-purpose register indirect memory addressing.

3.5.5 Direct Addressing

Direct addressing instructions are instructions used for access between I/O and a general-purpose register or between I/O and memory. Specifying an I/O address directly in an instruction instead of using register indirect addressing enables highly efficient high-speed access. Also, some direct addressing instructions can perform register indirect memory addressing with register increment/decrement operations.

3.5.6 Bit Search

A bit search instruction searches 32-bit data beginning from the MSB to obtain the bit location of the first "1" or "0" found in the register. A bit search instruction can also make a comparison with the MSB value and obtain the bit location of a value different from the first MSB found in a register.

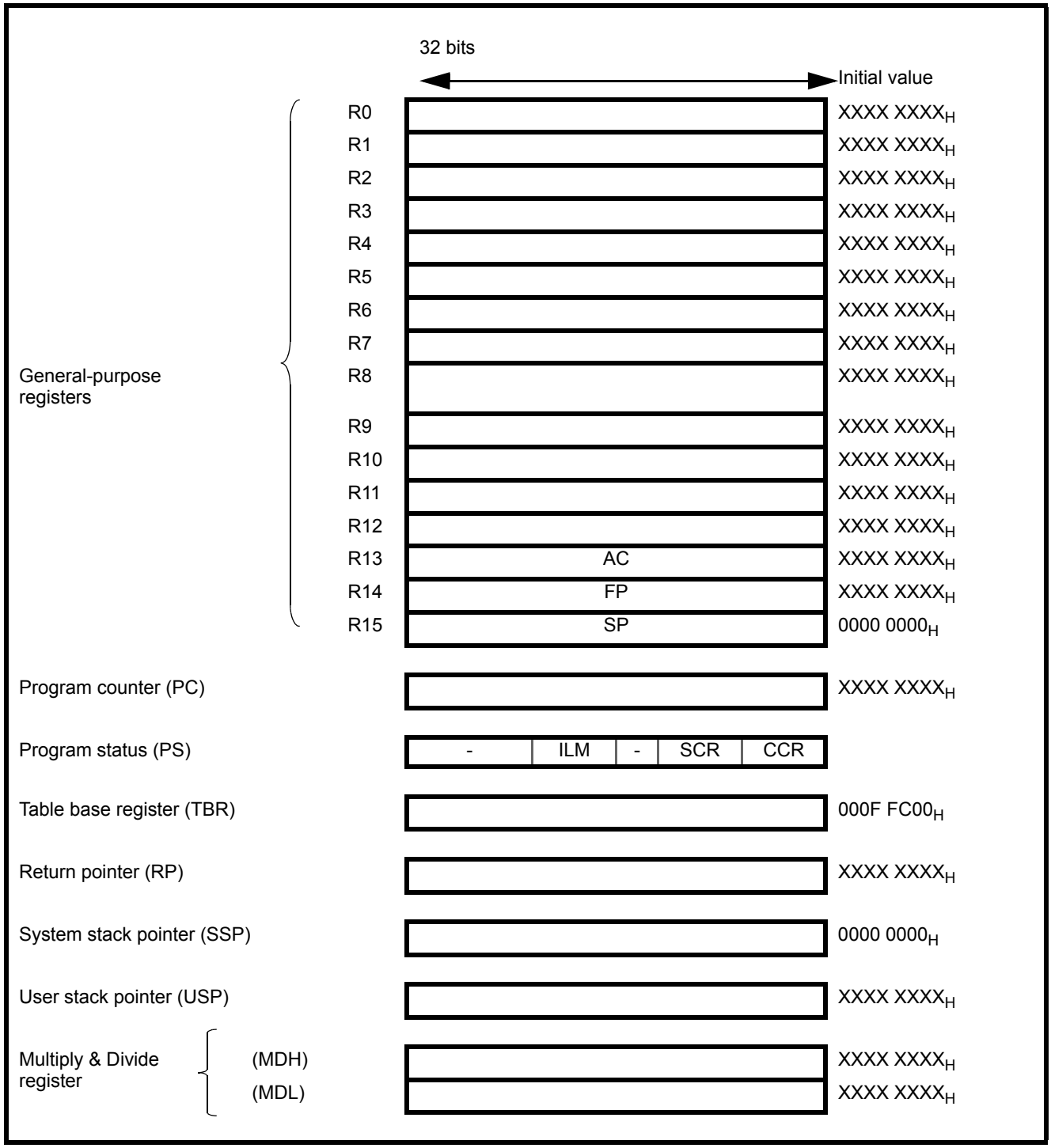
3.5.7 Other

Other available instructions include those for setting flags in the PS register, performing stack operations, and making a carry/zero extension. Also included in these instructions are function entry/exit instructions supporting high-level languages and multi-load/store instructions for registers.

3.6 Basic Programming Model

Figure 3-2 shows the basic programming model.

Figure 3-2. Basic Programming Model



3.7 Registers

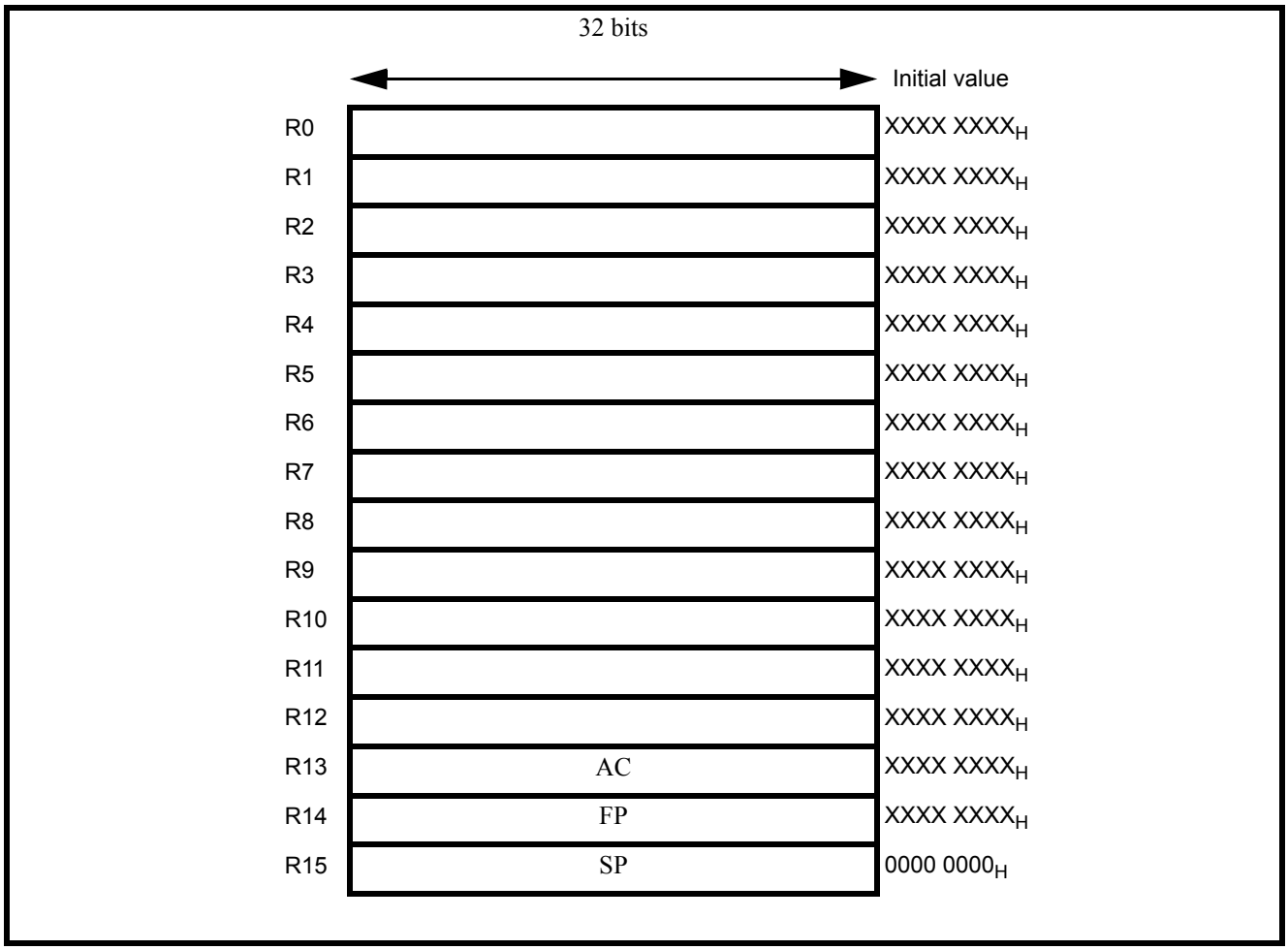
The register configuration consists of general-purpose registers and dedicated registers for specific purposes.

3.7.1 General-purpose Registers (R0 to R15)

Registers R0 to R15 are general-purpose registers. They are used as accumulators and memory access pointers in a variety of operations.

Figure 3-3 shows the bit configuration of the general-purpose registers (R0 to R15).

Figure 3-3. Bit Configuration Of The General-purpose Registers (R0 to R15)



Of the 16 registers, the following registers are assumed to have specific purposes, and certain instructions have therefore been enhanced. For details of the initial values at the reset time, see .

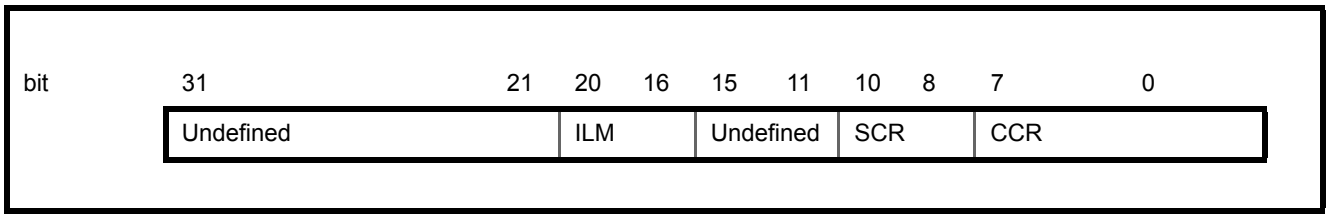
- R13: Virtual accumulator (AC)
- R14: Frame pointer (FP)
- R15: Stack pointer (SP)

3.7.2 Program Status Register (PS)

This register retains the program status, and it is divided into 3 parts: interrupt level mask register (ILM), system condition code register (SCR), and condition code register (CCR).

Figure 3-4 shows the bit configuration of the program status register (PS).

Figure 3-4. Bit Configuration Of The Program Status Register (PS)



- [bit31 to bit21, bit15 to bit11]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is always read.

- [bit20 to bit16] Interrupt level mask register (ILM)
  See "Interrupt level mask register (ILM)".
- [bit10 to bit8] System condition code register (SCR)
  See "System condition register (SCR)".
- [bit7 to bit0] Condition code register (CCR)
  See "Condition code register (CCR)".

Condition code register (CCR)

Figure 3-5 shows the bit configuration of the condition code register (CCR).

Figure 3-5. Bit Configuration Of The Condition Code Register (CCR)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	S	I	N	Z	V	C
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	X	X	X	X

R/W: Read/Write

-: Undefined

X: Undefined

- [bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is always read.

- [bit5]: S (Stack flag)

  This bit specifies a stack pointer operating as general-purpose register 15 (R15).

S	Explanation
0	The system stack pointer (SSP) is operating as general-purpose register 15 (R15). The bit is automatically cleared to "0" when EIT occurs. (However, the value before the bit is cleared is saved to the stack.)
1	The user stack pointer (USP) is operating as general-purpose register 15 (R15).

  This bit is cleared to "0" when the system is reset.

  "0" must be written when the RETI instruction is executed.

- [bit4]: I (Interrupt enable flag)

  This bit controls enabling/disabling of user interrupt requests.

I	Explanation
0	Disables user interrupt requests. The bit is automatically cleared to "0" when the INT instruction is executed. (However, the value before the bit is cleared is saved to the stack.)
1	Enables user interrupt requests. The mask processing of user interrupt requests is controlled with the value retained by the interrupt level mask register (ILM).

  This bit is cleared to "0" when the system is reset.

- [bit3]: N (Negative flag)

  This bit indicates a carry for an operation result recognized as an integer represented by a 2's complement.

N	Explanation
0	Indicates that the operation result is a positive value.
1	Indicates that the operation result is a negative value.

  The initial state set by a reset is undefined.

- [bit2]: Z (Zero flag)

  This bit indicates whether the result of an operation is "0".

Z	Explanation
0	Indicates that the operation result is not "0".
1	Indicates that the operation result is "0".

  The initial state set by a reset is undefined.

- [bit1]: V (Overflow flag)

  This bit indicates whether an overflow occurred as a result of an operation by interpreting each operand used for the operation as integers represented by 2's complements.

V	Explanation
0	No overflow occurred as a result of the operation.
1	An overflow occurred as a result of the operation.

  The initial state set by a reset is undefined.

- [bit0]: C (Carry flag)

  This bit indicates whether a carry or borrow from the most significant bit occurred as a result of an operation.

C	Explanation
0	No carry or borrow occurred.
1	A carry or borrow occurred.

  The initial state set by a reset is undefined.

System condition register (SCR)

Figure 3-6 shows the bit configuration of the system condition register (SCR).

Figure 3-6. Bit Configuration of the System Condition Register (SCR)

bit	10	9	8
	D1	D0	T
Attribute	R/W	R/W	R/W
Initial value	X	X	0

R/W: Read/Write

X: Undefined

- [bit10, bit9]: D1, D0 (Step division flag)

  These bits retain in-process data during step division execution.

  Do not change these bits while division processing is being executed.

  To execute any other processing during step division, save and return the value of the program status register (PS). Doing so ensures a restart of step division.

  The initial state set by a reset is undefined.

  Notes:

  - The bits are set with the reference of the dividend and divisor by execution of the DIV0S instruction.
  - They are forcibly cleared by execution of the DIV0U instruction.

- [bit8]: T (Step trace trap flag)

  This bit specifies whether the step trace trap is enabled.

T	Explanation
0	The step trace trap is disabled.
1	The step trace trap is enabled. All user interrupt requests are disabled.

  This bit is cleared to "0" when the system is reset.

  Emulators use the step trace trap function. The step trace trap cannot be used in a user program together with an emulator.

Interrupt level mask register (ILM)

This register retains the interrupt level mask value. The value retained by the register is used for the level mask.

Figure 3-7 shows the bit configuration of the interrupt level mask register (ILM).

Figure 3-7. Bit Configuration of the Interrupt Level Mask Register (ILM)

bit	20	19	18	17	16
	ILM4	ILM3	ILM2	ILM1	ILM0
Attribute	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	1	1	1

R/W: Read/Write

An interrupt request that is input to the CPU is accepted only if the corresponding interrupt level is higher than the level specified by this register.

The highest level is "0" (00000_B), and the lowest is "31" (11111_B).

A limited range of values can be set from programs.

- Original value in a range of 16 to 31: A value ranging from 16 to 31 can be specified as a new value. If a value ranging from 0 to 15 is set for an instruction, (specified-value + 16) is transferred when the instruction is executed.
- Original value in a range of 0 to 15: Any value ranging from 0 to 31 can be specified.

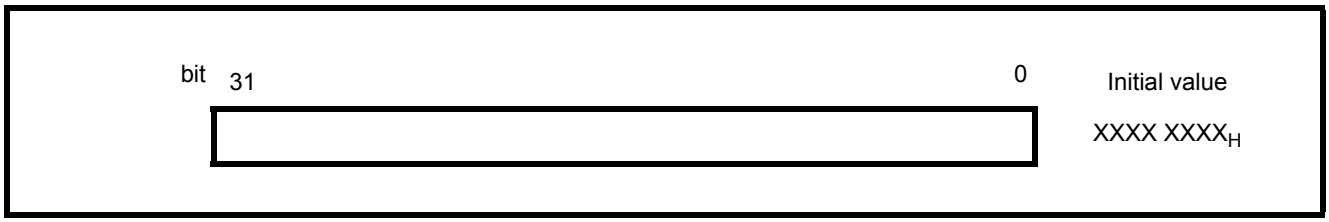
These bits are initialized to "15" (01111_B) by a reset.

3.7.3 Program Counter (PC)

This register is the program counter (PC) indicating the address of the instruction being executed.

Figure 3-8 shows the bit configuration of the program counter (PC).

Figure 3-8. Bit Configuration of the Program Counter (PC)



bit0 is set to "0" when an instruction that entails a PC update is executed.

It is prohibited to specify an odd-numbered location as the branch destination address, and to set bit0 to "1".

The instruction would have to be located at an address that is a multiple of 2.

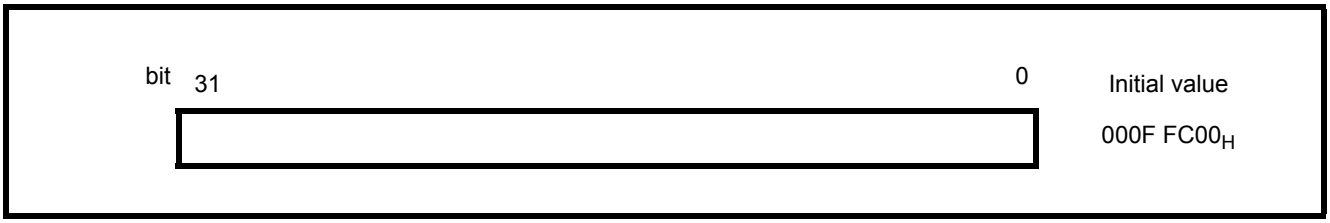
The initial value following a reset is undefined, and the program start address is set by a reset vector fetch.

3.7.4 Table Base Register (TBR)

This register retains the start address of the vector table used for EIT processing.

Figure 3-9 shows the bit configuration of the table base register (TBR).

Figure 3-9. Bit Configuration of the Table Base Register (TBR)



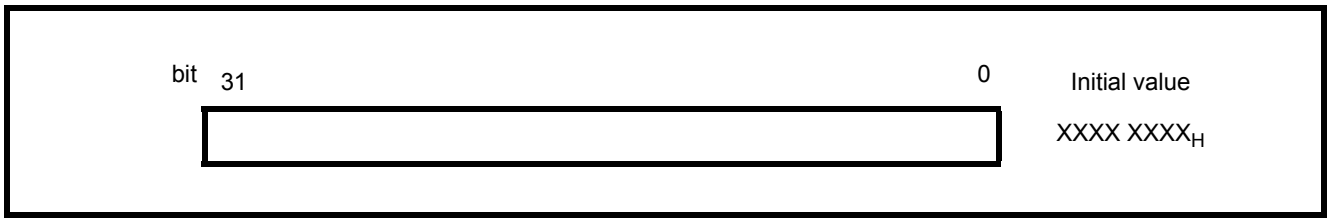
The initial value following a reset is "000F $FC00_H$".

3.7.5 Return Pointer (RP)

This pointer retains the return destination address when returning from a subroutine.

Figure 3-10 shows the bit configuration of the return pointer (RP).

Figure 3-10. Bit Configuration of the Return Pointer (RP)



The value of the program counter (PC) is transferred to this register when the CALL instruction is executed.

The register contents are transferred to the program counter (PC) when the RET instruction is executed.

3.7.6 System Stack Pointer (SSP)

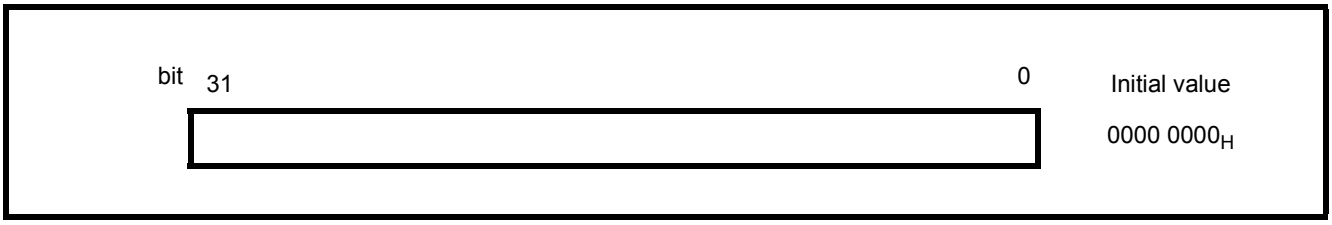
This pointer operates as R15 when the S flag of the condition code register (CCR) is "0".

Also, the system stack pointer (SSP) can be specified explicitly.

It can be used as a stack pointer specifying the stack for saving the program status register (PS) and the program counter (PC) when EIT occurs.

Figure 3-11 shows the bit configuration of the system stack pointer (SSP).

Figure 3-11. Bit Configuration of the System Stack Pointer (SSP)



The initial value following a reset is "0000 0000$_H$".

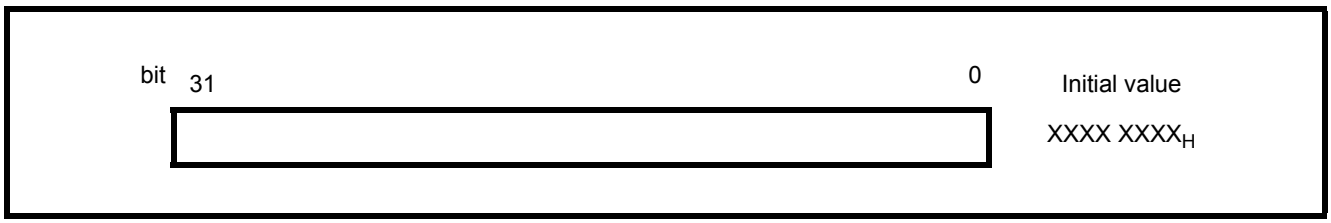
3.7.7 User Stack Pointer (USP)

This pointer operates as R15 when the S flag of the condition code register (CCR) is "1".

Also, the user stack pointer (USP) can be specified explicitly.

Figure 3-12 shows the bit configuration of the user stack pointer (USP).

Figure 3-12. Bit Configuration of the User Stack Pointer (USP)



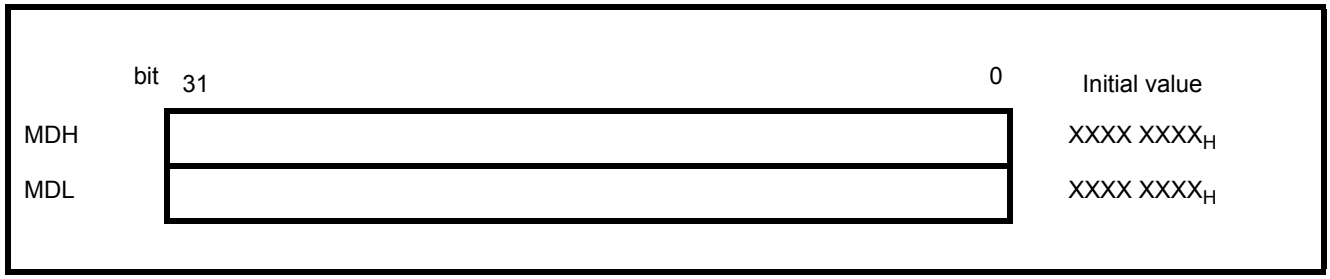
The initial value following a reset is undefined.

This pointer cannot be used in the RETI instruction.

3.7.8 Multiply & Divide Registers

These registers are used for multiplication and division, and each register has a length of 32 bits.

Figure 3-13. Bit Configuration of the Multiply & Divide Registers



The initial value following a reset is undefined.

- In multiplication

  In multiplication of 32 bits × 32 bits, the result of an operation with a length of 64 bits is stored in the Multiply & Divide registers at the following locations:

  - MDH: Upper 32 bits
  - MDL: Lower 32 bits

  In multiplication of 16 bits × 16 bits, the result is stored as follows:

  - MDH: Undefined
  - MDL: 32-bit result

- In division

  The dividend is stored in MDL at the start of calculation.

  In division according to the DIV0S, DIV0U, DIV1, DIV2, DIV3, or DIV4S instruction, the result is stored in MDH and MDL:

  - MDH: Remainder
  - MDL: Quotient

3.8 Data Configuration

Data is arranged in the FR80 family CPUs in the following two ways:

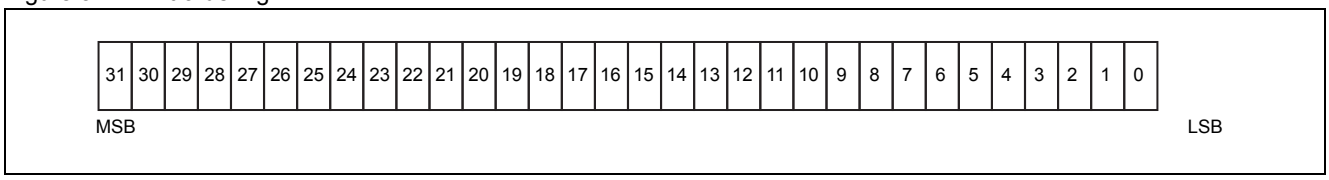
- Bit Ordering
- Byte Ordering

3.8.1 Bit Ordering

The FR80 family CPUs use little endian for bit ordering.

Figure 3-14 shows the bit ordering.

Figure 3-14. Bit ordering

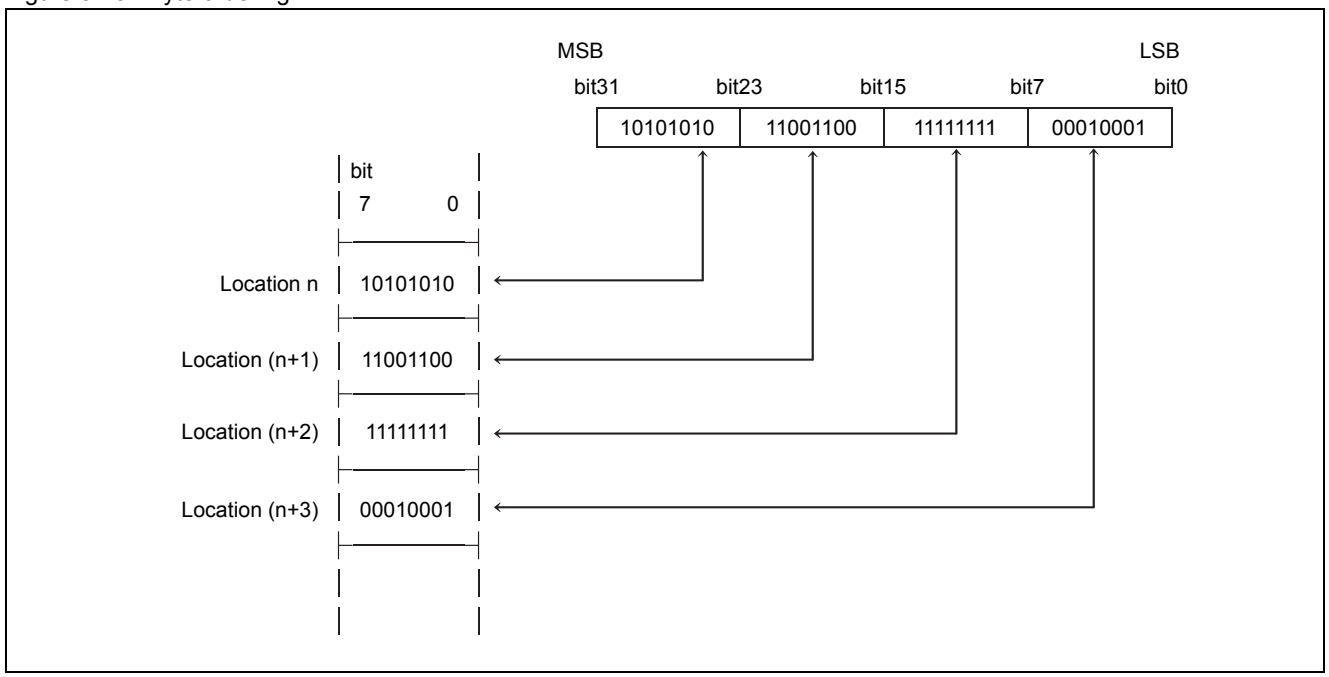


3.8.2 Byte Ordering

The FR80 family CPUs use big endian for byte ordering.

Figure 3-15 shows the byte ordering.

Figure 3-15. Byte ordering



3.8.3 Word Alignment

Program access

Programs for the FR80 family CPUs must be located at addresses that are multiples of 2. bit0 of the program counter (PC) is set to "0" when an instruction that entails the program counter (PC) update is executed. It is prohibited to specify an odd-numbered location as the branch destination address, and to set bit0 to "1".

The instruction would have to be located at an address that is a multiple of 2.

There is no odd-numbered address exception.

Data access

For an accessing of data in the FR80 family, set the address depending on the size of the data accessed as shown below. (The address is not aligned by the hardware.)

- Word access: The address is a multiple of 4 (the lowest 2 bits are set to "00").
- Half word access: The address is a multiple of 2 (the lowest bit is set to "0").
- Byte access: ----

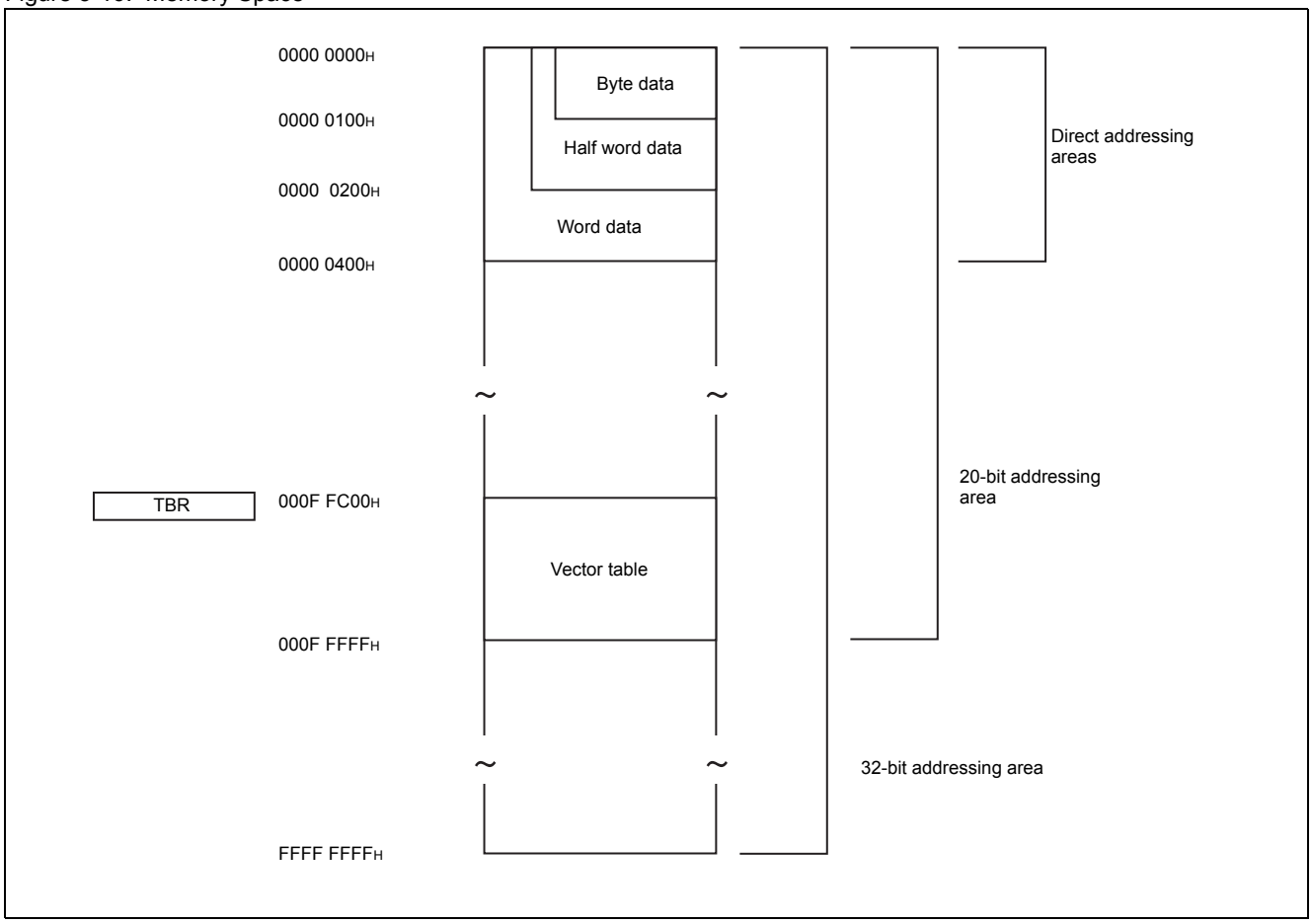
During a word or half word data access, set the above address for the result from a calculation of the effective address.

3.9 Addressing

The memory space consists of linear 32-bit addresses.

Figure 3-16 shows the memory space.

Figure 3-16. Memory Space



3.9.1 Direct Addressing Areas

The memory space areas listed below are areas for I/O. Direct addressing enables these areas to be specified directly as operand addresses in instructions.

The size of an address area that can be specified by a direct address varies depending on the data length.

- Byte data (8 bits): 0 to 0x0FF
- Half word data (16 bits): 0 to 0x1FF
- Word data (32 bits): 0 to 0x3FF

3.9.2 20-bit Addressing Area

20-bit addressing area: 0 to 0xFFFFF

If all the program and data areas are located in the 20-bit addressing area, programs will be more compact and therefore have high performance after compilation.

An example of expansion of a normal 20-bit branch macro instruction is shown below.

```
BRA20           label20,Ri
                ↓                       Code size
LDI:20          #label20,Ri             ;  4 bytes
JMP             @Ri                     ;  2 bytes
                                        ---------
                                        Total: 6 bytes
```

For details, see the "FR Family SOFTUNE C/C++ Compiler Manual for V6".

3.9.3 32-bit Addressing Area

32-bit addressing area: 0 to 0xFFFFFFFF

If the program and data areas are located beyond the 20-bit addressing area, the code sizes of programs will be larger than those of programs created in the 20-bit addressing area.

An example of expansion of a normal 32-bit branch macro instruction is shown below.

```
BRA32           label32,Ri
                ↓                       Code size
LDI:32          #label32,Ri             ;  6 bytes
JMP             @Ri                     ;  2 bytes
                                        ---------
                                        Total: 8 bytes
```

For details, see the "FR Family SOFTUNE C/C++ Compiler Manual for V6".

3.9.4 Vector Table Initial Area

The area from 000F $FC00_H$ to 000F $FFFF_H$ is the EIT vector table initial area.

The vector table used for EIT processing can be placed at an arbitrary address by changing the table base register (TBR) accordingly, but the initial address following a reset is the above address.

3.10 Branch Instructions

Operation with delay slots and operation without delay slots can be specified for branch instructions in the FR80 family CPUs.

3.10.1 Operation with Delay Slots

Instructions

The following instructions perform branch operations with delay slots:

JMP:D	@Ri	/	CALL:D	label12	/	CALL:D	@Ri	/	RET:D	
BRA:D	label9	/	BNO:D	label9	/	BEQ:D	label9	/	BNE:D	label9
BC:D	label9	/	BNC:D	label9	/	BN:D	label9	/	BP:D	label9
BV:D	label9	/	BNV:D	label9	/	BLT:D	label9	/	BGE:D	label9
BLE:D	label9	/	BGT:D	label9	/	BLS:D	label9	/	BHI:D	label9

Explanation of operation

The instruction that is located immediately following a branch instruction (the location is called a "delay slot") is executed before branching, and an instruction at the branch destination is executed after that. Because the instruction in the delay slot is executed before the branch operation, the apparent execution speed is 1 cycle. Such being the case, if no valid instruction can be entered in the delay slot, the NOP instruction must be placed there instead.

Example:

```
;          Order of instructions
           ADD       R1, R2;
           BRA:D     LABEL     ; Branch instruction
           MOV       R2, R3    ; Delay slot          ..... Executed before branching
           ...
LABEL:     ST        R3, @R4   ; Branch destination
```

The conditional branch instruction that is located in the delay slot is executed whether the branch condition is satisfied or not.

Although the sequence of execution of some instructions seems to be inverted for delay branch instructions, the sequence is inverted only when the program counter (PC) is updated. Any other operations, such as updating or referencing a register, are executed in the sequence described.

Concrete explanations are given below:

1. Ri referenced by the JMP:D @Ri / CALL:D @Ri instruction is not affected even when updated by the instruction in a delay slot.

   Example:

```
LDI:32    #Label, R0

JMP:D     @R0          ; Branching to Label

LDI:8     #0, R0       ; The branch destination address is not affected.

...
```

2. The return pointer (RP) referenced by the RET:D instruction is not affected even when the instruction in a delay slot updates the return pointer (RP).

   Example:

```
RET:D                  ; Branching to the address indicated by the RP
                         specified beforehand

MOV       R8, RP       ; The return operation is not affected.

...
```

3. The flag referenced by the Bcc:D rel instruction is not affected by the instruction in a delay slot either.

   Example:

```
ADD       #1, R0       ; Flag change

BC:D      Overflow     ; Branching according to the execution result of the
                         above instruction

ANDCCR    #0           ; This flag update is not referenced in the above
                         branch instruction.

...
```

4. When the RP is referenced in an instruction in the delay slot of the CALL:D instruction, the updated contents are read by the CALL:D instruction.

   Example:

```
CALL:D    Label        ; RP update and branching

MOV       RP, R0       ; Transfer of the RP of the execution result for the
                         above CALL:D

...
```

Instructions that can be placed in delay slots

Only instructions that satisfy the following conditions can be executed in delay slots:

- 1-cycle instruction
- Not a branch instruction
- Instruction that does not affect operations even if the order of execution is changed

Step trace trap

No step trace trap occurs between execution of a branch instruction with a delay slot and the delay slot.

Interrupts

No interrupt is accepted between execution of a branch instruction with a delay slot and the delay slot.

Undefined instruction exception

If the instruction except for BNO:D instruction in a delay slot is undefined, no undefined instruction exception occurs. In such cases, the undefined instruction operates as the NOP instruction.

Note: Do not place an undefined instruction in a delay slot of BNO:D instruction.

3.10.2 Operation without Delay Slots

Instructions

The following instruction performs branch operations without delay slots:

JMP	@Ri	/	CALL	label12	/	CALL	@Ri	/	RET	
BRA	label9	/	BNO	label9	/	BEQ	label9	/	BNE	label9
BC	label9	/	BNC	label9	/	BN	label9	/	BP	label9
BV	label9	/	BNV	label9	/	BLT	label9	/	BGE	label9
BLE	label9	/	BGT	label9	/	BLS	label9	/	BHI	label9

Explanation of operation

Instructions are executed in the order they are listed. No instruction that is coded immediately following a branch instruction is executed before branching.

Example:

```
;           Order of instructions
            ADD     R1, R2          ;
            BRA     LABEL           ; Branch instruction (without a delay slot)
            MOV     R2, R3          ; Not executed
            ...
LABEL       ST      R3, @R4         ; Branch destination
```

The number of execution cycles of a branch instruction without a delay slot is 2 cycles if there is branching and 1 cycle if there is no branching.

Such operation increases the instruction code efficiency compared with that of branch instructions with delay slots in which NOP is clearly written because appropriate instructions cannot be placed in the delay slots.

If valid instructions can be placed in delay slots, select operation with delay slots; otherwise, select operation without delay slots. Doing so can balance execution speed with code efficiency.

3.11 EIT (Exception, Interrupt, Trap)

EIT stands for Exception, Interrupt, and Trap. It indicates that the event that occurred results in suspension of execution of the current program, and the execution of another program.

An exception is an event that occurs in connection with the context being executed. The processing is re-executed beginning with the instruction that causes an exception.

An interrupt is an event that occurs independently of the context being executed. The source of events is hardware.

A trap is an event that occurs in connection with the context being executed. Some traps occur as instructed in programs such as a system call. The instruction following the instruction that generates a trap is re-executed first.

Features

- Multi-EIT support
- Level mask function for interrupts (A user can use 15 levels.)
- Trap instructions (INT/INTE)
- EIT for emulator activation (hardware/software)

3.11.1 EIT Sources

EIT sources include the following:

- Reset
- User interrupt (peripheral functions, external interrupts)
- Delay interrupt
- Undefined instruction exception
- Trap instruction (INT)
- Trap instruction (INTE)
- Step trace trap

3.11.2 Return from EIT

The return from each EIT is through the RETI instruction.

3.11.3 Interrupt Level

The interrupt levels are 0 to 31, and they are controlled in units of 5 bits.

Table 3-2 lists the assignment of each level.

Table 3-2. Interrupt Level Assignment Table

Level		Interrupt Type	Remarks
Binary number	Decimal number		
00000	0	(Reserved for system)	If the original value of the interrupt level mask register (ILM) is in a range of 16 to 31, no value in this range can be specified for the interrupt level mask register (ILM) from the program.
. . .	. . .	. . .	
. . .	. . .	. . .	
00011	3	(Reserved for system)	
00100	4	INTE instruction Step trace trap	
00101	5	(Reserved for system)	
. . .	. . .	. . .	
. . .	. . .	. . .	
01100	14	(Reserved for system)	
01101	15	(Reserved for system)	
10000	16	Interrupt request	When the interrupt level mask register (ILM) is set, user interrupts must be disabled.
10001	17	Interrupt request	
. . .	. . .	. . .	
. . .	. . .	. . .	
11110	30	Interrupt request	
11111	31	-	If the interrupt control register (ICR) is set, interrupts are disabled.

The operations are enabled only if the level is in a range of 16 to 31.

The interrupt level does not affect undefined instruction exceptions and the INT instruction. It does not change the interrupt level mask register (ILM) either.

3.11.4 I Flag

This flag specifies whether interrupts are enabled or disabled. It is provided as bit4 of the condition code register (CCR) in the program status register (PS).

I	Explanation
0	The bit is automatically cleared to "0" when the INT instruction is executed. (However, the value that is saved to the stack is that immediately before the bit is cleared.)
1	The mask processing of user interrupt requests is controlled with the value retained by the interrupt level mask register (ILM).

Note: After an instruction changes the value of the I flag, interrupt requests can be accepted beginning from the instruction after the next instruction.
Therefore, to operate interrupts properly, NOP must be placed after the instruction that changes the I flag value.

- Enabling interrupts (I flag = 1)

Instruction execution		I flag	Interrupts	
↓	ORCCR #set_iflag	0	Disabled	
	NOP	1	Disabled	
	Instruction A	1	Enabled	↑ **Starts enabling interrupts**

- Disabling interrupts (I flag = 0)

Instruction execution		I flag	Interrupts	
↓	ANDCCR #clear_iflag	1	Enabled	
	NOP	0	Enabled	
	Instruction A	0	Disabled	↑ **Starts disabling interrupts**

If an interrupt is received while executing an instruction to set I flag to "0", there is a delay for 1 cycle from execution of an instruction for I flag and ILM to change. Therefore, I flag becomes "0" although processing moves to the interrupt processing routine.

At this time, if multiple interrupts are generated, I flag can not receive any interrupt because it is "0", and processing of multiple interrupts is not performed.

I flag itself is updated when executing an instruction. Therefore, a value of I flag after update is saved to the stack, and when the value of the stack is returned, the value of I flag after update is reflected to PS register.

To receive a new interrupt within the interrupt routine, it is required to set software to make I flag to "1" at the beginning of the interrupt routine.

3.11.5 Interrupt Level Mask Register (ILM)

This register retains the interrupt level mask value. The register is provided as bit20 to bit16 of the program status register (PS).

An interrupt request that is input to a CPU in the FR80 family CPUs is accepted only if the corresponding interrupt level is higher than the level specified by the interrupt level mask register (ILM).

The highest level is "0" (00000), and the lowest is "31" (11111).

A limited range of values can be set from programs. If the original value is in a range of 16 to 31, a value ranging from 16 to 31 can be specified as a new value. If a value ranging from 0 to 15 is set for an instruction, (specified-value + 16) is transferred when the instruction is executed.

If the original value is in a range of 0 to 15, any value ranging from 0 to 31 can be specified. Use the STILM instruction for this setting.

Note: After an instruction changes the value of the interrupt level mask register (ILM), interrupt requests can be accepted beginning from the instruction after the next instruction.
Therefore, to operate interrupts properly, NOP must be placed after the instruction that changes the interrupt level mask register (ILM).

Instruction execution		ILM	Interrupt Accepted	
↓	SETILM #set_ILM_B	A	A	
	NOP	B	A	
	Instruction C	B	B	↑
	Instruction D	B	B	**Starts enabling ILM=B.**

3.11.6 Level Mask for Interrupts

When an interrupt request is generated, the interrupt level of the interrupt source is compared with the level mask value retained by the interrupt level mask register (ILM). Then, if the following condition is satisfied, the source is masked and the request is not accepted:

Interrupt level of source $\geq$ **Level mask value**

3.11.7 Interrupt Control Register (ICR)

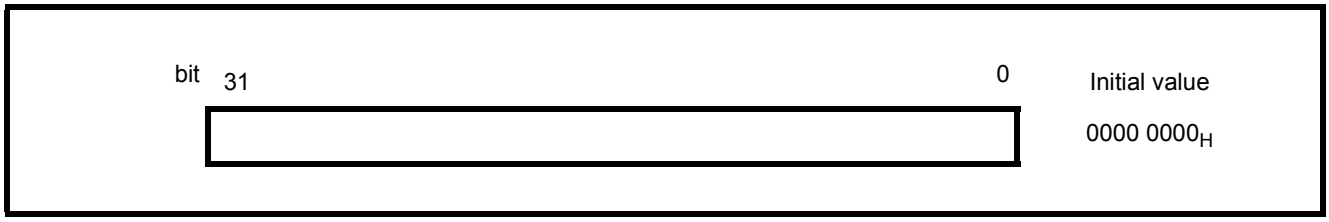
See "10. Interrupt Controller".

3.11.8 System Stack Pointer (SSP)

This pointer indicates the stack used for saving or restoring data, when EIT has been received or the return operation is performed.

Figure 3-17 shows the bit configuration of the system stack pointer (SSP).

Figure 3-17. Bit Configuration of the System Stack Pointer (SSP)



"8" is subtracted during EIT processing, and "8" is added at the time of return from EIT with the RETI instruction executed.

The initial value following a reset is "0000 0000_H".

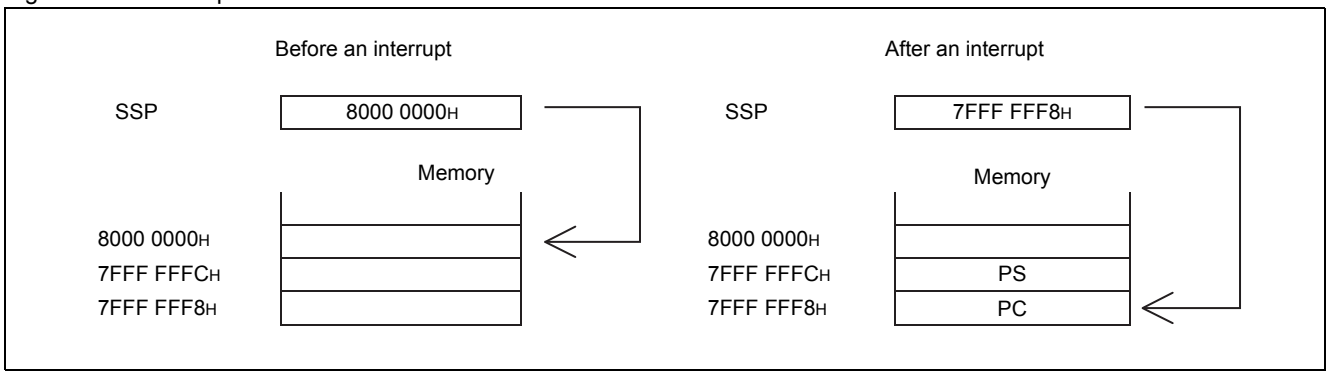
This pointer operates as general-purpose register R15 when the S flag of the condition code register (CCR) is "0".

3.11.9 Interrupt Stack

The interrupt stack is the area specified by the system stack pointer (SSP). It saves and restores the values of the program counter (PC) and the program status register (PS). After an interrupt, the value of the program counter (PC) is stored in the address specified by the system stack pointer (SSP), and the value of the program status register (PS) is stored in the address specified by the system stack pointer (SSP) plus 4.

Figure 3-18 shows the interrupt stack.

Figure 3-18. Interrupt Stack

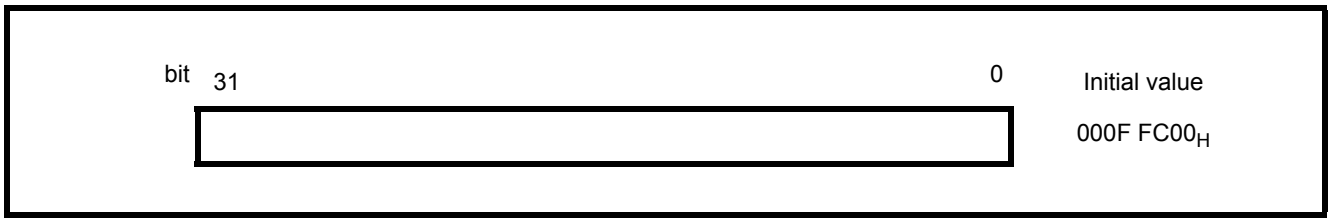


3.11.10 Table Base Register (TBR)

This register indicates the start address of the vector table used for EIT processing.

Figure 3-19 shows the bit configuration of the table base register (TBR).

Figure 3-19. Bit Configuration of the Table Base Register (TBR)



A vector address is the table base register (TBR) value plus the offset value assigned to each EIT source.

The initial value following a reset is "000F $FC00_H$".

3.11.11 EIT Vector Table

The vector area for EIT processing is the 1-KB area from the address specified by the table base register (TBR).

The size of 1 vector is 4 bytes, and the relationship between interrupt vector numbers and vector addresses is expressed as follows:

vctadr = TBR + vctofs

= TBR + (0x3FC - 4 × vct)

vctadr: Vector address vctofs: Vector offset vct: Interrupt vector number
TBR: Table base register

The lowest 2 bits of the addition result are always handled as "00".

The initial area of the vector table following a reset is the area from 000F $FC00_H$ to 000F $FFFF_H$.

Specific functions are assigned to part of the vectors.

3.11.12 Multi-EIT Processing

If multiple EIT sources occur at one time, the CPU selectively selects and accepts 1 EIT source, executes the EIT sequence, detects EIT sources again, and then repeats these actions. When no more detected EIT sources can be accepted, the CPU executes the handler instruction of the last EIT source accepted.

Therefore, if multiple EIT sources occur at one time, the sequence in which the handler of each source is executed depends on the following:

1. Priority in which EIT sources are accepted
2. The mask applied to other sources after a source is accepted

The sequence of execution depends on the above 2 elements.

The priority in which EIT sources are accepted is the order of selection of the source whose EIT sequence will be executed. In the EIT sequence, the program status register (PS) and the program counter (PC) are saved, the program counter (PC) is updated, and the other sources are masked as required. The handler of a source accepted earlier is not necessarily executed earlier.

Table 3-3 outlines the priority in which EIT sources are accepted.

Table 3-3. Priority in which EIT Sources are Accepted and Masking of Other Sources

Priority of Acceptance	Source	Masking of Other Sources	ILM
1	Reset	The other sources are abandoned.	15
2	Other than undefined instructions	All sources of lower priority	-
3	INT instruction	I flag = 0	-
4	INTE instruction	All sources of lower priority	4
5	User interrupt	ILM = Level of accepted source	ICR
6	Step trace trap	All sources of lower priority	4

With additional consideration given to the masking of other sources after an EIT source is accepted, the sequence of execution of the handlers of EIT sources that occur at one time is as shown below.

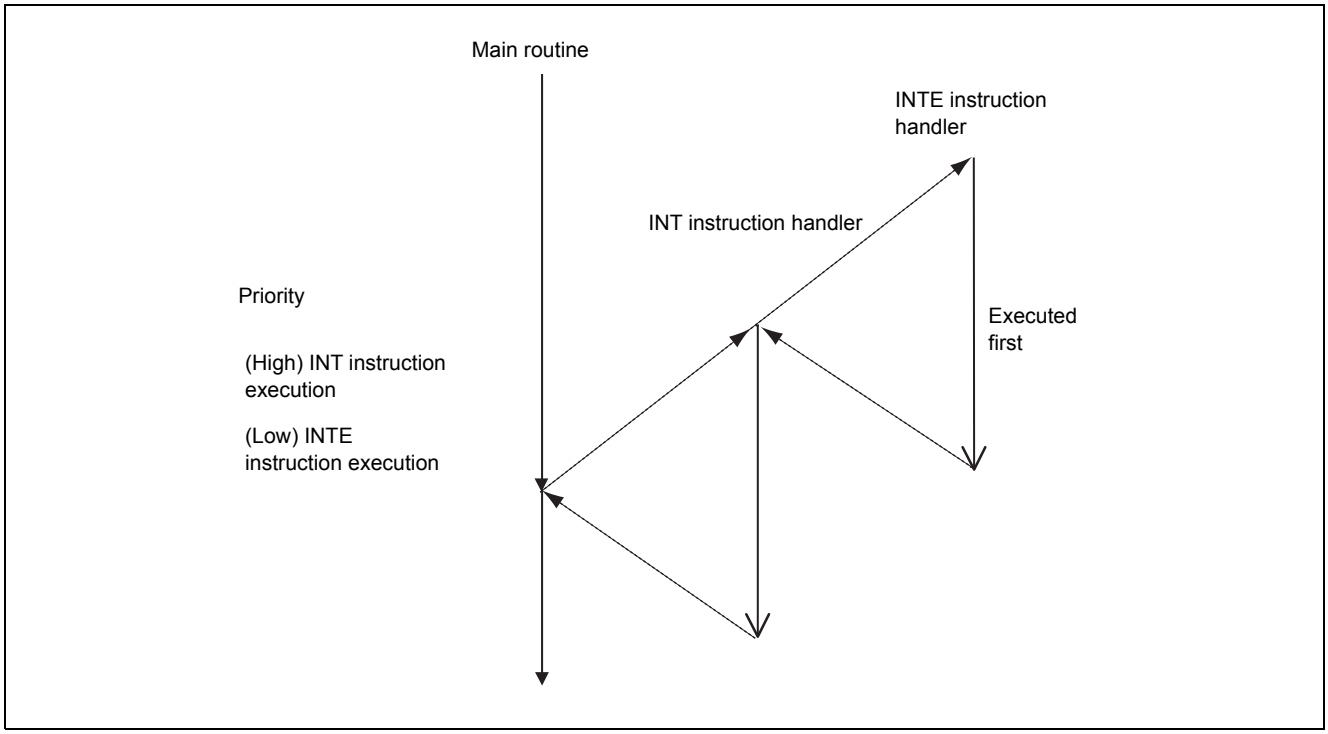
Table 3-4 lists the sequence of execution.

Table 3-4. Sequence of EIT handler execution

Priority of Acceptance	Source
1	Reset
2	Other than undefined instructions
3	INTE instruction
4	Step trace trap
5	INT instruction
6	User interrupt

Figure 3-20 shows multi-EIT processing.

Figure 3-20. Multi-EIT Processing



3.11.13 Operation

In the explanations in this section, the PC of the transfer source indicates the address of the instruction that detects each EIT source.

"Next instruction address" indicates the value corresponding to the case where each of the instructions below that detects EIT satisfies the respective condition shown:

- For LDI:32 instruction: PC + 6
- For LDI:20 instruction: PC + 4
- For other instructions: PC + 2

User interrupt operation

The sequence in which a generated user interrupt request is determined as accepted or not is shown below.

User interrupt requests are generated from peripheral functions, and an interrupt level is set for every interrupt request.

- Acceptance of interrupt requests
  1. The levels of interrupt requests generated simultaneously are compared, and the interrupt with the highest level (with the lowest numerical value) is selected.

     The value retained by the corresponding interrupt control register (ICR) is used for this comparison.
  2. If multiple interrupt requests generated at one time have the same interrupt level, the interrupt request with the lowest interrupt number is selected.
  3. An interrupt request with an interrupt level greater than or equal to the level mask value is masked and not accepted.

     If the level mask value is greater than the interrupt level, go to 4.
  4. In cases where the selected interrupt request can be masked, if the I flag is "0", the interrupt request is masked and not accepted. If the I flag is "1", the interrupt request is accepted.

  Under the above conditions, the interrupt request will be accepted when one instruction processing is completed.

  When an instruction that changes the I flag or interrupt level mask register (ILM) is executed, EIT control with the new acceptance condition becomes effective after 2 instructions.

  If an EIT request is detected at the same time that a user interrupt request is accepted, the CPU operates as follows using the interrupt number corresponding to the accepted interrupt request.

  Note: The parentheses () in "Operation" below indicate the address that a register points to.
- Operation

1	(TBR + vector offset of the accepted interrupt request)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	Next instruction address	→ (SSP)
6	Interrupt level of the accepted request	→ ILM
7	"0"	→ S flag
8	TMP	→ PC

After the interrupt sequence is completed, detection of any new EIT is performed before the first instruction of the handler is executed. If any EIT that occurred can be accepted at this point, the CPU switches to the EIT processing sequence.

3.11.14 INT Instruction Operation

The INT #u8 instruction generates a trap in software.

It generates a trap with the interrupt number specified in the operand.

- Operation

1	(TBR + 0x3FC - 4 × u8)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	PC + 2	→ (SSP)
6	"0"	→ I flag
7	"0"	→ S flag
8	TMP	→ PC

3.11.15 INTE Instruction Operation

The INTE instruction generates a trap in software for debugging.

- Operation

1	(TBR + 0x3D8)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	PC + 2	→ (SSP)
6	"00100_B"	→ ILM
7	"0"	→ S flag
8	TMP	→ PC

3.11.16 Step Trace Trap Operation

The step trace trap is a trap for debugging, and it is generated for each single instruction execution by setting the T flag of the program status register (PS). No step trace trap is generated immediately after execution of a branch instruction during execution of a delay branch instruction. It is generated after the instruction in the delay slot is executed.

- Step trace trap detection conditions
  1. T flag of the program status register (PS) = 1
  2. The instruction being executed is not a delay branch instruction.
  3. The CPU is in user mode.

  If the above conditions are satisfied, a break is set when one instruction operation processing is completed.
- Operation

1	(TBR + 0x3C4)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	Next instruction address	→ (SSP)
6	"00100_B"	→ ILM
7	"0"	→ S flag
8	TMP	→ PC

If the T flag = 1, user interrupts are disabled.

3.11.17 Undefined Instruction Exception Operation

When the instruction being decoded is detected as being undefined, an undefined instruction exception is generated.

- Undefined instruction exception detection conditions
  1. The instruction being decoded is detected as being undefined.
  2. The instruction is not in a delay slot (i.e., it does not immediately follow a delay branch instruction).

  If the above conditions are satisfied, an undefined instruction exception is generated and a break is set.
- Operation

1	(TBR + 0x3C4)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	PC	→ (SSP)
6	"0"	→ S flag
7	TMP	→ PC

The address of the instruction that detects an undefined instruction exception is saved as the program counter (PC).

3.11.18 RETI Instruction Operation

The RETI is an instruction to return from the EIT processing routine.

- Operation

1	(R15)	→ PC
2	R15 + 4	→ R15
3	(R15)	→ PS
4	R15 + 4	→ R15

The S flag must be "0" when the RETI instruction is executed.

3.11.19 Delay Slots and EIT

The delay slots of branch instructions have the following restrictions concerning EIT.

- Interrupts, traps

  No interrupt or trap occurs between execution of a branch instruction with a delay slot and the delay slot.
- Exceptions

  If the instruction in a delay slot is undefined, no undefined instruction exception occurs. In such cases, the undefined instruction operates as the NOP instruction.

4. Clock Generating Parts

This chapter explains the clock generating parts that generate the source clock (SRCCLK), which is the source of all internal clocks in this device.

4.1 Overview

4.2 Configuration

4.3 Pins

4.4 Registers

4.5 Explanation of Operations

4.1 Overview

The source clock (SRCCLK) is generated as the source of internal clocks used in operating this device.

This section explains generation and oscillation control of the source clock (SRCCLK) and selection of a clock as the source clock (SRCCLK).

Overview

This device operates with various internal clocks. The various internal clocks are generated by dividing the source clock (SRCCLK).

The following 3 clocks can be selected for the source clock (SRCCLK):

- Main clock (MCLK)
- PLL clock (PLLCLK)
- Sub clock (SBCLK)

The clock generating parts control the following:

- Main clock (MCLK) generation
  - Controls the oscillation of the main clock (MCLK).
  - Sets the oscillation stabilization wait time of the main clock (MCLK).
  - Controls the main timer or generation of main timer interrupt requests.
- Sub clock (SBCLK) generation
  - Controls the oscillation of the sub clock (SBCLK).
  - Sets the oscillation stabilization wait time of the sub clock (SBCLK).
  - Controls the sub timer or generation of sub timer interrupt requests.
- PLL clock (PLLCLK) generation
  - Controls the oscillation of the PLL clock (PLLCLK).
  - Sets the oscillation stabilization wait time of the PLL clock (PLLCLK).
  - Sets the PLL multiple rate (the main clock (MCLK) multiple rate for generating the PLL clock (PLLCLK)). The multiple rate can be set only for the main clock (MCLK), but not for the subclock (SBCLK).
- Source clock (SRCCLK) selection

  Selects one of 3 clocks for use as the source clock (SRCCLK).

4.2 Configuration

The clock generating parts consist of the clock generating parts themselves and the source clock (SRCCLK) selection block.

4.2.1 Clock Generating Parts

There are 3 clock generating parts. Any of the clocks generated by the clock generating parts can be selected for the source clock (SRCCLK).

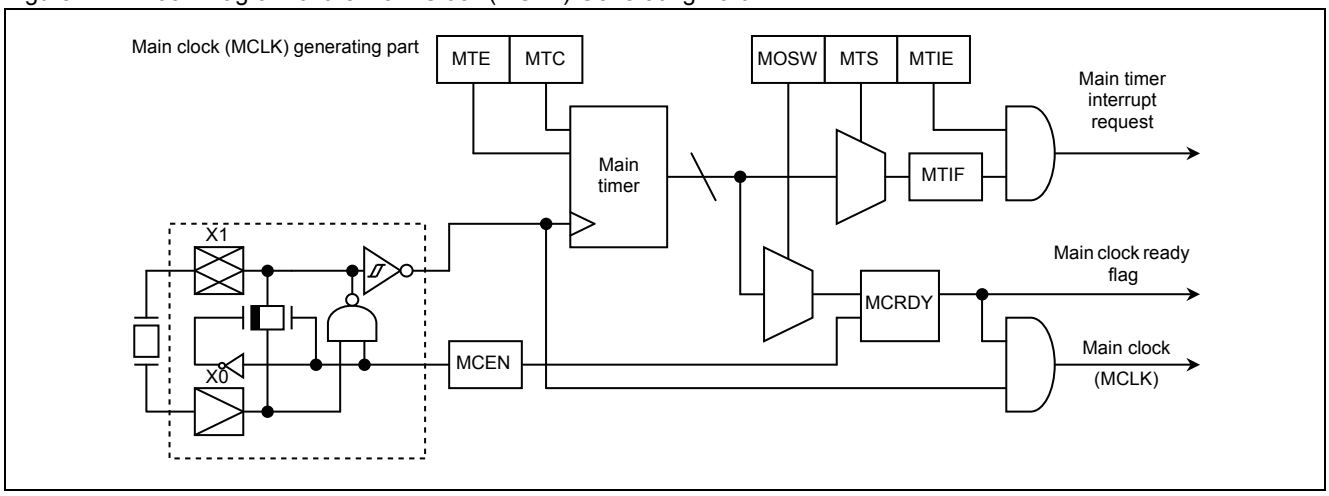
Main clock (MCLK) generating part

This part uses inputs from the X0 pin and X1 pin (main oscillator) to generate the main clock (MCLK).

The main clock (MCLK) is used to generate the PLL clock (PLLCLK).

Figure 4-1 shows a block diagram of the main clock (MCLK) generating part.

Figure 4-1. Block Diagram of the Main Clock (MCLK) Generating Part



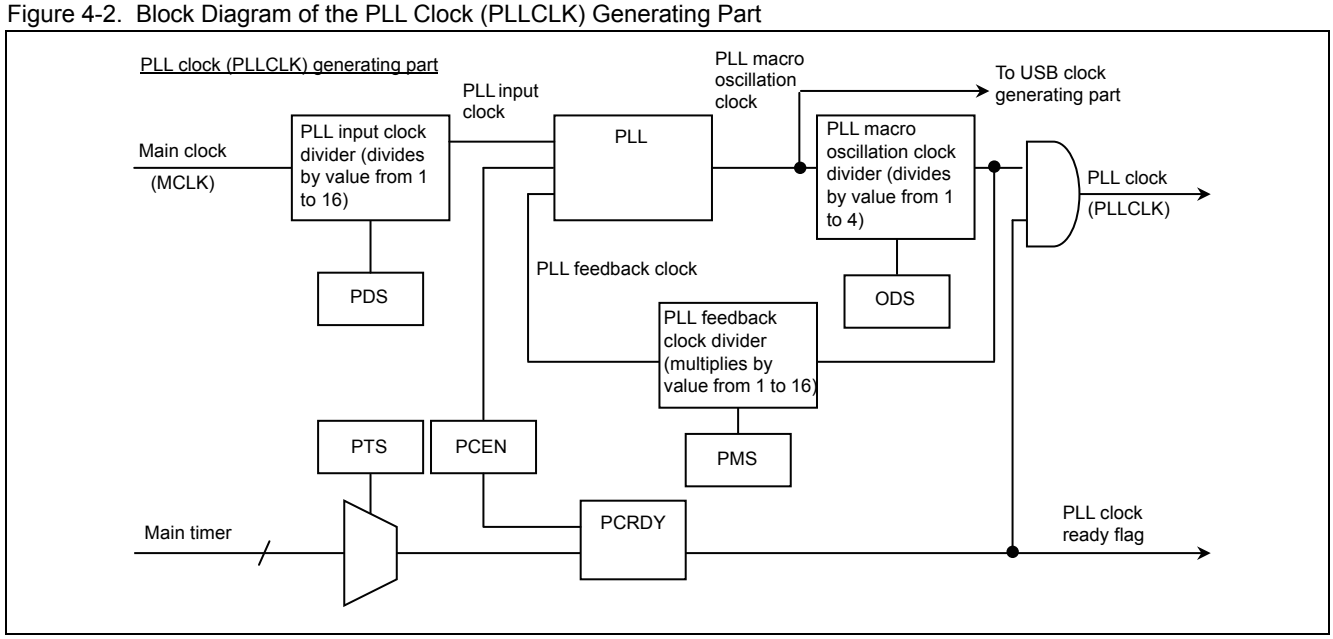
- Main timer
  The main timer operates with the main clock (MCLK). For details, see "6. Main Timer".

PLL clock (PLLCLK) generating part

This part multiplies the main clock (MCLK) to generate the PLL clock (PLLCLK).

Figure 4-2 shows a block diagram of the PLL clock (PLLCLK) generating part.

Figure 4-2. Block Diagram of the PLL Clock (PLLCLK) Generating Part



- PLL

  Clock multiplication circuit
- PLL input clock divider

  This divider divides the main clock (MCLK) to generate the PLL input clock.
- PLL feedback clock divider

  This divider divides the PLL clock (PLLCLK) generated by dividing the PLL macro oscillation clock in order to generate the PLL feedback clock.
- PLL macro oscillation clock divider

  This divider divides the PLL macro oscillation clock to generate the PLL clock (PLLCLK).

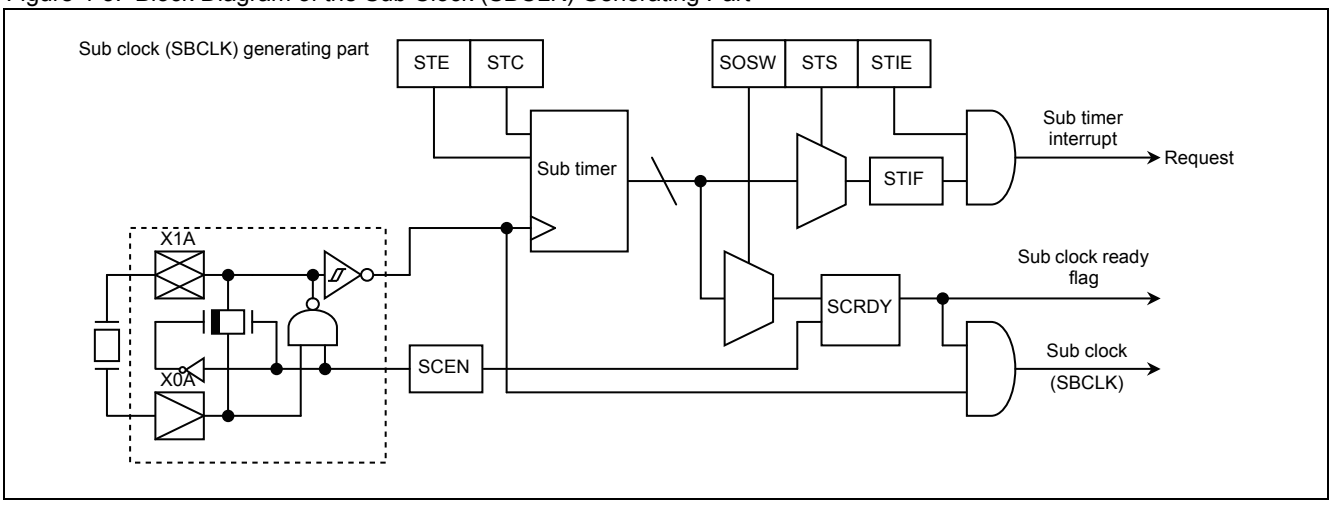
Sub clock (SBCLK) generating part

This part uses inputs from the X0A pin and X1A pin (sub oscillator) to generate the sub clock (SBCLK).

The sub clock (SBCLK) is the oscillation output as is.

Figure 4-3 shows a block diagram of the sub clock (SBCLK) generating part.

Figure 4-3. Block Diagram of the Sub Clock (SBCLK) Generating Part



- Sub timer

  The sub timer operates with the sub clock (SBCLK). For details, see "7. Sub Timer".

4.2.2 Source Clock (SRCCLK) Selection Block

This section explains selection of the source clock (SRCCLK). The source clock (SRCCLK) is selected from the following 3 clock sources:

- Main clock (MCLK) divided by 2
- PLL clock (PLLCLK)
- Sub clock (SBCLK)

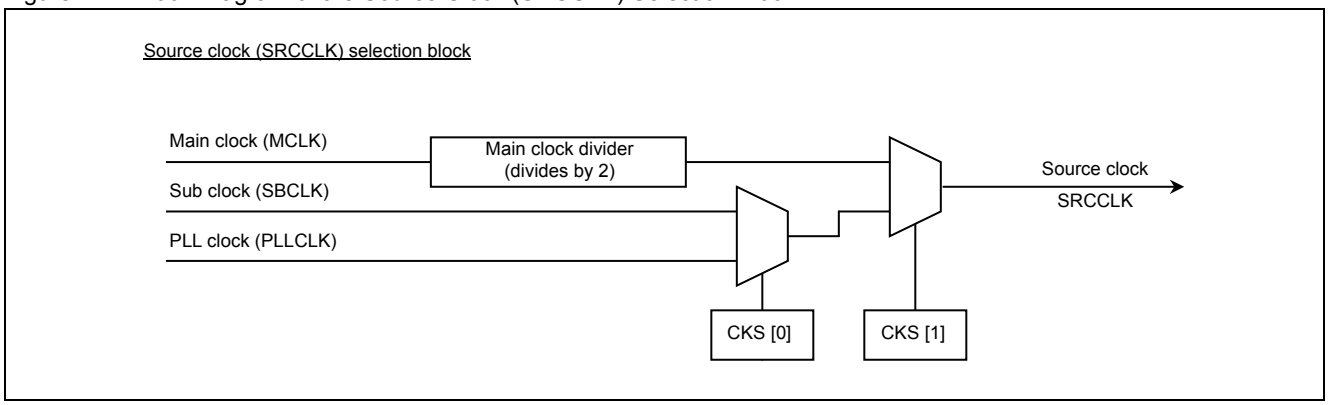
When an initialization reset (INIT) is generated, the settings of the source clock (SRCCLK) are initialized, and the main clock (MCLK) divided by 2 is set for the source clock (SRCCLK).

Change it to an arbitrary source clock (SRCCLK) with the setting of the clock source select register (CSELR) after the start of program operation.

Block diagram of the source clock (SRCCLK) selection block

Figure 4-4 shows a block diagram of the source clock (SRCCLK) selection block.

Figure 4-4. Block Diagram of the Source Clock (SRCCLK) Selection Block



- Main clock divider (divides by 2)
  The divider divides the main clock (MCLK) by 2 and sets the resultant value for the source clock (SRCCLK).
- CKS1 and CKS0 bits
  These bits are the source clock (SRCCLK) selection bits in the clock source select register (CSELR).

4.3 Pins

This section explains the pins of the clock generating parts.

Overview

- X0 and X1 pins

  These pins are used to generate the main clock (MCLK).
- X0A and X1A pins

  These pins are used to generate the sub clock (SBCLK).

  They are used to connect the oscillator to an external unit.

  The pins are multiplexed pins. For details of using the X0A and X1A pins of the sub clock (SBCLK), see "2.4 Setting Method for Pins".

4.4 Registers

This section explains the configuration and functions of registers of the clock generating parts.

Registers of the clock generating parts

Table 4-1 lists the registers of the clock generating parts.

Table 4-1. Registers of the Clock Generating Parts

Abbreviated Register Name	Register Name	Reference
CSELR	Clock source select register	4.4.1
CMONR	Clock source monitor register	4.4.2
CSTBR	Clock stabilization time select register	4.4.3
PLLCR	PLL configuration register	4.4.4

4.4.1 Clock Source Select Register (CSELR)

This register controls the clock source and selects the source clock (SRCCLK).

Figure 4-5 shows the bit configuration of the clock source select register (CSELR).

Figure 4-5. Bit Configuration of the Clock Source Select Register (CSELR)

bit	7	6	5	4	3	2	1	0
	SCEN	PCEN	MCEN	Reserved	Reserved	Reserved	CKS1	CKS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (at INIT)	0	0	1	0	0	0	0	0
Initial value (at RST)	[1]	[1]	[1]	0	0	0	[1]	[1]

R/W: Read/Write

[1]: Uninitialized bit

Notes:

- When this register is read, the actual setting value is not necessarily read. To verify that the value specified for this register has actually been made effective, read the clock source monitor register (CMONR).
- Before changing this register, verify that the value specified for this register is the same as the value of the clock source monitor register (CMONR).
- Writing of this register is ignored during switching of the clocks (CKS1, CKS0 ≠ CKM1, CKM0).

[bit7]: SCEN (Sub clock oscillation enable bit)

This bit controls the oscillation of the sub clock (SBCLK).

Written Value	Explanation	Remarks
0	The oscillation of the sub clock (SBCLK) is stopped.	The X0A or X1A pin can be used as a port (PK0, PK1).
1	The sub clock (SBCLK) starts oscillating.	The X0A and X1A pins are used to generate the sub clock (SBCLK).

Notes:

- If the sub clock (SBCLK) is selected with the CKS1 and CKS0 bits (CKS1, CKS0=11) as the source clock (SRCCLK), this bit cannot be changed.
- The sub timer is cleared when "0" is written to the bit.
- In stop mode, the oscillation of the sub clock (SBCLK) is stopped regardless of the value of the bit.

[bit6]: PCEN (PLL clock oscillation enable bit)

This bit controls the oscillation of the PLL clock (PLLCLK).

Written Value	Explanation
0	The oscillation of the PLL clock (PLLCLK) is stopped.
1	The PLL clock (PLLCLK) starts oscillating.

Notes:

- Write "0" to this bit to stop the oscillation of the PLL clock (PLLCLK) before entering stop mode.
- The bit cannot be changed under any of the following conditions:
  - When the PLL clock (PLLCLK) is selected with the CKS1 and CKS0 bits (CKS1, CKS0 = 10) as the source clock (SRCCLK)
  - When the oscillation of the main clock (MCLK) is stopped, or the oscillation stabilization wait time is in effect (MCRDY bit = 0 in the clock source monitor register (CMONR))
- This bit is changed to "0" when the MCEN bit (MCEN = 0) is specified to stop the oscillation of the main clock (MCLK).
- Do not change this bit from "0" to "1" while the main timer is being cleared (MTC bit = 1 in the main timer control register (MTMCR)).
- If this bit is changed from "0" to "1" to enable the oscillation of the PLL clock (PLLCLK), the main timer is cleared.

  In such cases, "1" is read from the MTC bit in the main timer control register (MTMCR).

[bit5]: MCEN (Main clock oscillation enable bit)

This bit controls the oscillation of the main clock (MCLK).

Written Value	Explanation
0	The oscillation of the main clock (MCLK) is stopped.
1	The main clock (MCLK) starts oscillating.

Notes:

- If any of the following is selected with the CKS1 or CKS0 bit as the source clock (SRCCLK), this bit cannot be changed.
  - The main clock (MCLK) is selected (CKS1, CKS0 = 00 or 01).
  - The PLL clock (PLLCLK) is selected (CKS1, CKS0 = 10).
- The main timer is cleared when "0" is written to this bit.
- In stop mode, the oscillation of the main clock (MCLK) is stopped regardless of the value of the bit.

[bit4 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: CKS1, CKS0 (Source clock select bits)

These bits select the source clock (SRCCLK).

CKS1	CKS0	Explanation
0	0	Main clock (MCLK) divided by 2
0	1	
1	0	PLL clock (PLLCLK)
1	1	Sub clock (SBCLK)

A clock whose oscillation is stopped or that has entered the oscillation stabilization wait time cannot be selected as the source clock (SRCCLK).

Furthermore, no switching from the PLL clock (PLLCLK) to the sub clock (SBCLK) or from the sub clock (SBCLK) to the PLL clock (PLLCLK) is possible.

lists the conditions for changes of this bit.

Table 4-2. CKS1 and CKS0 bit change conditions

Value before Change		Changeable Value [CKS1:CKS0]	Change Condition Bit Clock Source Monitor Register (CMONR)	Unchangeable Value [CKS1:CKS0]
CKS1	CKS0			
0	0	00, 01	MCRDY = 1	11
		10	PCRDY = 1	
0	1	00, 01	MCRDY = 1	10
		11	SCRDY = 1	
1	0	00	MCRDY = 1	01, 11
		10	PCRDY = 1	
1	1	01	MCRDY = 1	00, 10
		11	SCRDY = 1	

Do not write the unchangeable values listed in . For the procedures for switching the source clock (SRCCLK), see "4.5.2 Switching the Source Clock (SRCCLK)".

4.4.2 Clock Source Monitor Register (CMONR)

This register displays the clock source and state of the source clock (SRCCLK).

The value specified for the clock source select register (CSELR) can be verified by reading this register to verify whether it is actually effective.

Figure 4-6 shows the bit configuration of the clock source monitor register (CMONR).

Figure 4-6. Bit Configuration of the Clock Source Monitor Register (CMONR)

bit	7	6	5	4	3	2	1	0
	SCRDY	PCRDY	MCRDY	Reserved	Reserved	Reserved	CKM1	CKM0
Attribute	R	R	R	R	R	R	R	R
Initial value (at INIT)	0	0	1	0	0	0	0	0
Initial value (at RST)	[1]	[1]	[1]	0	0	0	[1]	[1]

R: Read only

*: Uninitialized bit

Notes:

- When changing a set value of the clock source select register (CSELR), be sure to read this register and verify that the read value is the same as the set value of the clock source select register (CSELR).
- Do not change the clock source select register (CSELR) unless the set value of the clock source select register (CSELR) matches the register value.

[bit7]: SCRDY (Sub clock ready bit)

This bit displays the sub clock (SBCLK) state.

Read Value	Explanation
0	The oscillation is stopped, or the oscillation stabilization wait time is in effect.
1	The oscillation stabilization is in effect. This clock can be used as the source clock (SRCCLK).

Notes:

- If this bit is "0", the sub clock (SBCLK) cannot be selected as the source clock (SRCCLK).
- After the SCEN bit in the clock source select register (CSELR) is changed from "1" to "0", this bit may be read as having a value of "1".

[bit6]: PCRDY (PLL clock ready bit)

This bit displays the PLL clock (PLLCLK) state.

Read Value	Explanation
0	The oscillation is stopped, or the oscillation stabilization wait time is in effect.
1	The oscillation stabilization is in effect. This clock can be used as the source clock (SRCCLK).

Notes:

- If this bit is "0", the PLL clock (PLLCLK) cannot be selected as the source clock (SRCCLK).
- After the PCEN bit in the clock source select register (CSELR) is changed from "1" to "0", this bit may be read as having a value of "1".

[bit5]: MCRDY (Main clock ready bit)

This bit displays the main clock (MCLK) state.

Read Value	Explanation
0	The oscillation is stopped, or the oscillation stabilization wait time is in effect.
1	The oscillation stabilization is in effect. This clock can be used as the source clock (SRCCLK).

Notes:

- If this bit is "0", neither the main clock (MCLK) nor the PLL clock (PLLCLK) can be selected as the source clock (SRC-CLK).
- After the MCEN bit in the clock source select register (CSELR) is changed from "1" to "0", this bit may be read as having a value of "1".

[bit4 to bit2]: Reserved bits

In case of reading	"0" is read.

[bit1, bit0]: CKM1, CKM0 (Source clock display bits)

These bits display the clock selected as the source clock (SRCCLK).

CKM1	CKM0	Explanation
0	0	The main clock (MCLK) divided by 2 is selected.
0	1	
1	0	The PLL clock (PLLCLK) is selected.
1	1	The sub clock (SBCLK) is selected.

4.4.3 Clock Stabilization Time Select Register (CSTBR)

This register sets the oscillation stabilization wait time of the clock source.

The oscillation stabilization wait time set in this register is used under the following conditions with the ready bit being "1" for the relevant clock:

- When returning from stop mode or watch mode
- When the main oscillation is stopped and an initialize reset (INIT) is generated
- When clock oscillation is enabled after being stopped

  The ready bits are as follows:
  - Sub clock: SCRDY bit
  - PLL clock: PCRDY bit
  - Main clock: MCRDY bit

Figure 4-7 shows the bit configuration of the clock stabilization select register (CSTBR).

Figure 4-7. Bit Configuration of the Clock Stabilization Time Select Register (CSTBR)

bit	7	6	5	4	3	2	1	0
	Reserved	SOSW2	SOSW1	SOSW0	MOSW3	MOSW2	MOSW1	MOSW0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
$\overline{\text{INIT}}$ pin = "L" level	0	0	0	0	0	0	0	0
Initial value (at INIT)	0	0	0	0	[1]	[1]	[1]	[1]
Initial value (at RST)	0	[1]	[1]	[1]	[1]	[1]	[1]	[1]

R/W: Read/Write

[1]: Uninitialized bit

Note: When the main oscillation is stopped and an initialize reset (INIT) is generated the main oscillation stabilization wait time after operation is restarted is the initial value of this register.

[bit7]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit6 to bit4]: SOSW2 to SOSW0 (Sub clock oscillation stabilization wait select bits)

These bits select the oscillation stabilization wait time of the sub clock (SBCLK).

SOSW2	SOSW1	SOSW0	Sub Clock (SBCLK) Oscillation Stabilization Wait Time	At 32.768 kHz
0	0	0	2^8 × Sub clock (SBCLK) period	About 7.8 ms
0	0	1	2^9 × Sub clock (SBCLK) period	About 15.6 ms
0	1	0	2^{10} × Sub clock (SBCLK) period	About 31.3 ms
0	1	1	2^{11} × Sub clock (SBCLK) period	62.5 ms
1	0	0	2^{12} × Sub clock (SBCLK) period	125.0 ms
1	0	1	2^{13} × Sub clock (SBCLK) period	250.0 ms
1	1	0	2^{14} × Sub clock (SBCLK) period	500.0 ms
1	1	1	2^{15} × Sub clock (SBCLK) period	1 s

Notes:

- The times listed in the table are calculated values. Use these values only as a guide because the actual times may include some errors depending on the oscillation state.
- Writing to this bit is ignored when the following conditions are satisfied (in the oscillation stabilization wait time of the sub clock (SBCLK)):
  - SCRDY bit = 0 in the clock source monitor register (CMONR)
  - SCEN bit = 1 in the clock source select register (CSELR)

[bit3 to bit0]: MOSW3 to MOSW0 (Main clock oscillation stabilization select bits)

These bits select the oscillation stabilization wait time of the main clock (MCLK).

MOSW3	MOSW2	MOSW1	MOSW0	Main Clock (MCLK) Oscillation Stabilization Wait Time	At 4 MHz	At 8 MHz	At 48 MHz
0	0	0	0	2^1 × Main clock (MCLK) period	500 ns	250 ns	About 42 ns
0	0	0	1	2^5 × Main clock (MCLK) period	8 µs	4 µs	About 667 ns
0	0	1	0	2^6 × Main clock (MCLK) period	16 µs	8 µs	About 1 µs
0	0	1	1	2^7 × Main clock (MCLK) period	32 µs	16 µs	About 3 µs
0	1	0	0	2^8 × Main clock (MCLK) period	64 µs	32 µs	About 5 µs
0	1	0	1	2^9 × Main clock (MCLK) period	128 µs	64 µs	About 11 µs
0	1	1	0	2^{10} × Main clock (MCLK) period	256 µs	128 µs	About 21 µs
0	1	1	1	2^{11} × Main clock (MCLK) period	512 µs	256 µs	About 43 µs
1	0	0	0	2^{12} × Main clock (MCLK) period	About 1 ms	512 µs	About 85 µs
1	0	0	1	2^{13} × Main clock (MCLK) period	About 2 ms	About 1 ms	About 171 µs
1	0	1	0	2^{14} × Main clock (MCLK) period	About 4 ms	About 2 ms	About 341 µs
1	0	1	1	2^{15} × Main clock (MCLK) period	About 8 ms	About 4 ms	About 683 µs
1	1	0	0	2^{17} × Main clock (MCLK) period	About 33 ms	About 16 ms	About 3 ms
1	1	0	1	2^{19} × Main clock (MCLK) period	About 131 ms	About 66 ms	About 11 ms
1	1	1	0	2^{21} × Main clock (MCLK) period	About 524 ms	About 262 ms	About 44 ms
1	1	1	1	2^{23} × Main clock (MCLK) period	About 2 s	About 1 s	About 175 ms

Notes:

- The times listed in the table are calculated values. Use these values only as a guide because the actual times may include some errors depending on the oscillation state.
- Specify an oscillation stabilization wait time as 25µs or longer for a product equipped with a regulator.
- Writing to this bit is ignored when the following conditions are satisfied (in the oscillation stabilization wait time of the main clock (MCLK)):
  - MCRDY bit = 0 in the clock source monitor register (CMONR)
  - MCEN bit = 1 in the clock source select register (CSELR)

4.4.4 PLL Configuration Register (PLLCR)

This register sets the multiple rate for generating the PLL clock (PLLCLK) from the main clock (MCLK).

For the calculation of the clock frequency and the multiple rate related to generating the PLL clock (PLLCLK), see "4.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK)".

Figure 4-8 shows the bit configuration of the PLL configuration register (PLLCR).

Figure 4-8. Bit Configuration of the PLL Configuration Register (PLLCR)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	ODS1	ODS0	PMS3	PMS2	PMS1	PMS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (at INIT)	0	0	0	0	0	0	0	0
Initial value (at RST)	0	0	[1]	[1]	[1]	[1]	[1]	[1]

bit	7	6	5	4	3	2	1	0
	PTS3	PTS2	PTS1	PTS0	PDS3	PDS2	PDS1	PDS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (at INIT)	1	1	1	1	0	0	0	0
Initial value (at RST)	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

R/W: Read/Write

[1]: Uninitialized bit

Note: Writing to this bit is ignored when the oscillation of the PLL clock (PLLCLK) is enabled (PCEN = 1 in the clock source select register (CSELR)).

[bit15, bit14]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit13, bit12]: ODS1, ODS0 (PLL macro oscillation clock division rate select bits)

These bits select the division rate from the PLL macro oscillation clock to the PLL clock (PLLCLK).

ODS1	ODS0	Explanation
0	0	PLL clock (PLLCLK) = PLL macro oscillation clock / 1
0	1	PLL clock (PLLCLK) = PLL macro oscillation clock / 2
1	0	PLL clock (PLLCLK) = PLL macro oscillation clock / 3
1	1	PLL clock (PLLCLK) = PLL macro oscillation clock / 4

[bit11 to bit8]: PMS3 to PMS0 (PLL clock multiple rate select bits)

These bits select the multiple rate from the PLL input clock to the PLL clock (PLLCLK).

PMS3	PMS2	PMS1	PMS0	PLL Clock (PLLCLK) Multiple Rate
0	0	0	0	PLL clock (PLLCLK) = PLL input clock × 1
0	0	0	1	PLL clock (PLLCLK) = PLL input clock × 2
0	0	1	0	PLL clock (PLLCLK) = PLL input clock × 3
0	0	1	1	PLL clock (PLLCLK) = PLL input clock × 4
0	1	0	0	PLL clock (PLLCLK) = PLL input clock × 5
0	1	0	1	PLL clock (PLLCLK) = PLL input clock × 6
0	1	1	0	PLL clock (PLLCLK) = PLL input clock × 7
0	1	1	1	PLL clock (PLLCLK) = PLL input clock × 8
1	0	0	0	PLL clock (PLLCLK) = PLL input clock × 9
1	0	0	1	PLL clock (PLLCLK) = PLL input clock × 10
1	0	1	0	PLL clock (PLLCLK) = PLL input clock × 11
1	0	1	1	PLL clock (PLLCLK) = PLL input clock × 12
1	1	0	0	PLL clock (PLLCLK) = PLL input clock × 13
1	1	0	1	PLL clock (PLLCLK) = PLL input clock × 14
1	1	1	0	PLL clock (PLLCLK) = PLL input clock × 15
1	1	1	1	PLL clock (PLLCLK) = PLL input clock × 16

[bit7 to bit4]: PTS3 to PTS0 (PLL clock oscillation stabilization wait time select bits)

These bits select the oscillation stabilization wait time of the PLL clock (PLLCLK).

PTS3	PTS2	PTS1	PTS0	PLL Clock (PLLCLK) Oscillation Stabilization Wait Time	At 4 MHz	At 8 MHz	At 48 MHz
1	0	0	0	2^9 × Main clock (MCLK) period	128.0 µs	64.0 µs	About 10.7 µs
1	0	0	1	2^{10} × Main clock (MCLK) period	256.0 µs	128.0 µs	About 21.3 µs
1	0	1	0	2^{11} × Main clock (MCLK) period	512.0 µs	256.0 µs	About 42.7 µs
1	0	1	1	2^{12} × Main clock (MCLK) period	About 1 ms	512.0 µs	About 85.3 µs
1	1	0	0	2^{13} × Main clock (MCLK) period	About 2 ms	About 1 ms	About 170.7 µs
1	1	0	1	2^{14} × Main clock (MCLK) period	About 4 ms	About 2 ms	About 341.3 µs
1	1	1	0	2^{15} × Main clock (MCLK) period	About 8 ms	About 4 ms	About 682.7 µs
1	1	1	1	2^{16} × Main clock (MCLK) period	About 16.4 ms	About 8 ms	About 1.4 ms

Notes:

- The times listed in the table are calculated values. Use these values only as a guide because the actual times may include some errors depending on the oscillation state.
- Always write "1" to the PTS3 bit.

[bit3 to bit0]: PDS3 to PDS0 (PLL input clock division select bits)

These bits select the main clock (MCLK) division rate for generating the PLL input clock.

PDS3	PDS2	PDS1	PDS0	PLL Input Clock Division Selection
0	0	0	0	PLL input clock = Main clock (MCLK) / 1
0	0	0	1	PLL input clock = Main clock (MCLK) / 2
0	0	1	0	PLL input clock = Main clock (MCLK) / 3
0	0	1	1	PLL input clock = Main clock (MCLK) / 4
0	1	0	0	PLL input clock = Main clock (MCLK) / 5
0	1	0	1	PLL input clock = Main clock (MCLK) / 6
0	1	1	0	PLL input clock = Main clock (MCLK) / 7
0	1	1	1	PLL input clock = Main clock (MCLK) / 8
1	0	0	0	PLL input clock = Main clock (MCLK) / 9
1	0	0	1	PLL input clock = Main clock (MCLK) / 10
1	0	1	0	PLL input clock = Main clock (MCLK) / 11
1	0	1	1	PLL input clock = Main clock (MCLK) / 12
1	1	0	0	PLL input clock = Main clock (MCLK) / 13
1	1	0	1	PLL input clock = Main clock (MCLK) / 14
1	1	1	0	PLL input clock = Main clock (MCLK) / 15
1	1	1	1	PLL input clock = Main clock (MCLK) / 16

4.5 Explanation of Operations

This section explains the operations of each clock source and how the source clocks are switched.

4.5.1 Explanation of Clock Source Operations

This section explains mainly oscillation control of the clock sources.

Main clock (MCLK)

This clock is generated with inputs from the X0 pin and X1 pin (main oscillator). It is used to generate the PLL clock.

The main clock is used in operating the main timer. (See "6. Main Timer".)

- Conditions for stopping oscillation

  The oscillation of the main clock (MCLK) stops under any of the following conditions:

  - When stop mode is in effect
  - When the sub clock (SBCLK) is selected for the source clock (SRCCLK) and the oscillation of the main clock (MCLK) is stopped (that is, when the following conditions are satisfied):
    1. CKS1 or CKS0 bit in the clock source select register (CSELR) = 11
    2. MCEN bit in the clock source select register (CSELR) = 0

  Supplying of the main clock (MCLK) starts after all the above oscillation stop conditions are cleared and the oscillation stabilization wait time specified by the MOSW3 to MOSW0 bits in the clock stabilization time select register (CSTBR) has elapsed.

- Selecting the oscillation stabilization wait time

  Supplying of the main clock (MCLK) starts after a wait for the oscillation of the main clock to stabilize once the oscillation has been enabled.

  The MOSW3 to MOSW0 bits in the clock stabilization time select register (CSTBR) specify the oscillation stabilization wait time of the main clock (MCLK).

  Input at the "L" level to the $\overline{\text{INIT}}$ pin initializes the MOSW3 to MOSW0 bits, returning the oscillation stabilization wait time to its initial value. In such cases, the initial value is $2^1 \times$ Main clock (MCLK) period.

  The MOSW3 to MOSW0 bits are not initialized by any other reset that occurs.

  Note: Specify an oscillation stabilization wait time as 25μs or longer for products equipped with regulators.

- End of the oscillation stabilization wait time

  The main clock (MCLK) is supplied at the end of the oscillation stabilization wait time.

  Checking the following values enables you to verify whether the main clock (MCLK) has entered the oscillation stabilization wait time while operation of the main clock (MCLK) is enabled.

Oscillation Stabilization Wait State Display	Oscillation Stabilization State Display
MCRDY = 0 in the clock source monitor register (CMONR)	MCRDY = 1 in the clock source monitor register (CMONR)

PLL clock (PLLCLK)

This high-performance clock multiplies and generates the main clock (MCLK).

- Conditions for stopping oscillation

The oscillation of the PLL clock (PLLCLK) stops under any of the following conditions:

- When the oscillation of the main clock (MCLK) is stopped, or the oscillation stabilization wait time is in effect
- (PCEN bit = 0 in the clock source select register (CSELR))
- When the following conditions are satisfied and a clock other than the PLL clock (PLLCLK) is selected for the source clock (SRCCLK):
  1. CKS1 or CKS0 bit in the clock source select register (CSELR) = a value other than 10
  2. PCEN bit in the clock source select register (CSELR) = 0

  Supplying of the PLL clock (PLLCLK) starts after all the above oscillation stop conditions are cleared and the oscillation stabilization wait time specified by the PTS3 to PTS0 bits in the PLL configuration register (PLLCR) has elapsed.

  Input at the "L" level to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the PCEN bit in the clock source select register (CSELR) to "0" and stops the oscillation of the PLL clock (PLLCLK). (To start the oscillation after such initialization, set the PCEN bit in the clock source select register (CSELR) to "1".)

- Selecting an oscillation stabilization wait time

  Supplying of the PLL clock (PLLCLK) starts after a wait for the oscillation of the PLL clock to stabilize once the oscillation has been enabled.

  The PTS3 to PTS0 bits in the PLL configuration register (PLLCR) specify the oscillation stabilization wait time of the PLL clock (PLLCLK).

  Input at the "L" level to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the PTS3 to PTS0 bits, returning the oscillation stabilization wait time to its initial value. In such cases, the initial value is $2^{16} \times$ Main clock (MCLK) period.

  To change the oscillation stabilization wait time, set the PTS3 to PTS0 bits, and then write "1" to the PCEN bit in the clock source select register (CSELR).

- End of the oscillation stabilization wait time

  The PLL clock (PLLCLK) is supplied at the end of the oscillation stabilization wait time.

  Checking the following values enables you to verify whether the PLL clock (PLLCLK) has entered the oscillation stabilization wait time while operation of the PLL clock (PLLCLK) is enabled.

Oscillation stabilization wait state display	Oscillation stabilization state display
PCRDY = 0 in the clock source monitor register (CMONR)	PCRDY = 1 in the clock source monitor register (CMONR)

Sub clock (SBCLK)

This clock is generated with inputs from the X0A pin and X1A pin (sub oscillator). The sub clock (SBCLK) is the oscillation output as is.

The sub clock is used in operating the sub timer (See "7. Sub Timer").

- Conditions for stopping oscillation

  The oscillation of the sub clock (SBCLK) stops under any of the following conditions:

  - When input to the $\overline{\text{INIT}}$ pin is at the "L" level
  - When stop mode is in effect
  - When any clock other than the sub clock (SBCLK) is selected for the source clock (SRCCLK) and the oscillation of the sub clock (SBCLK) is stopped (that is, when the following conditions are satisfied):
    1. CKS1 or CKS0 bit in the clock source select register (CSELR) = a value other than 11
    2. SCEN bit in the clock source select register (CSELR) = 0
  - The pins are set to use ports (the pins are multiplexed for the sub clock (SBCLK) generating part and the ports).

  Supplying of the sub clock (SBCLK) starts after all the above oscillation stop conditions are cleared and the oscillation stabilization wait time specified by the SOSW2 to SOSW0 bits in the clock stabilization time select register (CSTBR) has elapsed.

  Input at the "L" level to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the SCEN bit in the clock source select register (CSELR) to "0" and stops the oscillation of the sub clock (SBCLK). (To start the oscillation after such initialization, set the SCEN bit in the clock source select register (CSELR) to "1".)

- Selecting an oscillation stabilization wait time

  Supplying of the sub clock (SBCLK) starts after a wait for the oscillation of the sub clock to stabilize once the oscillation has been enabled.

  The SOSW2 to SOSW0 bits in the clock stabilization time select register (CSTBR) specify the oscillation stabilization wait time of the sub clock (SBCLK).

  Input at the "L" to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the SOSW2 to SOSW0 bits, returning the oscillation wait time to its initial value. In such cases, the initial value is $2^8 \times$ Sub clock (SBCLK) period.

  To change the oscillation stabilization wait time, set the SOSW2 to SOSW0 bits.

- End of the oscillation stabilization wait time

  The sub clock (SBCLK) is supplied at the end of the oscillation stabilization wait time.

  Checking the following values enables you to verify whether the sub clock (SBCLK) has entered the oscillation stabilization wait time while operation of the sub clock (SBCLK) is enabled.

Oscillation stabilization wait state display	Oscillation stabilization state display
SCRDY = 0 in the clock source monitor register (CMONR)	SCRDY = 1 in the clock source monitor register (CMONR)

4.5.2 Switching the Source Clock (SRCCLK)

This section explains switching of the source clock (SRCCLK).

Overview

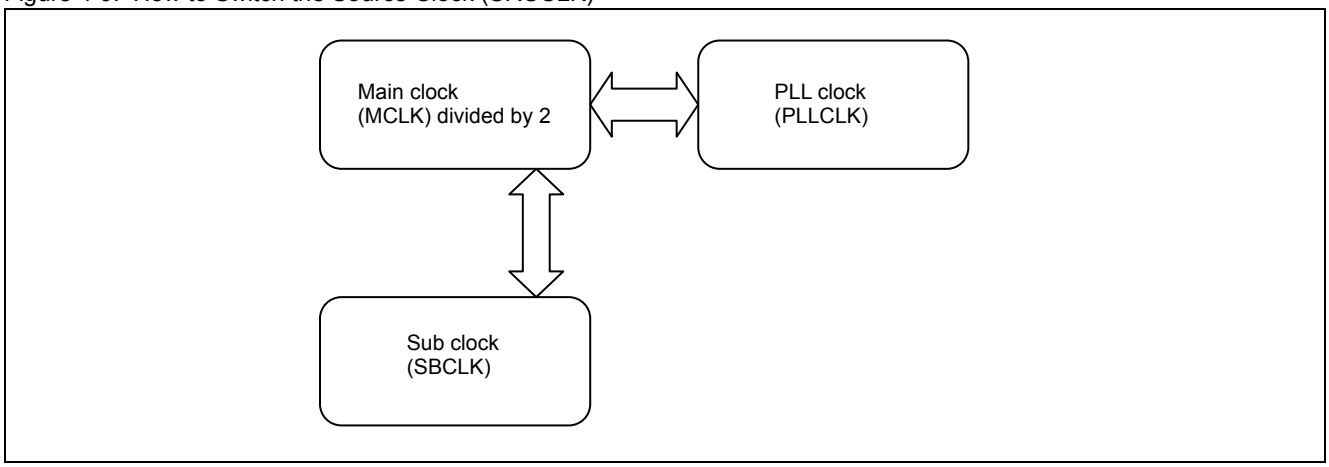
When "L" is input to the $\overline{INIT}$ pin or an initialization reset (INIT) is generated, the settings of the source clock (SRCCLK) are initialized, and the main clock (MCLK) divided by 2 is set for the source clock (SRCCLK).

The CKS1 and CKS0 bits of the clock source select register (CSELR) can be used to select the source clock (SRCCLK) from the clock sources after the start of program operation.

For this change to the source clock (SRCCLK), no switch from the PLL clock (PLLCLK) to the sub clock (SBCLK) or from the sub clock (SBCLK) to the PLL clock (PLLCLK) is possible. To do so, specify the main clock (MCLK) divided by 2, and then switch it.

Figure 4-9 shows how to switch the source clock (SRCCLK).

Figure 4-9. How to Switch the Source Clock (SRCCLK)



Note: Even if the source clock (SRCCLK) is switched, the oscillation enable settings (the values of the SCEN bit, PCEN bit, and MCEN bit in the clock source select register (CSELR)) of each clock are maintained. Stop the oscillation as necessary.

Procedures

- Switching from the main clock (MCLK) divided by 2 to the PLL clock (PLLCLK)

  To switch the source clock (SRCCLK) from the main clock (MCLK) divided by 2 to the PLL clock (PLLCLK), make settings by following the procedure below.

  1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 00 or 01) of the clock source monitor register (CMONR) to verify that the main clock (MCLK) divided by 2 is selected.
  2. Set the PLL multiple rate and the PLL clock (PLLCLK) oscillation stabilization wait time in the PLL configuration register (PLLCR).
  3. Set the PCEN bit (PCEN=1) in the clock source select register (CSELR) to start the oscillation of the PLL clock (PLLCLK).
  4. Check the PCRDY bit (PCRDY = 1) in the clock source monitor register (CMONR) to verify that the oscillation of the PLL clock (PLLCLK) has stabilized.
  5. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 10) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the PLL clock (PLLCLK).
  6. Check the CKM1 and CKM0 bits (CKM1, CKM = 10) in the clock source monitor register (CMONR) to verify that the source clock (SRCCLK) was switched to the PLL clock (PLLCLK).

Note: If the oscillation of the PLL clock (PLLCLK) has been enabled, steps 2 to 4 can be omitted.

- Switching from the PLL clock (PLLCLK) to the main clock (MCLK) divided by 2

  To switch the source clock (SRCCLK) from the PLL clock (PLLCLK) to the main clock (MCLK) divided by 2, make settings by following the procedure below.

  1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 10) in the clock source monitor register (CMONR) to verify that the PLL clock (PLLCLK) is selected.
  2. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 00) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the main clock (MCLK) divided by 2.
  3. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 00) in the clock source monitor register (CMONR) to verify that the source clock (SRCCLK) was switched to the main clock (MCLK) divided by 2.

- Switching from the main clock (MCLK) divided by 2 to the sub clock (SBCLK)

  To switch the source clock (SRCCLK) from the main clock (MCLK) divided by 2 to the sub clock (SBCLK), make settings by following the procedure below.

  1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 01) in the clock source monitor register (CMONR) to verify that the main clock (MCLK) divided by 2 is selected.
  2. Set the oscillation stabilization wait time of the sub clock (SBCLK) in the SOSW2 to SOSW0 bits in the clock stabilization time select register (CSTBR).
  3. Set the SCEN bit (SCEN=1) in the clock source select register (CSELR) to start the oscillation of the sub clock (SBCLK).
  4. Check the SCRDY bit (SCRDY = 1) in the clock source monitor register (CMONR) to verify that the oscillation of the sub clock (SBCLK) has stabilized.
  5. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 11) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the sub clock (SBCLK).
  6. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 11) in the clock source monitor register (CMONR) to verify that the source clock (SRCCLK) was switched to the sub clock (SBCLK).

  Note: If the oscillation of the sub clock (SBCLK) has been enabled, steps 2 to 4 can be omitted.

- Switching from the sub clock (SBCLK) to the main clock (MCLK) divided by 2

  To switch the source clock (SRCCLK) from the sub clock (SBCLK) to the main clock (MCLK) divided by 2, make settings by following the procedure below.

  1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 11) in the clock source monitor register (CMONR) to verify that the sub clock (SBCLK) is selected.
  2. Set the oscillation stabilization wait time of the main clock (MCLK) in the MOSW2 to MOSW0 bits in the clock stabilization time select register (CSTBR).
  3. Set the MCEN bit (MCEN=1) in the clock source select register (CSELR) to start the oscillation of the main clock (MCLK).
  4. Check the MCRDY bit (MCRDY = 1) in the clock source monitor register (CMONR) to verify that the oscillation of the main clock (MCLK) has stabilized.
  5. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 01) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the main clock (MCLK).
  6. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 01) in the clock source monitor register (CMONR) to verify that the source clock (CRCCLK) was switched to the main clock (MCLK).

  Note: If the oscillation of the main clock (MCLK) has been enabled, steps 2 to 4 can be omitted.

4.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK)

This section explains how to calculate the clock frequency and the multiple rate related to generating the PLL clock (PLLCLK).

PLL input clock frequency

= (Main oscillation frequency)/(Division rate set in the PDS bit in the PLL configuration register (PLLCR))

PLL multiple rate

= (Division rate set in the ODS bit in the PLL configuration register (PLLCR)) × (Multiple rate set in the PMS bit in the PLL configuration register (PLLCR))

PLL macro oscillation clock frequency

= (PLL input clock frequency) × PLL multiple rate

PLL clock (PLLCLK) frequency

= (PLL input clock frequency) × (Multiple rate set in the PMS bit in the PLL configuration register (PLLCR))

Table 4-3 lists sample settings of the PLL clock (PLLCLK).

Table 4-3. Sample Settings of the PLL Clock (PLLCLK)

Main Oscillation Frequency	PLL Configuration Register (PLLCR)			PLL Input Clock Frequency	PLL Multiple Rate ODS × PMS	PLL Macro Oscillation Clock Frequency	PLL Clock Frequency
	PDS3 to PDS0	ODS1, ODS0	PMS3 to PMS0				
48MHz	0010	10	0001	16 MHz	Multiplied by 6	96 MHz	32 MHz
4MHz	0000	10	0111	4 MHz	Multiplied by 24	96 MHz	32 MHz
8MHz	0000	10	0011	8 MHz	Multiplied by 12	96 MHz	32 MHz

Notes:

- The following conditions must be satisfied by the specified PLL input clock, PLL multiple rate, PLL macro oscillation clock, and Source Clock.

PLL Input Clock Frequency	4 to 24 MHz
PLL Multiple Rate	Multiplied by 2 to 24
PLL Macro Oscillation Clock Frequency	96 to 100 MHz
Source Clock (when PLL clock is selected)	24 to 33 MHz

- It is prohibited to set ODS=00 and PMS=0000 (PLL multiply rate=1).

5. Clock Division Control Part

This chapter explains the clock division control part that generates internal clocks.

5.1 Overview

5.2 Internal Clocks

5.3 Configuration

5.4 Registers

5.5 Division Rate

5.1 Overview

Internal clocks are generated by dividing the source clock (SRCCLK) input from a clock generating part.

The clock division control part divides the source clock (SRCCLK) and generates internal clocks to supply them to the CPU, bus, and/or peripheral functions.

Table 5-1 lists the internal clocks that are generated. These clocks are collectively called internal clocks.

Table 5-1. Internal Clocks that are Generated

Clock Name	Generation Source Clock
Base clock (BCLK)	Source clock (SRCCLK) divided by a value from 1 to 8
CPU clock (CCLK)	Base clock (BCLK) divided by 1 (undivided)
On-chip bus clock (HCLK)	Base clock (BCLK) divided by 1 (undivided)
Peripheral clock (PCLK)	Base clock (BCLK) divided by a value from 1 to 16
IP bus clock (IPCLK)	Base clock (BCLK) divided by a value from 1 to 8

For details of the source clock (SRCCLK), see "4. Clock Generating Parts".

5.2 Internal Clocks

This section explains the internal clocks.

Base clock (BCLK)

This clock is the generation source of all internal clocks.

The DIVB2 to DIVB0 bits of the divide clock configuration register 0 (DIVR0) are used when this clock is generated by dividing the source clock (SRCCLK) by a value ranging from 1 to 8.

The clock can decrease at once the operating frequency of the entire device.

It is stopped in one of the following low-power dissipation modes:

- Main timer mode
- Watch mode
- Stop mode

CPU clock (CCLK)

This clock is supplied to the CPU in this device and generated from the base clock (BCLK).

Since it is generated without dividing the base clock (BCLK), the operating frequency is always the same as that for the base clock (BCLK).

It is stopped in one of the following low-power dissipation modes:

- Doze mode (during a stop time)
- Sleep mode
- Main timer mode
- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
CPU clock (CCLK)	CPU (instruction execution block) Watchdog timer 1

On-chip bus clock (HCLK)

This clock is supplied to the on-chip bus and each circuit connected to the on-chip bus. It is generated from the base clock (BCLK).

Since it is generated without dividing the base clock (BCLK), the operating frequency is always the same as that for the base clock (BCLK).

It is stopped in one of the following low-power dissipation modes:

- Bus sleep mode
- Main timer mode
- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
On-chip bus clock (HCLK)	DMA controller (DMAC)

Peripheral clock (PCLK)

This clock is supplied to the peripheral buses and each peripheral function connected to the buses.

The DIVP3 to DIVP0 bits of divide clock configuration register 2 (DIVR2) are used when this clock is generated by dividing the base clock (BCLK) by a value ranging from 1 to 16.

It is stopped in one of the following low-power dissipation modes regardless of the setting:.

- Main timer mode
- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
Peripheral clock (PCLK)	Peripheral bus Clock control part Reset controller Watchdog timer 0 Interrupt controller External interrupt Delay interrupt 16-bit reload timer Each peripheral function

IP clock (IPCLK)

For the clock supplied by IP bus and each circuit connected to it, the clock is generated by dividing the on-chip bus clock (HCLK).

The division from undivided to divided by 2 to 8 can be done by the bit settings for bit2-0:DIVIP [2:0] bit of the DIVR0 register.

The clock stops at the following low-power consumption modes.

- Main timer mode
- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
IP clock (IPCLK)	OSDC

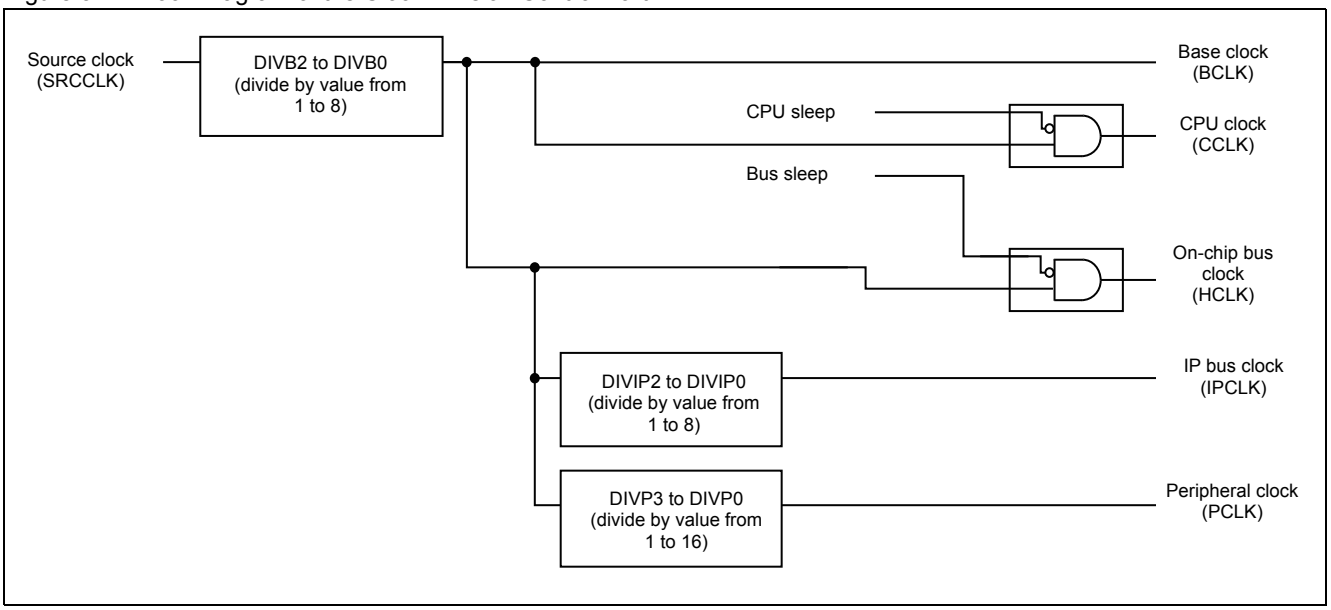
5.3 Configuration

The source clock input from a clock generating part is divided by the value specified in a register and output to a circuit.

Block diagram of the clock division control part

Figure 5-1 is a block diagram of the clock division control part.

Figure 5-1. Block Diagram of the Clock Division Control Part



5.4 Registers

This section explains the configuration and functions of registers of the clock division control part.

Registers of the clock division control part

Table 5-2 lists the registers of the clock division control part.

Table 5-2. Registers of the Clock Division Control Part

Abbreviated Register Name	Register Name	Reference
DIVR0	Divide clock configuration register 0	5.4.1
DIVR2	Divide clock configuration register 2	5.4.2

5.4.1 Divide Clock Configuration Register 0 (DIVR0)

This register sets the source clock (SRCCLK) division rate for generating the base clock (BCLK).

Figure 5-2 shows the bit configuration of divide clock configuration register 0 (DIVR0).

Figure 5-2. Bit Configuration of Divide Clock Configuration Register 0 (DIVR0)

bit	7	6	5	4	3	2	1	0
	DIVB2	DIVB1	DIVB0	Reserved	Reserved	DIVIP2	DIVIP1	DIVIP0
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	1	1

R/W: Read/Write

[bit7 to bit5]: DIVB2 to DIVB0 (base clock division configuration bits)

These bits set the division rate for generating the base clock (BCLK) from the source clock (SRCCLK).

Since the CPU clock (CCLK) and the on-chip bus clock (HCLK) are generated without dividing the base clock (BCLK), the frequency is the same as that for the base clock (BCLK).

DIVB2	DIVB1	DIVB0	Explanation
0	0	0	Divided by 1 (undivided)
0	0	1	Divided by 2
0	1	0	Divided by 3
0	1	1	Divided by 4
1	0	0	Divided by 5
1	0	1	Divided by 6
1	1	0	Divided by 7
1	1	1	Divided by 8

[bit4, bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2 to bit0]: DIVIP2 to DIVIP0: IP clock division setting

The division at the part for generating from the base clock (BCLK) to the IP clock (IPCLK) is set according to the following settings.

DIVIP2	DIVIP1	DIVIP0	Description
0	0	0	Divided by 1 (undivided)
0	0	1	Divided by 2
0	1	0	Divided by 3
0	1	1	Divided by 4
1	0	0	Divided by 5
1	0	1	Divided by 6
1	1	0	Divided by 7
1	1	1	Divided by 8

5.4.2 Divide Clock Configuration Register 2 (DIVR2)

This register sets the base clock (BCLK) division rate for generating the peripheral clock (PCLK).

Figure 5-3 shows the bit configuration of divide clock configuration register 2 (DIVR2).

Figure 5-3. Bit Configuration of Divide Clock Configuration Register 2 (DIVR2)

bit	7	6	5	4	3	2	1	0
	DIVP3	DIVP2	DIVP1	DIVP0	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	1	0	0	0	0

R/W: Read/Write

[bit7 to bit4]: DIVP3 to DIVP0 (Peripheral clock division configuration bits)

These bits set the division rate for generating the peripheral clock (PCLK) from the base clock (BCLK).

DIVP3	DIVP2	DIVP1	DIVP0	Explanation
0	0	0	0	Divided by 1 (undivided)
0	0	0	1	Divided by 2
0	0	1	0	Divided by 3
0	0	1	1	Divided by 4
0	1	0	0	Divided by 5
0	1	0	1	Divided by 6
0	1	1	0	Divided by 7
0	1	1	1	Divided by 8
1	0	0	0	Divided by 9
1	0	0	1	Divided by 10
1	0	1	0	Divided by 11
1	0	1	1	Divided by 12
1	1	0	0	Divided by 13
1	1	0	1	Divided by 14
1	1	1	0	Divided by 15
1	1	1	1	Divided by 16

[bit3 to bit0]: Reserved bits

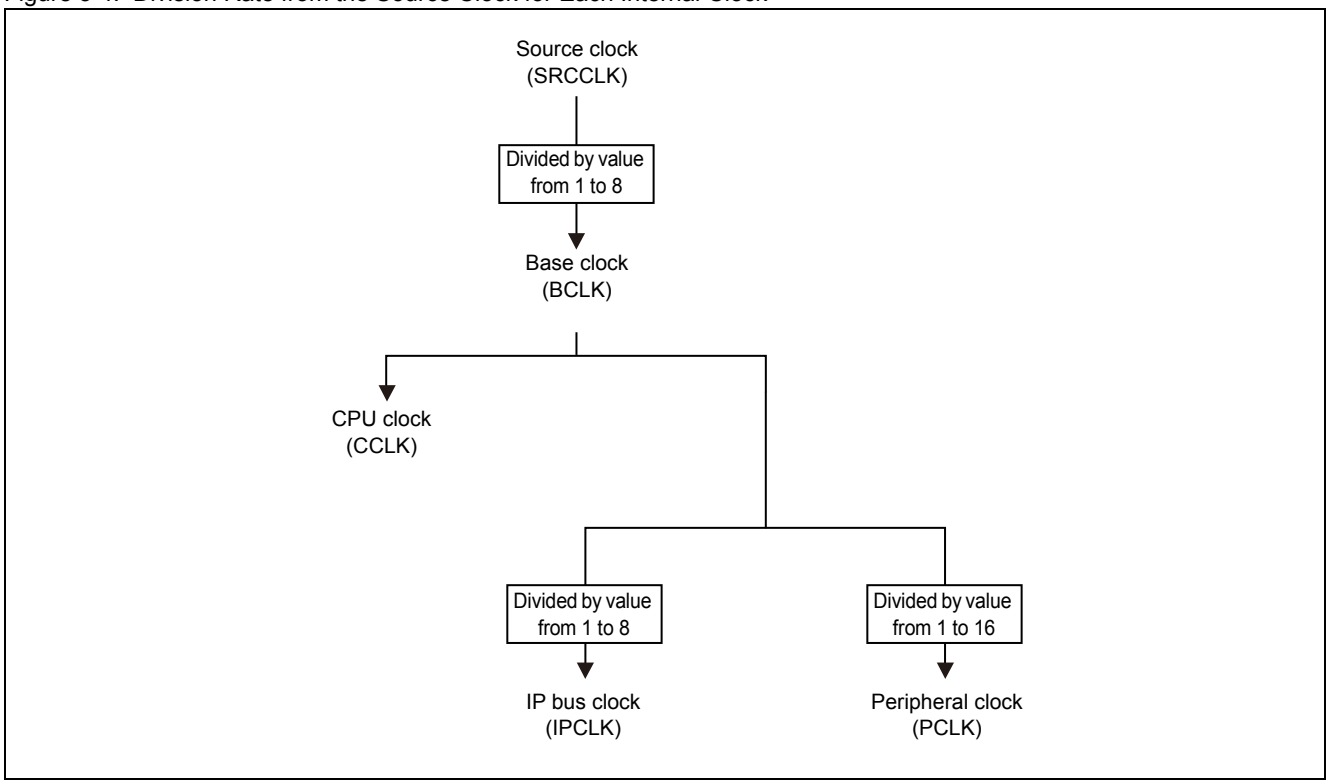
In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is red.

5.5 Division Rate

The clock division control part can set the division rate for each internal clock.

Figure 5-4 shows the division rate from the source clock for each internal clock.

Figure 5-4. Division Rate from the Source Clock for Each Internal Clock



Division rates after initialization

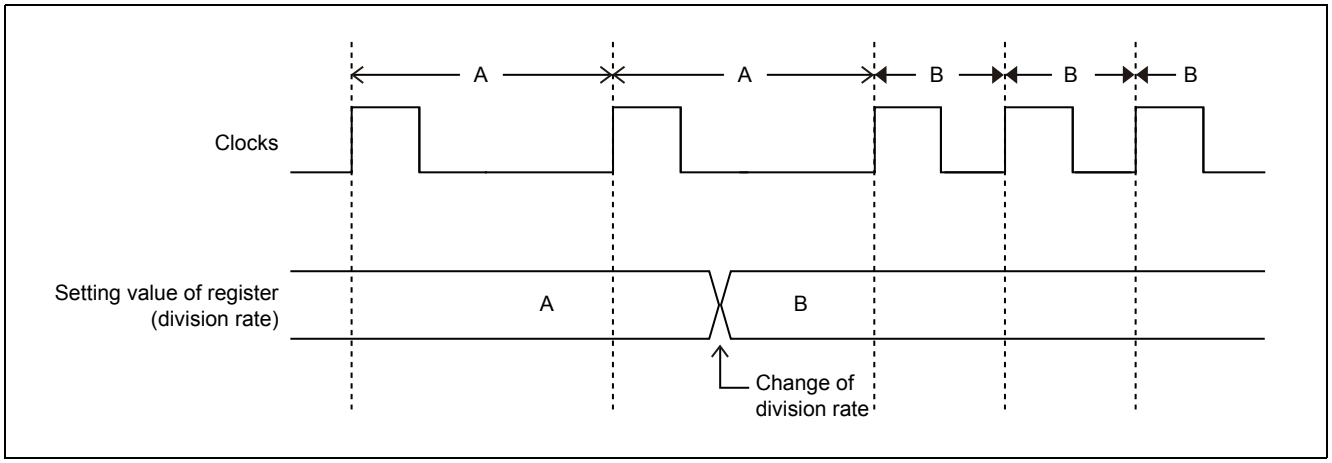
Table 5-3 shows the division of internal clocks after a reset.

Table 5-3. Division rates after a reset

Clock Name	Division Rate after Initialization
Base clock (BCLK)	Source clock (SRCCLK) divided by 1
CPU clock (CCLK)	Base clock (BCLK) divided by 1
On-chip bus clock (HCLK)	Base clock (BCLK) divided by 1
Peripheral clock (PCLK)	Base clock (BCLK) divided by 4
IP bus clock (IPCLK)	Base clock (BCLK) divided by 4

Changing the division rate

After the division rate setting is changed, the changed division rate is enabled at the next rising edge of the clock.



6. Main Timer



This chapter explains the functions and operations of the main timer function.

6.1 Overview

6.2 Configuration

6.3 Registers

6.4 Interrupts

6.5 An Explanation of Operations and Setting Procedure Examples

6.1 Overview

The main timer operates with the main clock (MCLK).

The main timer is used to generate the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

The main timer counts the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

When main clock (MCLK) oscillation is stable, the main timer can also be used as an interval timer for generating an interrupt request at regular intervals.

The main timer is cleared when:

- "1" is written to the MTC bit of the main timer control register (MTMCR).

  "1" is read from the MTC bit of the main timer control register (MTMCR) until the main timer is cleared after "1" is written to the MTC bit.
- Main clock (MCLK) oscillation is stopped.

  (The MCEN bit of the clock source select register (CSELR) is 0.)
- In stop mode
- The main timer is stopped with the MTE bit (MTE = 0) of the main timer control register (MTMCR).

If main timer operation is disabled, the timer is stopped during periods other than the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

6.2 Configuration

This section explains the main timer configuration.

Main timer block diagram

For the main timer block diagram, see "Main clock (MCLK) generating part" in "4. Clock Generating Parts".

Clocks

Table 6-1 shows the clocks used by the main timer.

Table 6-1. Clocks used by the Main Timer

Clock Name	Description
Operation clock	Main clock (MCLK)

6.3 Registers

This section explains the configuration and functions of registers used by the main timer.

Registers of main timer

Table 6-2 shows the registers used by the main timer.

Table 6-2. Main Timer Registers

Abbreviated Register Name	Register Name	Reference
MTMCR	Main timer control register	6.3.1

6.3.1 Main Timer Control Register (MTMCR)

This register controls the main timer.

Figure 6-1 shows the bit configuration of the main timer control register (MTMCR).

Figure 6-1. Bit Configuration of Main Timer Control Register (MTMCR)

bit	7	6	5	4	3	2	1	0
	MTIF	MTIE	MTC	MTE	MTS3	MTS2	MTS1	MTS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	1	1	1	1

R/W: Read/Write

Notes:

- This register can be rewritten only when the main clock (MCLK) is oscillating stably (The MCRDY bit of the clock source monitor register (CMONR) is 1).

  Note that the MTIE bit can be rewritten even when the MCRDY bit is "0".
- Software reset must be executed when both the MTE and MTC bits are "0". For details of the software reset, see "9. Reset".

[bit7]: MTIF (main timer interrupt flag bit)

This flag indicates that the main timer overflows.

The main timer overflows when:

- The counter has finished counting the period that is set with the MTS3 to MTS0 bits.
- The oscillation stabilization wait time of the main clock (MCLK) has elapsed after the MCEN bit of the clock source select register (CSELR) was rewritten from "0" to "1".
- The oscillation stabilization wait time of the main clock (MCLK) has elapsed after the system returns from stop mode.

A main timer interrupt request occurs when this bit is set to "1" while the MTIE bit is "1".

MTIF	In case of reading	In case of writing
0	No overflow occurred.	This bit is cleared to "0".
1	An overflow occurred.	Ignored

This bit is also cleared to "0" when a DMA transfer is caused by a main timer interrupt request.

Notes:

- Disabling main timer operation with the MTE bit (MTE = 0) clears the main timer.
- When the MTIE bit is set to "0", this bit is not cleared even when a DMA transfer is caused by a main timer interrupt request.
- After this device is reset by input of an "L" level signal from the $\overline{\text{INIT}}$ pin, an "H" level signal may be input again from the $\overline{\text{INIT}}$ pin. In this case, this bit is not changed to "1" even after the oscillation stabilization wait time of the main clock (MCLK) elapses.
- If clearing the bit to "0" coincides with the occurrence of an overflow, the overflow occurrence is given priority and this bit remains "1".
- When a read-modify-write instruction is used, "1" is read.

[bit6]: MTIE (main timer interrupt enable bit)

The MTIE bit is used to specify whether to cause a main timer interrupt request when the main timer overflows (MTIF=1).

A main timer interrupt request occurs when the MTIF bit is set to "1" while this bit is "1".

Written Value	Explanation
0	Disables generation of main timer interrupt requests.
1	Enables generation of main timer interrupt requests.

[bit5]: MTC (main timer clear bit)

Clear the main timer.

The operating state of the main timer can be verified by reading this bit.

MTC	In case of writing	In case of reading
0	Ignored	In normal operation
1	Clear the main timer.	The main timer is being cleared.

Notes:

- When a read-modify-write instruction is used, "0" is read.
- Do not clear the main timer during oscillation stabilization wait time of the PLL clock (PLLCLK).
- This register can be rewritten only while main clock (MCLK) oscillation is stable. Therefore, if the following conditions are satisfied, the main timer cannot be cleared even when the bit is set to "1":
  - Main clock (MCLK) is oscillating (the MCEN bit of the clock source select register (CSELR) is 1).
  - The main clock (MCLK) is in oscillation stopped/oscillation stabilization wait state (The MCRDY bit of the clock source monitor register (CMONR) is 0).
- Writing "1" to this bit at the same time that the MTE bit is changed from "0" to "1" clears the main timer and then starts main timer operation.
- Do not write "1" to this bit when it is "1".
- As long as the MTC bit is "0", the MTIF bit may become "1".

[bit4]: MTE (main timer operation enable bit)

This bit enables/disables (stops) the operation of the main timer.

Written Value	Explanation
0	Disables (stops) the operation of the main timer.
1	Enables the operation of the main timer.

Notes:

- If the operation of the main timer is disabled (stopped), the main timer is stopped during periods other than the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).
- Disabling (stopping) the operation of the main timer clears the main timer. While the main timer is cleared, "1" is read from the MTC bit. As long as the MTC bit is "0", the MTIF bit may become "1".
- Do not change this bit from "1" to "0" during oscillation stabilization wait time of the PLL clock (PLLCLK).
- Do not write "1" to this bit when the MTC bit is "1".

[bit3 to bit0]: MTS3 to MTS0 (main timer period select bits)

These bits are used to select an overflow period of the main timer.

The main timer overflows when it finishes counting the period specified with these bits.

MTS3	MTS2	MTS1	MTS0	Overflow Period	4 MHz	8 MHz	48 MHz
1	0	0	0	2^9 × Main clock cycle	128.0 µs	64.0 µs	About 10.7 µs
1	0	0	1	2^{10} × Main clock cycle	256.0 µs	128.0 µs	About 21.3 µs
1	0	1	0	2^{11} × Main clock cycle	512.0 µs	256.0 µs	About 42.7 µs
1	0	1	1	2^{12} × Main clock cycle	About 1 ms	512.0 µs	About 85.3 µs
1	1	0	0	2^{13} × Main clock cycle	About 2 ms	About 1 ms	About 170.7 µs
1	1	0	1	2^{14} × Main clock cycle	About 4 ms	About 2 ms	About 341.3 µs
1	1	1	0	2^{15} × Main clock cycle	About 8 ms	About 4 ms	About 682.7 µs
1	1	1	1	2^{16} × Main clock cycle	About 16.4 ms	About 8 ms	About 1.4 ms

Always write "1" to the MTS3 bit.

Notes:

- Change the values of these bits after stopping the main timer using the MTE bit (MTE = 0).
- While the MTIE bit is set to "1", a main timer interrupt request is generated when the main timer overflows. Set these bits so that the main timer overflow period exceeds 5T (T: peripheral clock (PCLK) period).

6.4 Interrupts

A main timer interrupt request is generated when the main timer overflows.

Table 6-3 outlines the interrupts that can be used with the main timer.

Table 6-3. Interrupts of the Main Timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Main timer interrupt request	MTIF=1 for MTMCR	MTIE=1 for MTMCR	Write "0" to the MTIF bit for MTMCR

MTMCR: Main Timer Control Register (MTMCR)

Notes:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For information on the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For the setting of interrupt levels, see "10. Interrupt Controller".

6.5 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the main timer. Also, examples of procedures for setting the operating state are shown.

6.5.1 Main Timer Operation

Overview

The main timer counts the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

When main clock (MCLK) oscillation is stable, the main timer can also be used as an interval timer for generating an interrupt request at regular intervals.

If main timer operation is disabled with the MTE bit (MTE = 0) of the main timer control register (MTMCR), the timer is stopped during periods other than the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

Operation

The main timer operates as follows:

1. Enable the main timer operation by the MTE bit of the main timer control register (MTMCR) (MTE = 1).
2. The main timer starts counting in synchronization with the main clock (MCLK).

   The main timer continues counting while the MTE bit of the main timer control register (MTMCR) is "1".
3. The main timer counts up to the value set in the MTS3 to MTS0 bits of the main timer control register (MTMCR).
   a. The MTIF bit of the main timer control register (MTMCR) changes to "1".
   b. If the MTIE bit of the main timer control register (MTMCR) is "1" at this time, a main timer interrupt request is generated.
   c. To clear the main timer interrupt request, write "0" to the MTIF bit. The MTIF bit is cleared to "0".

If main timer operation is disabled with the MTE bit (MTE=0) of the main timer control register (MTMCR) during main timer operation, the main timer stops counting and clears the counter value. For more information, see "Clearing the timer".

Clearing the timer

The main timer is cleared when:

- "1" is written to the MTC bit of the main timer control register (MTMCR).

  "1" is read from the MTC bit of the main timer control register (MTMCR) until the main timer is cleared after "1" is written to the MTC bit.
- Main clock (MCLK) oscillation is stopped.

  (The MCEN bit of the clock source select register (CSELR) is 0).
- In stop mode
- The main timer is stopped with the MTE bit (MTE = 0) of the main timer control register (MTMCR).

Note:

The main timer control register (MTMCR) can be rewritten only when the oscillation of the main clock (MCLK) is stable. Therefore, even if "1" is written to the MTC bit of the main timer control register (MTMCR) when the following conditions are satisfied, the main timer cannot be cleared:

- Main clock (MCLK) oscillation is oscillating (the MCEN bit of the clock source select register (CSELR) is 1).
- The main clock (MCLK) is in oscillation stopped/oscillation stabilization wait state (The MCRDY bit of the clock source monitor register (CMONR) is 0).

Interrupt setting procedure

An example of the procedure for setting the main timer control register (MTMCR) is shown below.

1. Set the MTIE bit to disable main timer interrupts (MTIE=0).
2. Set the MTIF bit to clear the main timer interrupt flag (MTIF=0).
3. Set the MTE bit to disable main timer operation (MTE=0).
4. Read the MTC bit to verify that the main timer has been cleared (MTC=0).
5. Set the timer period in the MTS3 to MTS0 bits.
6. Set the MTIE bit to enable main timer interrupts (MTIE=1).
7. Set the MTE bit to enable main timer operation (MTE=1).

   When the period that is set in the MTS3 to MTS0 bits elapses, a main timer interrupt request is generated and processing moves to the interrupt processing routine.
8. Set the MTIF bit to clear the main timer interrupt flag (MTIF=0).
9. Read the MTIF bit once to complete clearing the main timer interrupt flag.

   Issue the RETI instruction to return to normal program processing from the interrupt processing routine.

Note: When "0" is written to the MTIF bit, the main timer interrupt flag is not cleared soon. After reading the MTIF bit once to complete clearing the flag, it can be returned by the RETI instruction.

6.5.2 Transition to Stop Mode

Before transition to the stop mode, generation of main timer interrupt requests must be disabled.

Follow the procedure below for transition to the stop mode:

1. Set the PCEN bit of the clock source select register (CSELR) to stop PLL clock (PLLCLK) oscillation (PCEN=0).
2. Set the MTIE bit of the main timer control register (MTMCR) to disable generation of main timer interrupt requests (MTIE=0).
3. Set the MTE bit of the main timer control register (MTMCR) to disable main timer operation (MTE = 0).
4. Read the MTC bit of the main timer control register (MTMCR) to verify that the main timer is not being cleared (MTC=0).
5. Set the MTIF bit of the main timer control register (MTMCR) to clear the main timer interrupt flag (MTIF=0).
6. Set the oscillation stabilization wait time of the main clock (MCLK) in the MOSW3 to MOSW0 bits of the clock stabilization time select register (CSTBR).
7. Transition to stop mode

Note: Before transition to stop mode, be sure to stop PLL clock (PLLCLK) oscillation.

7. Sub Timer

This chapter explains the functions and operations of the sub timer.

7.1 Overview

7.2 Configuration

7.3 Registers

7.4 Interrupts

7.5 An Explanation of Operations and Setting Procedure Examples

7.1 Overview

The sub timer operates based on the sub clock (SBCLK).

It is used to generate the sub clock (SBCLK) oscillation stabilization wait time.

The sub timer counts the oscillation stabilization wait time of the sub clock (SBCLK).

When sub clock (SBCLK) oscillation is stable, the sub timer can also be used as an interval timer for generating an interrupt request at regular intervals.

The sub timer is cleared when:

- "1" is written to the STC bit of the sub timer control register (STMCR).

  "1" is read from the STC bit of the sub timer control register (STMCR) until the sub timer is cleared after "1" is written to the STC bit.
- Sub clock (SBCLK) oscillation is stopped.

  (The SCEN bit of the clock source select register (CSELR) is 0.)
- In stop mode
- The sub timer is stopped with the STE bit (STE=0) of the sub timer control register (STMCR).

If sub timer operation is disabled, the timer is stopped during periods other than the oscillation stabilization wait time of the sub clock (SBCLK).

7.2 Configuration

This section explains the sub timer configuration.

Sub timer block diagram

For details of the sub timer block diagram, see "Sub clock (SBCLK) generating part" in "4. Clock Generating Parts".

Clocks

Table 7-1 shows the clocks used by the sub timer.

Table 7-1. Clocks Used by the Sub Timer

Clock Name	Description
Operation clock	Sub clock (SBCLK)

7.3 Registers

This section explains the configuration and functions of registers used by the sub timer.

Registers of sub timer

The registers used by the sub timer are listed in .

Table 7-2. Sub Timer Registers

Abbreviated Register Name	Register Name	Reference
STMCR	Sub timer control register	7.3.1

7.3.1 Sub Timer Control Register (STMCR)

This register controls the sub timer.

Figure 7-1 shows the bit configuration of the sub timer control register (STMCR).

Figure 7-1. Bit Configuration of Sub Timer Control Register (STMCR)

bit	7	6	5	4	3	2	1	0
	STIF	STIE	STC	STE	Reserved	STS2	STS1	STS0
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

R/W: Read/Write

Notes:

- This register can be rewritten only when the sub clock (SBCLK) is oscillating stably. (The SCRDY bit of the clock source monitor register (CMONR) is 1.)

  Note that the STIE bit can be rewritten even when the SCRDY bit is "0".
- Software reset must be executed when both the STE and STC bits are "0". For details of the software reset, see "9. Reset".

[bit7]: STIF (sub clock timer interrupt flag bit)

This flag indicates that the sub timer caused an overflow.

The sub timer overflows when:

- The counter has finished counting the period that is set with the STS2 to STS0 bits.
- The oscillation stabilization wait time of the sub clock (SBCLK) has elapsed after the SCEN bit of the clock source select register (CSELR) was rewritten from "0" to "1".
- The oscillation stabilization wait time of the sub clock (SBCLK) has elapsed after the system returns from stop mode.

A sub timer interrupt request occurs when this bit is set to "1" while the STIE bit is "1".

STIF	In case of reading	In case of writing
0	No overflow occurred.	This bit is cleared to "0".
1	An overflow occurred.	Ignored

This bit is also cleared to "0" when a DMA transfer is caused by a sub timer interrupt request.

Notes:

- Disabling sub timer operation with the STE bit (STE = 0) clears the sub timer.
- When the STIE bit is set to "0", this bit is not cleared even when a DMA transfer is caused by a sub timer interrupt request.
- If clearing the bit to "0" coincides with the occurrence of an overflow, the overflow occurrence is given priority and this bit remains "1".
- When a read-modify-write instruction is used, "1" is read

[bit6]: STIE (sub timer interrupt enable bit)

The STIE bit is used to specify whether to cause a sub timer interrupt request when the sub timer overflows (STIF=1).

A sub timer interrupt request occurs when the STIF bit is set to "1" while this bit is "1".

Written Value	Explanation
0	Disables generation of sub timer interrupt requests.
1	Enables generation of sub timer interrupt requests.

[bit5]: STC (sub timer clear bit)

This bit clears the sub timer.

The operating state of the sub timer can be verified by reading this bit.

STC	In case of writing	In case of reading
0	Ignored	In normal operation
1	Clear the sub timer.	The sub timer is being cleared.

Notes:

- When a read-modify-write instruction is used, "0" is read.
- This register can be rewritten only while sub clock (SBCLK) oscillation is stable. Therefore, if the following conditions are satisfied, the sub timer cannot be cleared even when the bit is set to "1":
  - Sub clock (SBCLK) is oscillating (the SCEN bit of the clock source select register (CSELR) is 1).
  - The sub clock (SBCLK) is in oscillation stopped/oscillation stabilization wait state. (The SCRDY bit of the clock source monitor register (CMONR) is 0.)
- Writing "1" to this bit at the same time that the STE bit is changed from "0" to "1" clears the sub timer and then starts sub timer operation.
- Do not attempt to write "1" to this bit when it is "1".
- As long as the STC bit is "0", the STIF bit may become "1".

[bit4]: STE (sub timer operation enable bit)

This bit controls the sub timer operation.

Written Value	Explanation
0	Disables (stops) the operation of the sub timer.
1	Enables the operation of the sub timer.

Notes:

- If the operation of the sub timer is disabled (stopped), the sub timer is stopped during periods other than the oscillation stabilization wait time of the sub clock (SBCLK).
- Disabling (stopping) the operation of the sub timer clears the sub timer. While the sub timer is cleared, "1" is read from the STC bit. As long as the STC bit is "0", the STIF bit may become "1".
- Do not write "1" to this bit when the STC bit is "1".

[bit3]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2 to bit0]: STS2 to STS0 (sub timer period select bits)

These bits are used to select an overflow period of the sub timer.

The sub timer overflows when it finishes counting the period specified with these bits.

STS2	STS1	STS0	Overflow Period	At 32768Hz
0	0	0	2^8 × Sub clock cycle	About 7.8 ms
0	0	1	2^9 × Sub clock cycle	About 15.6 ms
0	1	0	2^{10} × Sub clock cycle	About 31.3 ms
0	1	1	2^{11} × Sub clock cycle	62.5 ms
1	0	0	2^{12} × Sub clock cycle	125.0 ms
1	0	1	2^{13} × Sub clock cycle	250.0 ms
1	1	0	2^{14} × Sub clock cycle	500.0 ms
1	1	1	2^{15} × Sub clock cycle	1 s

Notes:

- Change the values of these bits after stopping the sub timer using the STE bit (STE = 0).
- While the STIE bit is set to "1", a sub timer interrupt request is generated when the sub timer overflows.
  Set these bits so that the sub timer overflow period is 5T (T: peripheral clock (PCLK) period) or more than that.

7.4 Interrupts

A sub timer interrupt request is generated when the sub timer overflows.

Table 7-3 outlines the interrupts that can be used with the sub timer.

Table 7-3. Interrupts of the Sub Timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Sub timer interrupt request	STIF=1 for STMCR	STIE=1 for STMCR	Write "0" to the STIF bit for STMCR

STMCR: sub timer control register (STMCR)

Notes:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the interrupt level settings, see "10. Interrupt Controller".

7.5 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the sub timer. Also, examples of procedures for setting the operating state are shown.

7.5.1 Sub timer operation

Overview

The sub timer counts the oscillation stabilization wait time of the sub clock (SBCLK).

When sub clock (SBCLK) oscillation is stable, the sub timer can also be used as an interval timer for generating an interrupt request at regular intervals.

If sub timer operation is disabled with the STE bit (STE = 0) of the sub timer control register (STMCR), the timer is stopped during periods other than the oscillation stabilization wait time of the sub clock (SBCLK).

Operation

The sub timer operates as follows:

1. The STE bit of the sub timer control register (STMCR) enables (STE = 1) sub timer operation.
2. The sub timer starts counting in synchronization with the sub clock (SBCLK).

   The sub timer continues counting while the STE bit of the sub timer control register (STMCR) is "1".
3. The sub timer counts up to the value specified in the STS2 to STS0 bits of the sub timer control register (STMCR).
   a. The STIF bit of the sub timer control register (STMCR) changes to "1".
   b. If the STIE bit of the sub timer control register (STMCR) is "1" at this time, a sub timer interrupt request is generated.
   c. To clear the sub timer interrupt request, write "0" to the STIF bit. The STIF bit is cleared to "0".

If sub timer operation is disabled with the STE bit (STE = 0) of the sub timer control register (STMCR) during sub timer operation, the sub timer stops counting and clears the counter value. For more information, see "Clearing the timer".

Clearing the timer

The sub timer is cleared when:

- "1" is written to the STC bit of the sub timer control register (STMCR).

  "1" is read from the STC bit of the sub timer control register (STMCR) until the sub timer is cleared after "1" is written to the STC bit.
- Sub clock (SBCLK) oscillation is stopped.

  (The SCEN bit of the clock source select register (CSELR) is 0.)
- In stop mode
- The sub timer is stopped with the STE bit (STE = 0) of the sub timer control register (STMCR).

  The sub timer is stopped for periods other than the oscillation stabilization wait time of the sub clock (SBCLK).

Note:

The sub timer control register (STMCR) can be rewritten only while the oscillation of the sub clock (SBCLK) is stable. Therefore, even if "1" is written to the STC bit of the sub timer control register (STMCR) when the following conditions are satisfied, the sub timer cannot be cleared:

- Sub clock (SBCLK) is oscillating. (The SCEN bit of the clock source select register (CSELR) is 1.)
- The sub clock (SBCLK) is in oscillation stopped/oscillation stabilization wait state. (The SCRDY bit of the clock source monitor register (CMONR) is 0.)

Interrupt setting procedure

An example of the procedure for setting the sub timer control register (STMCR) is shown below.

1. Set the STIE bit to disable sub timer interrupts (STIE = 0).
2. Set the STIF bit to clear the sub timer interrupt flag (STIF = 0).
3. Set the STE bit to disable sub timer operation (STE = 0).
4. Read the STC bit to verify that the sub timer is operating normally (STC=0).
5. Set the timer period in the STS2 to STS0 bits.
6. Set the STIE bit to enable sub timer interrupts (STIE = 1).
7. Set the STE bit to enable sub timer operation (STE = 1).

   When the period that is set in the STS2 to STS0 bits elapses, a sub timer interrupt request is generated and processing moves to the interrupt processing routine.
8. Set the STIF bit to clear the sub timer interrupt flag (STIF = 0).
9. Read the STIF bit once to complete clearing the sub timer interrupt flag.

   Issue the RETI instruction to return to normal program processing from the interrupt processing routine.

Note: When "0" is written to the STIF bit, the sub timer interrupt flag is not cleared soon. After reading the STIF bit once to complete clearing the flag, it can be returned by the RETI instruction.

7.5.2 Transition to Stop Mode, and Watch Mode

Before transition to stop mode, interrupt operation by the sub timer must be disabled.

Follow the procedure below for transition to the stop mode:

1. Set the PCEN bit of the clock source select register (CSELR) to stop PLL clock (PLLCLK) oscillation (PCEN=0).
2. Set the STIE bit of the sub timer control register (STMCR) to disable sub timer interrupts (STIE = 0).
3. Set the STE bit of the sub timer control register (STMCR) to disable sub timer operation (STE = 0).
4. Read the STC bit of the sub timer control register (STMCR) to confirm that the sub timer is not being cleared (STC=0).
5. Set the STIF bit of the sub timer control register (STMCR) to clear the sub timer interrupt flag (STIF = 0).
6. Set the oscillation stabilization wait time of the sub clock (SBCLK) in the SOSW2 to SOSW0 bits of the clock stabilization time select register (CSTBR).
7. Transition to stop mode

Note: Before transition to the stop mode, be sure to stop PLL clock oscillation.

8. Low-power Dissipation Mode

This chapter explains the functions and operations of low-power dissipation mode.

8.1 Overview

8.2 Configuration

8.3 Registers

8.4 An Explanation of Operations and Setting Procedure Examples

8.5 Notes on Use

8.1 Overview

This series can use low-power dissipation mode to reduce power dissipation.

Overview

This series can control power dissipation in the following way.

- Clock control
  - Clock division

    By changing the division ratio of each operation clock, operation frequency can be reduced.
  - Stop clock

    This allows the user to specify a specific clock to stop the clock.
- Doze mode

  This mode intermittently operates the CPU repeatedly at a set operation rate.
- Sleep mode

  This mode operates only peripheral functions. One of the following two modes can be selected.
  - CPU sleep mode

    This mode stops the operation of the CPU.
  - Bus sleep mode

    This mode stops the CPU and on-chip bus.
- Standby mode

  One of the following three modes can be selected.
  - Main timer mode

    This mode stops all the operations other than the main clock oscillation.

    The sub clock oscillation can be specified arbitrarily.
  - Watch mode

    This mode stops all the operations other than the sub clock oscillation.
  - Stop mode

    This mode stops all operations including the oscillation of all clocks.

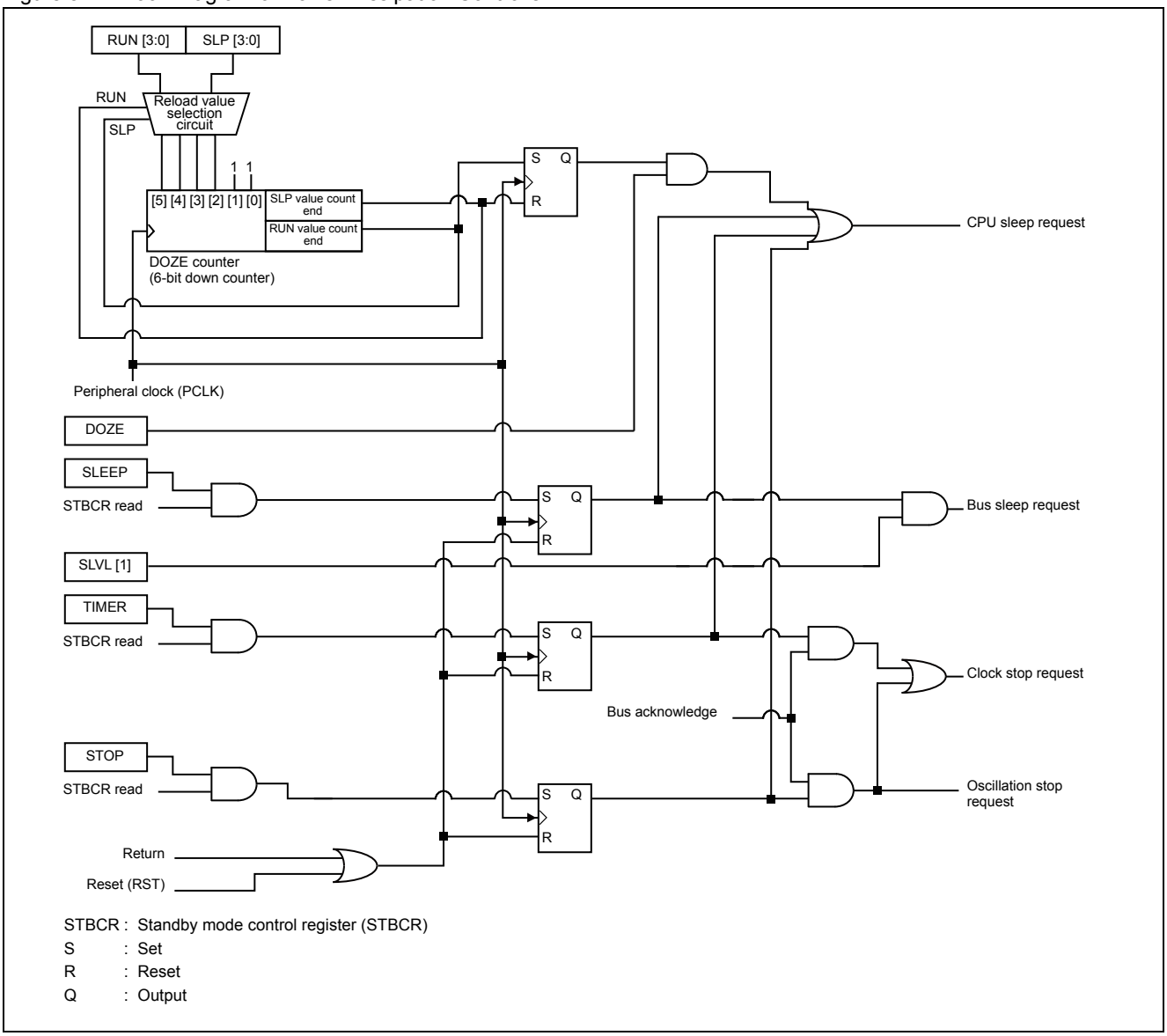
8.2 Configuration

The configuration of the power dissipation controller is shown below.

Block diagram of power dissipation controller

Figure 8-1 is a block diagram of the power dissipation controller.

Figure 8-1. Block Diagram of Power Dissipation Controller



- Standby mode control register (STBCR)
  This register controls low-power dissipation mode.
- Sleep rate configuration register (SLPRR)
  This register configures the operation state (RUN state) rate and sleep state rate (sleep rate) in doze mode.
- Reload value selection circuit
  A circuit for selecting to reload either the operation state (RUN state) rate or sleep state rate (Sleep rate) which has been set in the sleep rate configuration register (SLPRR).

Clocks

Table 8-1 shows the clock used in the power dissipation controller.

Table 8-1. Clock Used in Power Dissipation Controller

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-

8.3 Registers

This section explains the configurations and functions of the registers that are required for controlling power dissipation.

List of registers that control power dissipation

Table 8-2 is a list of registers that control power dissipation.

Table 8-2. List of Registers that Control Power Dissipation

Abbreviated Register Name	Register Name	Reference
STBCR	Standby mode control register	8.3.1
SLPRR	Sleep rate configuration register	8.3.2

8.3.1 Standby Mode Control Register (STBCR)

This register controls low-power dissipation mode.

Figure 8-2 shows the bit configuration of the standby mode control register (STBCR).

Figure 8-2. Bit Configuration of the Standby Mode Control Register (STBCR)

bit	7	6	5	4	3	2	1	0
	STOP	TIMER	SLEEP	DOZE	Reserved	Reserved	SLVL1	SLVL0
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	1	1

R/W: Read/Write

[bit7]: STOP (Stop mode enable bit)

This bit enables transition to stop mode.

Written Value	Explanation
0	Does not transit to stop mode.
1	Transits to stop mode.

If this register is read after this bit enables transition to stop mode, power dissipation mode moves to stop mode.

If the return resource from stop mode occurs, this bit is cleared to "0". For information on return resource from stop mode, see "Return from stop mode" in "8.4.6 Operation in Stop Mode".

[bit6]: TIMER (Main timer mode/watch mode enable bit)

This bit enables transition to main timer mode/watch mode.

Written Value	Explanation
0	Does not transit to main timer mode/watch mode.
1	Transits to main timer mode/watch mode.

If this register is read after this bit enables transition to main timer mode/watch mode, power dissipation mode moves to main timer mode/watch mode.

If, however, transition to stop mode is enabled with the STOP bit (STOP = 1), the setting of this bit is ignored even when transition to main timer mode/watch mode is enabled by writing "1" to this bit.

If the return resource from main timer mode/watch mode occurs, this bit is cleared to "0". For information on return resource from main timer mode, see "Return from the main timer mode" in "8.4.4 Operation in Main Timer Mode". For information on return resource from watch mode, see "Return from the watch mode" in "8.4.5 Operation in Watch Mode".

[bit5]: SLEEP (Sleep mode enable bit)

This bit enables transition to sleep mode.

Written Value	Explanation
0	Does not transit to sleep mode.
1	Transits to sleep mode.

If this register is read after this bit enables transition to sleep mode, power dissipation mode moves to sleep mode.

If, however, transition to stop mode/main timer mode/watch mode is enabled with the STOP bit/TIMER bit (STOP/TIMER = 1), the setting of this bit is ignored even when transition to sleep mode is enabled by writing "1" to this bit.

If the return resource from sleep mode occurs, this bit is cleared to "0". For information on return resource from sleep mode, see "Return from sleep mode" in "8.4.3 Operation in Sleep Mode".

[bit4]: DOZE (Doze mode enable bit)

This bit enables transition to doze mode.

Written Value	Explanation
0	Does not transit to doze mode (CPU intermittent sleep).
1	The CPU transits to doze mode (CPU intermittent sleep).

While the SLVL1 bit is set to "0", if the return resource from doze mode occurs, this bit is cleared to "0". For information on return resource from doze mode, see "Return from doze mode" in "8.4.2 Operation in Doze Mode".

[bit3, bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: SLVL1, SLVL0 (Standby level setting bits)

The meaning of the value to be written to this bit varies depending on the low-power dissipation mode to move to.

Low-power Dissipation Mode	SLVL1	SLVL0	Explanation
Stop mode/ Main timer mode/watch mode	0	0	Does not place the output from each pin in Hi-Z in stop mode/main timer mode/watch mode.
	0	1	
	1	0	Places the output from each pin in Hi-Z in stop mode/main timer mode/ watch mode.
	1	1	
Sleep mode	0	0	When moving to sleep mode, power dissipation mode moves to CPU sleep mode (stops only the operation of the CPU).
	0	1	
	1	0	When moving to sleep mode, power dissipation mode moves to bus sleep mode (stops operations of the CPU and on-chip bus).[1]
	1	1	
Doze mode	0	0	When interrupt request occur, the DOZE bit is cleared to "0".
	0	1	
	1	0	When interrupt request occur, the DOZE bit is not cleared to "0".
	1	1	

[1]: During DMA transfer, the on-chip bus operates.

Notes:

- For information on pins of which the output can be placed in Hi-Z in stop mode/main timer mode/watch mode, see "A.4 Pin State in Each CPU State".
- The setting value of SLVL0 bit has no effect on the operation.

8.3.2 Sleep Rate Configuration Register (SLPRR)

This register configures the operation state (RUN state) rate and sleep state rate (sleep rate) in doze mode.

Figure 8-3 shows the bit configuration of the sleep rate configuration register (SLPRR).

Figure 8-3. Bit Configuration of the Sleep Rate Configuration Register (SLPRR)

bit	7	6	5	4	3	2	1	0
	RUN3	RUN2	RUN1	RUN0	SLP3	SLP2	SLP1	SLP0
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

R/W: Read/Write

Note: If this register is rewritten in doze mode, the rewritten setting is reflected at the next stop/activation timing.

[bit7 to bit4]: RUN3 to RUN0 (Operation period bits)

These bits set the period during which the CPU operates in doze mode.

The CPU operation period is calculated from the value that is set to these bits as follows.

(Value of this bit + 1) × **4** × $\mathbf{t_{CYCP}}$

$\mathbf{t_{CYCP}}$: Period of the peripheral clock (PCLK)

For details of operation period, see "8.4.2 Operation in Doze Mode".

[bit3 to bit0]: SLP3 to SLP0 (Sleep state period bits)

These bits set the period of sleep state in doze mode.

The sleep state period is calculated from the value that is set to these bits as follows.

(Value of this bit + 1) × **4** × $\mathbf{t_{CYCP}}$

$\mathbf{t_{CYCP}}$: Period of the peripheral clock (PCLK)

For details of the sleep state period, see "8.4.2 Operation in Doze Mode".

Notes:

- A delay may occur when the CPU accepts the sleep request. In this case, the sleep period will be shorter than that obtained from the above calculation formula.
- If the sleep state period is short, the CPU may not enter the sleep state depending on the operating status of the CPU.

8.4 An Explanation of Operations and Setting Procedure Examples

This section explains the operation and use of low-power dissipation mode and includes examples of the procedure for setting this mode.

Overview

You can reduce power dissipation by changing the division ratio of the operation clock or stopping the operation clock.

You can also use the following low-power dissipation modes:

- Doze mode

  This mode intermittently operates the CPU repeatedly at a set operation rate.

  By repeating operation and stop of the CPU alternately in the set period, the average power dissipation of the CPU can be reduced.

- Sleep mode

  In this mode, only the peripheral functions operate while the CPU and on-chip bus are stopped.

  One of the following two modes can be selected.

  - CPU sleep mode

    This mode stops the operation of the CPU.

  - Bus sleep mode

    This mode stops the CPU and on-chip bus.

- Standby mode

  This mode stops the entire device to put it in a standby state.

  One of the following three modes can be selected.

  - Main timer mode
  - Watch mode
  - Stop mode

8.4.1 Operation When Clock Control Is Set

Power dissipation and CPU performance can be optimized by adjusting the operation clocks that are built in this series.

Overview

To reduce power dissipation by controlling the clock, the following two methods are available.

- Clock division

  By changing the division ratio of each operation clock, the operation frequency can be reduced.
- Stop clock

  This allows the user to specify a specific clock to stop.

Clock division

By changing the division ratio of each operation clock, power dissipation can be reduced. The division ratio of the operation clock can be individually set.

Table 8-3 shows each operation clock and settable division ratio.

Table 8-3. Operation Clock and Settable Division Ratio

Operation Clock	Division Ratio
Base clock (BCLK)	Source clock (SRCCLK) divided by 1 to 8.
Peripheral clock (PCLK)	Base clock (BCLK) divided by 1 to 16.
IP bus clock (IPCLK)	Base clock (BCLK) divided by 1 to 8.

Note: The division method or condition differs depending on the operation clock. For information on the division of the operation clock, see "5. Clock Division Control Part".

8.4.2 Operation in Doze Mode

This mode intermittently operates the CPU in order to reduce the average power dissipation by the CPU.

Overview

Using doze mode enables reducing the average power dissipation by the CPU by operating and stopping the CPU alternately at a set interval. Maintain performance while reducing power dissipation by changing the sleep rate according to the processing load.

Setting the period

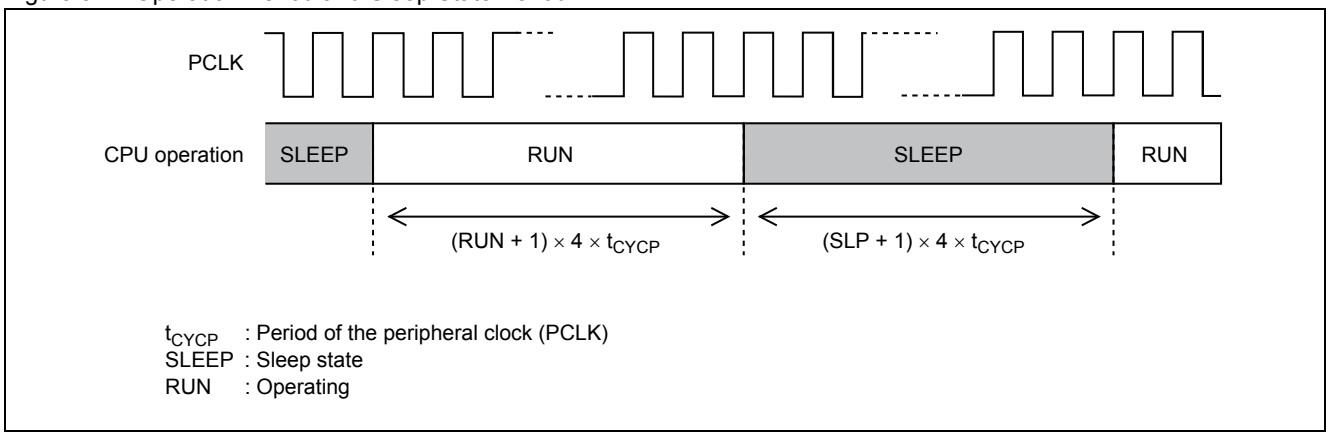
If you set the CPU operation period in the RUN3 to RUN0 bits and sleep state period in the SLP3 to SLP0 bits of the sleep rate configuration register (SLPRR), the period will be calculated from the set value using the following calculation formula.

$$\mathbf{(RUN + 1) \times 4 \times t_{CYCP} + (SLP + 1) \times 4 \times t_{CYCP}}$$

RUN: Value for the RUN3 to RUN0 bits
SLP: Value for the SLP3 to SLP0 bits
$\mathbf{t_{CYCP}}$: Period of the peripheral clock (PCLK)

Figure 8-4 shows each cycle.

Figure 8-4. Operation Period and Sleep State Period



Notes:

- The above calculation formula does not contain delay time for the CPU to accept the sleep request. Therefore an error may occur.
- If the setting of the sleep state period is short, the CPU may not enter the sleep state depending on the operating status of the CPU.

Transition

If "1" is written to the DOZE bit in the standby mode control register (STBCR) after the cycle is set, doze mode is entered and the CPU starts intermittent operation by alternately running and stopping according to the setting configured in the sleep rate configuration register (SLPRR).

To return from doze mode, write "0" in the DOZE bit of standby mode control register (STBCR).

Note: If the sleep rate configuration register (SLPRR) is rewritten in doze mode, the rewritten setting is reflected at the next stop/operation transition timing.

Return from doze mode

The CPU returns from doze mode in either of the following cases.

- This device is reset.
- "0" is written to the DOZE bit of standby mode control register (STBCR).
- An interrupt request is generated when the SLVL1 bit of standby mode control register (STBCR) is "0".

Except the above cases, the configuration is retained so that you can use doze mode even after returning from sleep mode, main timer mode, watch mode, or stop mode.

8.4.3 Operation in Sleep Mode

This mode is used to reduce power dissipation in the event wait state.

If sleep mode is entered, it continues until a return resource occurs. When a return resource occurs, it returns to the program operation after two or three clock period.

Overview

Using sleep mode can significantly reduce power dissipation in the event wait state by stopping the CPU and on-chip bus while allowing only the peripheral functions to operate.

The following two modes are available for sleep mode.

- CPU sleep mode

  This mode stops only the operation of the CPU.

  Because the clock continues to be delivered to the DMA controller (DMAC) or to the on-chip bus, operations of these devices continue.

  Though the power dissipation is larger than that in bus sleep mode, quick response can be given to the DMA transfer request.

- Bus sleep mode

  This mode stops the operation of the CPU and on-chip bus.

  It also disables the clock delivery to the DMAC controller (DMAC) or on-chip bus. For information on disabling clock, see "5. Clock Division Control Part".

  However, if the DMA transfer request is accepted, the clock delivery to the DMA controller (DMAC) or on-chip bus will be tentatively resumed to allow DMA transfer.

  After the DMA transfer is completed, the clock delivery will be disabled again.

  While this mode is slower in responding to the DMA transfer request than in CPU sleep mode, it can reduce power dissipation.

Setting

Table 8-4 shows the settings required before changing to sleep mode.

Table 8-4. Setting Register

Registers	Bit	Explanation
Standby mode control register (STBCR)	SLVL1	Sets whether to change to CPU sleep mode or to bus sleep mode 0 = Change to CPU sleep mode 1 = Change to bus sleep mode

Transition

By following the steps below, power dissipation mode moves to sleep mode.

1. Write "0" to the STOP bit, write "0" to the TIMER bit, and write "1" to the SLEEP bit of standby mode control register (STBCR).
2. Read standby mode control register (STBCR).

Note: To prevent the CPU from executing the next instruction before moving to sleep mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_sleep, R0     ; SLEEP bit=1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to sleep mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

Return from sleep mode

The CPU returns from sleep mode in either of the following cases.

- This device is reset.
- An interrupt request is generated (whose interrupt level is other than "31").

  For information on the interrupt level, see "10. Interrupt Controller".

Notes:

- If the interrupt request is not accepted by the CPU when returning from sleep mode due to the interrupt request, the program is executed starting from the next instruction after entering sleep mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.
- In bus sleep mode, if a DMA transfer request is generated, the on-chip bus clock (HCLK) delivery is tentatively resumed to perform DMA transfer. The on-chip bus clock (HCLK) delivery is again disabled after DMA transfer is completed.

8.4.4 Operation in Main Timer Mode

Main timer mode is categorized as a standby mode. Standby mode stops the entire device to put it in a standby state. By doing so, it can significantly reduce power dissipation in the external event wait state. The permitted clock oscillation, however, operates, allowing less reduction in power dissipation than in stop mode.

In main timer mode, select the main clock (MCLK) oscillation as a source clock (SRCCLK) for the CPU.

If main timer mode is entered, it continues until a return resource occurs. When a return resource occurs, it returns to the program operation after two or three clock period.

Overview

In main timer mode, because main clock (MCLK) oscillation is permitted as a source clock (SRCCLK) for the CPU, the count operation of the main timer is executed.

The sub clock (SBCLK) oscillation can be specified arbitrarily.

Setting

Table 8-5 shows the settings required before changing to main timer mode.

Table 8-5. Setting Register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects main clock (MCLK) for the CPU source clock (SRCCLK) (CKS1, CKS0=00 or 01)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
	SCEN	Specify sub clock (SBCLK) oscillation. 0=Stop oscillation 1=Start oscillation
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in main timer mode 0 = Retain the state in effect before main timer mode is entered 1 = Hi-Z

Note: When moving to main timer mode, if the SLVL1 bit of the standby mode control register (STBCR) is set to "0" while setting doze mode, the DOZE bit is cleared to "0" on returning from main timer mode to end doze mode.

Transition

By following the steps below, power dissipation mode moves to main timer mode.

1. Write "0" to the STOP bit, write "1" to the TIMER bit, and write "0" to the SLEEP bit in the standby mode control register (STBCR).
2. Read the standby mode control register (STBCR).

Note: To prevent the CPU from executing the next instruction before moving to main timer mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_timer, R0     ; TIMER bit = 1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to main timer mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

Return from the main timer mode

The CPU returns from main timer mode in either of the following cases.

- This device is reset.
- Below interrupt requests are generated (whose interrupt level is other than "31").
  - Main timer interrupt
  - Sub timer interrupt
  - Watch counter interrupt
  - External interrupt
  - An interrupt by the WKUP bit of the USB function

For the interrupt level, see "10. Interrupt Controller".

Note: If the interrupt request is not accepted by the CPU when returning from main timer mode due to the interrupt request, the program is executed starting from the next instruction after entering main timer mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.

8.4.5 Operation in Watch Mode

Watch mode is categorized as a standby mode. Standby mode stops the entire device to put it in a standby state. By doing so, it can significantly reduce power dissipation in the external event wait state. The permitted clock oscillation, however, operates, allowing less reduction in power dissipation than in stop mode.

In watch mode, select the sub clock (SBCLK) oscillation as a source clock (SRCCLK) for the CPU.

If watch mode is entered, it continues until a return resource occurs. When a return resource occurs, it returns to the program operation after two or three clock period.

Overview

In watch mode, because sub clock (SBCLK) oscillation is permitted as a source clock (SRCCLK) for the CPU, the count operation of the sub timer and watch counter is executed.

Setting

Table 8-6 shows the settings required before changing to watch mode.

Table 8-6. Setting Register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects sub clock (SBCLK) for the CPU source clock (SRCCLK) (CKS1, CKS0=11)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
	MCEN	Stops main clock (MCLK) oscillation (MCEN = 0)
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in watch mode 0 = Retain the state in effect before watch mode is entered 1 = Hi-Z

Note: When moving to watch mode, if the SLVL1 bit of the standby mode control register (STBCR) is set to "0" while setting doze mode, the DOZE bit is cleared to "0" on returning from watch mode to end doze mode.

Transition

By following the steps below, power dissipation mode moves to watch mode.

1. Write "0" to the STOP bit, write "1" to the TIMER bit, and write "0" to the SLEEP bit in the standby mode control register (STBCR).
2. Read the standby mode control register (STBCR).

Note: To prevent the CPU from executing the next instruction before moving to watch mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_timer, R0     ; TIMER bit = 1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to watch mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

Return from the watch mode

The CPU returns from watch mode in either of the following cases.

- This device is reset.
- Below interrupt requests are generated (whose interrupt level is other than "31").
  - Sub timer interrupt request
  - Watch counter interrupt request
  - External interrupt request
  - An interrupt by the WKUP bit of the USB function.

For the interrupt level, see "10. Interrupt Controller".

Note: If the interrupt request is not accepted by the CPU when returning from watch mode due to the interrupt request, the program is executed starting from the next instruction after entering watch mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.

8.4.6 Operation in Stop Mode

Stop mode is categorized as a standby mode. Standby mode stops the entire device to put it in a standby state. By doing so, it can significantly reduce power dissipation in the external event wait state.

Stop mode stops all operations including the oscillation of all clocks to minimize power dissipation.

Overview

Using stop mode can minimize power dissipation by stopping the oscillation of all clocks.

To return to the program operation after the return request is generated, however, a certain amount of oscillation stabilization wait time is required.

Setting

The setting may differ depending on the source clock of the CPU (SRCCLK) before entering stop mode and after returning from stop mode.

- If the source clock (SRCCLK) of the CPU before/after stop mode is a sub clock (SBCLK)

  Table 8-7 shows the settings required before changing to stop mode.

Table 8-7. Setting Register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects sub clock (SBCLK) for the CPU source clock (SRCCLK) (CKS1, CKS0=11)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in stop mode 0 = Retain the state in effect before stop mode is entered 1 = Hi-Z

Note: At transition to stop mode, if the SLVL1 bit of standby mode control register (STBCR) is set to "0" while doze mode has been set, the DOZE bit is cleared to "0" when the CPU returns from stop mode to end doze mode.

- If the source clock (SRCCLK) of the CPU before/after stop mode is a main clock (MCLK)

  Table 8-8 shows the settings required before changing to stop mode.

Table 8-8. Setting Register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects the main clock (MCLK) as a source clock (SRCCLK) of the CPU (CKS1, CKS0=00/01)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in stop mode 0 = Retain the state in effect before stop mode is entered 1 = Hi-Z

Note: At transition to stop mode, if the SLVL1 bit of standby mode control register (STBCR) is set to "0" while doze mode has been set, the DOZE bit is cleared to "0" when the CPU returns from stop mode to end doze mode.

Transition

By following the steps below, power dissipation mode moves to stop mode.

1. Write "1" to the STOP bit write "0" to the TIMER bit, and write "0" to the SLEEP bit in the standby mode control register (STBCR).
2. Read the standby mode control register (STBCR).

Note: To prevent the CPU from executing the next instruction before moving to stop mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_stop, R0      ; STOP bit = 1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to stop mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

Return from stop mode

The CPU returns from stop mode in either of the following cases.

- This device is reset.
- Below interrupt requests are generated (whose interrupt level is other than "31").
  - External interrupt
  - An interrupt by the WKUP bit of the USB function

For information on the interrupt level, see "10. Interrupt Controller".

Note: If the interrupt request is not accepted by the CPU when returning from stop mode due to the interrupt request, the program is executed starting from the next instruction after entering stop mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.

8.5 Notes on Use

Note the following points on using low-power dissipation mode.

- If the interrupt request is generated when low-power dissipation mode is switched to the following modes, the switching is disabled.
  - Doze mode
  - Sleep mode
  - Main timer mode
  - Watch mode
  - Stop mode
- For instance, sleep mode is not entered in the following cases. Move to sleep mode after clearing the interrupt request.
  - In sleep mode, when returning from sleep mode due to an interrupt request that has not been accepted by the CPU, an operation to move to sleep mode is performed again without clearing the interrupt request.

9. Reset

This chapter explains the functions and operations of reset.

9.1 Overview

9.2 Configuration

9.3 Pins

9.4 Registers

9.5 Explanation of Operations

9.6 Operating State and Transition

9.1 Overview

This section explains "reset" to initialize the internal circuit.

Overview

This device has the following three types of reset resource.

- $\overline{\text{INIT}}$ pin input
- Watchdog reset 0
- Watchdog reset 1
- Software reset

If either one of the reset resources occurs, operation of all the programs and internal circuits is stopped for initialization.

This state is called a reset state.

If the reset resource is released, operation of the programs and the hardware starts.

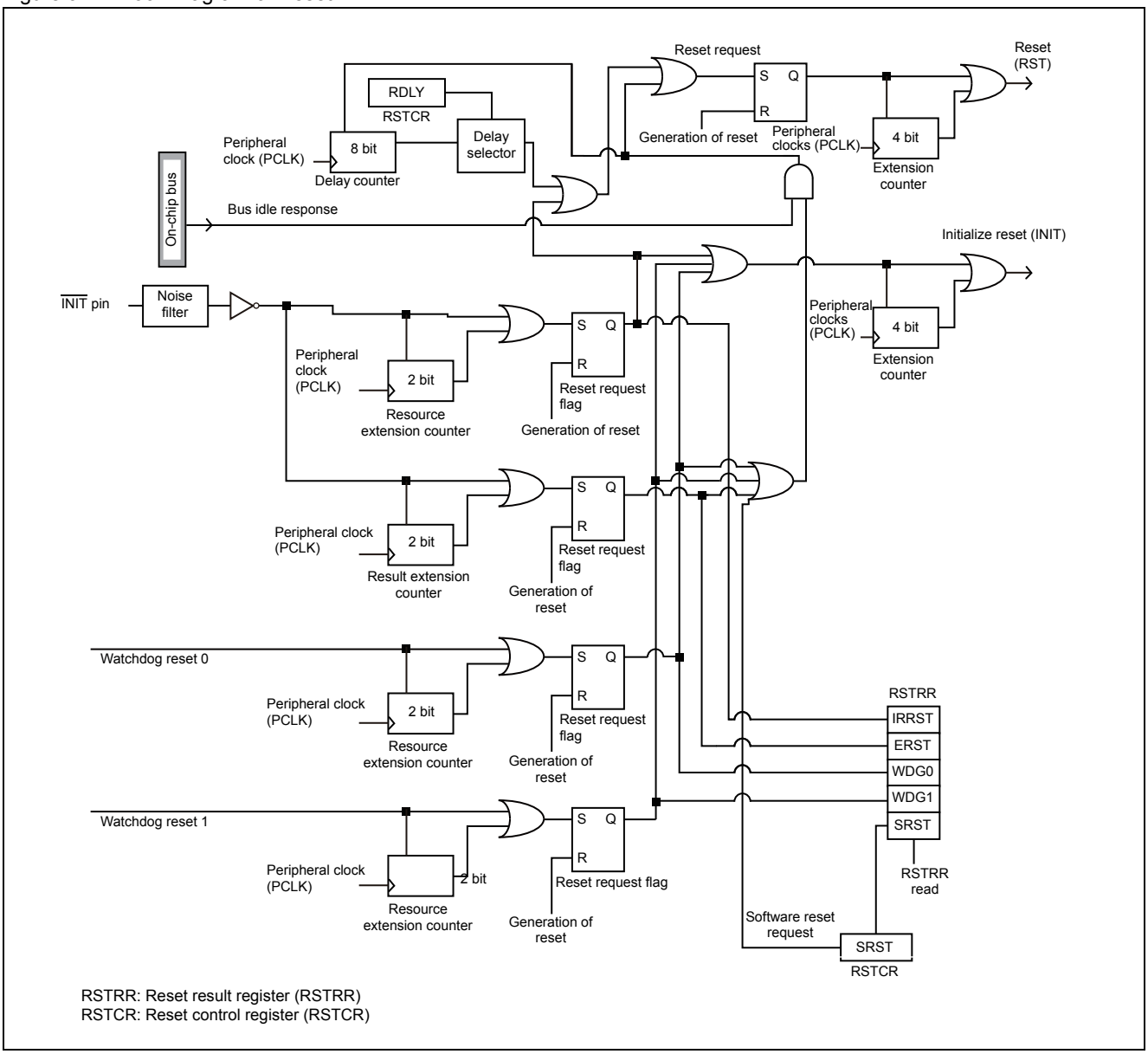
9.2 Configuration

The configuration of reset is shown.

Block diagram of reset

Figure 9-1 is a block diagram of reset.

Figure 9-1. Block Diagram of Reset



RSTRR: Reset result register (RSTRR)
RSTCR: Reset control register (RSTCR)

- Reset result register (RSTRR)
  This register indicates the reset resource.
- Reset control register (RSTCR)
  This register controls issuing of reset.
- Delay counter
  This counter counts the period from generation of the reset request until the bus enters the idle state.
  If the bus does not enter the idle state within a certain period of time, the initialize reset (INIT) is forcibly issued.
- Result extension counter
  This counter counts the amount of time for the reset resource to be extended. Each reset resource will be retained until reset is issued.

Clocks

Table 9-1 shows clocks to be used for reset.

Table 9-1. Clocks Used for Reset

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

9.3 Pins

This section explains the pins that are used for reset.

Overview

The following pins are used for reset.

- $\overline{\text{INIT}}$ pins

  The external input pins are used to input the reset request.

9.4 Registers

This section explains the configuration and functions of registers used for reset.

List of registers used for reset

Table 9-2 shows the list of registers used for reset.

Table 9-2. List of Registers Used for Reset

Abbreviated Register Name	Register Name	Reference
RSTRR	Reset result register	9.4.1
RSTCR	Reset control register	9.4.2

9.4.1 Reset Result Register (RSTRR)

This register stores the reset resource.

It stores all the reset resources that have occurred since the power was turned on until this register is read.

Figure 9-2 shows the bit configuration of the reset result register (RSTRR).

Figure 9-2. Bit Configuration of the Reset Result Register (RSTRR)

bit	7	6	5	4	3	2	1	0
	IRRST	ERST	WDG1	WDG0	Undefined	Undefined	Undefined	SRST
Attribute	R	R	R	R	R	R	R	R
Initial value	* This differs depending on the reset resource.							

R: Read only

*: The initial values are as follows:

Reset Resource	Initial Value
$\overline{\text{INIT}}$ pin input	11XXXXXX
Watchdog reset 0	XXX1XXXX
Timeout of the watchdog reset 0	1XX1XXXX
Watchdog reset 1	XX1XXXXX
Timeout of the watchdog reset 1	1X1XXXXX
Software reset	XXXXXXX1
Timeout for software reset	1XXXXXX1
Register reading	00000000

X: Each bit is initialized by a specific reset factor.
It is not initialized by other factors.

Please refer to "Flow of reset result determination" of "9.5.2 Reset Resource" for the determination of the Reset Result.

Notes:

- If this register is read, all the bits are cleared.
- The value when the power supply is turned on is undefined.

[bit7]: IRRST (Irregular reset bit)

A reset is issued without waiting for completion of bus access. This is called an irregular reset. If an irregular reset occurs, the contents of the memory may be damaged.

If either a reset by the $\overline{\text{INIT}}$ pin input or a reset timeout occurs, this bit changes to "1".

Read Value	Explanation
0	No irregular reset is detected. The memory contents are guaranteed to be damage free.
1	An irregular reset is detected. The contents of the memory may have been damaged during the last reset.

For details of the irregular reset, see "9.5.4 Irregular Reset".

[bit6]: ERST (Reset pin input bit)

This bits indicates whether the reset by an $\overline{\text{INIT}}$ pin input has occurred.

Read Value	Explanation
0	Reset by an $\overline{\text{INIT}}$ pin input has not occurred.
1	Reset by an $\overline{\text{INIT}}$ pin input has occurred.

[bit5]: WDG1 (Watchdog reset 1 bit)

This bit indicates whether the watchdog reset 1 has occurred.

If a reset timeout occurred in watchdog timer 1, the IRRST bit also changes to "1".

Read Value	Explanation
0	A watchdog reset 1 has not occurred.
1	A watchdog reset 1 has occurred.

[bit4]: WDG0 (Watchdog reset 0 bit)

This bit indicates whether the watchdog reset 0 has occurred.

If a reset timeout occurred in watchdog timer 0, the IRRST bit also changes to "1".

Read Value	Explanation
0	A watchdog reset 0 has not occurred.
1	A watchdog reset 0 has occurred.

[bit3 to bit1]: Undefined bits

In case of reading	A value is undefined.

[bit0]: SRST (Software reset bit)

This bit indicates whether a software reset (RSTCR:SRST) has occurred.

If a reset timeout occurred in the software reset (RSTCR:SRST), the IRRST bit also changes to "1".

Read Value	Explanation
0	A software reset (RSTCR:SRST) has not occurred.
1	A software reset (RSTCR:SRST) has occurred.

9.4.2 Reset Control Register (RSTCR)

This register controls issuing of reset.

Figure 9-3 shows the bit configuration of the reset control register (RSTCR).

Figure 9-3. Bit Configuration of the Reset Control Register (RSTCR)

bit	7	6	5	4	3	2	1	0
	RDLY2	RDLY1	RDLY0	Reserved	Reserved	Reserved	Reserved	SRST
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

R/W: Read/Write

[bit7 to bit5]: RDLY2 to RDLY0 (Reset issue delay bit)

These bits set the delay time for reset issuing, meaning the length of time that it takes for all the busses to become idle after acceptance of the reset request (delay cycle).

RDLY2	RDLY1	RDLY0	Explanation
0	0	0	Peripheral clock (PCLK) × 2 cycles
0	0	1	Peripheral clock (PCLK) × 4 cycles
0	1	0	Peripheral clock (PCLK) × 8 cycles
0	1	1	Peripheral clock (PCLK) × 16 cycles
1	0	0	Peripheral clock (PCLK) × 32 cycles
1	0	1	Peripheral clock (PCLK) × 64 cycles
1	1	0	Peripheral clock (PCLK) × 128 cycles
1	1	1	Peripheral clock (PCLK) × 256 cycles

Notes:

- The values of each bit are initialized by reset. Writing after reset is possible only once.
- If a low value is set for the delay cycle, a irregular reset due to the reset timeout will likely occur. In contrast, if a high value is set for the delay cycle, it may take long for the reset to be issued after the reset resource occurs.
- For information on the irregular reset, see "9.5.4 Irregular Reset".

[bit4 to bit1]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit0]: SRST (Software reset bit)

A software reset request occurs if the reset control register (RSTCR) is read after "1" is written to this bit.

Written Value	Explanation
0	A reset request has not occurred.
1	A reset request has occurred by reading this register.

Notes:

- After "1" is written to this bit, any subsequent writing in the reset control register (RSTCR) is ignored until reset occurs.
- Before generating a software reset request by writing "1" to SRST bit, switch the source clock to the main clock (MCLK) divided by 2.

9.5 Explanation of Operations

This section explains the operation of reset.

9.5.1 Reset Types

Three types of resets are provided for this device, whose reset resources and contents for initialization differ from one another.

- Power-on reset (SINIT)

  This reset is used to initialize the unstable state of the division circuit.

  At the same time, initialize reset (INIT) and reset (RST) are issued.

Reset resource	Input "L" level to $\overline{\text{INIT}}$ pin
Target of initialization	Oscillation stabilization wait time of the main clock (MCLK)
Reset that concurrently occurs	■ Initialize reset (INIT) ■ Reset (RST)

- Initialize reset (INIT)

  Initializes the following registers to reset the clock control settings.

  - ❒ Clock source select register (CSELR)
  - ❒ Clock source monitor register (CMONR)
  - ❒ PLL configuration register (PLLCR)
  - ❒ Clock stabilization time select register (CSTBR)

  Reset (RST) is issued at the same time.

Reset resource	■ $\overline{\text{INIT}}$ pin input ■ Reset time out ■ Watchdog reset 0 ■ Watchdog reset 1
Target of initialization	■ Source clock = Main clock (MCLK) divided by 2 ■ Clock oscillation = Main clock oscillates, sub/PLL clock stopped ■ Division rate of the PLL macro oscillation clock ■ Multiplying factor of the PLL clock (PLLCLK) ■ Oscillation stabilization wait time of the PLL clock ■ Division rate of the PLL input clock ■ Oscillation stabilization wait time of the sub clock
Reset that concurrently occurs	Reset (RST)

- Reset (RST)

  This reset initializes the program operation.

Reset resource	■ $\overline{\text{INIT}}$ pin input ■ Reset time out ■ Watchdog reset 0 ■ Watchdog reset 1 ■ Software reset
Target of initialization	All the register settings and hardware other than those that are initialized by the power-on reset (SINIT) and initialize reset (INIT).
Reset that concurrently occurs	No

9.5.2 Reset Resource

There are four types of reset resource. The level of the reset that is issued differs depending on the reset resource.

In addition, whether there is an occurrence of the irregular reset that issues initialize reset (INIT) without verifying completion of bus access, also depends on the reset resource.

- $\overline{\text{INIT}}$ pin input

  An initialize reset (INIT) request occurs while "L" level is input in the $\overline{\text{INIT}}$ pin.

Generation source	"L" level is input in the $\overline{\text{INIT}}$ pin
Cancellation source	"H" level is input in the $\overline{\text{INIT}}$ pin
Reset level	Issues all of the three resets: power-on reset (SINIT), initialize reset (INIT), and reset (RST)
Corresponding flag	ERST bit of the reset result register (RSTRR) = 1
Operation	Issues the power-on reset (SINIT), initialize reset (INIT), and reset (RST) without waiting for a completion of bus access (irregular reset).

- Watchdog reset 0

  The watchdog reset 0 request is generated if the period set for the watchdog timer 0 elapses. If the watchdog reset 0 request is generated, the initialize reset (INIT) is issued.

Generation source	The period set for the watchdog timer 0 elapses
Cancellation source	Automatically canceled after the initialize reset (INIT) is issued.
Reset level	Issues the initialize reset (INIT) and reset (RST)
Corresponding flag	WDG0 bit of the reset result register (RSTRR) = 1
Operation	■ Issues an initialize reset (INIT) and reset (RST) after the completion of bus access is verified. ■ Forcibly issues an initialize reset (INIT) and reset (RST) if a reset timeout occurs before completion of bus access (irregular reset).

- Watchdog reset 1

  The watchdog reset 1 request is generated if the period set for the watchdog timer 1 elapses. If the watchdog reset 1 request is generated, the initialize reset (INIT) is issued.

Generation source	The period set for the watchdog timer 1 elapses
Cancellation source	Automatically canceled after the initialize reset (INIT) is issued.
Reset level	Issues the initialize reset (INIT) and reset (RST)
Corresponding flag	WDG1 bit of the reset result register (RSTRR) = 1
Operation	■ Issues an initialize reset (INIT) and reset (RST) after the completion of bus access is verified. ■ Forcibly issues an initialize reset (INIT) and reset (RST) if a reset timeout occurs before completion of bus access (irregular reset).

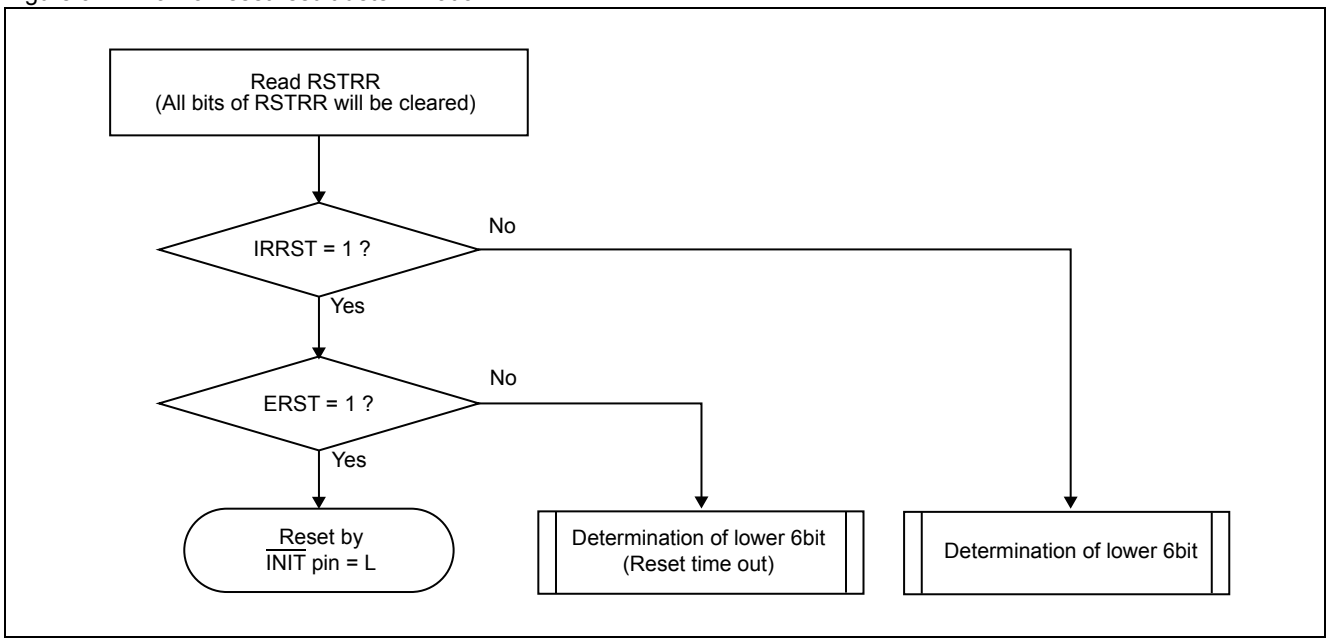
- Software reset (RSTCR:SRST)

  If the reset control resister (RSTCR) is read after "1" is written to the SRST bit of the reset control register (RSTCR), a reset (RST) request is generated.

Generation source	The reset control register (RSTCR) is read after "1" is written to the SRST bit of the reset control register (RSTCR). **Note:** Set the main clock (MCLK) to the source clock (SRCCLK) before writing "1" to SRST bit.
Cancellation source	Automatically canceled after the reset (RST) is issued.
Reset level	Issues only reset (RST)
Corresponding flag	SRST bit of the reset result register (RSTRR) = 1
Operation	■ Issues reset (RST) after verifying completion of bus access. ■ Forcibly issues an initialize reset (INIT) and reset (RST) if a reset timeout occurs before completion of bus access (irregular reset).

Flow of reset result determination

Figure 9-4. Flow of reset result determination



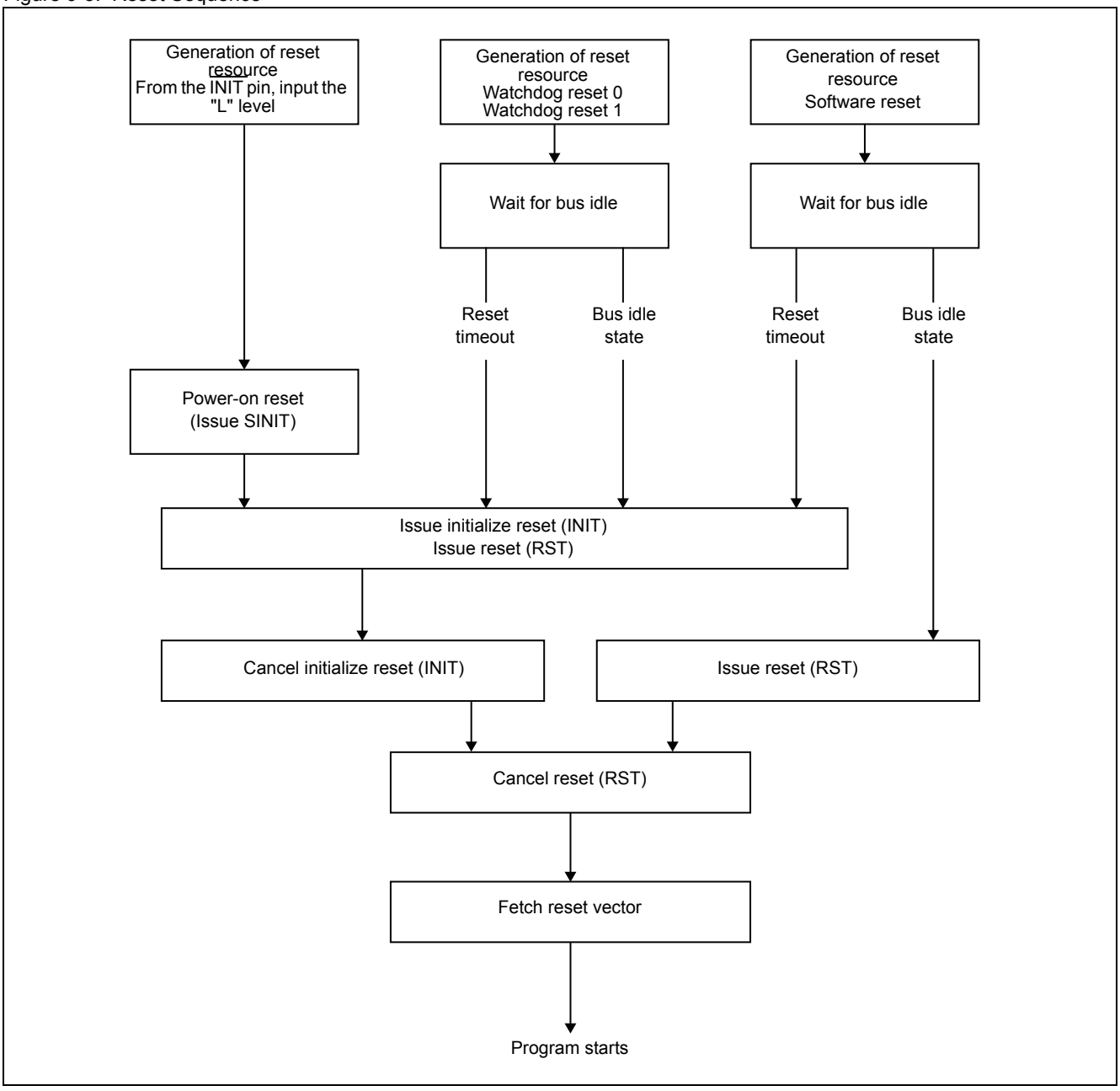
9.5.3 Operation of Reset

Flow of reset operation

A series of operations from the generation of reset, through reset state, until the CPU starts operation is called a reset sequence.

Figure 9-5 shows the reset sequence.

Figure 9-5. Reset Sequence



1. Retrieval and extension of reset resource

   The generated reset resource is asynchronously retrieved and retained until reset is issued.

   2 bits of resource extension counter retains the reset resource for at least 4Ts (T: Peripheral clock (PCLK) period).

2. Generation of the reset request

   Reports the generated reset request to the internal bus controller to perform the following processing.

   a. Stops the program operation of the CPU (same as for sleep mode).

   b. Verifies that the idle request has been reported to all busses.

   At the same time, the delay counter starts counting.

3. Acceptance of reset request and issue of reset

   After all processing for the reset request is completed, the reset request is accepted.

   An irregular reset is issued if a reset timeout occurs due to an overflow of the delay counter before response of the completion from the bus.

4. Issue of reset

   a. Input "L" level to $\overline{\text{INIT}}$ pin

      Issues a power-on reset (SINIT), initialize reset (INIT), and reset (RST).

   b. Watchdog reset 0

      Issues initialize reset (INIT) and reset (RST).

   c. Watchdog reset 1

      Issues initialize reset (INIT) and reset (RST).

   d. Reset time out

      Issues initialize reset (INIT) and reset (RST).

   e. Software reset (RSTCR:SRST)

      Issues reset (RST).

5. Cancellation of reset resource

   If the reset resource is canceled, the reset request is extended for a period of 4Ts (T: Peripheral clock (PCLK)). The request is then retained for 16 Ts (T: Peripheral clock (PCLK)) reset period. Therefore, the minimum cycle of reset issue is 20 Ts.

6. Cancellation of reset

   When the reset cycle ends, reset is canceled and the hardware starts operation.

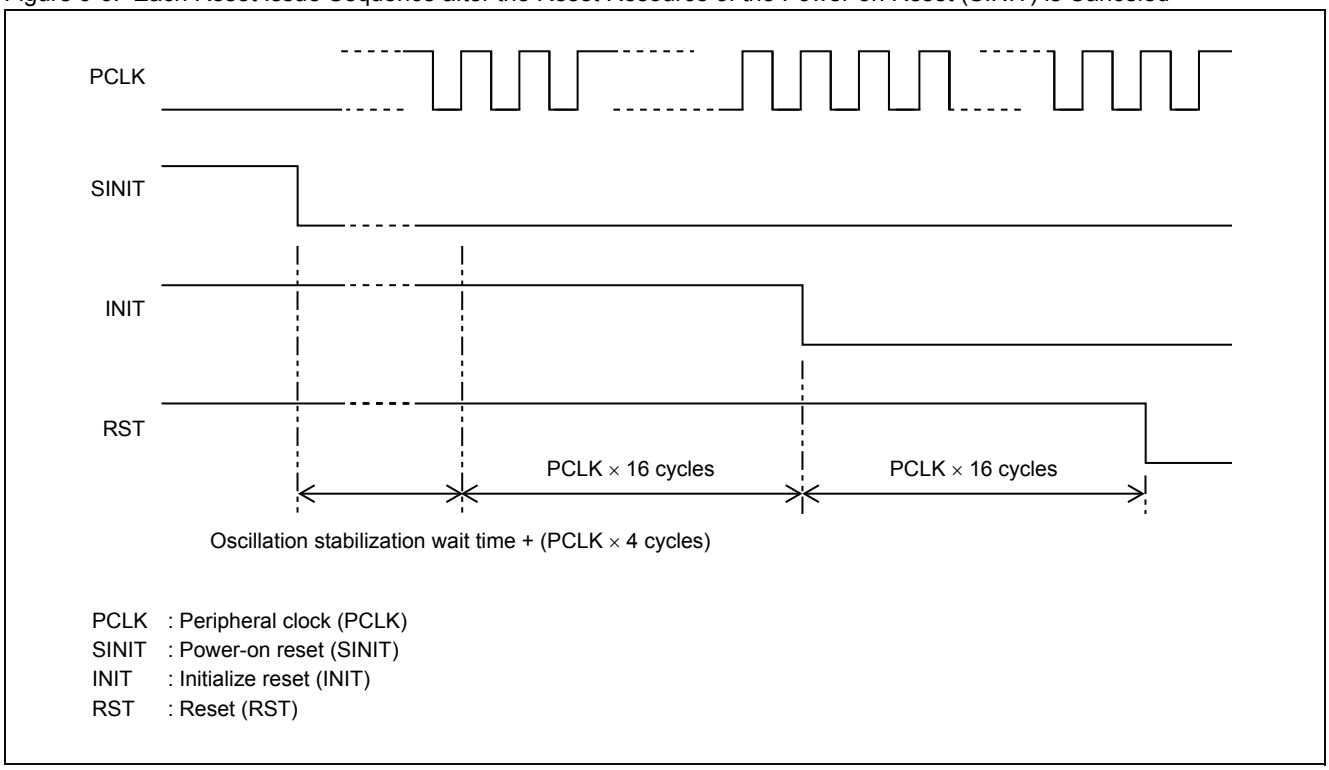
7. Retrieval of the reset vector (fetch)

   The CPU starts fetching the reset vector (000F $FFFC_H$). The CPU retrieves the fetched reset vector in the program counter (PC) to start program operation.

Power-on reset (SINIT)

Initialize reset (INIT) and reset (RST) are also issued at the same time as the power-on reset (SINIT) is issued.
Figure 9-6 shows the respective reset issue sequence after the reset resource of the power-on reset (SINIT) is canceled.

Figure 9-6. Each Reset Issue Sequence after the Reset Resource of the Power-on Reset (SINIT) is Canceled

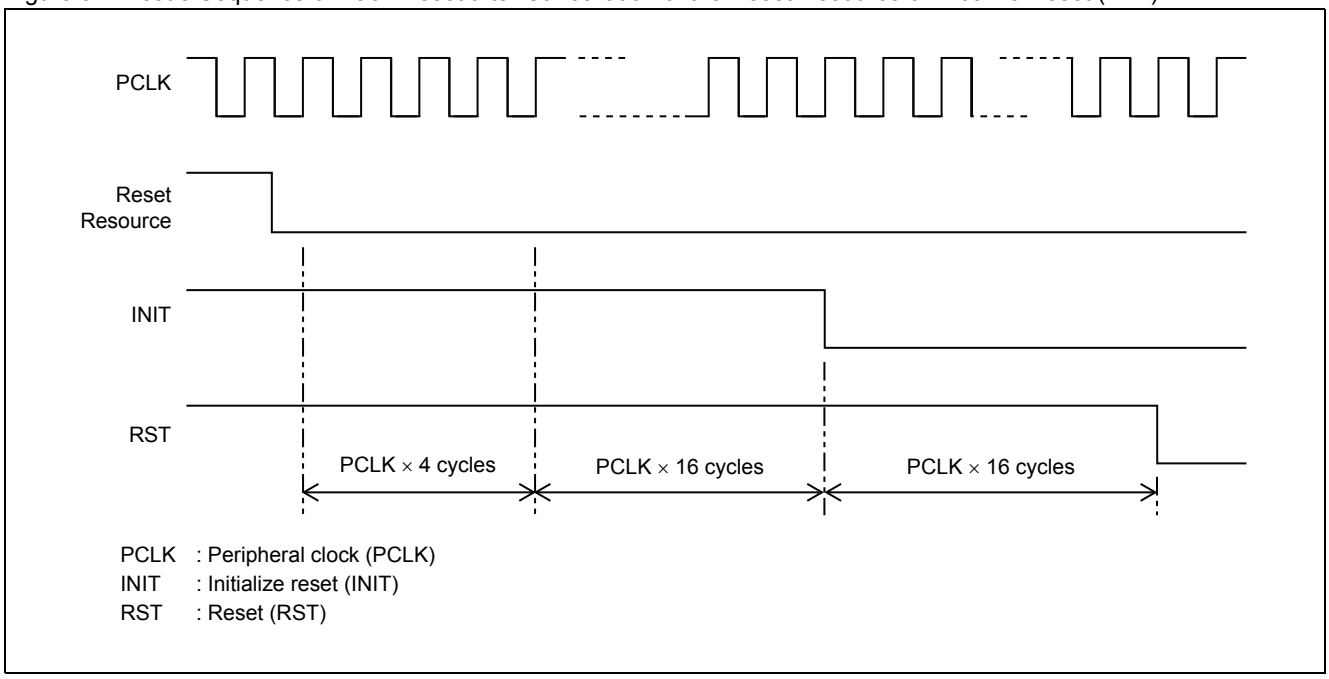


Initialize reset (INIT)

When initialize reset (INIT) is issued, reset (RST) is also issued at the same time.

Figure 9-7 shows the issue sequence of the respective resets after the reset resource of initialize reset (INIT) is canceled.

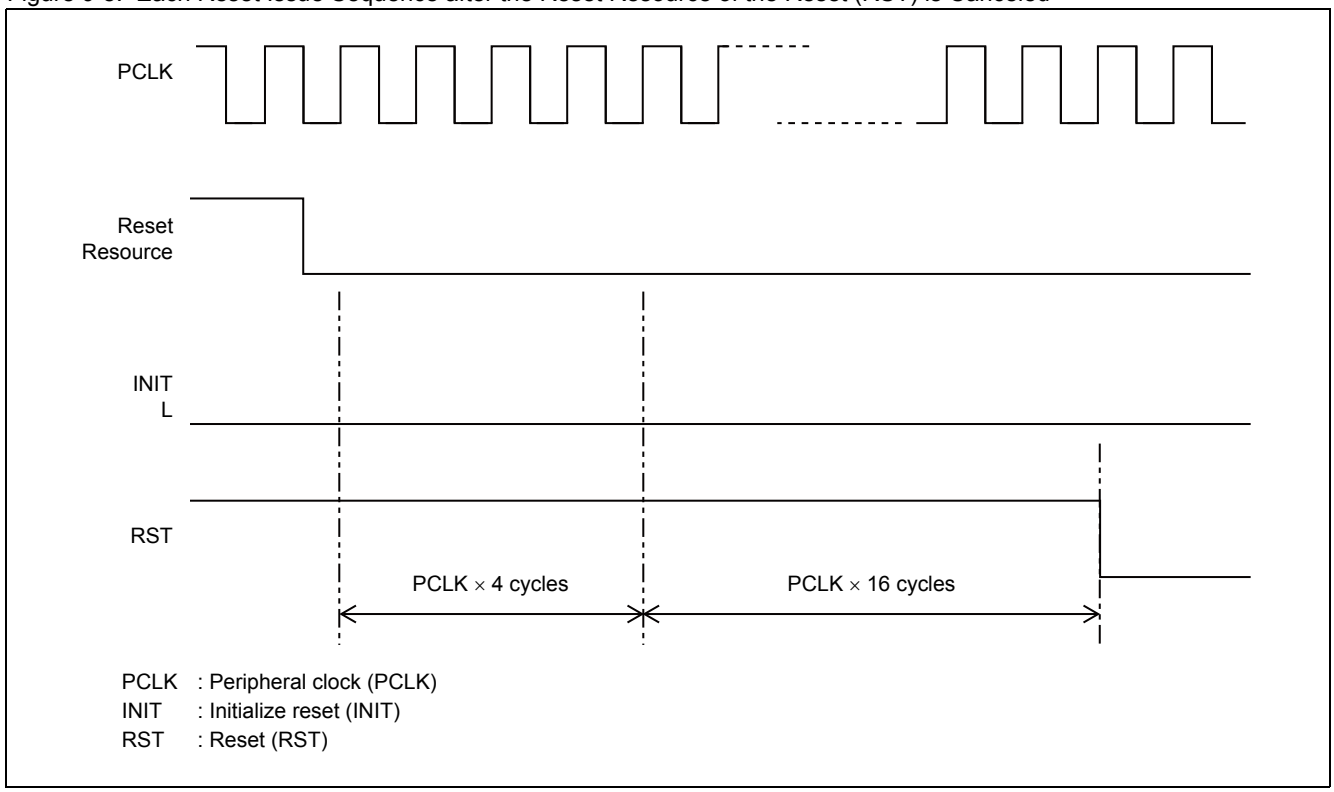
Figure 9-7. Issue Sequence of Each Reset after Cancellation of the Reset Resource of Initialize Reset (INIT)



Reset (RST)

Figure 9-8 shows the respective reset issue sequence after the reset resource of reset (RST) is canceled.

Figure 9-8. Each Reset Issue Sequence after the Reset Resource of the Reset (RST) is Canceled



9.5.4 Irregular Reset

Irregular reset occurs in the following cases.

- When an $\overline{\text{INIT}}$ pin input (INIT) is used
- When a reset timeout occurs

  (The delay counter overflows before the response from the bus is received during watchdog reset 0 /watchdog reset 1 / software reset (RSTCR: SRST).)

If irregular reset occurs, the following processes are executed.

- Initialize reset (INIT) is issued.
- The IRRST bit of the reset result register (RSTRR) changes to "1".

Note: When irregular reset occurs, the bus access may be performed at the time of reset input. In this case, the contents of the memory may be damaged.

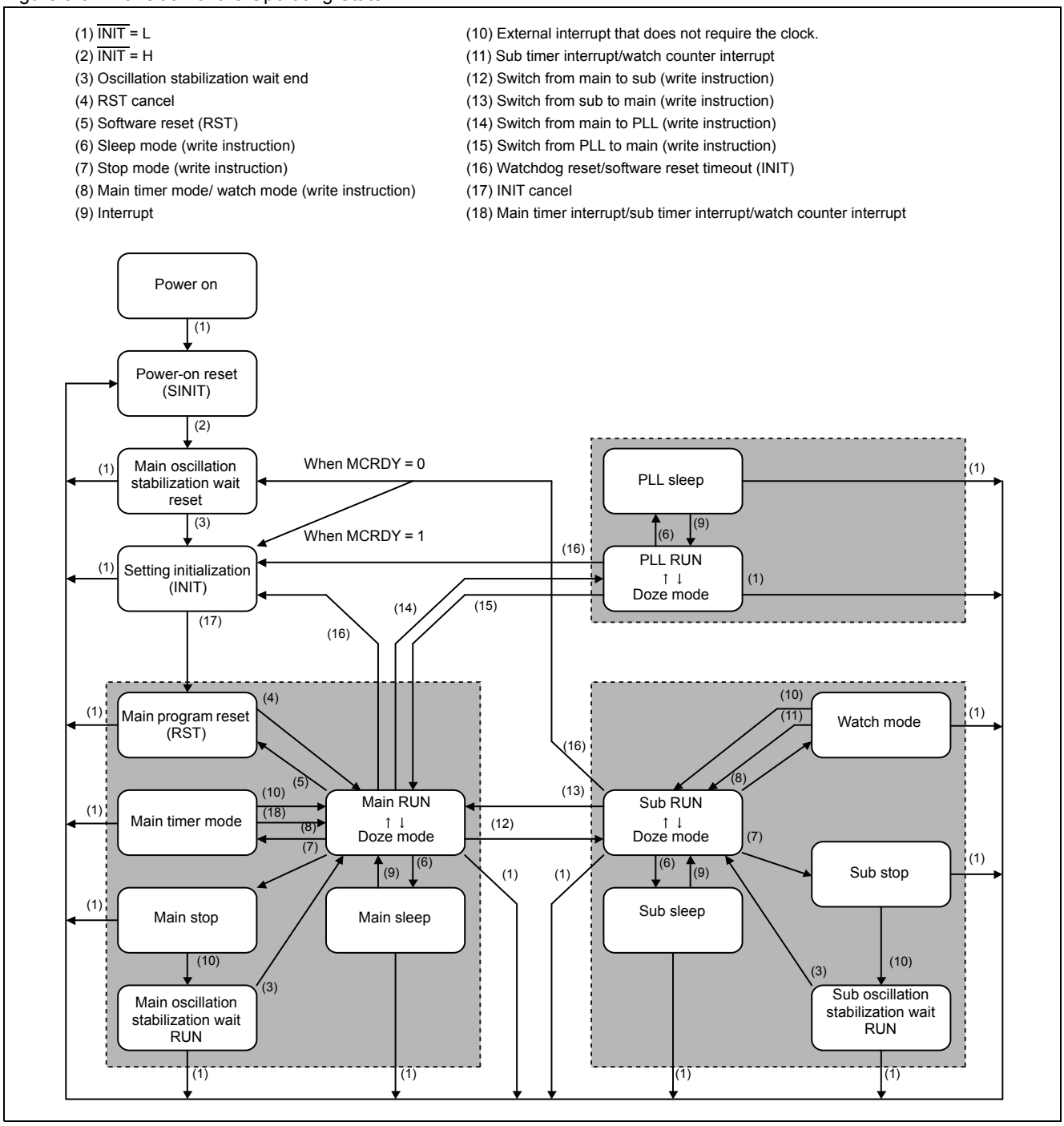
9.6 Operating State and Transition

This section explains each operating state and how to control it.

Operating state

Figure 9-9 shows transition of the operating state.

Figure 9-9. Transition of the Operating State



- RUN state (normal operation)

  Program is running.

  All the internal clocks are delivered and all the circuits are enabled.

  The Hi-Z control of the external pins in stop state, main timer mode state and watch mode state is canceled.

- Sleep state

  Program is stopped. Transition occurs by program operation.

  Only program execution of the CPU is stopped. The peripheral circuits are enabled.

  The built-in memories and external busses are suspended until the DMA controller (DMAC) request is received.

  In bus sleep mode, the internal bus is suspended until the DMA controller (DMAC) request is received.

  - If a valid interrupt request is generated, the device undergoes transition to the RUN state (normal operation).
  - If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

- Watch mode state

  The device is in a suspended state. Transition occurs by the program operation.

  Internal circuits other than the oscillation circuits (sub clock (SBCLK)) are stopped.

  The external pins can be uniformly set to Hi-Z (excluding certain pins).

  - If an external interrupt request is generated, the device undergoes transition to the RUN state (normal operation).
  - If a sub timer interrupt, or watch counter interrupt request is generated, it undergoes transition to the RUN state (normal operation).
  - If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

  Note: Stop oscillation of the main clock (MCLK) and PLL clock (PLLCLK) before transition to watch mode.

- Main timer mode state

  The device is in a suspended state. Transition occurs by the program operation.

  Internal circuits other than the oscillation circuits (main clock (MCLK) and sub clock (SBCLK)) are stopped.

  The external pins can be uniformly set to Hi-Z (excluding certain pins).

  - If an external interrupt is generated, the device undergoes transition to the RUN state (normal operation).
  - If a main timer interrupt, sub timer interrupt, and watch counter interrupt requests are generated, it undergoes transition to the RUN state (normal operation).
  - If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

  Note: Stop oscillation of the PLL clock (PLLCLK) before transition to main timer mode.

- Stop state

  The device is in a suspended state. Transition occurs by the program operation.

  All the internal circuits are suspended.

  The external pins can be uniformly set to Hi-Z (excluding certain pins).

  - If an external interrupt request is generated, the device undergoes transition to the oscillation stabilization wait RUN state.
  - If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

  Note: Stop oscillation of the PLL clock (PLLCLK) before transition to the stop state.

- Oscillation stabilization wait RUN state

  The device is in a suspended state. Transition to this state occurs after the device returns from the stop state.

  All the internal circuits are suspended (excluding timer operation for clock stabilization wait).

  While all the internal clocks are stopped, oscillation circuits that have been enabled operate.

  - When the oscillation stabilization wait time elapses, the device undergoes transition to the RUN state (normal operation).
  - If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.
- Oscillation stabilization wait reset (RST) state

  The device is in a suspended state. Transition occurs after the device returns from power-on reset (SINIT).

  All the internal circuits are suspended (excluding timer operation for oscillation stabilization wait).

  While all the internal clocks are suspended, the main oscillation circuit operates.

  - When the oscillation stabilization wait time elapses, the device undergoes transition to the initialize reset (INIT) state.
  - If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.
- Program reset (RST) state

  Program is in the initialized state. Transition occurs when a reset (RST) request is accepted or after the initialize reset (INIT) state ends.

  The program execution of the CPU is suspended and the program counter is initialized. The peripheral circuits are initialized (excluding certain circuits).

  All the internal clocks as well as the oscillation circuits that have been enabled and the PLL clock (PLLCLK) operate.

  - The reset (RST) request for the internal circuits is generated. When the reset (RST) request disappears, transition to the RUN state (normal operation) occurs.
  - If "L" level is input in the $\overline{\text{INIT}}$ pin, the device undergoes transition to the power-on reset (SINIT) state.
- Initialize reset (INIT) state

  This is the state in which all settings are initialized. Transition occurs when the initialize reset (INIT) request is accepted.

  The program execution of the CPU is suspended and the program counter is initialized. All the peripheral circuits are initialized. The main clock (MCLK) oscillation circuit operates (while the sub clock (SBCLK) oscillation circuit and PLL clock (PLLCLK) oscillation circuit stop operation). All the internal clocks stop while the "L" level is being input in the $\overline{\text{INIT}}$ pin. Otherwise, they operate.

  Initialize reset (INIT) and reset (RST) are output to the internal circuit.

  - When the initialize reset (INIT) request disappears, this state is canceled and transition to the program reset (RST) state occurs.
  - If "L" is input in the $\overline{\text{INIT}}$ pin, the device undergoes transition to the power-on reset (SINIT) state.

Priority of state transition requests

state transition requests are prioritized in the following order in any states. However, since some requests are generated only in the particular states, they are enabled only in those states.



Highest priority	Power-on reset (SINIT) request	
	Initialize reset (INIT) request	
	Oscillation stabilization wait time end	Occurs only in the oscillation stabilization wait reset state and oscillation stabilization wait RUN state
	Reset (RST) request	
	Valid interrupt request	Occurs only in the RUN, sleep, stop, and watch mode state
	Stop mode request (register write)	Occurs only in the RUN state
	Watch mode request (register write)	Occurs only in the RUN state
Lowest priority	Sleep mode request (register write)	Occurs only in the RUN state

10. Interrupt Controller

This chapter explains the functions and operations of the interrupt controller.

10.1 Overview

10.2 Configuration

10.3 Registers

10.4 An Explanation of Operations and Setting Procedure Examples

10.5 Notes on Use

10.1 Overview

The interrupt controller determines the priority of an interrupt request and sends the request to the CPU.

Overview

The interrupt control has the following functions:

- Accepts interrupt requests from peripheral functions.
- Determines the priority of sending interrupt requests to the CPU according to the interrupt level and interrupt vector.
- Sends the highest priority interrupt request to the CPU.
- Sends the interrupt vector number of the highest priority interrupt request to the CPU.
- Generates a request for returning from sleep mode or stop mode according to an interrupt request with an interrupt level other than "1111".

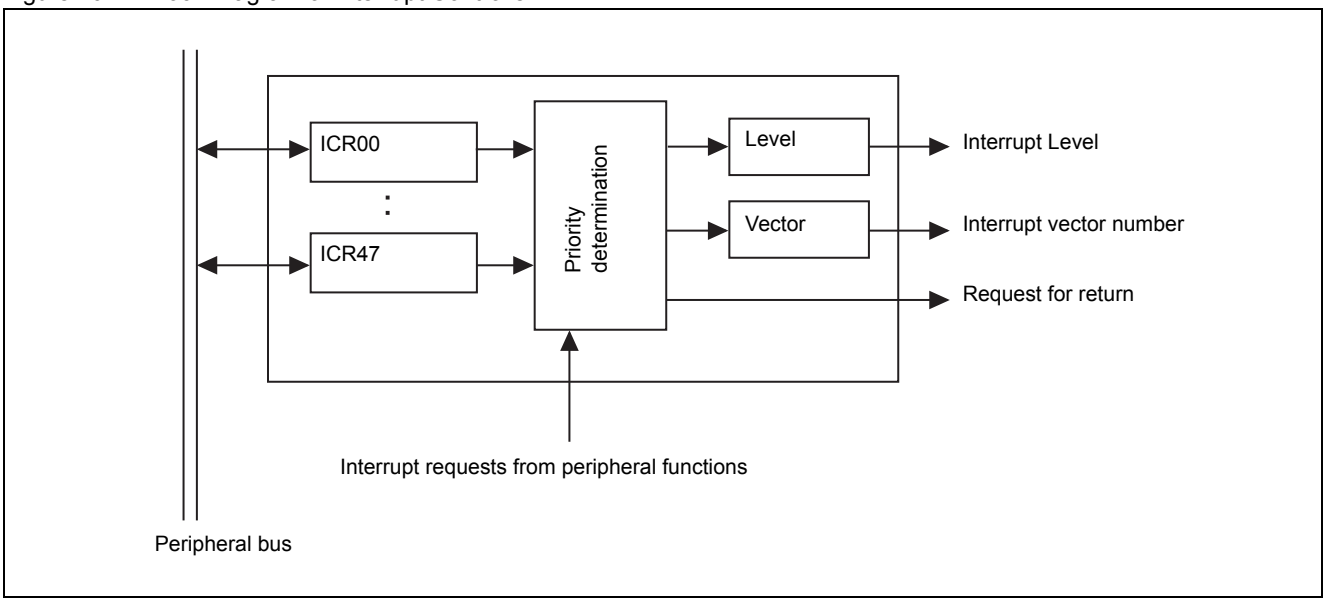
10.2 Configuration

This section explains the interrupt controller configuration.

Block diagram of interrupt controller

Figure 10-1 shows a block diagram of the interrupt controller.

Figure 10-1. Block Diagram of Interrupt Controller



- Interrupt priority determination circuit

  This circuit determines the priority of an incoming interrupt request. It also generates a request to return from sleep mode or stop mode.
- Interrupt level generating circuit

  This circuit transmits the interrupt level of an interrupt request to the CPU.
- Interrupt vector generating circuit

  This circuit sends the interrupt vector of an interrupt request to the CPU.
- Interrupt control registers (ICR00 to ICR47)

  These registers are used to set the interrupt levels of interrupt requests.

Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

10.3 Registers

This section explains the configurations and functions of the registers used by the interrupt controller.

Interrupt controller registers

Figure 10-2 lists the interrupt controller registers.

Table 10-1. Interrupt Controller Registers

Abbreviated Register Name	Register Name	Reference
ICR00 to ICR47	Interrupt control registers 00 to 47	10.3.1

10.3.1 Interrupt Control Register (ICR00 to ICR47)

These registers are used to set interrupt levels. This register is provided for input of each interrupt.

Figure 10-2 shows the bit configuration of the interrupt control registers (ICR00 to ICR47).

Figure 10-2. Bit Configuration of Interrupt Control Registers (ICR00 to ICR47)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	IL4	IL3	IL2	IL1	IL0
Attribute	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

R/W: Read/Write

R: Read only

[bit7 to bit5]: Undefined bits

In case of writing	Ignored
In case of reading	"1" is read.

[bit4 to bit0]: IL4 to IL0 (interrupt level control bits)

These bits specify the interrupt level of an interrupt request.

When reset, the bits are initialized to IL4 to IL0=11111("11111_B" is level 31 interrupt disabled).

IL4	IL3	IL2	IL1	IL0	Interrupt Level	
1	0	0	0	0	16	Highest level that can be set
1	0	0	0	1	17	
1	0	0	1	0	18	
1	0	0	1	1	19	
1	0	1	0	0	20	
1	0	1	0	1	21	
1	0	1	1	0	22	
1	0	1	1	1	23	
1	1	0	0	0	24	
1	1	0	0	1	25	
1	1	0	1	0	26	
1	1	0	1	1	27	
1	1	1	0	0	28	
1	1	1	0	1	29	
1	1	1	1	0	30	Lowest level that can be set
1	1	1	1	1	31	Interrupt Disabled

Notes:

- If the interrupt level that is set in this register is lower than the mask level in the CPU interrupt level mask register (ILM), the interrupt request is masked on the CPU side.
- The interrupt control register (ICR00 to ICR47) in which an interrupt level is set varies depending on the peripheral function. For information on the correspondence between the peripheral function and interrupt control register (ICR00 to ICR47), see "A.3 Interrupt Vectors".
- IL4 bit is fixed to "1" and IL3 to IL0 can be set.

10.4 An Explanation of Operations and Setting Procedure Examples

This section explains the operations of the interrupt controller.

10.4.1 Explanation of Operations of Interrupt Controller

This section explains the three types of operations of the interrupt controller.

- Specifying interrupt levels using interrupt control registers (ICR00 to ICR47)
- Determining the priorities of interrupt requests
- Generating a request to return from sleep mode or stop mode

Specifying an interrupt level

The procedure for setting interrupt levels using interrupt control registers (ICR00 to ICR47) is shown below:

1. Set an interrupt level in the interrupt control register (ICR00 to ICR47) with the interrupt vector number corresponding to the peripheral function for which an interrupt request needs to be generated.
   For information on the correspondence between interrupt control numbers and interrupt requests, see "A.3 Interrupt Vectors".
2. Enable generation of interrupt requests on the peripheral function for which an interrupt request needs to be generated.
3. Activate the relevant peripheral function.

Determining the priorities of interrupt requests

The interrupt controller sends the interrupt level and interrupt vector number of the highest priority interrupt request, among the interrupt requests that are concurrently generated, to the CPU.

The criteria for determining the priorities of interrupt requests are shown in order of determining:

1. Is the interrupt level of the interrupt request "30" or lower (Level 31 is "Interrupt Disabled").
2. Is the value of the interrupt level of the interrupt request the smallest.
3. If the interrupt level is the same, is the interrupt vector number of the interrupt request the smallest.

If no interrupt request meets the above criteria, interrupt level "31" (11111_B) that indicates no interrupt request is output to the CPU.

Generating a request to return from sleep mode

If an interrupt request with an interrupt level other than "31" is generated, the interrupt controller generates a request to the clock control part to return from sleep mode.

Generating a request to return from stop mode

If an external interrupt request /USB function with an interrupt level other than "31" is generated, the interrupt controller generates a request to the clock control part to return from stop mode.

After return from the stop mode, the interrupt priority determination circuit resumes operation only after the operation of clock begins. The CPU thus executes instructions until the interrupt priority determination circuit produces results.

Note: For interrupts that are not used as causes of return from stop mode, set interrupt level "31" (Interrupt Disabled) in the corresponding interrupt control registers (ICR00 to ICR47).

10.5 Notes on Use

Note the following points about using the interrupt controller.

Note on the program

- For interrupt requests that should not be used to generate a request to return from sleep mode or stop mode, set interrupt level "31" (Interrupt Disabled) in the corresponding interrupt control registers (ICR00 to ICR47).

Notes on operations

- If the interrupt level that is set in an interrupt control register (ICR00 to ICR47) is lower than the mask level in the CPU interrupt level mask register (ILM), the interrupt request is masked on the CPU side.

11. Interrupt Request Batch-Read Function

This section explains the interrupt request batch-read function.

11.1 Overview

11.2 Configuration

11.3 Registers

11.4 Notes on Use

11.1 Overview

The interrupt request batch-read function reads multiple interrupt requests assigned to one interrupt vector all at once.

The bit search instruction of an FR80 family CPUs can be used to quickly check which interrupt requests have been generated.

This function allows the user to check at one time whether interrupt requests that use the same interrupt vector number have been generated.

Note: This function cannot clear the interrupt request flag. Use the register of each peripheral function to clear the interrupt request flag.

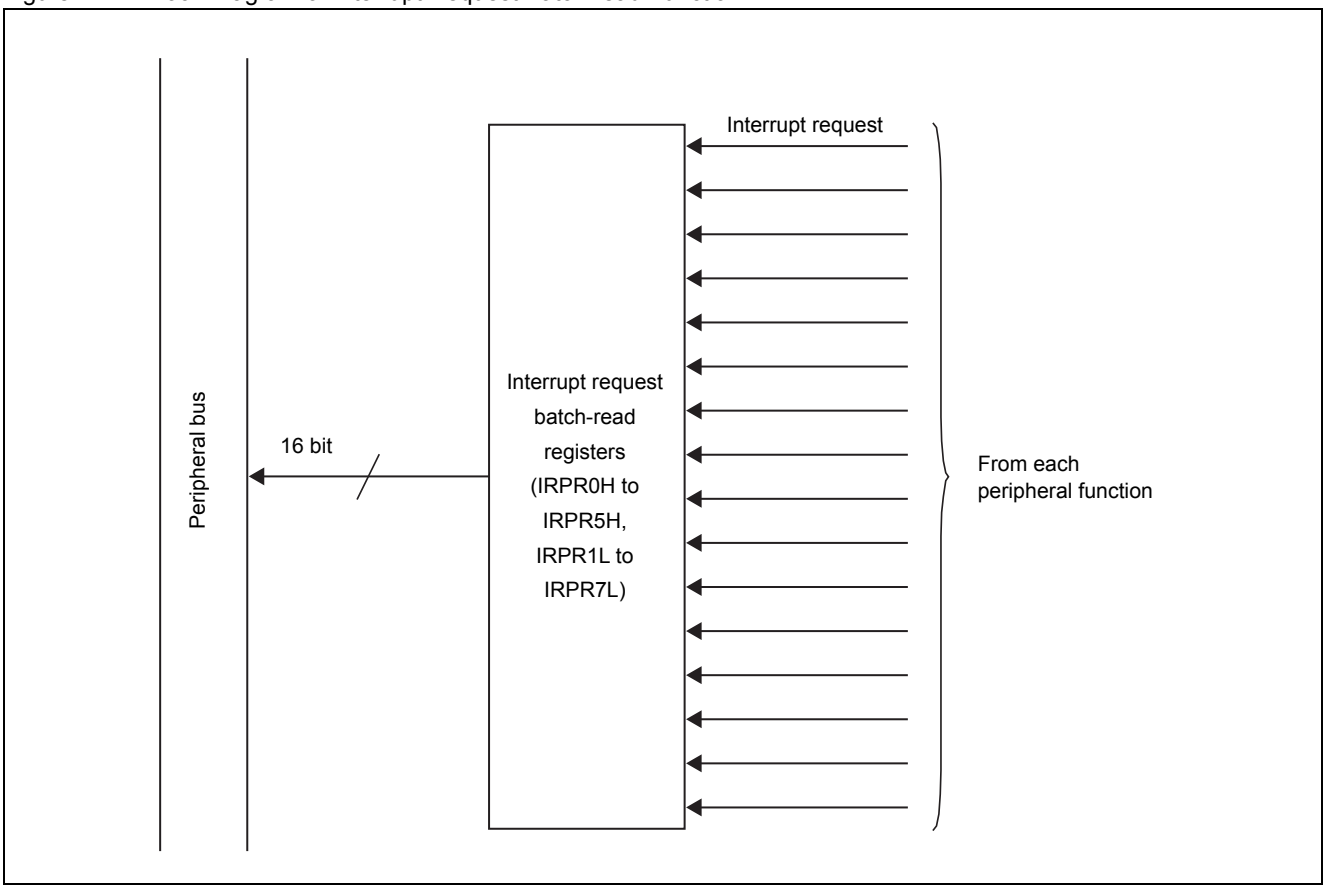
11.2 Configuration

This section shows the configuration of the interrupt request batch-read function.

Block diagram of interrupt request batch-read function

Figure 11-1 is a block diagram of the interrupt request batch-read function.

Figure 11-1. Block Diagram of Interrupt Request Batch-read Function



Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

11.3 Registers

This section explains the configuration and functions of registers used by the interrupt request batch-read function.

Registers for interrupt request batch-read function

Table 11-1 lists the registers for the interrupt request batch-read function.

Table 11-1. Registers for the Interrupt Request Batch-read Function

Abbreviated Register Name	Register Name	Reference
IRPR0H	Interrupt request batch-read register 0 upper	11.3.1
IRPR1H/ IRPR1L	Interrupt request batch-read register 1 upper/lower	11.3.2
IRPR2L	Interrupt request batch-read register 2 lower	11.3.3
IRPR3H	Interrupt request batch-read register 3 upper	11.3.4
IRPR4H	Interrupt request batch-read register 4 upper	11.3.5
IRPR5H/ IRPR5L	Interrupt request batch-read register 5 upper/lower	11.3.6, 11.3.7
IRPR7L	Interrupt request batch-read register 7 lower	11.3.8

11.3.1 Interrupt Request Batch-Read Register 0 Upper (IRPR0H)

The interrupt requests of 16-bit reload timer ch.0 to ch.2 are assigned to interrupt vector number 20 (decimal). This register can be read to check the channel on which an interrupt request has been generated.

Figure 11-2 shows the bit configuration of interrupt request batch-read register 0 upper (IRPR0H).

Figure 11-2. Bit Configuration of Interrupt Request Batch-read Register 0 Upper (IRPR0H)

Interrupt request batch-read register 0 upper (IRPR0H)

bit	15	14	13	12	11	10	9	8
	RTIR0	RTIR1	RTIR2	Undefined	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

The bit corresponding to the channel on which an interrupt request has been generated is set to "1".

Bit number	Bit	Value	Explanation
bit15	RTIR0	0	No interrupt request in reload timer ch.0
		1	Interrupt request in reload timer ch.0
bit14	RTIR1	0	No interrupt request in reload timer ch.1
		1	Interrupt request in reload timer ch.1
bit13	RTIR2	0	No interrupt request in reload timer ch.2
		1	Interrupt request in reload timer ch.2
bit12 to bit8	Undefined	"0" is read.	

11.3.2 Interrupt Request Batch-Read Register 1 Upper/Lower (IRPR1H/ IRPR1L)

Interrupt vector number 39 (decimal) is used for multifunction serial interface channels ch.8 to ch.11. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11-3 shows the bit configuration of interrupt request batch-read register 1 upper/lower (IRPR1H/ IRPR1L).

Figure 11-3. Bit Configuration of Interrupt Request Batch-read Register 1 Upper/lower (IRPR1H/ IRPR1L)

Interrupt request batch-read register 1 upper (IRPR1H)

bit	15	14	13	12	11	10	9	8
	RXIR8	TXIR8	ISIR8	Undefined	RXIR9	TXIR9	ISIR9	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Interrupt request batch-read register 1 lower (IRPR1L)

bit	7	6	5	4	3	2	1	0
	RXIR10	TXIR10	ISIR10	Undefined	RXIR11	TXIR11	ISIR11	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	RXIR8	0	No UART/CSIO/I^{2}C receive interrupt request on ch.8
		1	UART/CSIO/I^{2}C receive interrupt request on ch.8
bit14	TXIR8	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.8
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.8
bit13	ISIR8	0	No I^{2}C status interrupt request on ch.8
		1	I^{2}C status interrupt request on ch.8
bit12	Undefined	"0" is read.	
bit11	RXIR9	0	No UART/CSIO/I^{2}C receive interrupt request on ch.9
		1	UART/CSIO/I^{2}C receive interrupt request on ch.9
bit10	TXIR9	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.9
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.9
bit9	ISIR9	0	No I^{2}C status interrupt request on ch.9
		1	I^{2}C status interrupt request on ch.9
bit8	Undefined	"0" is read.	
bit7	RXIR10	0	No UART/CSIO/I^{2}C receive interrupt request on ch.10
		1	UART/CSIO/I^{2}C receive interrupt request on ch.10
bit6	TXIR10	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.10
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.10
bit5	ISIR10	0	No I^{2}C status interrupt request on ch.10
		1	I^{2}C status interrupt request on ch.10
bit4	Undefined	"0" is read.	
bit3	RXIR11	0	No UART/CSIO/I^{2}C receive interrupt request on ch.11
		1	UART/CSIO/I^{2}C receive interrupt request on ch.11
bit2	TXIR11	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.11
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.11
bit1	ISIR11	0	No I^{2}C status interrupt request on ch.11
		1	I^{2}C status interrupt request on ch.11
bit0	Undefined	"0" is read.	

11.3.3 Interrupt Request Batch-Read Register 2 Lower (IRPR2L)

Interrupt vector number 41 (decimal) is used for the following peripheral functions:

- Main timer
- Sub timer
- Watch counter

This register can be read to check the peripheral function from which an interrupt request has been generated.

Figure 11-4 shows the bit configuration of interrupt request batch-read register 2 lower (IRPR2L).

Figure 11-4. Bit Configuration of Interrupt Request Batch-read Register 2 Lower (IRPR2L)

bit	7	6	5	4	3	2	1	0
	MCIR	SCIR	TCIR	Undefined	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit7	MCIR	0	No main timer interrupt request
		1	Main timer interrupt request
bit6	SCIR	0	No sub timer interrupt request
		1	Sub timer interrupt request
bit5	TCIR	0	No watch counter interrupt request
		1	Watch counter interrupt request
bit4 to bit0	Undefined	"0" is read.	

11.3.4 Interrupt Request Batch-Read Register 3 Upper (IRPR3H)

Interrupt vector number 44 (decimal) is used for 32-bit input capture channels ch.0 to ch.3. This register can be read to check on which channel an interrupt request has been generated.

Figure 11-5 shows the bit configuration of interrupt request batch-read register 3 upper (IRPR3H).

Figure 11-5. Bit Configuration of Interrupt Request Batch-read Register 3 Upper (IRPR3H)

bit	15	14	13	12	11	10	9	8
	ICIR0	ICIR1	ICIR2	ICIR3	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	ICIR0	0	No interrupt request on 32-bit input capture ch.0
		1	Interrupt request on 32-bit input capture ch.0
bit14	ICIR1	0	No interrupt request on 32-bit input capture ch.1
		1	Interrupt request on 32-bit input capture ch.1
bit13	ICIR2	0	No interrupt request on 32-bit input capture ch.2
		1	Interrupt request on 32-bit input capture ch.2
bit12	ICIR3	0	No interrupt request on 32-bit input capture ch.3
		1	Interrupt request on 32-bit input capture ch.3
bit11 to bit8	Undefined	"0" is read.	

11.3.5 Interrupt Request Batch-Read Register 4 Upper (IRPR4H)

Interrupt vector number 45 (decimal) is used for 32-bit output compare channels ch.0 to ch.3. This register can be read to check on which channel an interrupt request has been generated.

Figure 11-6 shows the bit configuration of interrupt request batch-read register 4 upper (IRPR4H).

Figure 11-6. Bit Configuration of Interrupt Request Batch-read Register 4 Upper (IRPR4H)

bit	15	14	13	12	11	10	9	8
	OCIR0	OCIR1	OCIR2	OCIR3	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	OCIR0	0	No interrupt request on 32-bit output compare ch.0
		1	Interrupt request on 32-bit output compare ch.0
bit14	OCIR1	0	No interrupt request on 32-bit output compare ch.1
		1	Interrupt request on 32-bit output compare ch.1
bit13	OCIR2	0	No interrupt request on 32-bit output compare ch.2
		1	Interrupt request on 32-bit output compare ch.2
bit12	OCIR3	0	No interrupt request on 32-bit output compare ch.3
		1	Interrupt request on 32-bit output compare ch.3
bit11 to bit8	Undefined	"0" is read.	

11.3.6 Interrupt Request Batch-Read Register 5 Upper (IRPR5H)

Interrupt vector number 50 (decimal) is used for base timer channels ch.4 and ch.5. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11-7 shows the bit configuration of interrupt request batch-read register 5 upper (IRPR5H).

Figure 11-7. Bit Configuration of Interrupt Request Batch-read Register 5 Upper (IRPR5H)

bit	15	14	13	12	11	10	9	8
	BT0IR4	BT1IR4	BT0IR5	BT1IR5	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	BT0IR4	0	No interrupt request 0 generated on base timer ch.4
		1	Interrupt request 0 generated on base timer ch.4
bit14	BT1IR4	0	No interrupt request 1 generated on base timer ch.4
		1	Interrupt request 1 generated on base timer ch.4
bit13	BT0IR5	0	No interrupt request 0 generated on base timer ch.5
		1	Interrupt request 0 generated on base timer ch.5
bit12	BT1IR5	0	No interrupt request 1 generated on base timer ch.5
		1	Interrupt request 1 generated on base timer ch.5
bit11 to bit8	Undefined	"0" is read.	

Interrupt requests 0 and 1 vary depending on the mode of the base timer operation.

Modes of base timer operation	Interrupt request 0	Interrupt request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

11.3.7 Interrupt Request Batch-Read Register 5 Lower (IRPR5L)

Interrupt vector number 51 (decimal) is used for base timer channels ch.6 and ch.7. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11-8 shows the bit configuration of interrupt request batch-read register 5 lower (IRPR5L).

Figure 11-8. Bit Configuration of Interrupt Request Batch-read Register 5 Lower (IRPR5L)

bit	7	6	5	4	3	2	1	0
	BT0IR6	BT1IR6	BT0IR7	BT1IR7	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit7	BT0IR6	0	No interrupt request 0 generated on base timer ch.6
		1	Interrupt request 0 generated on base timer ch.6
bit6	BT1IR6	0	No interrupt request 1 generated on base timer ch.6
		1	Interrupt request 1 generated on base timer ch.6
bit5	BT0IR7	0	No interrupt request 0 generated on base timer ch.7
		1	Interrupt request 0 generated on base timer ch.7
bit4	BT1IR7	0	No interrupt request 1 generated on base timer ch.7
		1	Interrupt request 1 generated on base timer ch.7
bit3 to bit0	Undefined	"0" is read.	

Interrupt requests 0 and 1 vary depending on the mode of the base timer operation.

Modes of base timer operation	Interrupt request 0	Interrupt request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

11.3.8 Interrupt Request Batch-Read Register 7 Lower (IRPR7L)

Interrupt vector number 61 (decimal) is used for DMA controller (DMAC) channels ch.4 to ch.7. This register can be read to check on which channel an interrupt request has been generated.

Figure 11-9 shows the bit configuration of interrupt request batch-read register 7 lower (IRPR7L).

Figure 11-9. Bit Configuration of Interrupt Request Batch-read Register 7 Lower (IRPR7L)

bit	7	6	5	4	3	2	1	0
	DMAC4	DMAC5	DMAC6	DMAC7	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When one of the following interrupt requests is generated on DMA controller (DMAC) ch.4 to ch.7, the bit corresponding to the generated interrupt request is set to "1".

- Normal end interrupt request
- Abnormal end interrupt request
- Transfer stop interrupt request

Bit number	Bit	Value	Explanation
bit7	DMAC4	0	No interrupt request on DMAC ch.4
		1	Interrupt request on DMAC ch.4
bit6	DMAC5	0	No interrupt request on DMAC ch.5
		1	Interrupt request on DMAC ch.5
bit5	DMAC6	0	No interrupt request on DMAC ch.6
		1	Interrupt request on DMAC ch.6
bit4	DMAC7	0	No interrupt request on DMAC ch.7
		1	Interrupt request on DMAC ch.7
bit3 to bit0	Undefined	"0" is read.	

11.4 Notes on Use

Note the following points about using the interrupt request batch-read function.

Notes on operations

- Writing to the interrupt request batch-read register (IRPR0 to IRPR7) is disabled. To cancel an interrupt request, clear the interrupt request flag bit of the corresponding function register.

12. Delay Interrupt

This chapter explains the functions and operations of the delay interrupt function.

12.1 Overview

12.2 Configuration

12.3 Registers

12.4 An Explanation of Operations and Setting Procedure Examples

12.5 Notes on Use

12.1 Overview

The delay interrupt function generates task switching interrupts used by a real-time OS.

Overview

The delay interrupt function generates task switching interrupt requests used by a real-time OS such as REALOS. Software can use delay interrupts to generate interrupt requests to the CPU or cancel them.

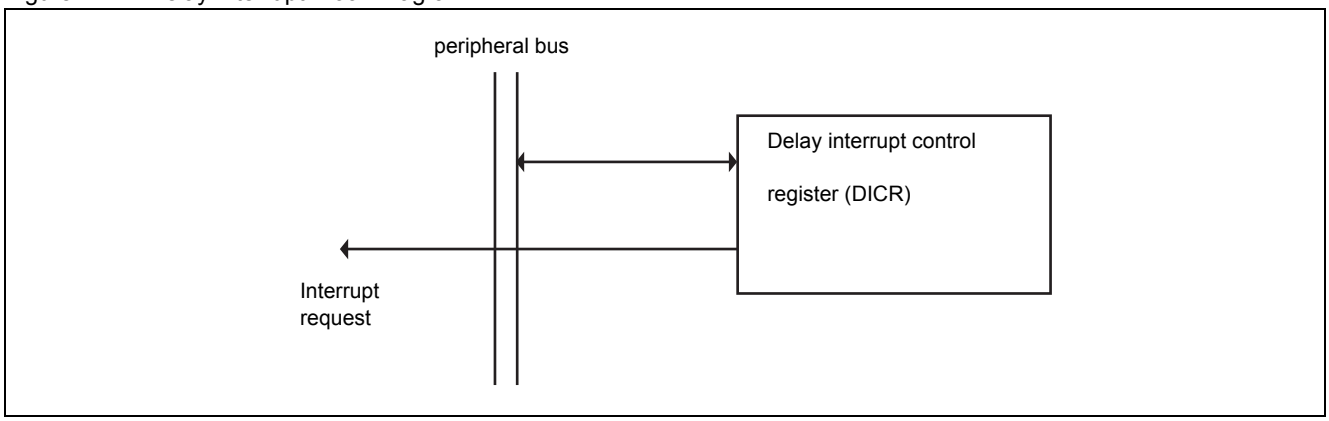
12.2 Configuration

This section explains the configuration of delay interrupts.

Delay interrupt block diagram

Figure 12-1 shows a delay interrupt block diagram.

Figure 12-1. Delay Interrupt Block Diagram



- Delayed interrupt control register (DICR)
  This register controls delay interrupts.

Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

12.3 Registers

This section explains the configuration and functions of the register used for delay interrupts.

Delay interrupt register

Table 12-1 shows the delay interrupt register.

Table 12-1. Delay Interrupt Register

Abbreviated Register Name	Register Name	Reference
DICR	Delayed interrupt control register	12.3.1

12.3.1 Delayed Interrupt Control Register (DICR)

This register controls delay interrupts.

Figure 12-2 shows the bit configuration of the delayed interrupt control register (DICR).

Figure 12-2. Bit Configuration of Delayed Interrupt Control Register (DICR)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	DLYI
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	0

R/W: Read/Write

[bit7 to bit1]: Undefined bits

In case of writing	Ignored
In case of reading	"1" is read.

[bit0]: DLYI (delay interrupt control bit)

This bit is used to enable generation of delay interrupt requests or cancel the delay interrupt requests.

Written Value	Explanation
0	Cancels delay interrupt source or generates no delay interrupt request
1	Generation of delay interrupt requests.

Note: This bit is used in the same way as other interrupt request flags. Clear this bit in the interrupt processing routine and switch tasks accordingly.

12.4 An Explanation of Operations and Setting Procedure Examples

This section explains delay interrupt operations and the setting procedure for delay interrupts.

12.4.1 Explanation of Delay Interrupt Operations

Software can use delay interrupts to generate interrupt requests to the CPU or cancel them.

Table 12-2 lists the conditions for generating delay interrupts.

Table 12-2. Interrupt Request Generation Conditions

Interrupt request	Delay interrupt request
Interrupt request generation	Write "1" to the DLYI bit of the delayed interrupt control register (DICR).
Interrupt request enabled	None (interrupts always enabled)
Clearing an interrupt request	Write "0" to the DLYI bit of the delayed interrupt control register (DICR).

Notes:

- Delay interrupts cannot be used for DMA transfer requests.
- For information on interrupt vector numbers, see "A.3 Interrupt Vectors".
- Use an interrupt control register (ICR47) to specify the interrupt level corresponding to the interrupt vector number.For information on the setting of interrupt levels, see "10. Interrupt Controller".

12.5 Notes on Use

Note the following points about using delay interrupts.

Notes on the program

- The delay interrupt control bit can be used in the same way as other interrupt request flags. Clear this bit in the interrupt routine and switch tasks accordingly.
- Delay interrupts cannot be used for DMA transfer requests.

13. I/O Ports

This chapter explains the functions and operations of the I/O ports.

13.1 Overview

13.2 Configuration

13.3 Pins

13.4 Registers

13.5 Notes on Use

13.1 Overview

Pins of this series that are not used for the peripheral functions can be used as I/O ports.

This series is equipped with 50 I/O ports.

Overview

The I/O ports have the following features:

- Each pin can be specified as an I/O port used only as an input port or output port.
- Each pin can be specified as a pin used as an I/O port or a pin for a peripheral function.
  Also, one of the I/O modes listed below can be selected depending on the register settings:
  Table 13-1 lists the I/O modes.

Table 13-1. I/O Modes

I/O mode	Access to PDR	
Port input mode	In case of reading (except RMW instructions)	The levels of external pins are read.
	In case of reading (RMW instructions)	The PDR value is read.
	In case of writing	The written value is stored in a PDR.
Port output mode	In case of reading (except RMW instructions)	The PDR value is read.
	In case of reading (RMW instructions)	The PDR value is read.
	In case of writing	The written value is stored in a PDR and output to an external pin.
Peripheral function output mode [1]	In case of reading (except RMW instructions)	The output level from a peripheral function or the PDR value is read.
	In case of reading (RMW instructions)	The PDR value is read.
	In case of writing	The written value is stored in a PDR.

PDR: Port data register (PDR0 to PDRK)
RMW instruction: Read-modify-write instruction
[1]: The value that is read varies depending on the register settings.

- A pull-up resistor can be set for each pin.
- If Hi-Z is set to a pin with the CPU in standby mode (stop mode/watch mode/main timer mode), input is fixed at "0". However, input is not fixed at "0" for external interrupt requests whose generation is enabled and it can be used.
- A peripheral function can be assigned to any pin available for peripheral functions, if more than one pin is available, and peripheral function output from the pin can be enabled/disabled.
  However, if the peripheral function has more than one I/O, each I/O must be set to individual ports belonging to the same group.
  Example: Ch.0 multifunction serial interface settings

Serial Data Output	Serial Clock I/O	Serial Data Input	Valid Port
SOUT0 pin (Port 0)	SCK0 pin (Port 0)	SIN0 pin (Port 0)	Port 0
		SIN0_1 pin (Port 1)	Setting prohibited
	SCK0_1 pin (Port 1)	SIN0 pin (Port 0)	
		SIN0_1 pin (Port 1)	
SOUT0_1 pin (Port 1)	SCK0 pin (Port 0)	SIN0 pin (Port 0)	
		SIN0_1 pin (Port 1)	
	SCK0_1 pin (Port 1)	SIN0 pin (Port 0)	
		SIN0_1 pin (Port 1)	Port 1

13.2 Configuration

This series has the following 2 types of built-in I/O port:

- Ordinary I/O ports
- Analog input multifunction I/O ports

Overview

2 types of built-in I/O port that this series has are described below.

- Ordinary I/O ports

  These I/O ports have basic configurations in which the ports are used also for I/O of peripheral functions. Each port consists of the following blocks:

  - Port function registers (PFR0 to PFR7)
  - Port data direction registers (DDR0 to DDRK)
  - Extended port function registers (EPFR0 to EPFR34)
  - Pull-up resistor control registers (PCR0 to PCR7)
  - Port data registers (PDR0 to PDRK)

- Analog input multifunction I/O ports

  These I/O ports are used also for analog input of the 10-bit A/D converter. Each port consists of an analog input enable block and the ordinary I/O port blocks.

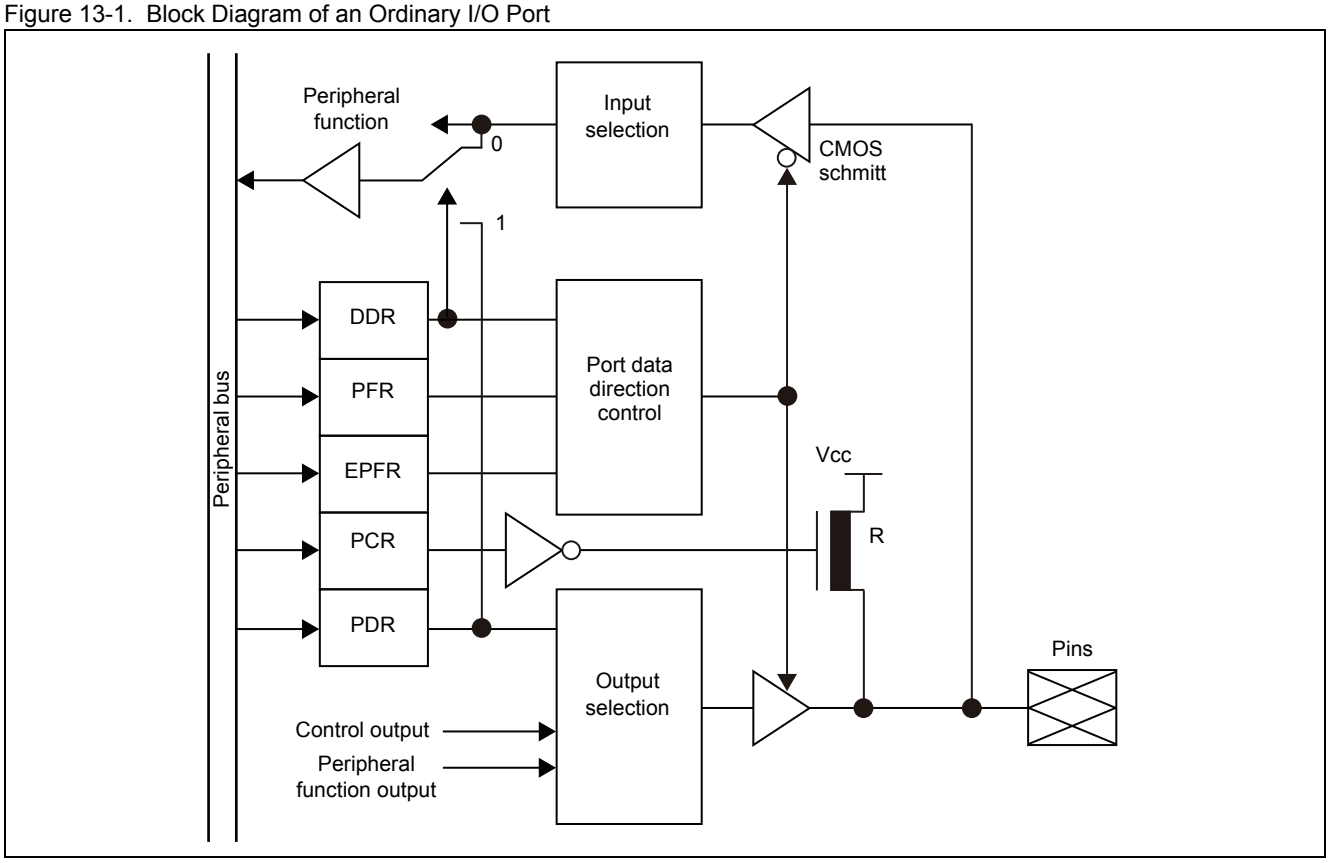
  The analog input multifunction ports are P77 to P70.

Block diagrams

- Ordinary I/O ports

  Figure 13-1 is a block diagram of an ordinary I/O port.

Figure 13-1. Block Diagram of an Ordinary I/O Port



- Port data direction registers (DDR0 to DDRK)

  These registers set the I/O directions of pins used as general-purpose ports.

  For a pin for a peripheral function, these registers set the contents read from a port data register (PDR0 to PDRK).
- Port function registers (PFR0 to PFR7)

  These registers select how to use individual pins.
- Extended port function registers (EPFR0 to EPFR34)

  These registers set the pin to which a peripheral function is assigned from among the multiple pins available for peripheral functions. Peripheral function output from such pins is enabled/disabled according to the registers.
- Pull-up resistor control registers (PCR0 to PCR7)

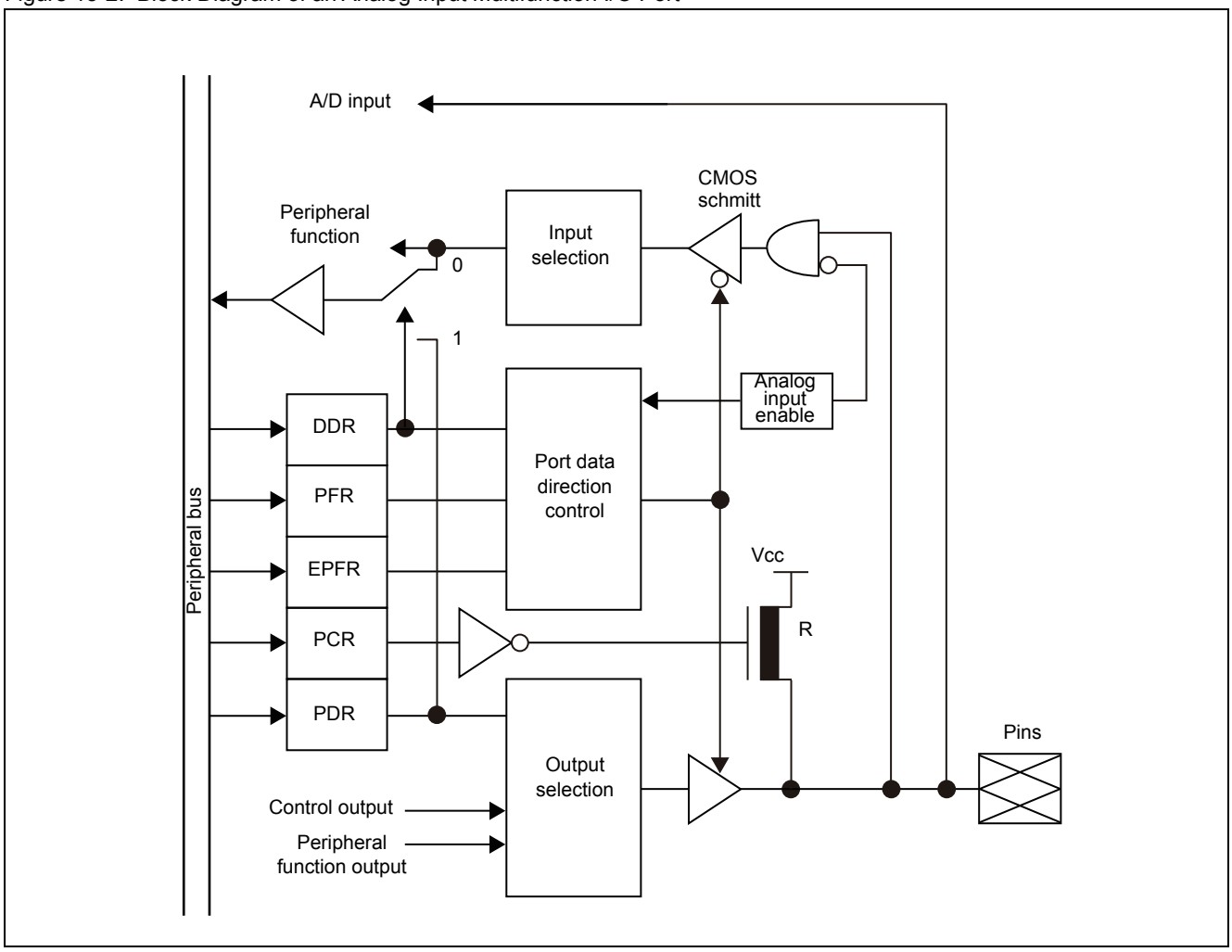
  These registers set pull-up resistors. With one register provided for each port, a pull-up resistor can be connected to each pin.
- Port data registers (PDR0 to PDRK)

  These registers store output data. The meanings of read and written values vary depending on the mode of the port.

- Analog input multifunction I/O port

  Figure 13-2 is a block diagram of an analog input multifunction I/O port.

Figure 13-2. Block Diagram of an Analog Input Multifunction I/O Port



The analog input multifunction I/O port consists of the blocks that are components of each ordinary I/O port and the analog input enable block.

This block enables analog input from pins for which input is enabled by the A/D channel enable register (ADCHE).

Notes:

- The analog input multifunction ports are P77 to P70.
- In serial write mode selected by the MD1 and MD0 pins (MD1, MD0 = 01), digital input is enabled and analog input is disabled only for P75 (AN5 pin).

Clocks

Table 13-2 lists the clocks used for I/O ports.

Table 13-2. Clocks used for I/O Ports

Clock name	Description
Operation clock	Peripheral clock (PCLK)

13.3 Pins

This section explains the pins of I/O ports.

Overview

Up to 50 I/O ports are provided, and they are categorized into port 0 to port K.

The I/O ports belonging to a port with the same suffix can be read/written at the same time.

- P00 to P07 (port 0)
- P10 to P17 (port 1)
- P20 to P27 (port 2)
- P30 to P37 (port 3)
- P50 to P57 (port 5)
- P70 to P77 (port 7)
- PK0, PK1 (port K)

13.4 Registers

This section explains the configuration and functions of the registers used for I/O ports.

List of registers for I/O ports

Table 13-3 lists the registers for I/O ports.

Table 13-3. Registers for I/O Ports

Port	Abbreviated Register Name	Register Name	Reference
Common	EPFR0 to EPFR34	Extended port function register 0 to 34	13.4.3
	ADCHE	A/D channel enable register	13.4.6
0	DDR0	Port data direction register 0	13.4.1
	PFR0	Port function register 0	13.4.2
	PCR0	Pull-up resistor control register 0	13.4.5
	PDR0	Port data register 0	13.4.4
1	DDR1	Port data direction register 1	13.4.1
	PFR1	Port function register 1	13.4.2
	PCR1	Pull-up resistor control register 1	13.4.5
	PDR1	Port data register 1	13.4.4
2	DDR2	Port data direction register 2	13.4.1
	PFR2	Port function register 2	13.4.2
	PDR2	Port data register 2	13.4.4
3	DDR3	Port data direction register 3	13.4.1
	PFR3	Port function register 3	13.4.2
	PDR3	Port data register 3	13.4.4
5	DDR5	Port data direction register 5	13.4.1
	PFR5	Port function register 5	13.4.2
	PCR5	Pull-up resistor control register 5	13.4.5
	PDR5	Port data register 5	13.4.4
7	DDR7	Port data direction register 7	13.4.1
	PFR7	Port function register 7	13.4.2
	PCR7	Pull-up resistor control register 7	13.4.5
	PDR7	Port data register 7	13.4.4
K	DDRK	Port data direction register K	13.4.1
	PDRK	Port data register K	13.4.4

13.4.1 Port Data Direction Registers (DDR0 to DDRK)

These registers set the I/O directions of pins used as general-purpose ports.

For a pin for a peripheral function, these registers set the contents read from a port data register (PDR0 to PDRK).

The meaning of a read/written value of the port data register (PDR0 to PDRK) varies depending on the setting of each bit in this port data direction register and the settings of a port function register (PFR0 to PFR7).

Figure 13-3 shows the bit configuration of the port data direction registers (DDR0 to DDRK).

Figure 13-3. Bit Configuration of the Port Data Direction Registers (DDR0 to DDRK)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
DDR0	DDR07	DDR06	DDR05	DDR04	DDR03	DDR02	DDR01	DDR00	0000 0000	R/W
DDR1	DDR17	DDR16	DDR15	DDR14	DDR13	DDR12	DDR11	DDR10	0000 0000	R/W
DDR2	DDR27	DDR26	DDR25	DDR24	DDR23	DDR22	DDR21	DDR20	0000 0000	R/W
DDR3	DDR37	DDR36	DDR35	DDR34	DDR33	DDR32	DDR31	DDR30	0000 0000	R/W
DDR5	DDR57	DDR56	DDR55	DDR54	DDR53	DDR52	DDR51	DDR50	0000 0000	R/W
DDR7	DDR77	DDR76	DDR75	DDR74	DDR73	DDR72	DDR71	DDR70	0000 0000	R/W
DDRK	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	DDRK1	DDRK0	XXXX XX00	R/W

R/W: Read/Write

X: Undefined

Each bit sets the I/O direction of the corresponding port.

Written Value	Explanation
0	Input direction
1	Output direction

The meaning of a read/written value of a port data register (PDR0 to PDRK) varies depending on the setting of each bit in one of these port data direction registers and the settings of a port function register (PFR0 to PFR7).

Table 13-4 shows the relationship between the register settings and read/written values of the port data registers (PDR0 to PDRK).

Table 13-4. Relationship Between Register Settings and rEad/written Values of the Port Data Registers (PDR0 to PDRK)

Mode	DDR	PFR	PDR	
Port input mode	0	0	In case of reading (except RMW instructions)	The output level of an external pin is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR.
Port output mode	1	0	In case of reading (except RMW instructions)	The PDR value is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR and output to an external pin.
Peripheral function output mode [1]	0	1	In case of reading (except RMW instructions)	The output level from a peripheral function is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR.
	1	1	In case of reading (except RMW instructions)	The PDR value is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR.

[1]: The functions of the output pins of external functions must be assigned to the appropriate pins by the extended port function registers (EPFR0 to EPFR34), and output from the pins must be enabled.
DDR: Port data direction register (DDR0 to DDRK)
PFR: Port function register (PFR0 to PFR7)
PDR: Port data register (PDR0 to PDRK)
RMW instruction: Read-modify-write instruction

Notes:

- The input to a peripheral function is always connected to the pin assigned by an appropriate bit in an extended port function register (EPFR0 to EPFR34). Use port input mode for input to a peripheral function.

  However, when input from the 10-bit A/D converter is enabled, input is always fixed at "0", and output from the port is always fixed at Hi-Z.

  In serial write mode selected by the MD1 and MD0 pins (MD1, MD0 = 01), digital input is enabled and analog input is disabled only for P75 (AN5 pin).
- When this device is reset, the settings of these registers are reset to the initial value (00_H), and the I/O direction of every port becomes input.
- To use PK0 and PK1 as low-speed oscillation pins, be sure to set the I/O directions of the ports to input (DDRK0 = 0, DDRK1 = 0) in port data direction register K (DDRK).

  (If PK0 and PK1 is used as a low-speed oscillation pin when the I/O direction of the related port has been set to output, the PDR value is output from the pin when low-speed oscillation is disabled.)

13.4.2 Port Function Registers (PFR0 to PFR7)

These registers select how to use individual pins.

The meaning of a read/written value of a port data register (PDR0 to PDRK) varies depending on the setting of each bit in one of these port function registers and the settings of a port data direction register (DDR0 to DDRK).

For details, see "13.4.1 Port Data Direction Registers (DDR0 to DDRK)".

Figure 13-4 shows the bit configuration of the port function registers (PFR0 to PFR7).

Figure 13-4. Bit Configuration of the Port Function Registers (PFR0 to PFR7)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PFR0	PFR07	PFR06	PFR05	PFR04	PFR03	PFR02	PFR01	PFR00	0000 0000	R/W
PFR1	PFR17	PFR16	PFR15	PFR14	PFR13	PFR12	PFR11	PFR10	0000 0000	R/W
PFR2	PFR27	PFR26	PFR25	PFR24	PFR23	PFR22	PFR21	PFR20	0000 0000	R/W
PFR3	PFR37	PFR36	PFR35	PFR34	PFR33	PFR32	PFR31	PFR30	0000 0000	R/W
PFR5	PFR57	PFR56	PFR55	PFR54	PFR53	PFR52	PFR51	PFR50	0000 0000	R/W
PFR7	PFR77	PFR76	PFR75	PFR74	PFR73	PFR72	PFR71	PFR70	0000 0000	R/W

R/W: Read/Write

X: Undefined

The port function registers specify each pin as either a pin used as a general-purpose port or a pin for the peripheral function specified by an extended port function register (EPFR0 to EPFR34).

Written Value	Explanation
0	General-purpose port
1	Peripheral function

The following function and I/O settings can be made for each pin according to the settings of bits in one of these registers and the corresponding bits in an extended port function register (EPFR0 to EPFR34):

PFR	EPFR	Function of Corresponding Pin	Output from Peripheral Functions	Input to Peripheral Functions	Port Output
0	0	Port	Disabled	Enabled	-
1	Sets the function assigned to the output pin of a peripheral function.	Output pin of a peripheral function	Enabled	Enabled	Disabled
	Sets the function not assigned to the output pin of a peripheral function	Port	Disabled	Disabled	- (depending on DDR)

PFR: Corresponding bit in a port function register (PFR0 to PFR7)
EPFR: Corresponding bit in an extended port function register (EPFR0 to EPFR34)

Notes:

- When this device is reset, the settings of these registers are reset to the initial value (00_H), and all ports are set to operate as input ports.
- If this register specifies a pin as a general-purpose port, the corresponding pin will operate as a general-purpose port even if a peripheral function has been assigned to that pin in one of the extended port function registers (EPFR0 to EPFR34).
- When analog input is enabled through the settings of the A/D channel enable register (ADCHE), input from ports and other functions is fixed at "0" regardless of the settings of these registers.
- The input to a peripheral function is always connected to the pin assigned by an appropriate bit in an extended port function register (EPFR0 to EPFR34). Use port input mode for input to a peripheral function.

  However, when input from the 10-bit A/D converter is enabled, input is always fixed at "0", and output from the port is always fixed at Hi-Z.

13.4.3 Extended Port Function Registers (EPFR0 to EPFR34)

These registers set the pin to which a function is assigned from among the multiple pins available for the function. Output from such pins is enabled/disabled according to the registers.

Figure 13-5 shows the bit configuration of the extended port function registers (EPFR0 to EPFR34).

Figure 13-5. Bit Configuration of the Extended Port Function Registers (EPFR0 to EPFR34)

	bit 7	6	5	4	3	2	1	0	Initial value
EPFR0	Undefined	Undefined	Undefined	OUT1E1	OUT1E0	Undefined	OUT0E1	OUT0E0	XXX0 0X00
EPFR1	Undefined	Undefined	Undefined	OUT3E1	OUT3E0	Undefined	OUT2E1	OUT2E0	XXX0 0X00
EPFR6	Undefined	SOUT0E1	SOUT0E0	Undefined	SCK0E1	SCK0E0	Undefined	SIN0E0	X00X 00X0
EPFR7	Undefined	Undefined	Undefined	Undefined	SOUT1E0	Undefined	SCK1E0	Undefined	XXXX 0X0X
EPFR8	Undefined	Undefined	Undefined	Undefined	SOUT2E0	Undefined	SCK2E0	Undefined	XXXX 0X0X
EPFR9	Undefined	Undefined	Undefined	Undefined	SOUT3E0	Undefined	SCK3E0	Undefined	XXXX 0X0X
EPFR10	Undefined	Undefined	Undefined	Undefined	RCIN	Undefined	Undefined	Undefined	XXXX 0XXX
EPFR14	Undefined	Undefined	Undefined	Undefined	SOUT8E0	Undefined	SCK8E0	Undefined	XXXX 0X0X
EPFR15	Undefined	Undefined	Undefined	Undefined	SOUT9E0	Undefined	SCK9E0	Undefined	XXXX 0X0X
EPFR16	Undefined	Undefined	Undefined	Undefined	SOUT10E0	Undefined	SCK10E0	Undefined	XXXX 0X0X
EPFR17	Undefined	Undefined	Undefined	Undefined	SOUT11E0	Undefined	SCK11E0	Undefined	XXXX 0X0X
EPFR19	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	XAE	XXXX XXX1
EPFR20	Undefined	Undefined	Undefined	TIOA1E0	Undefined	Undefined	TIOA0E0	Undefined	XXX0 XX0X
EPFR21	Undefined	Undefined	Undefined	TIOA3E0	Undefined	Undefined	TIOA2E0	Undefined	XXX0 XX0X
EPFR22	Undefined	Undefined	Undefined	TIOA5E0	Undefined	Undefined	TIOA4E0	Undefined	XXX0 XX0X
EPFR23	Undefined	Undefined	Undefined	TIOA7E0	Undefined	Undefined	TIOA6E0	Undefined	XXX0 XX0X
EPFR33	Undefined	Undefined	Undefined	TMO1E0	Undefined	Undefined	TMO0E0	Undefined	XXX0 XX0X
EPFR34	Undefined	Undefined	TMO2E0	Undefined	Undefined	Undefined	Undefined	Undefined	XX0X XXXX

Attribute: R/W (Read/Write) for all the bits

X: Undefined

Notes:

- The pins that are specified as general-purpose ports in settings of the port function registers (PFR0 to PFR7) are treated as general-purpose I/O ports regardless of the settings of these registers.
- When analog input is enabled through the settings of the A/D channel enable register (ADCHE), input from ports is fixed at "0" regardless of the settings of these registers or port function registers (PFR0 to PFR7).
- A single pin cannot be used as an output pin for multiple peripheral functions. Also, a single output function cannot be assigned to multiple pins.
- A single pin can be used as an input pin for multiple peripheral functions. However, a single input function cannot be assigned to multiple pins.
- If multiple functions are assigned to one pin, the order of priority is as follows:
  1. X0A/X1A
  2. Multifunction serial interface
  3. Base timer
  4. 16-bit reload timer
  5. 32-bit output compare
- The input to a peripheral function is always connected to the pin assigned by an appropriate bit in an extended port function register (EPFR0 to EPFR34). Use port input mode for input to a peripheral function.

  However, when input from the 10-bit A/D converter is enabled, input is fixed at "0".
- Before changing the pin to which peripheral function output is assigned through the settings of this register, make the following settings:
  - Set port input mode for the pin to which the function is currently assigned and the pin to which it will be assigned.
  - Disable the assigned peripheral function.
- Before changing the pin to which a peripheral function input is assigned through the settings of this register, disable the assigned peripheral function.

- Extended port function register 0 (EPFR0) to extended port function register 1 (EPFR1)

  [bit4, bit3, bit1, bit0]: OUTxE1, OUTxE0 (Output compare output pin select bits)

  2 output pins for 32-bit output compare are provided for each channel.

  These bits select the pins used by ch.0 to ch.3 for 32-bit output compare. The OUT0E1 to OUT0E0 bits correspond to ch.0, the OUT1E1 to OUT1E0 bits correspond to ch.1,..., and the OUT3E1 to OUT3E0 bits correspond to ch.3.

OUTxE1	OUTxE0	Port Number	Pin Name
0	0	-	Output disabled
	1	Port 0	OUTx pin
1	0	Port 1	OUTx_1 pin
	1	-	Setting prohibited

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.

- Extended port function register 6 (EPFR6)

[bit6, bit5]: SOUT0E1, SOUT0E0 (Serial interface ch.0 serial data pin select bits)

These bits select one pin from the SOUT0 and SOUT0_1 pins to assign the serial data output function of multifunction serial interface ch.0 to the pin.

SOUT0E1	SOUT0E0	Port Number	Pin Name
0	0	-	Output disabled (Input: SOUT0 pin (Port 0))
	1	Port 0	SOUT0 pin
1	0	Port 1	SOUT0_1 pin
	1	-	Setting prohibited

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.
- The pins selected by the following bits must be assigned to the same port number:
  - SOUT0E1, SOUT0E0 (serial data output pins)
  - SCK0E1, SCK0E0 (serial clock I/O pins)
  - SIN0E0 (serial data input pin)
- The serial data pins operate as input pins according to peripheral function settings. The input of a peripheral function is always connected to the selected pin, and if these bits are set to "00", the input is connected to the SOUT0 pin (port 0).

[bit3, bit2]: SCK0E1, SCK0E0 (Serial interface ch.0 serial clock pin select bits)

These bits select one pin from the SCK0 and SCK0_1 pins to assign the serial clock I/O function of multifunction serial interface ch.0 to the pin.

SCK0E1	SCK0E0	Port Number	Pin Name
0	0	-	Output disabled (Input: SCK0 pin (Port 0))
	1	Port 0	SCK0 pin
1	0	Port 1	SCK0_1 pin
	1	-	Setting prohibited

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.
- The pins selected by the following bits must be assigned to the same port number:
  - SOUT0E1, SOUT0E0 (serial data output pins)
  - SCK0E1, SCK0E0 (serial clock I/O pins)
  - SIN0E0 (serial data input pin)
- The input of a peripheral function is always connected to the selected pin, and if these bits are set to "00", the input is connected to the SCK0 pin (port 0).

[bit0]: SIN0E0 (Serial interface ch.0 serial data input select bits)

These bits select one pin from the SIN0 and SIN0_1 pins to assign the serial data input function of multifunction serial interface ch.0 to the pin.

SIN0E0	Port Number	Pin Name
0	Port 0	SIN0 pin
1	Port 1	SIN0_1 pin

Note:

The pins selected by the following bits must be assigned to the same port number:

- SOUT0E1, SOUT0E0 (serial data output pins)
- SCK0E1, SCK0E0 (serial clock I/O pins)
- SIN0E0 (serial data input pin)

- Extended port function register 7 (EPFR7) to extended port function register 9 (EPFR9)

[bit3]: SOUTxE0 (Serial interface ch.1 to ch.3 serial data pin select bits)

These bits select whether to enable the serial data output pin of each channel in multifunction serial interface ch.1 to ch.3. The SOUT1E0 bit correspond to ch.1, the SOUT2E0 bit correspond to ch.2, the SOUT3E0 bit correspond to ch.3.

SOUTxE0	Port Number	Pin name
0	-	Output disabled (Input: SOUTx pin (Port 0))
1	Port 0	SOUTx pin

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.
- Serial data pins operate as input pins according to peripheral function settings. The input of a peripheral function is always connected to the selected pin, and if these bits are set to "0", the input is connected to the SOUTx pin (port 0).

[bit1]: SCKxE0 (Serial interface ch.1 to ch.3 serial clock pin select bits)

These bits select whether to enable the serial clock I/O pin of each channel in multifunction serial interface ch.1 to ch.3. The SCK1E0 bit correspond to ch.1, the SCK2E0 bit correspond to ch.2, the SCK3E0 bit correspond to ch.3.

SCKxE0	Port Number	Pin name
0	-	Output disabled (Input: SCKx pin (Port 0))
1	Port 0	SCKx pin

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.
- The input of the serial clock is always connected to the selected pin, and if these bits are set to "0", the input is connected to the SCKx pin (port 0).

- Extended port function register 10 (EPFR10)

  [bit3]: RCIN (HDMI-CEC/ remote control I/O [RCIN/RCIN_1] function select bit)

  Selects the remote control input/output function.

RCIN	Description
0	Input/output disabled
1	RCIN/RCIN_1 input/output

Notes:

- The corresponding pins can be used as output pins for other functions when input/output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when input/output has been disabled by these bits, the pins can be used as general-purpose port.
- When HDMI-CEC/remote control reception is used, select either PFR54 or PFR23 for RCIN/RCIN_1 pin, and set "1" to this bit.
  Do not set "1" to PFR54 and PFR23 at the same time.

- Extended port function register 14 (EPFR14) to extended port function register 17 (EPFR17)

  [bit3]: SOUTxE0 (Serial interface ch.8 to ch.11 serial data pin select bits)

  These bits select whether to enable the serial data output pin of each channel in multifunction serial interface ch.8 to ch.11. The SOUT8E0 bit corresponds to ch.8, the SOUT9E0 bit corresponds to ch.9,... and the SOUT11E0 bit corresponds to ch.11.

SOUTxE0	Port Number	Pin Name
0	-	Output disabled (Input: SOUTx pin (Port 0))
1	Port 0	SOUTx pin

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.
- Serial data pins operate as input pins according to peripheral function settings. The input of a peripheral function is always connected to the selected pin, and if these bits are set to "0", the input is connected to the SOUTx pin (port 0).

[bit1]: SCKxE0 (Serial interface ch.8 to ch.11 serial clock pin select bits)

These bits select whether to enable the serial clock I/O pin of each channel in multifunction serial interface ch.8 to ch.11. The SCK8E0 bit corresponds to ch.8, the SCK9E0 bit corresponds to ch.9,... and the SCK11E0 bit corresponds to ch.11.

SCKxE0	Port Number	Pin Name
0	-	Output disabled (Input: SCKx pin (Port 0))
1	Port 0	SCKx pin

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.
- The input of the serial clock is always connected to the selected pin, and if these bits are set to "0", the input is connected to the SCKx pin (port 0).

- Extended port function register 19 (EPFR19)

  [bit0]: XAE (Clock oscillation I/O pin enable bit)

  This bit cuts off port input when the low-speed clock oscillation function is enabled. Always set XAE = 1 when the low-speed clock oscillation function is enabled.

Written Value	Explanation
0	Enables port input.
1	Disables port input.

Note: These pins can be used as general-purpose port when the low-speed oscillation function has been disabled by this bit.

- Extended port function register 20 (EPFR20) to extended port function register 23 (EPFR23)

  [bit4, bit1]: TIOAxE0 (Base timer ch.0 to ch.7 pin select bits)

  These bits select whether to enable the output pin of each channel in base timer ch.0 to ch.7. The TIOA0E0 bit correspond to ch.0, the TIOA1E0 bit correspond to ch.1,... and the TIOA7E0 bit correspond to ch.7.

TIOAxE0	Port Number	Pin name
0	-	Output disabled (Odd-numbered channel input: TIOAx pin (Port 0))
1	Port 0	TIOAx pin

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.
- The base timer output pins (TIOAx pins) of the odd-numbered channels (TIOAx pin) operate as input pins according peripheral function settings. The input of a peripheral function is always connected to the selected pin. If these bits are set to "0", the input is connected to the TIOAx pin (port 0).

- Extended port function register 33 (EPFR33)

  [bit4, bit1]: TMOxE0 (Reload timer ch.0 to ch.1 output pin select bits)

  These bits select one of the pins as the pin used by each of 16-bit reload timer ch.0 and ch.1. The TMO0E0 bit correspond to ch.0, and the TMO1E0 bit correspond to ch.1.

TMOxE0	Port Number	Pin Name
0	-	Output disabled
1	Port 0	TMOx pin

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.

- Extended port function register 34 (EPFR34)

  [bit5]: TMO2E0 (Reload timer ch.2 output pin select bits)

  This bit selects one of the pins as the pin used by 16-bit reload timer ch.2.

TMO2E0	Port Number	Pin Name
0	-	Output disabled
1	Port 0	TMO2 pin

Notes:

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as general-purpose port.

13.4.4 Port Data Registers (PDR0 to PDRK)

These registers store I/O data.

The values read from or written to these registers vary depending on the settings of a port data direction register (DDR0 to DDRK) and port function register (PFR0 to PFR7). For details of a read value or written value, see "13.4.1 Port Data Direction Registers (DDR0 to DDRK)".

Figure 13-6 shows the bit configuration of the port data registers (PDR0 to PDRK).

Figure 13-6. Bit Configuration of the Port Data Registers (PDR0 to PDRK)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PDR0	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00	XXXX XXXX	R/W
PDR1	PDR17	PDR16	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	XXXX XXXX	R/W
PDR2	PDR27	PDR26	PDR25	PDR24	PDR23	PDR22	PDR21	PDR20	XXXX XXXX	R/W
PDR3	PDR37	PDR36	PDR35	PDR34	PDR33	PDR32	PDR31	PDR30	XXXX XXXX	R/W
PDR5	PDR57	PDR56	PDR55	PDR54	PDR53	PDR52	PDR51	PDR50	XXXX XXXX	R/W
PDR7	PDR77	PDR76	PDR75	PDR74	PDR73	PDR72	PDR71	PDR70	XXXX XXXX	R/W
PDRK	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	PDRK1	PDRK0	XXXX XXXX	R/W

R/W: Read/Write

X: Undefined

Notes:

- If these registers are read with a read-modify-write instruction, the value of these registers is read regardless of the settings of the following registers:
  - Port data direction registers (DDR0 to DDRK)
  - Port function registers (PFR0 to PFR7)
- The value of these registers is not initialized even when this device is reset.

13.4.5 Pull-up Resistor Control Registers (PCR0 to PCR7)

These registers set pull-up resistors. One bit is provided for each of the pins for which pull-up resistors can be set, and a pull-up resistor can be set in the corresponding pin by writing "1" to the bit corresponding to the pin.

Figure 13-7 shows the bit configuration of the pull-up resistor control registers (PCR0 to PCR7).

Figure 13-7. Bit Configuration of the Pull-up Resistor Control Registers (PCR0 to PCR7)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PCR0	PCR07	PCR06	PCR05	PCR04	PCR03	PCR02	PCR01	PCR00	0000 0000	R/W
PCR1	PCR17	PCR16	PCR15	PCR14	PCR13	PCR12	PCR11	PCR10	0000 0000	R/W
PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	0000 0000	R/W
PCR7	PCR77	PCR76	PCR75	PCR74	PCR73	PCR72	PCR71	PCR70	0000 0000	R/W

R/W: Read/Write

X: Undefined

Each bit in the pull-up resistor control registers specifies whether a pull-up resistor is set for the assigned pin.

When a pull-up this register is set, the pull-up resistor is connected to the pin.

Written Value	Explanation
0	The pull-up resistor is not set.
1	The pull-up resistor is set.

Note:

Pull-up resistors are not set in the following cases regardless of the settings of these registers:

- In port output (in peripheral function output)
- In stop mode (with Hi-Z selected)

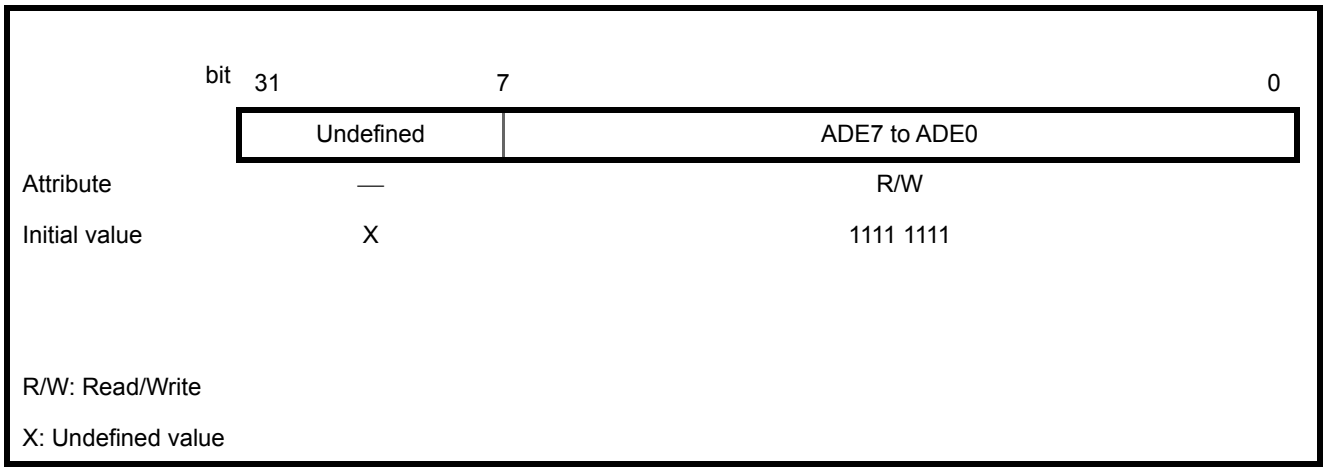
13.4.6 A/D Channel Enable Register (ADCHE)

This register specifies whether to input analog signals from the AN0 to AN7 pins.

One bit is provided for each of the pins for which A/D analog input can be set, and A/D analog input can be enabled for the corresponding pin by writing "1" to the bit corresponding to the pin.

Figure 13-8 shows the bit configuration of the A/D channel enable register (ADCHE).

Figure 13-8. Bit configuration of the A/D channel enable register (ADCHE)



[bit31 to bit8]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit7 to bit0]: ADE7 to ADE0 (Analog input enable bits)

These bits enables/disables analog signal input from the pin corresponding to the bit.

Written Value	Explanation
0	Disables analog signal input.
1	Enables analog signal input.

The ADE7 bit corresponds to ch.7, the ADE6 bit corresponds to ch.6, the ADE5 bit corresponds to ch.5, ..., the ADE1 bit corresponds to ch.1, and the ADE0 bit corresponds to ch.0.

Notes:

- To use any of the AN0 to AN7 pins as analog signal input pins of the 10-bit A/D converter, be sure to write "1" to the bits corresponding to the channels.
- When analog input is enabled through the settings of this register, input from ports and peripheral functions is fixed at "0" and output to them is fixed at Hi-Z regardless of the settings of the port function registers (PFR0 to PFR7) or extended port function registers (EPFR0 to EPFR34).

13.5 Notes on Use

Note the following points about using I/O ports:

- The order of priority of registers is as follows:
  1. A/D channel enable register (ADCHE)
  2. Port function registers (PFR0 to PFR7)
  3. Extended port function registers (EPFR0 to EPFR34)

  If settings are inconsistent, the setting with the higher order of priority is used.
- When analog input is enabled by the A/D channel enable register (ADCHE), input from ports is fixed at "0" and output from ports is fixed at Hi-Z.
- If multiple functions are assigned to one pin, the order of priority is as follows:
  1. X0A/X1A
  2. Multifunction serial interface
  3. Base timer
  4. 16-bit reload timer
  5. 32-bit output compare
- A single pin cannot be used as an output pin for multiple peripheral functions. Also, a single output function cannot be assigned to multiple pins.
- A single pin can be used as an input pin for multiple peripheral functions. However, a single input function cannot be assigned to multiple pins.
- If Hi-Z is set to a pin in standby mode (stop mode/watch mode/main timer mode), input is fixed at "0". However, input is not fixed at "0" for external interrupt requests whose generation is enabled and it can be used.
- Before changing the pin to which a peripheral function output is assigned, set port input mode for the relevant pins (the pin to which the function is currently assigned and the pin to which it will be assigned) and disable the assigned peripheral function.
- Before changing the pin to which a peripheral function input is assigned, disable the assigned peripheral function.
- To use PK0 and PK1 as low-speed oscillation pins, set the I/O directions of the ports to input (DDRK0 = 0, DDRK1 = 0) in port data direction register K (DDRK).
- The pin to which peripheral functions are assigned can be set, if the peripheral functions can be assigned to more than one pin, and peripheral function output from the pin can be enabled/disabled.

  However, if the peripheral function has more than one I/O, each I/O must be set to individual ports belonging to the same group.

Example: Ch.0 multifunction serial interface settings

Serial Data Output	Serial Clock I/O	Serial Data Input	Effective port
SOUT0 pin (Port 0)	SCK0 pin (Port 0)	SIN0 pin (Port 0)	Port 0
		SIN0_1 pin (Port 1)	Setting prohibited
	SCK0_1 pin (Port 1)	SIN0 pin (Port 0)	
		SIN0_1 pin (Port 1)	
SOUT0_1 pin (Port 1)	SCK0 pin (Port 0)	SIN0 pin (Port 0)	
		SIN0_1 pin (Port 1)	
	SCK0_1 pin (Port 1)	SIN0 pin (Port 0)	
		SIN0_1 pin (Port 1)	Port 1

14. External Interrupt Controllers

This chapter explains the functions and operations of external interrupt controllers.

14.1 Overview

14.2 Configuration

14.3 Pins

14.4 Registers

14.5 Explanation of Operations and Setting Procedure Examples

14.1 Overview

The external interrupt controllers detect edges/levels in external interrupt signals, and they control external interrupt requests.

This series has 16 built-in signal input pins for external interrupts.

Overview

An external interrupt controller generates an external interrupt request when it detects a preset edge/level in an external interrupt signal.

The edge/level to be detected can be selected from the following 4 types:

- "H" level
- "L" level
- Rising edge
- Falling edge

Also, external interrupt requests can be used for a return from sleep mode or standby mode (watch mode or stop mode).

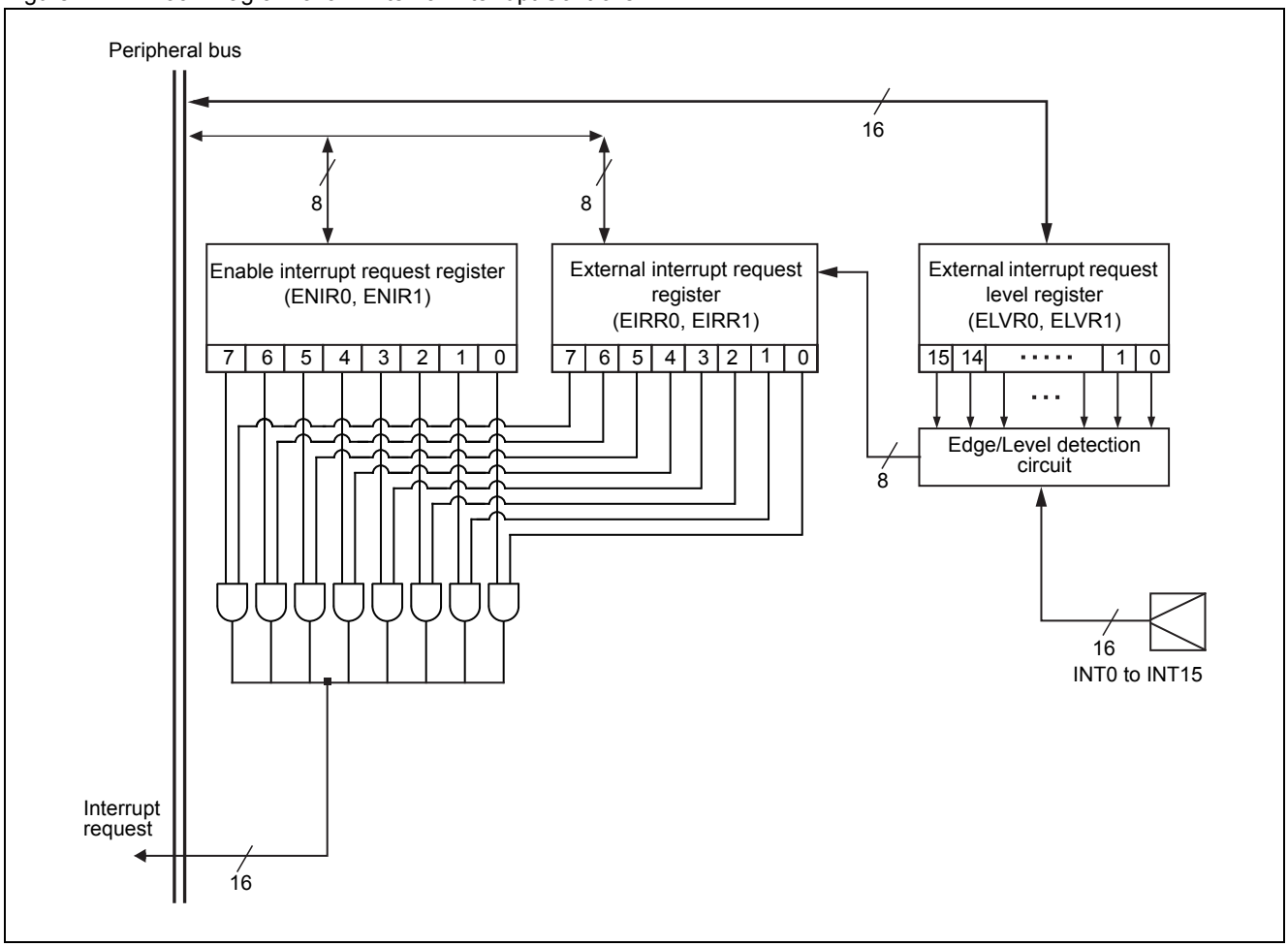
14.2 Configuration

This section shows the configuration of an external interrupt controller.

Block diagram of an external interrupt controller

Figure 14-1 is a block diagram of an external interrupt controller.

Figure 14-1. Block Diagram of an External Interrupt Controller



- External interrupt request level register (ELVR0, ELVR1)

  This register sets the edge/level used to determine whether a signal input to the INT0 to INT15 pins is for an external interrupt request.
- External interrupt request register (EIRR0, EIRR1)

  This register maintains the states of interrupt sources (indicating which pins have generated external interrupt requests).
- Enable interrupt request register (ENIR0, ENIR1)

  This register specifies whether external interrupt requests are enabled/disabled.
- Edge/Level detection circuit

  This circuit detects edges/levels in signals input to the INT0 to INT15 pins.

Clocks

Table 14-1 lists the clock used by the external interrupt controllers.

Table 14-1. Clock used by the external interrupt controllers

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

14.3 Pins

This section explains the pins of the external interrupt controllers.

Overview

The external interrupt controllers have the following pins:

- INT0 to INT15 pins

  These are external interrupt signal input pins.

  These pins are multiplexed pins. For details of using the INT0 to INT15 pins of the external interrupt controllers, see "2.4 Setting Method for Pins".

14.4 Registers

This section explains the configurations and functions of the registers for the external interrupt controllers.

List of registers for the external interrupt controllers

Table 14-2 lists the registers for the external interrupt controllers.

Table 14-2. Registers for the External Interrupt Controllers

Channel	Abbreviated Register Name	Register Name	Reference
Common	ELVR0	External interrupt request level register 0	14.4.1
	EIRR0	External interrupt request register 0	14.4.2
	ENIR0	Enable interrupt request register 0	14.4.3
	ELVR1	External interrupt request level register 1	14.4.1
	EIRR1	External interrupt request register 1	14.4.2
	ENIR1	Enable interrupt request register 1	14.4.3

14.4.1 External Interrupt Request Level Registers (ELVR0, ELVR1)

These registers set the edges/levels to be detected for external interrupt requests.

Figure 14-2 shows the bit configuration of the external interrupt request level registers (ELVR0, ELVR1).

Figure 14-2. Bit Configuration of the External Interrupt Request Level Registers (ELVR0, ELVR1)

External interrupt request level register 0 (ELVR0)

bit	15	14	13	12	11	10	9	8
	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4
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
	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
Initial value	0	0	0	0	0	0	0	0

External interrupt request level register 1 (ELVR1)

bit	15	14	13	12	11	10	9	8
	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12
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
	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
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

LB15 to LB0, LA15 to LA0 (Detection condition selection bits)

These bits select the edges/levels to be detected in signals for external interrupt requests. An external interrupt request is recognized upon detection of the edge/level selected by one of these bits.

The LB0 to LB15 bits correspond to the INT0 to INT15 bits, and the LA0 to LA15 bits similarly correspond to the INT0 to INT15 bits. For example, the INT0 pin is set with the LB0 and LA0 bits.

LB15 to LB0	LA15 to LA0	Explanation
0	0	"L" level detection
0	1	"H" level detection
1	0	Rising edge detection
1	1	Falling edge detection

To use an external interrupt request to return from standby mode, see "14.5.2 Return from Standby Mode".

Notes:

- For detection of an edge/level specified by these bits, the pulse width of the signal must be 3T or higher (T: Peripheral clock (PCLK) period). If a signal with a narrower pulse width is input, this device may not operate correctly.
- While "L" level detection/"H" level detection is set as the detection condition, the state of an interrupt source is maintained in the external interrupt request registers (EIRR0, EIRR1) even if the corresponding external interrupt request is canceled. Therefore, the external interrupt request remains at the interrupt controller, to which it has been output. To cancel the external interrupt request output to the interrupt controller, set "0" in the corresponding bit in the external interrupt request register (EIRR0, EIRR1).

  However, even when the external interrupt request register (EIRR0, EIRR1) is cleared, the external interrupt request remains as is while any signals at the effective level are input from the INT0 to INT15 pins.

  For diagrams illustrating operations that maintain the state of an interrupt source or clear an interrupt source, see "Canceling an external interrupt request" of "14.5 Explanation of Operations and Setting Procedure Examples".
- If the detection condition is changed by rewriting these bits, an incorrect interrupt source may be generated. To prevent incorrect interrupt sources from being generated when the detection condition has been changed, perform the following operations:
  1. Read the external interrupt request level register (ELVR0, ELVR1).
  2. Write "0" in the external interrupt request register (EIRR0, EIRR1) to clear the interrupt source.

14.4.2 External Interrupt Request Registers (EIRR0, EIRR1)

These registers maintain the states of interrupt sources of external interrupt requests (indicating which pins have generated the external interrupt requests).

Figure 14-3 shows the bit configuration of the external interrupt request registers (EIRR0, EIRR1).

Figure 14-3. Bit Configuration of the External Interrupt Request Registers (EIRR0, EIRR1)

External interrupt request register 0 (EIRR0)

bit	7	6	5	4	3	2	1	0
	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
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

External interrupt request register 1 (EIRR1)

bit	7	6	5	4	3	2	1	0
	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8
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

R/W: Read/Write

ER15 to ER0 (External interrupt request flag bits)

These bits indicate that external interrupt requests have been detected.

The ER0 to ER15 bits correspond to the INT0 to INT15 pins. For example, the ER0 bit is used to detect external interrupt requests from the INT0 pin, and the ER15 bit is used to detect external interrupt requests from the INT15 pin.

An external interrupt request is generated when "1" is set in any of the EN0 to EN15 bits of an enable interrupt request register (ENIR0, ENIR1) and the corresponding bit among the ER0 to ER15 bits becomes "1".

ER15 to ER0	In Case of Reading	In Case of Writing
0	No external interrupt request has been detected.	The interrupt source is cleared.
1	An external interrupt request has been detected.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- As long as a signal at the effective level is being input from any of the INT0 to INT15 pins when "L" level detection/"H" level detection has been set as the detection condition by an external interrupt request level register (ELVR0, ELVR1), "1" is set in the corresponding bit among the ER15 to ER0 bits even after the bit is cleared.

14.4.3 Enable Interrupt Request Registers (ENIR0, ENIR1)

These registers enable/disable external interrupt requests.

Figure 14-4 shows the bit configuration of the enable interrupt request registers (ENIR0, ENIR1).

Figure 14-4. Bit Configuration of the Enable Interrupt Request Registers (ENIR0, ENIR1)

Enable interrupt request register 0 (ENIR0)

bit	7	6	5	4	3	2	1	0
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
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

Enable interrupt request register 1 (ENIR1)

bit	7	6	5	4	3	2	1	0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
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

R/W: Read/Write

EN15 to EN0 (Interrupt enable bits)

These bits enable/disable external interrupts.

Each of the EN0 to EN15 bits corresponds to the respective bits of the external interrupt request registers (EIRR0, EIRR1).

Written Value	Explanation
0	Disables generation of external interrupt requests. The states of interrupt sources are maintained, but external interrupt requests are not output.
1	Enables generation of external interrupt requests. External interrupt requests are output.

14.5 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the external interrupt controllers and provides examples of setting procedures.

14.5.1 Operations of the External Interrupt Controllers

Overview

If external interrupts are enabled, an external interrupt controller outputs an external interrupt request when it detects a preset edge/level in a signal input to an external signal input pin.

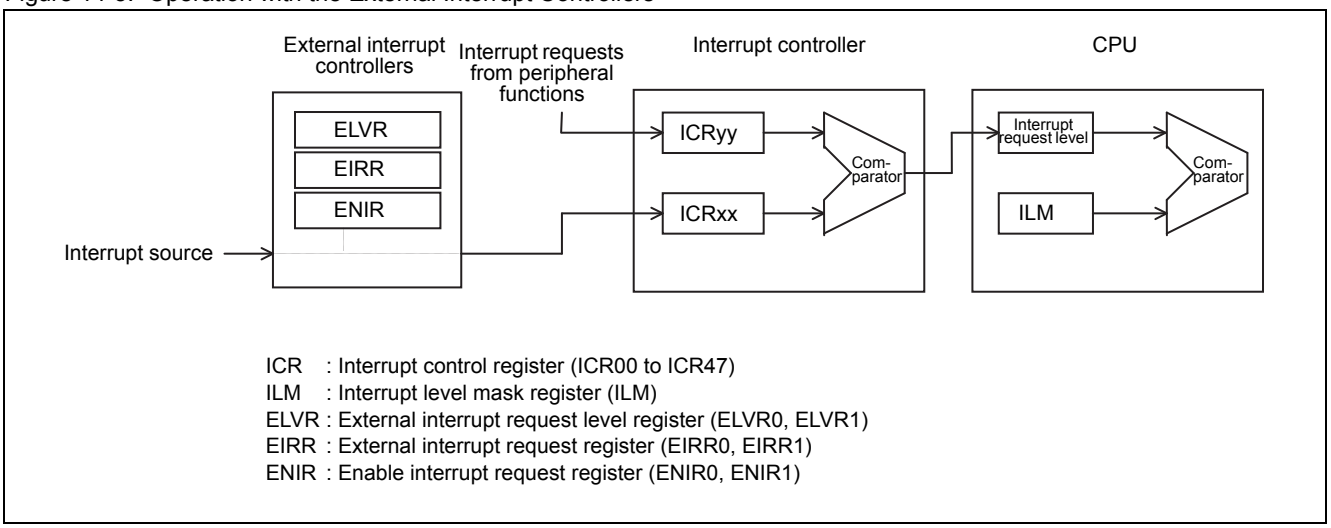
The edge/level to be detected can be selected from the following 4 types:

- "H" level
- "L" level
- Rising edge (Only when return from standby mode "L" level detection at the INT0 to INT7 pins, and rising edge detection at the INT8 to INT15 pins)
- Falling edge (Only when return from standby mode "H" level detection at the INT0 to INT7 pins, and falling edge detection at the INT8 to INT15 pins)

If an interrupt request from another peripheral device is generated at the same time, the interrupt controller determines their order of priority. An external interrupt is generated for the external interrupt request that has the higher priority.

Figure 14-5 shows operation with the external interrupt controllers.

Figure 14-5. Operation with the External Interrupt Controllers



Setting procedure

To set an external interrupt, follow the procedure below.

1. Disable external interrupts by using an enable interrupt request register (ENIR0, ENIR1).
2. Change the detection condition (effective edge /level) by using an external interrupt request level register (ELVR0, ELVR1).
3. Read the external interrupt request level register (ELVR0, ELVR1).
4. Clear interrupt sources by using an external interrupt request register (EIRR0, EIRR1).
5. Enable external interrupts by using the enable interrupt request register (ENIR0, ENIR1).

Notes:

- Before making settings for the external interrupt controller, disable external interrupts by using an enable interrupt request register (ENIR0, ENIR1).
- Before enabling output of external interrupt requests, clear interrupt sources by using an external interrupt request register (EIRR0, EIRR1).

Control operations

Each external interrupt controller issues external interrupt requests to the interrupt controller in the following sequence:

1. The external interrupt controller detects the edge/level specified by an external interrupt request level register (ELVR0, ELVR1) in a signal input to any of the INT0 to INT15 pins.
2. The external interrupt controller determines whether external interrupts are enabled by checking the enable interrupt request registers (ENIR0, ENIR1).
3. If external interrupts are enabled, the external interrupt controller outputs an external interrupt request to the interrupt controller.

Canceling an external interrupt request

While "L" level detection/"H" level detection is set as the detection condition for external interrupts, the state of an interrupt source is maintained in the external interrupt request registers (EIRR0, EIRR1) even if the corresponding external interrupt request is canceled. Therefore, the external interrupt remains at the interrupt controller, to which a request for it has been output.

To cancel the external interrupt request output to the interrupt controller, set "0" in the corresponding bit in an external interrupt request register (EIRR0, EIRR1). This operation clears the interrupt source, and the external interrupt request is canceled.

However, even when the external interrupt request register (EIRR0, EIRR1) is cleared, the external interrupt remains at the interrupt controller, to which for a request it has been output, while any signals at the effective level are input from the INT0 to INT15 pins.

Figure 14-6 shows the state of an interrupt source being maintained, and shows the clearing of an interrupt source.

Figure 14-6. Maintaining the State of an Interrupt Source

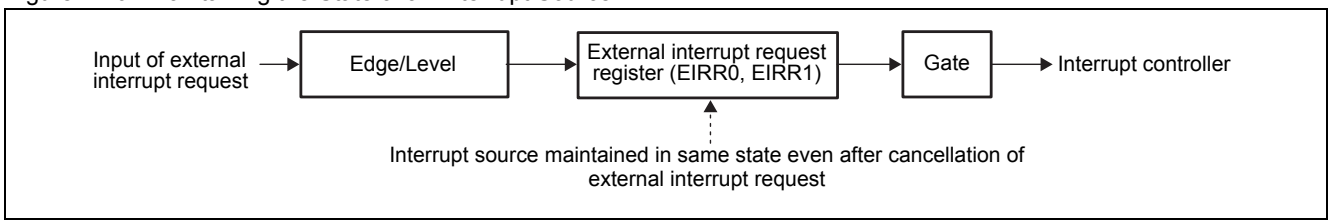
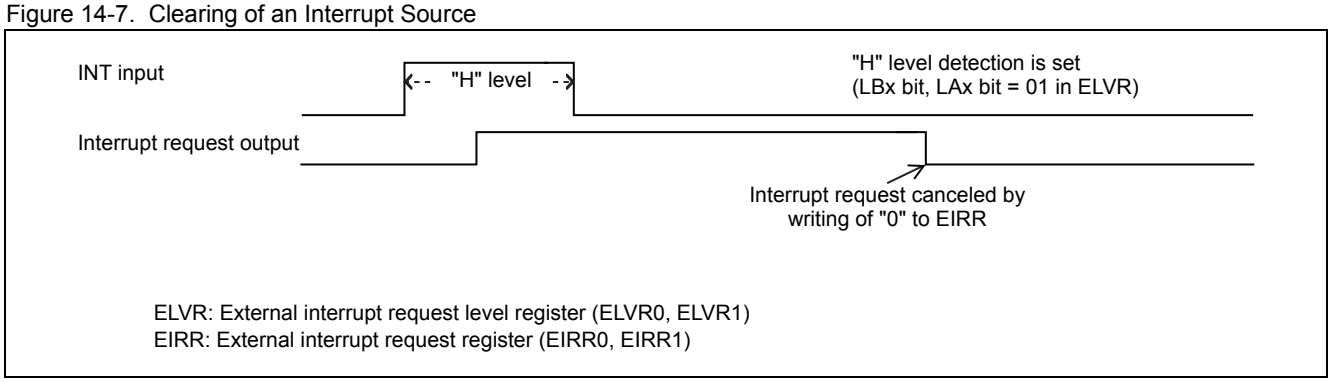


Figure 14-7. Clearing of an Interrupt Source



14.5.2 Return from Standby Mode

Overview

External interrupt requests can be used for a return from standby mode (watch mode or stop mode). A signal already input to any of the INT0 to INT15 pins in standby mode in asynchronous input can be used for a return from standby mode.

Settings

Before a transition to standby mode, the following setting for the INT0 to INT15 pins must be made with the enable interrupt request registers (ENIR0, ENIR1):

- Pins used for the return from standby mode: Enable interrupt request output.
- Pins not used for the return from standby mode: Disable interrupt request output.

Return operation

This device returns from standby mode when the effective level is detected in a signal input to the INT0 to INT15 pins in standby mode.

Table 14-3 shows the relationship between external interrupt request detection conditions and the levels for returning from standby mode.

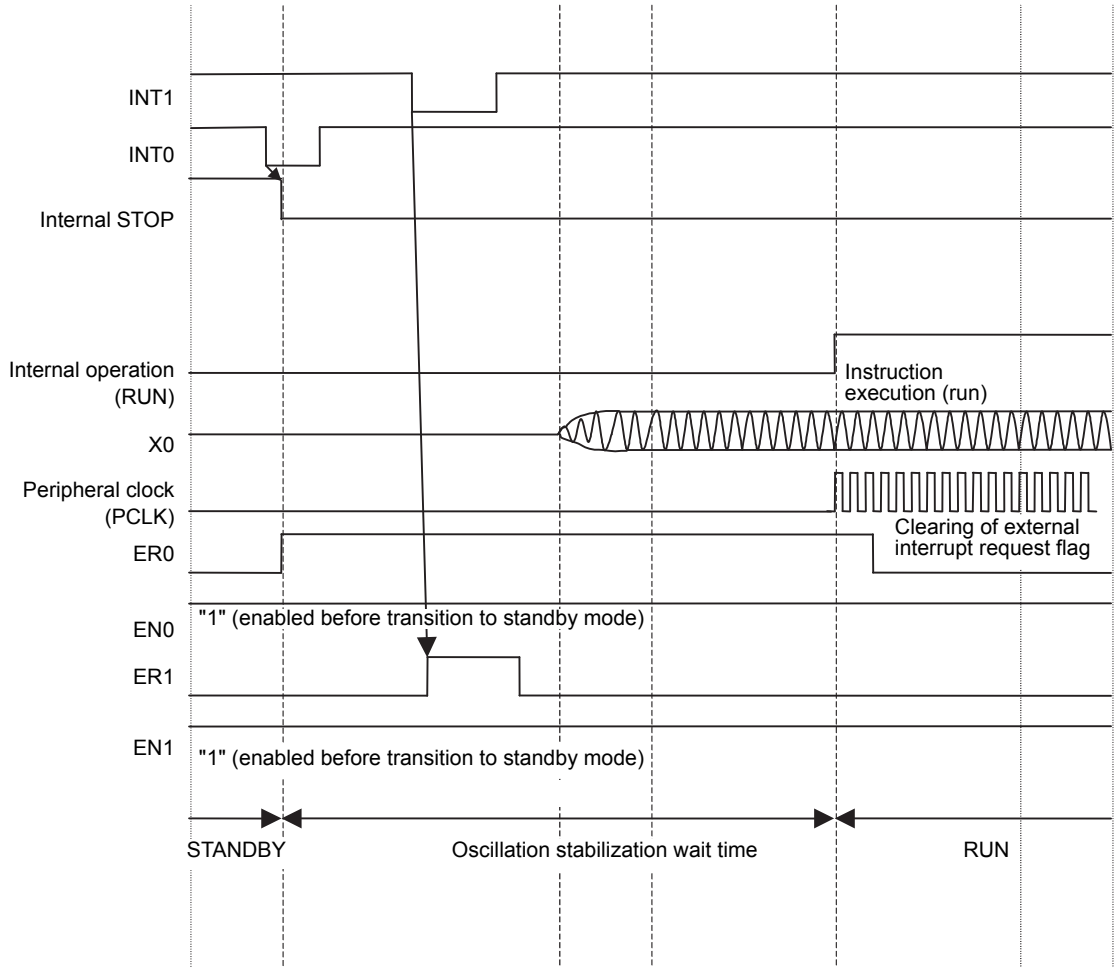
Table 14-3. Relationship Between External Interrupt Request Detection Conditions and the Levels for Returning from Standby Mode

Detection Condition	LB15 to LB0	LA15 to LA0	Level for Returning from Standby Mode
"L" level detection	0	0	"L" level detection
"H" level detection	0	1	"H" level detection
Rising edge detection	1	0	"L" level detection at the INT0 to INT7 pins, and rising edge detection at the INT8 to INT15 pins
Falling edge detection	1	1	"H" level detection at the INT0 to INT7 pins, and falling edge detection at the INT8 to INT15 pins

After this device returns from standby mode, other external interrupt requests cannot be recognized until the oscillation stabilization wait time has elapsed. To output an external interrupt request after this device returns from standby mode, input an external interrupt request signal after the oscillation stabilization wait time has elapsed.

Figure 14-8 shows an example of operation at the time of return from standby mode, where the INT0 and INT1 pins are used.

Figure 14-8. Operation when returning from standby mode



ER1, ER0 : ER1 and ER0 bits of external interrupt request register 0 (EIRR0)
EN1, EN0 : EN1 and EN0 bits of enable interrupt request register 0 (ENIR0)
STANDBY : Standby mode
RUN : Active

14.5.3 Return from Sleep Mode

Overview

External interrupt requests can be used for a return from sleep mode.

Settings

Before a transition to sleep mode, the following setting for the INT0 to INT15 pins must be made with the enable interrupt request registers (ENIR0, ENIR1):

- Pins used for the return from sleep mode: Enable interrupt request output.
- Pins not used for the return from sleep mode: Disable interrupt request output.

Return operation

This device returns from sleep mode when a signal at the specified level/edge is input to the INT0 to INT15 pins in sleep mode.

15. Watchdog Timer

This chapter explains the functions and operations of the watchdog timer.

15.1 Overview

15.2 Configuration

15.3 Registers

15.4 Explanation of Operations and Setting Procedure Examples

15.1 Overview

The watchdog timer is a monitoring timer used to determine whether software hangs up or performs other abnormal operations.

This product is equipped with 2 channels for watchdog timer.

Overview

If the watchdog timer is not cleared before the specified period has elapsed, it judges that software has hung up and outputs a reset request to the CPU. This reset request is called a watchdog reset request.

The operation of the watchdog timer requires that it be continually and periodically cleared before the specified period has elapsed. If an abnormal operation of software such as hanging up prevents it from being periodically cleared, it overflows and outputs a watchdog reset request.

- Watchdog timer 0
  - The watchdog timer counts cycles while a program is active on the CPU, and it stops counting while the CPU is stopped (in sleep mode, stop mode, or watch mode).
  - The watchdog timer can detect a transition to standby mode (watch mode/stop mode), and it can output a watchdog reset request to the CPU.
  - If an incorrect value is written to watchdog timer clear pattern register 0 (WDTCPR0), the watchdog timer outputs a watch reset request to the CPU.
  - The following period can be selected as the watchdog timer 0 period:peripheral clock (PCLK) $\times$ (2^9 to 2^{24})
- Watchdog timer 1
  - After releasing the reset of this device, it counts with CPU clock (CCLK).
  - Disable/enable of the counter operation can be controlled by HWDE pin.
  - If an incorrect value is written to watchdog timer clear pattern register 1 (WDTCPR1), the watchdog timer outputs a watch reset request to the CPU.
  - The period of the watchdog timer 1 is fixed to CPU clock (CCLK) $\times$ ($2^{23} \pm 2^{14}$) cycle.

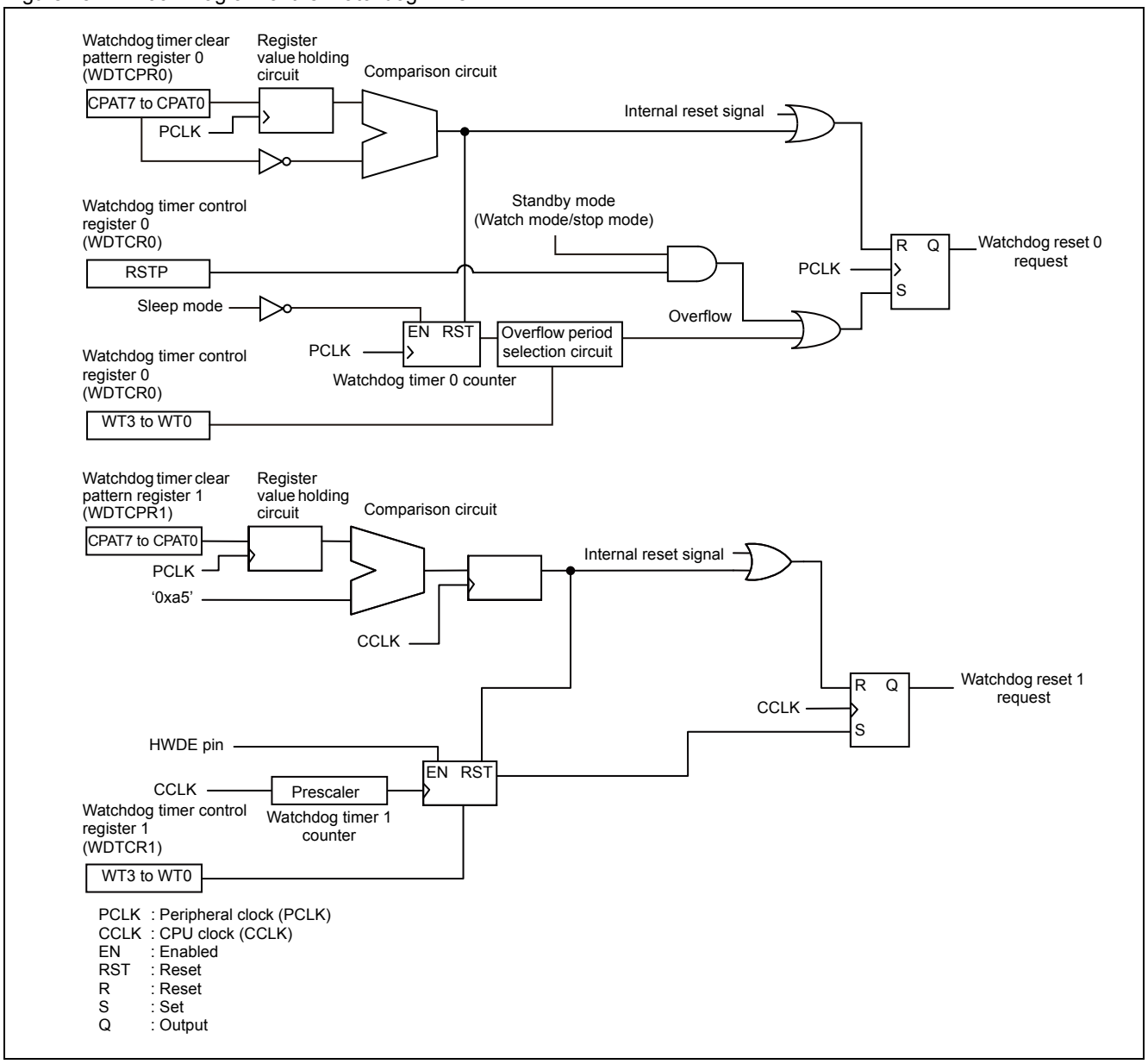
15.2 Configuration

This section shows the configuration of the watchdog timer.

Block diagram of the watchdog timer

Figure 15-1 is a block diagram of the watchdog timer.

Figure 15-1. Block Diagram of the Watchdog Timer



- Watchdog timer control register (WDTCR0, WDTCR1)
  This register controls the operation of the watchdog timer.
- Watchdog timer clear pattern register (WDTCPR0, WDTCPR1)
  This register activates and clears the watchdog timer.
- Watchdog timer 0
  This is a 24-bit up counter.
- Watchdog timer 1
  It consists of a 9-bit up counter and 14-bit prescaler.
- Register value holding circuit
  This circuit retains the value written in watchdog timer clear pattern register (WDTCPR0, WDTCPR1).
- Comparison circuit
  This circuit compares the value written in watchdog timer clear pattern register (WDTCPR0, WDTCPR1) with the previous value that was written.
- Overflow period selection circuit
  This circuit selects the overflow period of the watchdog timer.

Clocks

Table 15-1 lists the clock used by the watchdog timer 0.

Table 15-1. Clock used by the Watchdog Timer 0

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

Table 15-2 lists the clock used by the watchdog timer 1.

Table 15-2. Clock used by the Watchdog Timer 1

Clock Name	Description
Operation clock	CPU clock (CCLK)

15.3 Registers

This section explains the configuration and functions of registers for the watchdog timer.

List of registers for the watchdog timer

Table 15-3 lists the registers for the watchdog timer.

Table 15-3. Registers for the Watchdog Timer

Abbreviated Register Name	Register Name	Reference
WDTCR0	Watchdog timer control register 0	15.3.1
WDTCPR0	Watchdog timer clear pattern register 0	15.3.2
WDTCR1	Watchdog timer control register 1	15.3.3
WDTCPR1	Watchdog timer clear pattern register 1	15.3.4

15.3.1 Watchdog Timer Control Register 0 (WDTCR0)

This register controls the operation of the watchdog timer 0.

Figure 15-2 shows the bit configuration of watchdog timer control register 0 (WDTCR0).

Figure 15-2. Bit Configuration of Watchdog Timer Control Register 0 (WDTCR0)

bit	7	6	5	4	3	2	1	0
	Reserved	RSTP	Reserved	Reserved	WT3	WT2	WT1	WT0
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

R/W: Read/Write

Note: This register can be written only prior to activation of the watchdog timer 0.

[bit7]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit6]: RSTP (Stop mode detection reset enable bit)

This bit specifies whether to enable output of a watchdog reset request at the transition time of the CPU to standby mode (watch mode/stop mode) while the watchdog timer 0 is active.

Written Value	Explanation
0	Disables output of a watchdog reset request. The counting of the watchdog timer is suspended when a transition to standby mode (watch mode/stop mode) is detected, and it remains suspended until a return from standby mode.
1	Enables output of a watchdog reset request. A watchdog reset request is output when a transition to standby mode (watch mode/stop mode) is detected.

Note:

- To use standby mode (watch mode/stop mode), set "0" in this bit.
- This register can be written only before the watchdog timer 0 is activated. If "1" is set in this bit after the watchdog timer is activated, standby mode (watch mode/stop mode) is detected and a watchdog reset request is output. Therefore, standby mode becomes unusable.

[bit5, bit4]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit3 to bit0]: WT3 to WT0 (Watchdog timer period selection bits)

These bits select one of the following periods as the period from watchdog timer 0 clearing to watchdog reset request output.

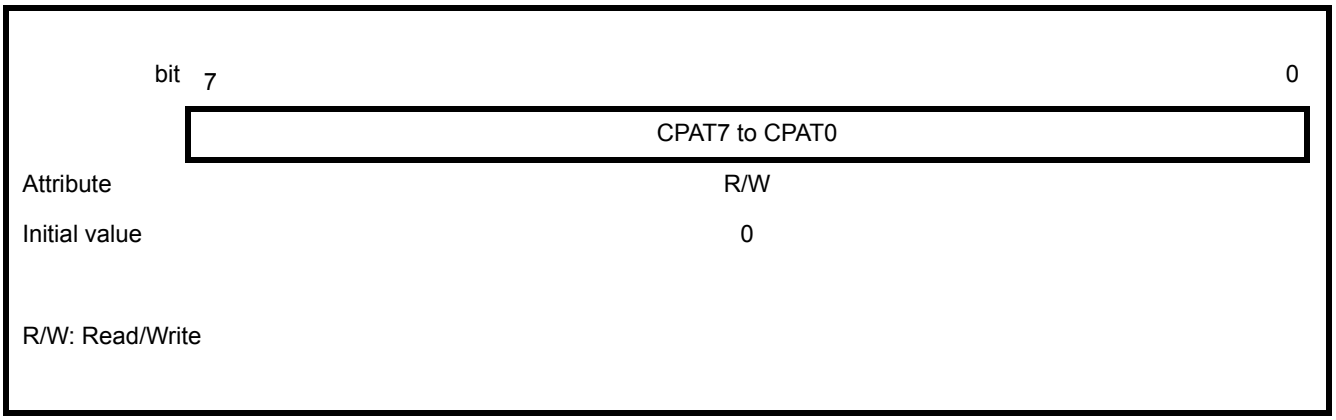
WT3 to WT0	Watchdog Timer Period
0000	PCLK x 2^{9}
0001	PCLK x 2^{10}
0010	PCLK x 2^{11}
0011	PCLK x 2^{12}
0100	PCLK x 2^{13}
0101	PCLK x 2^{14}
0110	PCLK x 2^{15}
0111	PCLK x 2^{16}
1000	PCLK x 2^{17}
1001	PCLK x 2^{18}
1010	PCLK x 2^{19}
1011	PCLK x 2^{20}
1100	PCLK x 2^{21}
1101	PCLK x 2^{22}
1110	PCLK x 2^{23}
1111	PCLK x 2^{24}
PCLK:Period of Peripheral clock (PCLK)	

15.3.2 Watchdog Timer Clear Pattern Register 0 (WDTCPR0)

This register activates and clears the watchdog timer 0.

Figure 15-3 shows the bit configuration of watchdog timer clear pattern register 0 (WDTCPR0).

Figure 15-3. Bit Configuration of Watchdog Timer Clear Pattern Register 0 (WDTCPR0)



[bit7 to bit0]: CPAT7 to CPAT0 bits

The watchdog timer 0 is activated when any value is written to this register after this device is reset.

To prevent a watchdog reset request from being output after the watchdog timer 0 is activated, the timer 0 must be cleared before the timer period has elapsed.

To clear the watchdog timer 0, invert the bit pattern written in these bits and write the inverted value to the bits.

For details of clearing the watchdog timer 0, see "Clearing the watchdog timer" in "15.4.1 Operations of the Watchdog Timer 0".

CPAT7 to CPAT0	In Case of Writing	In Case of Reading
Value obtained by inverting the written value	After being activated, the watchdog timer is cleared.	"0" is read.
Value other than that obtained by inverting the written value	A watchdog reset request is output immediately.	

15.3.3 Watchdog Timer Control Register 1 (WDTCR1)

This register controls the operation of the watchdog timer 1.

Figure 15-4 shows the bit configuration of watchdog timer control register 1 (WDTCR1).

Figure 15-4. Bit Configuration of Watchdog Timer Control Register 1 (WDTCR1)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	WT3	WT2	WT1	WT0
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	0	0	0	0

R: Read only

Note: This register cannot be rewritten.

[bit7 to bit4]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit3 to bit1]: WT3 to WT0 (Watchdog timer period selection bits)

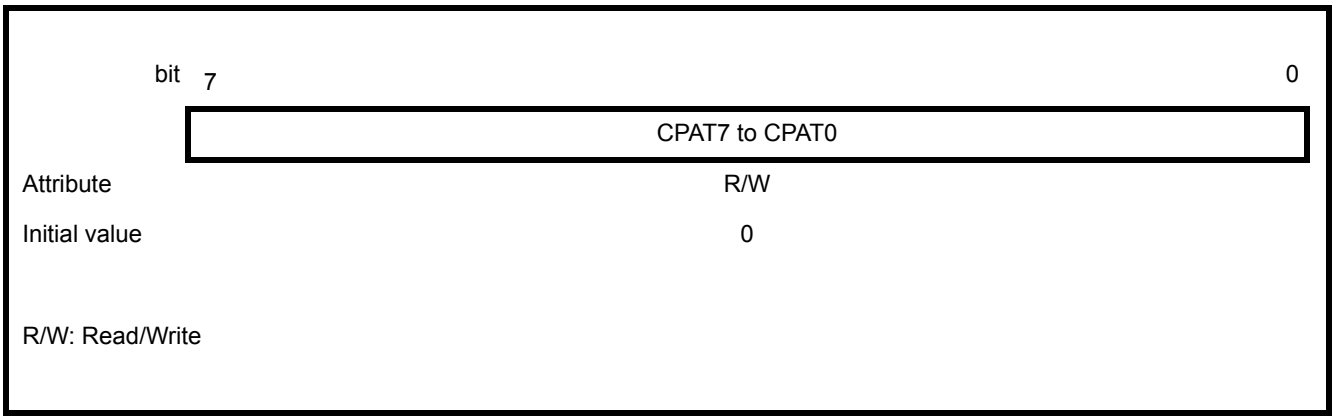
In case of reading	"0" is read.

15.3.4 Watchdog Timer Clear Pattern Register 1 (WDTCPR1)

This register activates and clears the watchdog timer 1.

Figure 15-5 shows the bit configuration of watchdog timer clear pattern register 1 (WDTCPR1).

Figure 15-5. Bit Configuration of Watchdog Timer Clear Pattern Register 1 (WDTCPR1)



[bit7 to bit0]: CPAT7 to CPAT0 bits

After releasing the reset of this device, the watchdog timer 1 will be activated.

To prevent a watchdog reset request from being output after the watchdog timer 1 is activated, the timer 1 must be cleared before the timer period has elapsed.

To clear the watchdog timer 1, write 0xa5 to this bit.

For details of clearing the watchdog timer 1, see "Clearing the watchdog timer 1" in "15.4.2 Operations of the Watchdog Timer 1".

CPAT7 to CPAT0	In Case of Writing	In Case of Reading
Writing 0xa5	After being activated, the watchdog timer is cleared.	"0" is read.
Written value other than 0xa5	A watchdog reset request is output immediately.	

15.4 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the watchdog timer. Also, examples of procedures for setting operating states are shown.

Overview

If the watchdog timer is not periodically cleared even though the program is designed to do so, a malfunction is judged to have occurred and the watchdog timer outputs a watchdog reset request to the CPU.

15.4.1 Operations of the Watchdog Timer 0

Settings

To use the watchdog timer, specify the following with watchdog timer control register 0 (WDTCR0) before activating the watchdog timer:

- Period from watchdog timer clearing to the watchdog reset request output (WT3 to WT0 bits)
- Whether to enable output of a watchdog reset request at the transition time of the CPU to standby mode (watch mode/stop mode) (RSTP)

Note:

- The watchdog timer performs counting only while the CPU is operating. Therefore, the WT3 to WT0 bits must be set based on the setting of the number of program steps and the clock division setting.
- To use standby mode (watch mode/stop mode), set "0" in the RSTP bit.
- If "1" is set in the RSTP bit after the watchdog timer is activated, standby mode (watch mode/stop mode) cannot be used.

Operations

The watchdog timer is activated when any value is written to the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) after this device is reset. The counter value changes in sync with the rising edge of the peripheral clock (PCLK) while the CPU is active.

Unless the watchdog timer is cleared before the period specified by the WT3 to WT0 bits of watchdog timer control register 0 (WDTCR0) has elapsed, a watchdog reset request is output to the CPU.

Also, the watchdog timer temporarily stops counting while the CPU is stopped, such as during doze mode or sleep mode.

The value of the watchdog timer is not cleared while the counting is temporarily stopped. When the counting resumes, it starts from the value at which it was stopped.

Note:

- Even during DMA transfer with the DMA controller (DMAC), the watchdog timer continues counting as long as the CPU is operating.
- Since the peripheral clock (PCLK) is stopped during the oscillation stabilization wait time of the CPU source clock (SRCCLK), the watchdog timer also stops counting during this time.
- Sampling of the CPU operation state is performed using the peripheral clock (PCLK). Therefore, a change in the operating state that does not last longer than the period of the peripheral clock (PCLK) may be ignored.

Clearing the watchdog timer

The watchdog timer can be cleared by inverting the value written in the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) at the watchdog timer activation time and writing the inverted value to these bits.

For example, if "55_H" is written in the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) at the watchdog timer activation time, the watchdog timer can be cleared by writing the inverted value "AA_H" to the bits.

Clearing of the watchdog timer can be subsequently repeated by alternately writing "55_H" and "AA_H" to the CPAT7 to CPAT0 bits.

However, a watchdog reset request is output to the CPU when any value other than the inverted values is written to the CPAT7 to CPAT0 bits.

Note: If it is difficult to maintain the value written in these bits, writing of a value to them can be followed by writing of its inverted value (e.g., writing "AA_H" then writing "55_H") every time the watchdog timer is cleared.

Output of a watchdog reset request

The watchdog timer outputs a watchdog reset request to the CPU in any of the following cases:

- The period specified by the WT3 to WT0 bits of watchdog timer control register 0 (WDTCR0) has elapsed (overflow).
- The value written in the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) is different from the value obtained by inverting the written value.
- There is a transition by the CPU to standby mode (watch mode/stop mode) (a watchdog reset request may be output depending on the setting of the RSTP bit of watchdog timer control register 0 (WDTCR0)).

For details of the operations after output of a watchdog reset request, see "9.5 Explanation of Operations".

15.4.2 Operations of the Watchdog Timer 1

Settings

To use the watchdog timer 1, set disable/enable of the watchdog timer 1 operation with the HWDE pin. If "L" is input to the HWDE pin, the operation of the timer is disabled, and the count is stopped.

Operations

When the HWDE pin is "H", the watchdog timer 1 is activated after the reset of this device is released, and it starts counting at the rising edge of the CPU clock (CCLK).

If the period specified by the WT3 to WT0 bits of watchdog timer 1 control register (WDTCR1) has elapsed when the watchdog timer is not cleared, a watchdog reset request is output to the CPU.

Note: The period is fixed to CCLK $\times 2^{23}$. There are $\pm 2^{14}$ variations.

Clearing the watchdog timer 1

The watchdog timer 1 can be cleared by writing "$A5_H$" to the CPAT7 to CPAT0 bits of the watchdog timer 1 clear register (WDTCPR1) while the watchdog timer 1 is activated.

However, when the value other than "$A5_H$" is written to the CPAT7 to CPAT0 bits, the watchdog reset request is output to the CPU.

Output of a watchdog reset request

The watchdog timer 1 outputs a watchdog reset request to the CPU in any of the following cases:

- The period specified by the WT3 to WT0 bits of watchdog timer 1 control register(WDTCR1) has elapsed (overflow).
- The value other than "$A5_H$" is written to the CPAT7 to CPAT0 bits in the watchdog timer 1 clear register (WDTCPR1).

For details of the operations after output of a watchdog reset request, see "9.5 Explanation of Operations".

16. Watch Counter

This chapter explains the functions and operations of the watch counter.

16.1 Overview

16.2 Configuration

16.3 Registers

16.4 Interrupts

16.5 Explanation of Operations and Setting Procedure Examples

16.6 Notes on Use

16.1 Overview

The watch counter is a timer that counts down starting from the specified value, and it generates an interrupt request at the time that the 6-bit down counter enters an underflow condition.

Interrupt requests can be generated at a period ranging from 125 ms to 64 s.

This series has 1 built-in channel for the watch counter.

Note: This function is not available when the sub clock (SBCLK) is not being used.

Overview

- The count clock can be selected from 4 types of clock, and interrupt requests can be set to be generated at an interval ranging from a minimum of 125 ms to a maximum of 64 s.

  Table 16-1 lists the count clocks and counting periods.

Table 16-1. Count Clocks and Counting Periods

Period of Count Clock	Counting Period (F_{CL} = 32.768 kHz)
$2^{12}/F_{CL}$	125 ms
$2^{13}/F_{CL}$	250 ms
$2^{14}/F_{CL}$	500 ms
$2^{15}/F_{CL}$	1 s

F_{CL}: Sub clock (SBCLK) frequency

- A number between 0 and 63 can be set as the value used for counting by the 6-bit down counter.

  If "60" is the count value used for a counting period of 1 second, an interrupt request is generated at an interval of 1 minute. If "0" is the count value used for a counting period of 1 second, an interrupt request is generated at an interval of 64 seconds.
- An interrupt request can be generated at the time that the 6-bit down counter enters an underflow condition.

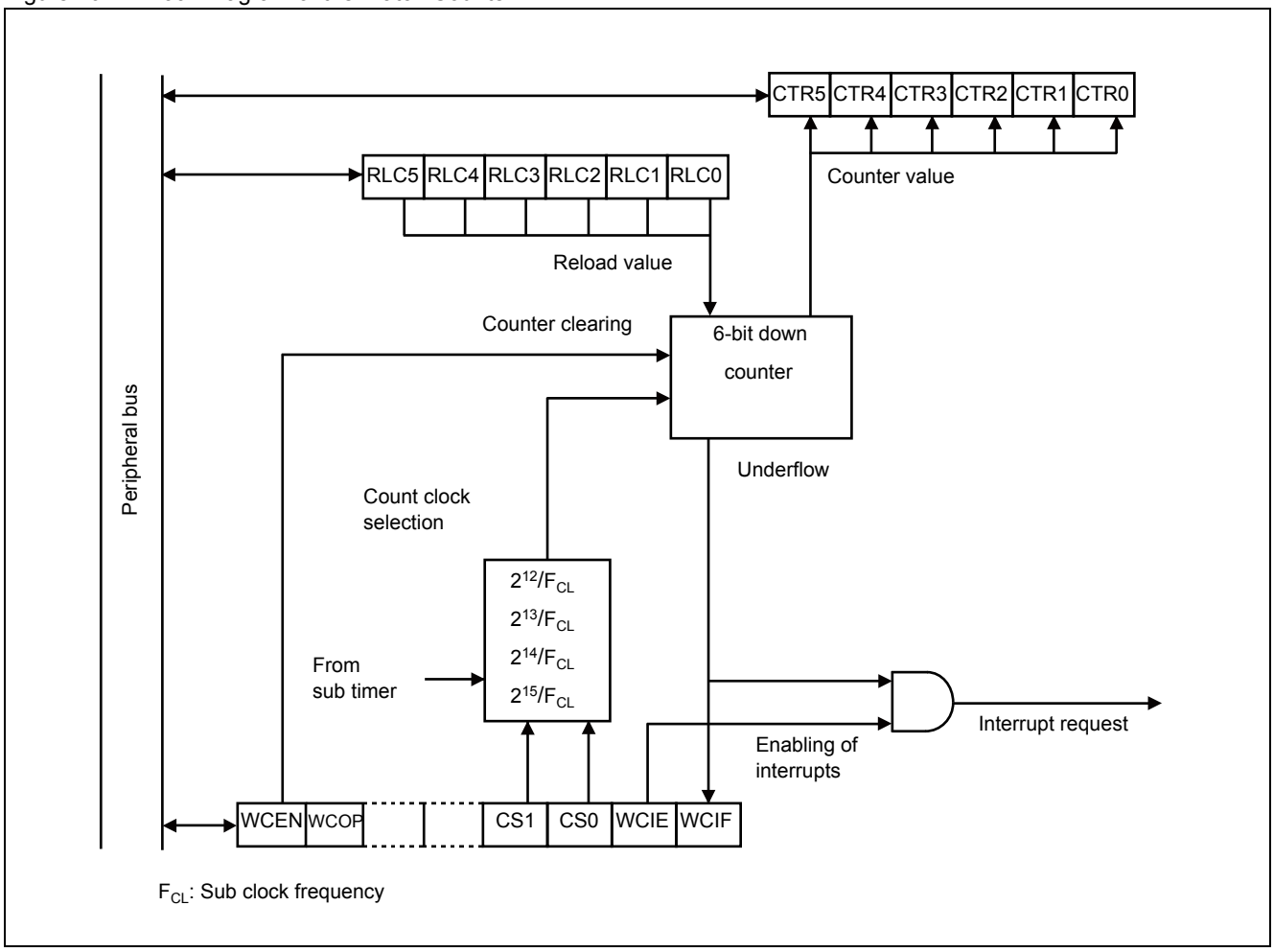
16.2 Configuration

This section shows the watch counter configuration.

Block diagram of the watch counter

Figure 16-1 is a block diagram of the watch counter.

Figure 16-1. Block Diagram of the Watch Counter



- 6-bit down counter

  This is the 6-bit down counter of the watch counter. It reloads the value set in the watch counter reload register (WCRL) and starts a countdown.
- Watch counter reload register (WCRL)

  This register specifies the value used by the watch counter to start counting. The 6-bit down counter counts down starting from the value set in this register.
- Watch counter read register (WCRD)

  This register reads the value in the 6-bit down counter. Also, the register can be read to check the count value.
- Watch counter control register (WCCR)

  This register controls the operation of the watch counter.

Clocks

Table 16-2 lists the clocks used by the watch counter.

Table 16-2. Clocks used by the watch counter

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Sub timer output	Sub timer period[1]

[1]: The sub timer period is specified by the STS2 to STS0 bits in the sub timer control register (STMCR). For details of the sub timer, see "7. Sub Timer".

16.3 Registers

This section explains the configurations and functions of the registers for the watch counter.

List of registers for the watch counter

Table 16-3 lists the registers for the watch counter.

Table 16-3. Registers for the Watch Counter

Abbreviated Register Name	Register Name	Reference
WCRL	Watch counter reload register	16.3.1
WCCR	Watch counter control register	16.3.2
WCRD	Watch counter read register	16.3.3

16.3.1 Watch Counter Reload Register (WCRL)

This register specifies the value used by the watch counter to start counting. The 6-bit down counter counts down starting from the value set in the register.

The register specifies the reload value for the 6-bit down counter. If the 6-bit down counter enters an underflow condition, the value in this register is reloaded in the 6-bit down counter, and the countdown is restarted.

Figure 16-2 shows the bit configuration of the watch counter reload register (WCRL).

Figure 16-2. Bit Configuration of the Watch Counter Reload Register (WCRL)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	RLC5	RLC4	RLC3	RLC2	RLC1	RLC0
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

-: Undefined

[bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit5 to bit0]: RLC5 to RLC0 (Counter reload value setting bits)

These bits set the reload value for the 6-bit down counter.

The 6-bit down counter counts downwards from the reload value and enters an underflow condition when its value reaches "1". If "0" is set in these bits, it performs 64 countdowns from "63" to "0".

Notes:

- If the value of these bits is changed to another value while the 6-bit down counter is active, an underflow occurs and the new value is then reloaded.
- If the value of these bits is changed to another value at the same time that an underflow interrupt request is generated, the correct value is not reloaded. Be sure to rewrite the value of these bits either when the watch counter is stopped or in the interrupt processing routine before an interrupt request is generated.
- To verify whether the reload value is correctly set, read this register.

16.3.2 Watch Counter Control Register (WCCR)

This register selects a count clock for the watch counter or enables/disables generation of interrupt requests. The register also enables/disables the operation of the watch counter.

Figure 16-3 shows the bit configuration of the watch counter control register (WCCR).

Figure 16-3. Bit Configuration of the Watch Counter Control Register (WCCR)

bit	7	6	5	4	3	2	1	0
	WCEN	WCOP	Undefined	Undefined	CS1	CS0	WCIE	WCIF
Attribute	R/W	R	-	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

[bit7]: WCEN (Watch counter operation enable bit)

This bit enables/disables the operation of the watch counter.

Written Value	Explanation
0	The watch counter is disabled/stopped. The value in the 6-bit down counter is cleared to "000000_B".
1	The watch counter is enabled/started.

Notes:

- Output of the sub timer is used for the count clock of the watch counter, and the peripheral clock (PCLK) is used for the settings of each register. Since the sub timer and peripheral clock (PCLK) are not synchronized, an error of up to 1T (T: Count clock period) may occur at the count start time, depending on the time at which "1" is written to this bit.
- Before writing "1" to this bit to start the operation of the watch counter, verify that the watch counter is stopped by checking the WCOP bit (WCOP = 0).

[bit6]: WCOP (Watch counter operating state flag bit)

This bit indicates the operating state of the watch counter.

Read Value	Explanation
0	The watch counter is stopped.
1	The watch counter is active.

[bit5, bit4]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit3, bit2]: CS1, CS0 (Count clock selection bits)

These bits set the count clock of the watch counter.

CS1	CS0	Count Clock
0	0	$2^{12}/F_{CL}$
0	1	$2^{13}/F_{CL}$
1	0	$2^{14}/F_{CL}$
1	1	$2^{15}/F_{CL}$

F_{CL} Sub clock (SBCLK) frequency

Note:

The following conditions must be satisfied when the information in these bits is changed:

- WCEN bit = 0 (watch counter operation disabled)
- WCOP bit = 0 (watch counter stopped)

[bit1]: WCIE (Interrupt request enable bit)

This bit specifies whether to generate an underflow interrupt request at the time that the 6-bit down counter enters an underflow condition (WCIF bit = 1).

Written Value	Explanation
0	Disables generation of an underflow interrupt request.
1	Enables generation of an underflow interrupt request.

[bit0]: WCIF (Interrupt request flag bit)

This bit indicates whether the 6-bit down counter has entered an underflow condition.

If "1" is set in the WCIE bit, an interrupt request is generated when "1" is set in this bit.

WCIF	In Case of Reading	In Case of Writing
0	The down counter has not entered an underflow condition.	This bit is cleared to "0".
1	The down counter has entered an underflow condition.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

16.3.3 Watch Counter Read Register (WCRD)

This register reads the value in the 6-bit down counter.

Figure 16-4 shows the bit configuration of the watch counter read register (WCRD).

Figure 16-4. Bit Configuration of the Watch Counter Read Register (WCRD)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	CTR5	CTR4	CTR3	CTR2	CTR1	CTR0
Attribute	-	-	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

-: Undefined

Note: If the 6-bit down counter is operating when its value is read, the register value must be read twice and verified to be the same value.

16.4 Interrupts

The 6-bit down counter enters an underflow condition when the value in the 6-bit down counter becomes "000001_B", and an underflow interrupt request is then generated.

Table 16-4 outlines the interrupts that can be used with the watch counter.

Table 16-4. Interrupts of the Watch Counter

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Underflow interrupt request	WCIF=1 for WCCR	WCIE=1 for WCCR	Write "0" to the WCIF bit for WCCR

WCCR: watch counter control register (WCCR)

Note:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling generation of interrupt requests.

  - Clear interrupt requests before enabling the generation of interrupt requests.
  - Clear interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- To set the interrupt level corresponding to the interrupt vector number, use an interrupt control register (ICR00 to ICR47). For details of setting interrupt levels, see "10. Interrupt Controller".

16.5 Explanation of Operations and Setting Procedure Examples

This section explains operations of the watch counter. Also, examples of procedures for setting the operating state are shown.

16.5.1 Operations of the Watch Counter

The watch counter is a timer that counts down starting from the value set in the watch counter reload register (WCRL), and it generates an interrupt request at the time that the 6-bit down counter enters an underflow condition.

To operate the watch counter, follow the procedure below.

1. Select a count clock by using the CS1 and CS0 bits of the watch counter control register (WCCR).
2. Set a count value to the RLC5 to RLC0 bits in the watch counter reload register (WCRL).
3. Enable the operation of the watch counter by using the WCEN bit (WCEN = 1) of the watch counter control register (WCCR).

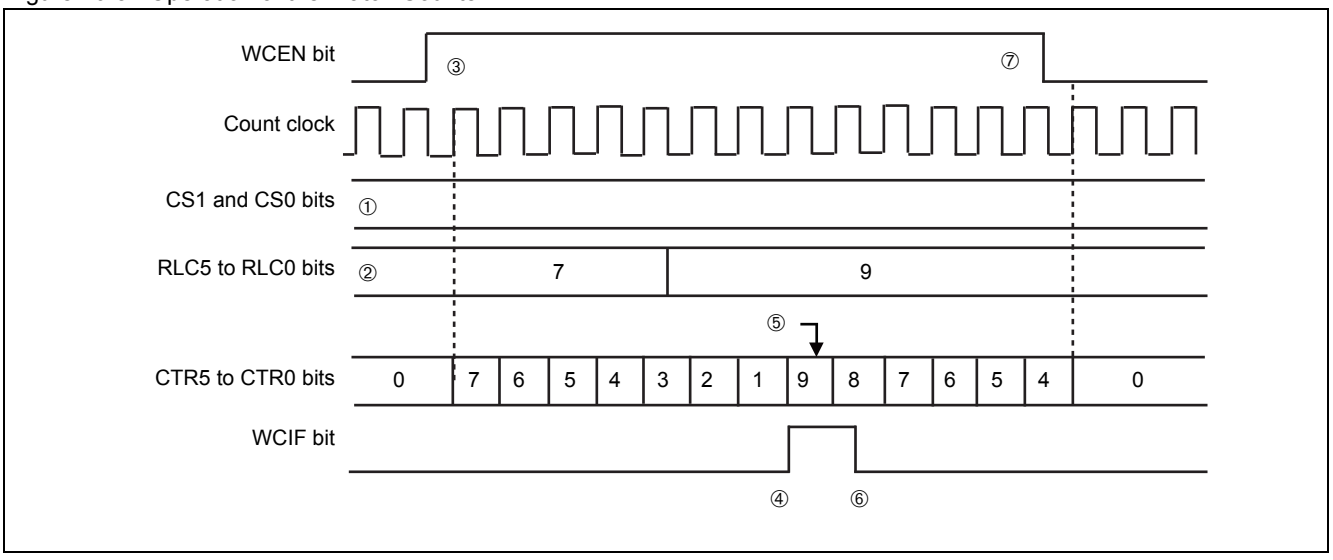
   Start a countdown. Counting is performed at the rising edge of the count clock.
4. If the 6 -bit down counter enters an underflow condition, the value of the WCIF bit in the watch counter control register (WCCR) is changed to "1".

   At this time, if generation of underflow interrupt requests has been enabled by the WCIE bit in the watch counter control register (WCCR), an underflow interrupt request is generated.

   Also, the value that is set in the RLC5 to RLC0 bits in the watch counter reload register (WCRL) is reloaded in the 6-bit down counter, and the countdown is restarted.
5. If the value of the RLC5 to RLC0 bits in the watch counter reload register (WCRL) is changed to another value while the watch counter is active, the watch counter is updated with the new value at the next reload time.
6. The underflow interrupt request is cleared when "0" is written to the WCIF bit in the watch counter control register (WCCR).
7. The 6-bit down counter is cleared to "000000_B" and the counting operation is stopped when "0" is written to the WCEN bit in the watch counter control register (WCCR).

Figure 16-5 shows the operation of the watch counter.

Figure 16-5. Operation of the Watch Counter



Note:

- Output of the sub timer is used for the count clock of the watch counter, and the peripheral clock (PCLK) is used for the settings of each register. Since the sub timer and peripheral clock (PCLK) are not synchronized, an error of up to 1T (T: Count clock period) may occur at the count start time, depending on the time at which "1" is written to the WCEN bit in the watch counter control register (WCCR).
- Since the count clock from the sub timer is also stopped when the sub clock (SBCLK) is stopped, the 6-bit down counter is stopped too. Even when the sub clock (SBCLK) starts operating again, the watch counter cannot count counter values correctly. Before using the watch counter when the sub clock (SBCLK) starts operating again, be sure to write "0" to the WCEN bit in the watch counter control register (WCCR) to clear the counter value to "000000_B".
- Even when the CPU is operating in watch mode, the watch counter continues operating as long as the sub timer is operating. The watch mode of the CPU can be canceled with the watch counter interrupt processing routine.
- If the sub timer is cleared while the watch counter is active, counting values correctly may become impossible. Stop the watch counter by using the WCEN bit (WCEN = 0) of the watch counter control register (WCCR), and then clear the sub timer.
- After the watch counter is stopped by writing "0" to the WCEN in the watch counter control register (WCCR), be sure to verify that the watch counter is stopped by checking the WCOP bit (WCOP = 0) in the watch counter control register (WCCR) before reactivating the watch counter by using the WCEN bit (WCEN = 1).

16.6 Notes on Use

Note the following points about using the watch counter.

Notes on operations

- If the sub timer is cleared while the watch counter is active, counting values correctly may become impossible. Stop the watch counter by using the WCEN bit (WCEN = 0) of the watch counter control register (WCCR), and then clear the sub timer.
- After the watch counter is stopped by the WCEN bit (WCEN = 0) in the watch counter control register (WCCR), be sure to verify that the watch counter is stopped by checking the WCOP bit (WCOP = 0) in the watch counter control register (WCCR) before reactivating the watch counter by using the WCEN bit (WCEN = 1).
- Since the watch counter uses output of the sub timer as the count clock, the setting of the sub timer must not be changed while the watch counter is active.
- The watch counter enters an underflow condition when it counts downwards from "000001_B". It counts downwards from the reload value to "1". If the value is set to "0", it performs 64 countdowns.

17. 32-bit Free-Run Timer

This chapter explains the functions and operations of the 32-bit free-run timer.

17.1 Overview

17.2 Configuration

17.3 Pins

17.4 Registers

17.5 Interrupts

17.6 An Explanation of Operations and Setting Procedure Examples

17.1 Overview

The 32-bit free-run timer is an up-counter that counts up to the predetermined value.

After counting up to the specified value, the free-run timer clears the value and starts counting again or generates an interrupt request. The count value is also used as the reference time for 32-bit output compare or 32-bit input capture.

This series microcontroller has 1 built-in channel for the 32-bit free-run timer.

Overview

The 32-bit free-run timer is part of the compare timer. The compare timer comprises the following three peripheral functions:

- 32-bit free-run timer (1 channel)
- 32-bit output compare (4 channels)
  See "19. 32-bit Output Compare".
- 32-bit input capture (4 channels)
  See "18. 32-bit Input Capture".

This chapter explains the 32-bit free-run timer.

- Count clock: One of the following can be selected:
  - Internal clock (peripheral clock)
    Can be selected from 9 types, which are peripheral clocks (PCLK) divided by 1, 2, 4, 8, 16, 32, 64,128, and 256.
  - External clock
- Interrupt request: Can be issued in the following cases:
  The count value of the 32-bit free-run timer matches the preset value (compare clear interrupt).

17.2 Configuration

The 32-bit free-run time is part of the compare timer. The following is a block diagram of the compare timer and the 32-bit free-run timer.

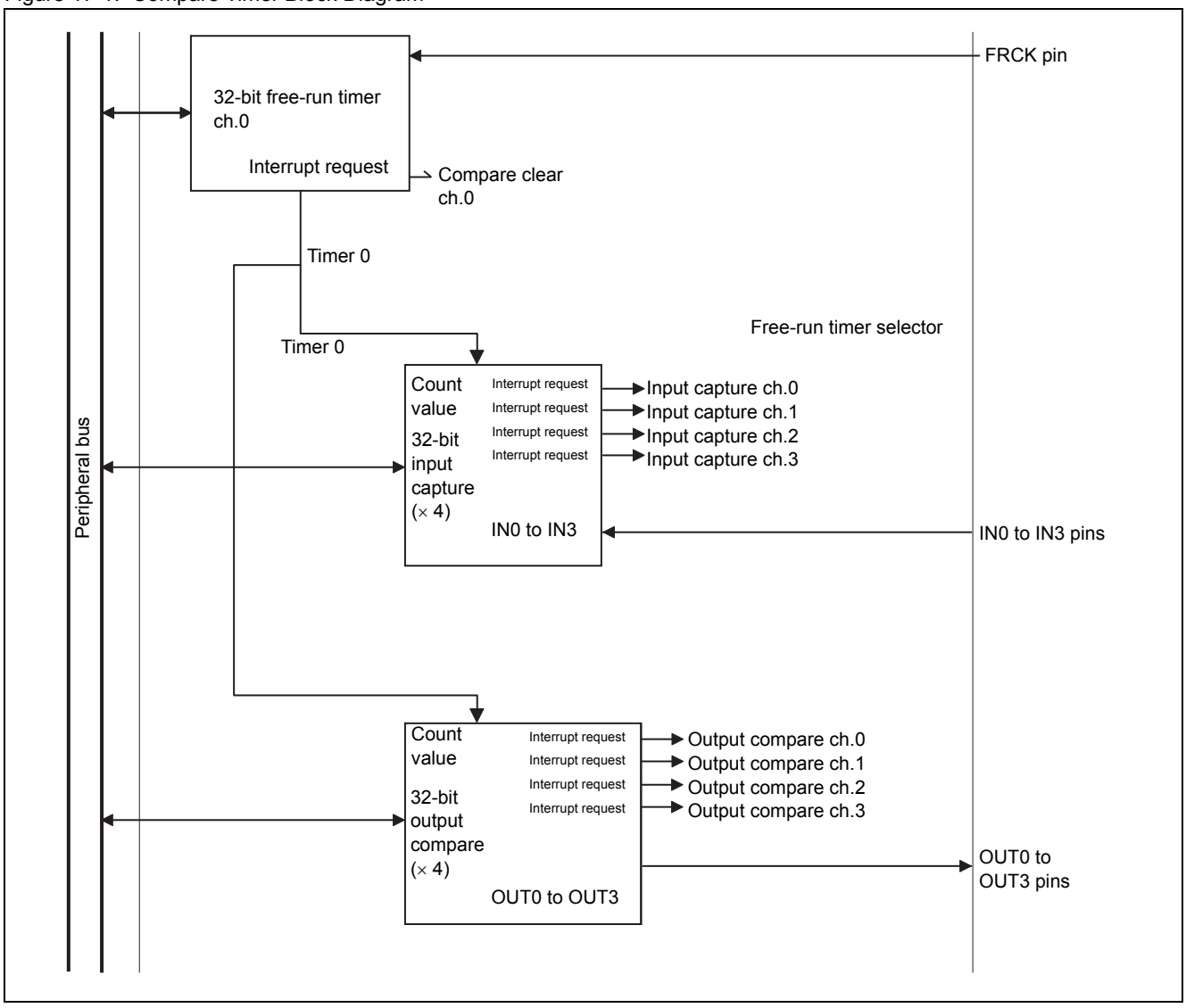
Compare timer block diagram

The compare timer consists of the following blocks.

- 32-bit free-run timer (1 channel)
- 32-bit input capture (4 channels)
- 32-bit output compare (4 channels)

Figure 17-1 is a compare timer block diagram.

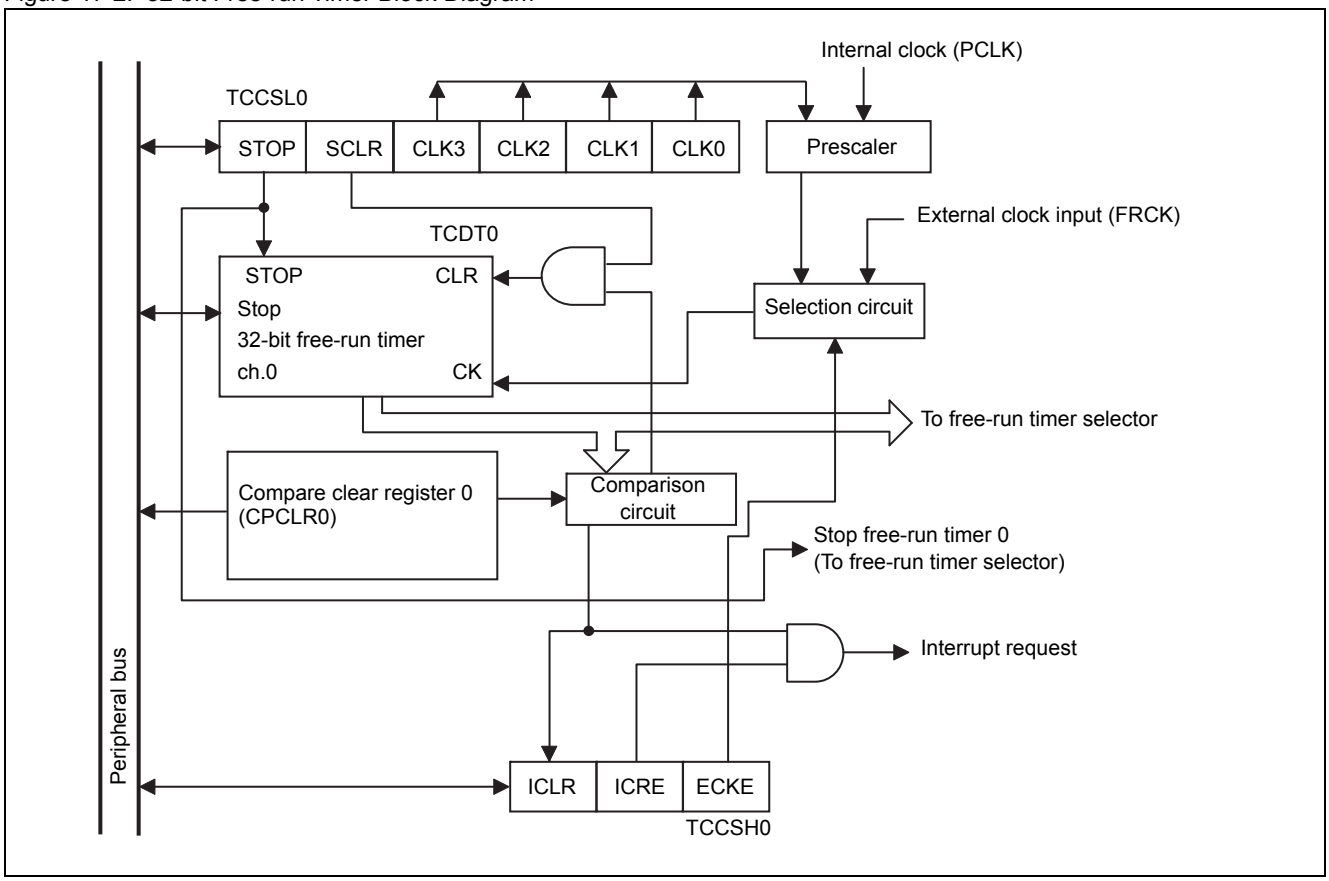
Figure 17-1. Compare Timer Block Diagram



32-bit free-run timer block diagram

is a block diagram of the 32-bit free-run timer.

Figure 17-2. 32-bit Free-run Timer Block Diagram



- 32-bit free-run timer

  This counter counts up to the value that is set in the compare clear register (CPCLR0)
- Timer status control register upper/lower (TCCSH0/TCCSL0)

  This register controls the operation of the 32-bit free-run timer.
- Compare clear register (CPCLR0)

  The 32-bit up counter counts up to the value that is set in this register.
- Timer data register (TCDT0)

  This register is used to set the value with which the timer starts counting or to read the current count value.
- Prescaler

  When the internal clock (peripheral clock) is selected for the count clock, the prescaler divides the peripheral clock (PCLK)
- Selection circuit

  The selection circuit selects whether to use the internal clock (peripheral clock) or external clock (FRCK) for the count clock.
- Comparison circuit

  The comparison circuit compares the count value of the 32-bit free-run timer and the value set in the compare clear register (CPCLR0).

Clocks

Table 17-1 lists the clocks used for the 32-bit free-run timer.

Table 17-1. Clocks used for 32-bit Free-run Timer

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Internal clock (peripheral clock)	Created through division of the peripheral clock (PCLK).
	External clock	Input from the FRCK pin

17.3 Pins

This section explains the pins used by the 32-bit free-run timer.

Overview

- FRCK pin

  These pins are 32-bit free-run timer external clock input pins. These pins are multiplexed pins.

  To use these pins as the FRCK pin of the 32-bit free-run timer, see "2.4 Setting Method for Pins".
- Relationship between pins and channels

Table 17-2 shows the relationship between channels and pins.

Table 17-2. Relationship Between Channels and Pins

Channel	Input Pin
0	FRCK

17.4 Registers

This section explains the configuration and functions of the registers used by the 32-bit free-run timer.

32-bit free-run timer registers

Table 17-3 lists the registers of the 32-bit free-run timer.

Table 17-3. 32-bit Free-run Timer Registers

Channel	Abbreviated Register Name	Register Name	Reference
0	CPCLR0	Compare clear register 0	17.4.1
	TCCSH0/TCCSL0	Timer status control register upper0/lower0	17.4.3
	TCDT0	Timer data register 0	17.4.2

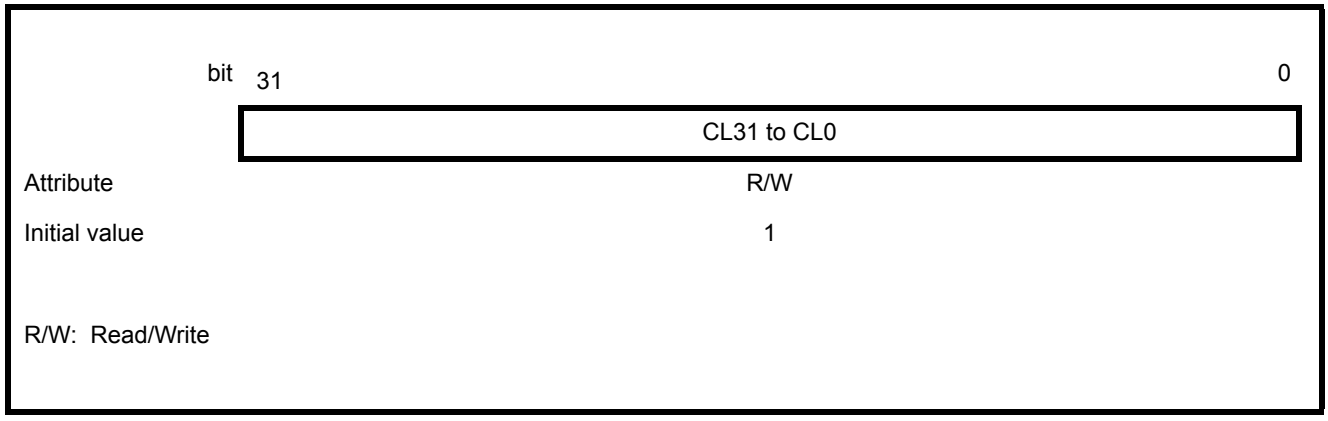
17.4.1 Compare Clear Register (CPCLR0)

This register sets the comparison value of the 32-bit free-run timer.

When the 32-bit free-run timer counts up and reaches the value that is set in this register, the count value of the 32-bit free-run timer is cleared to "0000 0000_H".

Figure 17-3 shows the bit configuration of the compare clear register (CPCLR0).

Figure 17-3. Bit Configuration of Compare Clear Register (CPCLR0)



Notes:

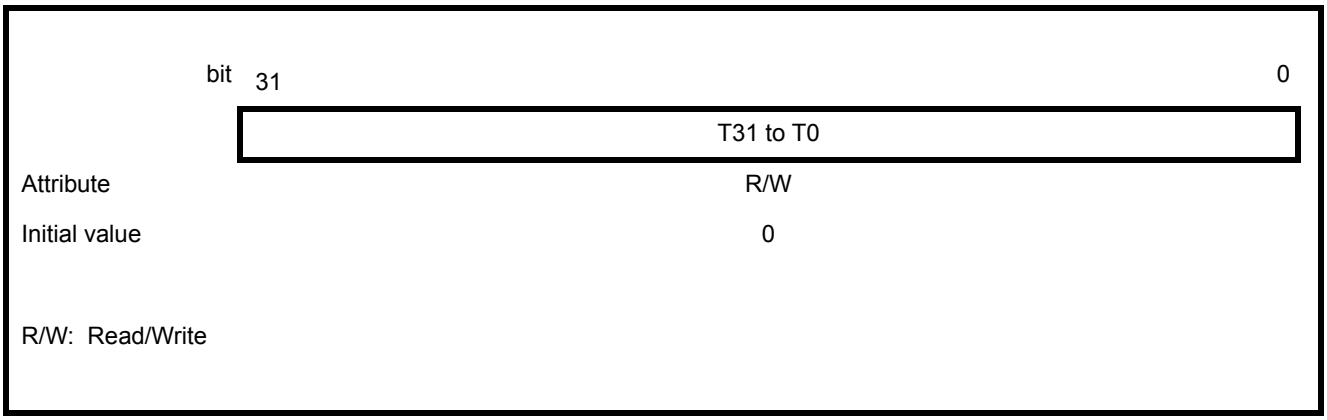
- Rewrite this register while the 32-bit free-run timer is stopped.
  The 32-bit free-run timer is stopped when the STOP bit of the timer status control register lower (TCCSL0) is "1".
- Be sure to access this register in units of words.

17.4.2 Timer Data Register (TCDT0)

This register is used to set the value with which the 32-bit free-run timer starts counting or to read the current count value.

Figure 17-4 shows the bit configuration of the timer data register (TCDT0).

Figure 17-4. Bit Configuration of Timer Data Register (TCDT0)



The 32-bit free-run timer counts up starting from the value written to this register. If this register is read, the count value of the 32-bit free-run timer is read.

Notes:

- Rewrite this register while the 32-bit free-run timer is stopped.
  The 32-bit free-run timer is stopped when the STOP bit of the timer status control register lower (TCCSL0) is "1".
- Be sure to access this register in units of half word.
- The write value and read value of this register are different.
- If one of the following occurs, the count value of the 32-bit free-run timer (the value of this register) is promptly cleared to "0000 0000_H".
  - This device is reset.
  - "1" is written to the SCLR bit of the timer status control register lower (TCCSL0).
  - The count value of the 32-bit free-run time matches the value of the compare clear register (CPCLR0).

17.4.3 Timer Status Control Register Upper/Lower (TCCSH0/TCCSL0)

This register controls the operation of the 32-bit free-run timer.

Figure 17-5 shows the bit configuration of the timer status control register upper/lower (TCCSH0/TCCSL0).

Figure 17-5. Bit Configuration of Timer Status Control Register Upper/lower (TCCSH0/TCCSL0)

Timer status control register upper (TCCSH0)

bit	15	14	13	12	11	10	9	8
	ECKE	Undefined	Undefined	Undefined	Undefined	Undefined	ICLR	ICRE
Attribute	R/W	-	-	-	-	-	R/W	R/W
Initial value	0	X	X	X	X	X	0	0

Timer status control register lower (TCCSL0)

bit	7	6	5	4	3	2	1	0
	Undefined	STOP	Undefined	SCLR	CLK3	CLK2	CLK1	CLK0
Attribute	-	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial value	X	1	X	0	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit15]: ECKE (Clock selection bit)

This bit selects the count clock of the 32-bit free-run timer.

Written Value	Explanation
0	Selects the internal clock (peripheral clock).
1	Selects an external clock.

An internal clock (peripheral clock) is generated by dividing the peripheral clock (PCLK). If an internal clock (peripheral clock) is selected, CLK3 to CLK0 bits must be used to select the division rate of the peripheral clock (PCLK).

An external clock is input through the FRCK pin. When an external clock is selected, the timer counts on both edges of the signal input through the FRCK pin.

Notes:

- The count clock changes as soon as this bit is changed.
- Rewrite this bit while the 32-bit free-run timer, 32-bit input capture, and 32-bit output compare are all stopped.

[bit14 to bit10]: Reserved bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit9]: ICLR (compare clear interrupt request flag bit)

This bit indicates that the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0).

If "1" is set in the ICRE bit when this bit is "1", a compare clear interrupt request is generated.

ICLR	In Case of Reading	In Case of Writing
0	The count value does not match the preset value.	This bit is cleared to "0".
1	The count value matches the preset value.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit8]: ICRE (compare clear interrupt request enable bit)

This bit is specifies whether to generate a compare clear interrupt request when the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0) (ICLR bit = 1).

Written Value	Explanation
0	Disables generation of compare clear interrupt requests.
1	Enables generation of compare clear interrupt requests.

[bit7]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit6]: STOP (timer operation enable bit)

This bit enables (starts) or disables (stops) the count operation of the 32-bit free-run timer.

Written Value	Explanation
0	Enables (starts) the count function.
1	Disables (stops) the count function.

Note: When the 32-bit free-run timer is stopped, the 32-bit output compare is also stopped.

[bit5]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit4]: SCLR (timer clear bit)

This bit clears the count value of the 32-bit free-run timer to "0000 0000_H".

SCLR	In Case of Writing	In Case of Reading
0	Does not clear the count value.	"0" is read.
1	Clears the count value.	

Note: When this bit is set to "1", the count value is cleared at the next count clock timing.

[bit3 to bit0]: CLK3 to CLK0 (clock frequency selection bits)

These bits select the division rate of the peripheral clock (PCLK) when the internal clock (peripheral clock) is selected for the count clock of the 32-bit free-run timer,

The count cycle is determined by using the division rate selected by these bits and the peripheral clock (PCLK) frequency.

Table 17-4 provides an example of count cycles that are set according to the relationship between the values written to these bits and the peripheral clock (PCLK).

Table 17-4. Example of Written Values and Count Cycles

CLK3	CLK2	CLK1	CLK0	PCLK Division Rate	PCLK Frequency				
					32 MHz	16 MHz	8 MHz	4 MHz	1 MHz
0	0	0	0	Divided by 1	31.25 ns	62.5 ns	125 ns	0.25 µs	1 µs
0	0	0	1	Divided by 2	62.5 ns	125 ns	0.25 µs	0.5 µs	2 µs
0	0	1	0	Divided by 4	125 ns	0.25 µs	0.5 µs	1 µs	4 µs
0	0	1	1	Divided by 8	0.25 µs	0.5 µs	1 µs	2 µs	8 µs
0	1	0	0	Divided by 16	0.5 µs	1 µs	2 µs	4 µs	16 µs
0	1	0	1	Divided by 32	1 µs	2 µs	4 µs	8 µs	32 µs
0	1	1	0	Divided by 64	2 µs	4 µs	8 µs	16 µs	64 µs
0	1	1	1	Divided by 128	4 µs	8 µs	16 µs	32 µs	128 µs
1	0	0	0	Divided by 256	8 µs	16 µs	32 µs	64 µs	256 µs

PCLK: Peripheral clock (PCLK)

Notes:

- Do not use any settings other than those listed in Table 17-4.
- The count clock changes as soon as this bit is rewritten.
- Rewrite this bit while the 32-bit free-run timer, 32-bit input capture, and 32-bit output compare are all stopped.

17.5 Interrupts

An interrupt request (compare clear interrupt request) is generated when the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0).

Table 17-5 outlines the interrupts that can be used with the 32-bit free-run timer.

Table 17-5. Interrupts of the 32-bit Free-run Timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Compare clear interrupt request	ICLR=1 for TCCSH	ICRE=1 for TCCSH	Write "0" to the ICLR bit for TCCSH

TCCSH: timer status control register upper (TCCSH0)

Notes:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use an interrupt control register (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "10. Interrupt Controller".

17.6 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the 32-bit free-run timer. Also, examples of procedures for setting the operating state are shown.

Overview

The 32-bit free-run timer uses an internal clock (peripheral clock) or an external clock as count clock and counts up starting from the value set in the timer data register (TCDT0) to the value set in the compare clear register (CPCLR0).

- Internal clock (peripheral clock)

  Can be selected from 9 types, which are peripheral clocks (PCLK) divided by 1, 2, 4, 8, 16, 32, 64,128, and 256.
- External clock

  The timer counts up at both edges. The count start timing varies depending on the initial value of the external clock input through the FRCK pin.

The count value of the 32-bit free-run timer is used as the reference time for 32-bit output compare or 32-bit input capture.

Timer clearing

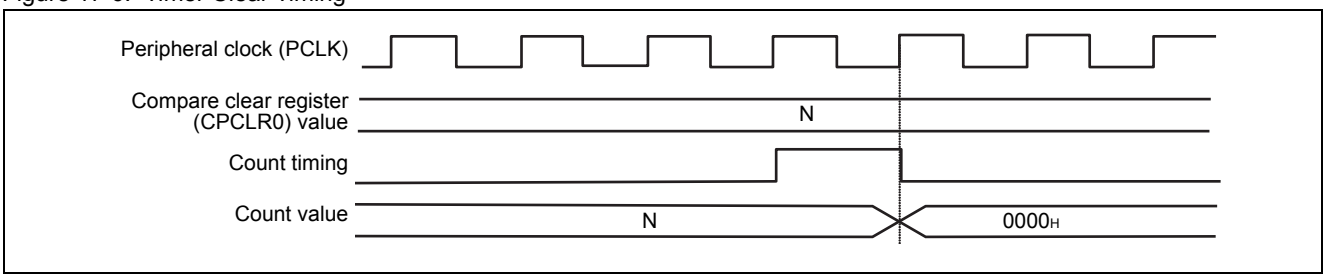
The count value of the 32-bit free-run timer is promptly cleared when one of the following conditions is met:

- This count value matches the value that is set in the compare clear register (CPCLR0).
- The SCLR bit of the timer status control register lower (TCCSL0) is set to 1 to clear the count value of the 32-bit free-run timer.
- "0000 0000_H" is written to the timer data register (TCDT0) while the 32-bit free-run timer is stopped.
- This device is reset.

When the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0), the count value is cleared in synchronization with the count timing.

Figure 17-6 shows the timer clear timing.

Figure 17-6. Timer Clear Timing



17.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected

A divided peripheral clock (PCLK) is used as the count clock.

Count operation

When the STOP bit of the timer status control register lower (TCCSL0) is set to 0 to enable the 32-bit free-run timer, the timer counts up starting from the value set in the timer data register (TCDT0) to the value set in the compare clear register (CPCLR0).

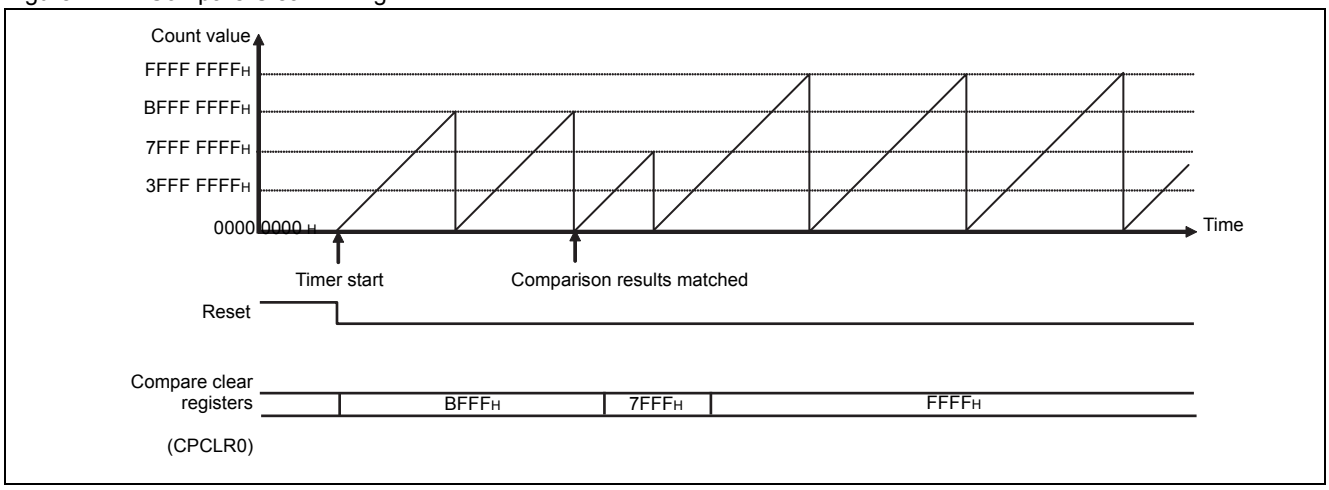
Compare clear

When the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0), the count value is cleared in synchronization with the count timing (compare clear).

After compare clear, the timer starts counting again.

Figure 17-7 shows the compare clear timing.

Figure 17-7. Compare Clear Timing



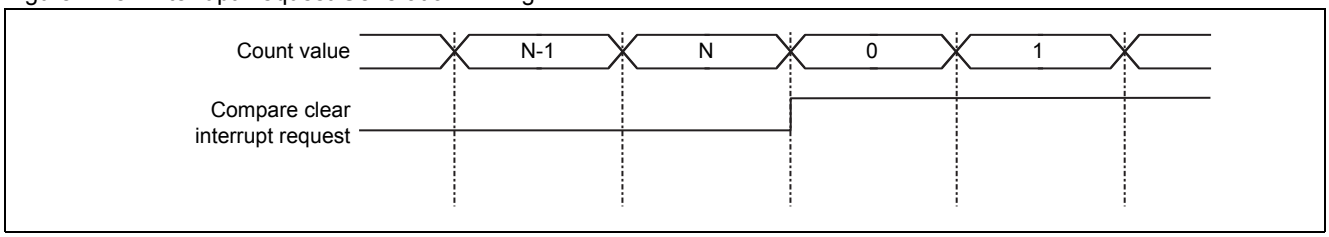
Interrupt processing

An interrupt request can be generated when the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0).

The interrupt request can be cleared by writing "0" to the ICLR bit of the timer status control register upper (TCCSH0).

Figure 17-8 shows the interrupt request generation timing.

Figure 17-8. Interrupt Request Generation Timing



17.6.2 Operation When an External Clock Is Selected

The external clock input through the FRCK pin is used as the count clock.

Count operation

Upon detection of a valid edge through the FRCK pin while the STOP bit of the timer status control register lower (TCCSL0) is set to 0 to enable the 32-bit free-run timer, the timer counts up starting from the value set in the timer data register (TCDT0) to the value set in the compare clear register (CPCLR0).

The count timing varies depending on the signal level input through the FRCK pin when the free-run timer is enabled.

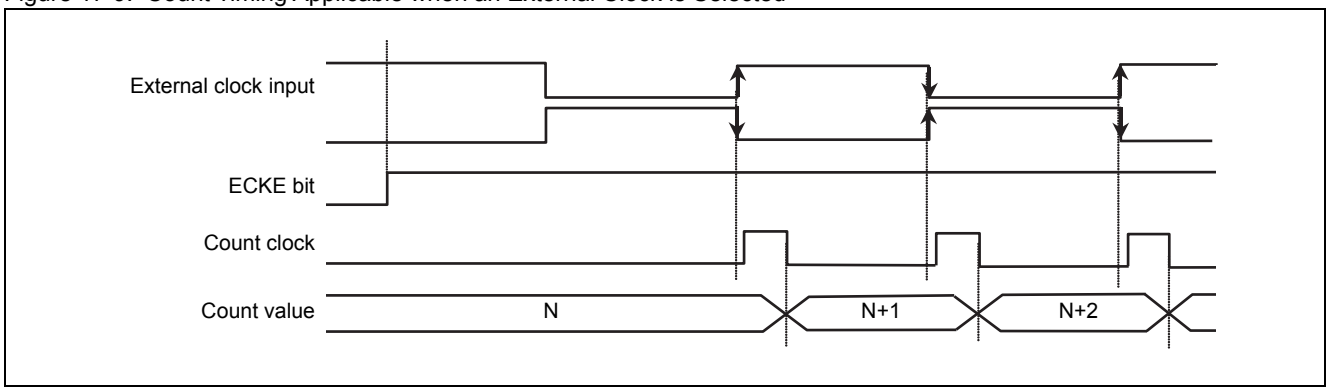
Table 17-6 lists the count timings applicable when an external clock is selected.

Table 17-6. Count Timings Applicable when an External Clock is Selected

Signal Level When Timer Is Enabled	Count Timing
"H" level	Starts counting at a rising edge and thereafter counts up at both edges.
"L" level	Starts counting at a falling edge and thereafter counts up at both edges.

Figure 17-9 shows the count timing applicable when an external clock is selected (ECKE=1).

Figure 17-9. Count Timing Applicable when an External Clock is Selected



Compare clear

Same as when an internal clock (peripheral clock) is selected. See "Compare clear" in "17.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected".

Interrupt processing

Same as when an internal clock (peripheral clock) is selected. See "Interrupt processing" in "17.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected".

18. 32-bit Input Capture

This chapter explains the functions and operations of the 32-bit input capture.

18.1 Overview

18.2 Configuration

18.3 Pins

18.4 Registers

18.5 Interrupts

18.6 An Explanation of Operations and Setting Procedure Examples

18.1 Overview

Upon detection of an input signal edge that is set in advance, the 32-bit input capture saves the value of the 32-bit free-run timer at the time.

This series microcontroller has 4 built-in input capture channels.

Overview

The 32-bit input capture is part of the compare timer. The compare timer comprises the following three functions:

- 32-bit free-run timer (1 channel)
  See "17. 32-bit Free-Run Timer".
- 32-bit output compare (4 channels)
  See "19. 32-bit Output Compare".
- 32-bit input capture (4 channels)

This chapter explains the 32-bit input capture.

- One of the following three triggers can be selected to save the value of the 32-bit free-run timer.
  - Rising edge
  - Falling edge
  - Both edges
- An interrupt request can be generated upon detection of an input signal edge that is set in advance.

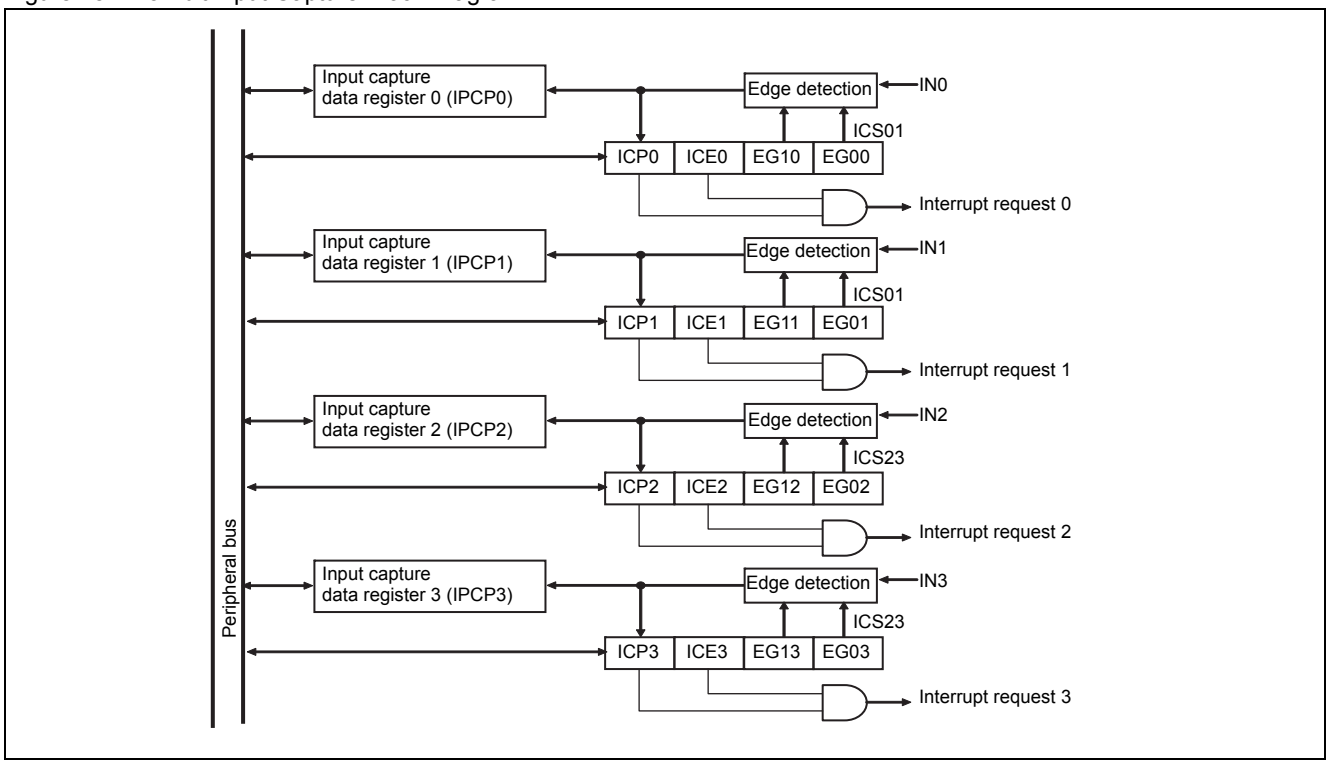
18.2 Configuration

This section explains the configuration of the 32-bit input capture.

32-bit input capture block diagram

Figure 18-1 is a block diagram of the 32-bit input capture.

Figure 18-1. 32-bit Input Capture Block Diagram



- Input capture data registers (IPCP0 to IPCP3)
  Free-run timer values are saved to these registers.
- Input capture status control registers (ICS01, ICS23)
  These registers are used to control the operation and state of the 32-bit input capture.

Note: For details of the compare timer block diagram, see "Compare timer block diagram" in "17. 32-bit Free-Run Timer".

Clocks

Table 18-1 lists the clock used for the 32-bit input capture.

Table 18-1. Clock used for 32-bit Input Capture

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

18.3 Pins

This section explains the pins used by the 32-bit input capture.

Overview

- IN0 to IN3 pins

  Input pins of 32-bit input capture. These pins are multiplexed pins. To use these pins as input pins of the 32-bit input capture, see "2.4 Setting Method for Pins".

Relationship between pins and channels

Table 18-2 lists the relationship between channels and pins.

Table 18-2. Relationship Between Channels and Pins

Channel	Input Pin
0	IN0
1	IN1
2	IN2
3	IN3

18.4 Registers

This section explains the configuration and functions of registers used by the 32-bit input capture.

Registers of 32-bit input capture

Table 18-3 lists the registers of the 32-bit input capture.

Table 18-3. Registers of 32-bit Input Capture

Channel	Abbreviated Register Name	Register Name	Reference
Common to 0 and 1	ICS01	Input capture status control register 01	18.4.1
Common to 2 and 3	ICS23	Input capture status control register 23	18.4.1
0	IPCP0	Input capture data register 0	18.4.2
1	IPCP1	Input capture data register 1	18.4.2
2	IPCP2	Input capture data register 2	18.4.2
3	IPCP3	Input capture data register 3	18.4.2

18.4.1 Input Capture Status Control Registers (ICS01, ICS23)

These registers are used to control the operation and state of the 32-bit input capture.

Figure 18-2 shows the bit configuration of the input capture status control register (ICS01, ICS23).

Figure 18-2. Bit Configuration of Input Capture Status Control Register (ICS01, ICS23)

bit	7	6	5	4	3	2	1	0
	ICPm	ICPn	ICEm	ICEn	EG1m	EG0m	EG1n	EG0n
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

R/W: Read/Write

[bit7, bit6]: ICPm, ICPn (interrupt request flag bit)

Each of these bits indicates that a valid edge has been detected at pins IN0 to IN3. When this bit is "1" while ICEm or ICEn bit is set to "1", an edge detection interrupt request is generated.

The ICPm bit corresponds to the odd-numbered channel, and the ICPn bit corresponds to the even-numbered channel.

ICPm, ICPn	In Case of Reading	In Case of Writing
0	A valid edge is not detected.	This bit is cleared to "0".
1	A valid edge is detected.	Ignored

Table 18-4 lists the relationship between the ICPm bits and ICPn bits and channels.

Table 18-4. Relationship between bits and channels

Input Capture Status Registers	ICPm Bit	Supported Channel	ICPn Bit	Supported Channel
ICS01	ICP1	ch.1	ICP0	ch.0
ICS23	ICP3	ch.3	ICP2	ch.2

Note: When a read-modify-write instruction is used, "1" is read.

[bit5, bit4]: ICEm, ICEn (interrupt request enable bits)

Each of these bits specifies whether to generate an edge detection interrupt request when a valid edge is detected through pins IN0 to IN3 (ICPm, ICPn=1).

The ICEm bit corresponds to the odd-numbered channel, and the ICEn bit corresponds to the even-numbered channel.

Written Value	Explanation
0	Disables generation of edge detection interrupt requests.
1	Enables generation of edge detection interrupt requests.

Table 18-5 shows the relationship between the ICEm bits and ICEn bits and channels.

Table 18-5. Relationship Between Bits and Channels

Input Capture Status Registers	ICEm Bit	Supported Channel	ICEn Bit	Supported Channel
ICS01	ICE1	ch.1	ICE0	ch.0
ICS23	ICE3	ch.3	ICE2	ch.2

[bit3, bit2]: EG1m, EG0m (edge selection bits)

These bits select a valid edge for the 32-bit input capture of the odd-numbered channel.

When the edge selected here is detected, the value of the 32-bit free-run timer is saved to the input capture data register (IPCP0 to IPCP3).

EG1m	EG0m	Explanation
0	0	No edge detected (input capture stopped)
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Table 18-6 shows the relationship between the EG1m and EG0m bits and channels.

Table 18-6. Relationship Between Bits and Channels

Input Capture Status Registers	EG1m, EG0m Bits	Supported Channel
ICS01	EG11, EG01	ch.1
ICS23	EG13, EG03	ch.3

Note: If a value other than "00" is written to these bits, the operation of the corresponding channel is enabled at the same time as a valid edge is selected.

[bit1, bit0]: EG1n, EG0n (edge selection bits)

These bits select a valid edge for the 32-bit input capture of the even-numbered channel.

When the edge selected here is detected, the value of the 32-bit free-run timer is saved to the input capture data register (IPCP0 to IPCP3).

EG1n	EG0n	Explanation
0	0	No edge detected (input capture stopped)
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

The bit names of EG1n and EG0n vary depending on the channel.

Table 18-7 shows the relationship between bits and channels.

Table 18-7. Relationship Between Bits and Channels

Input Capture Status Registers	EG1n, EG0n Bits	Supported Channel
ICS01	EG10, EG00	ch.0
ICS23	EG12, EG02	ch.2

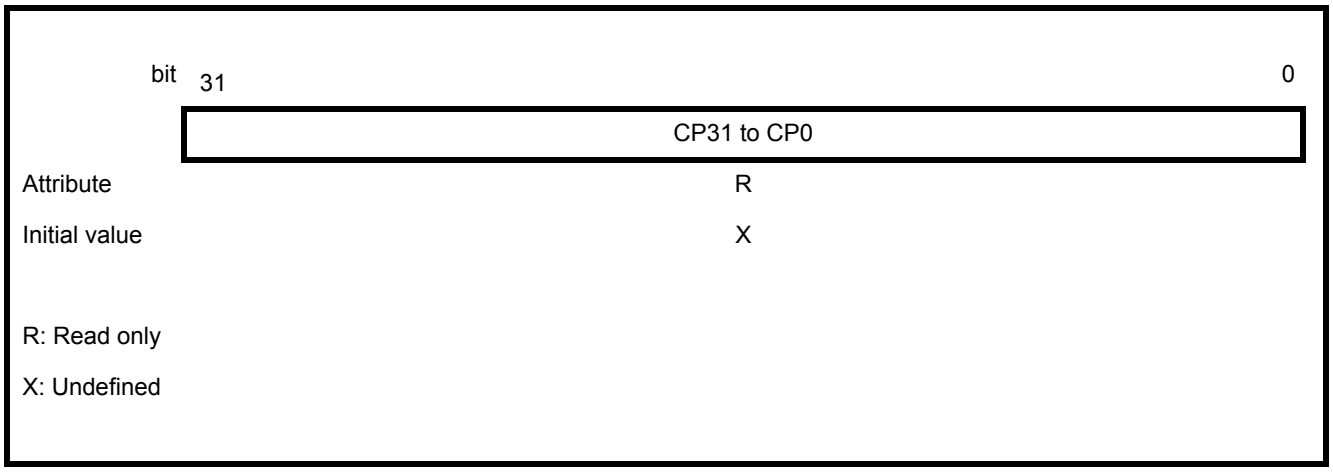
Note: If a value other than "00" is written to these bits, the operation of the corresponding channel is enabled at the same time as a valid edge is selected.

18.4.2 Input Capture Data Register (IPCP0 to IPCP3)

This register saves the value of the 32-bit free-run timer. When a valid edge is detected in the input signal through pins IN0 to IN3, the value of the 32-bit free-run timer is saved to this register.

Figure 18-3 shows the bit configuration of the input capture data register (IPCP0 to IPCP3).

Figure 18-3. Bit Configuration of Input Capture Data Register (IPCP0 to IPCP3)



Note: Be sure to read this register in units of words.

18.5 Interrupts

Upon detection of a valid edge in the input signal through pins IN0 to IN3, an interrupt request is generated (edge detection interrupt request).

Table 18-8 outlines the interrupts that can be used with the 32-bit input capture.

Table 18-8. Interrupts of the 32-bit Input Capture

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Edge detection interrupt request	Even-numbered channel: ICPn=1 for ICS Odd-numbered channel: ICPm=1 for ICS	Even-numbered channel: ICEn=1 for ICS Odd-numbered channel: ICEm=1 for ICS	Write "0" to the next bit. Even-numbered channel: ICPn bit for ICS Odd-numbered channel: ICPm bit for ICS

ICS: input capture status control register (ICS01, ICS23)

Notes:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests at the same time with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "10. Interrupt Controller".

18.6 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the 32-bit input capture. Also, examples of procedures for setting the operating state are shown.

18.6.1 Explanation of 32-bit Input Capture Operation

Upon detection of an input signal edge that is set in advance, the 32-bit input capture saves the value of the 32-bit free-run timer at the time.

Operation

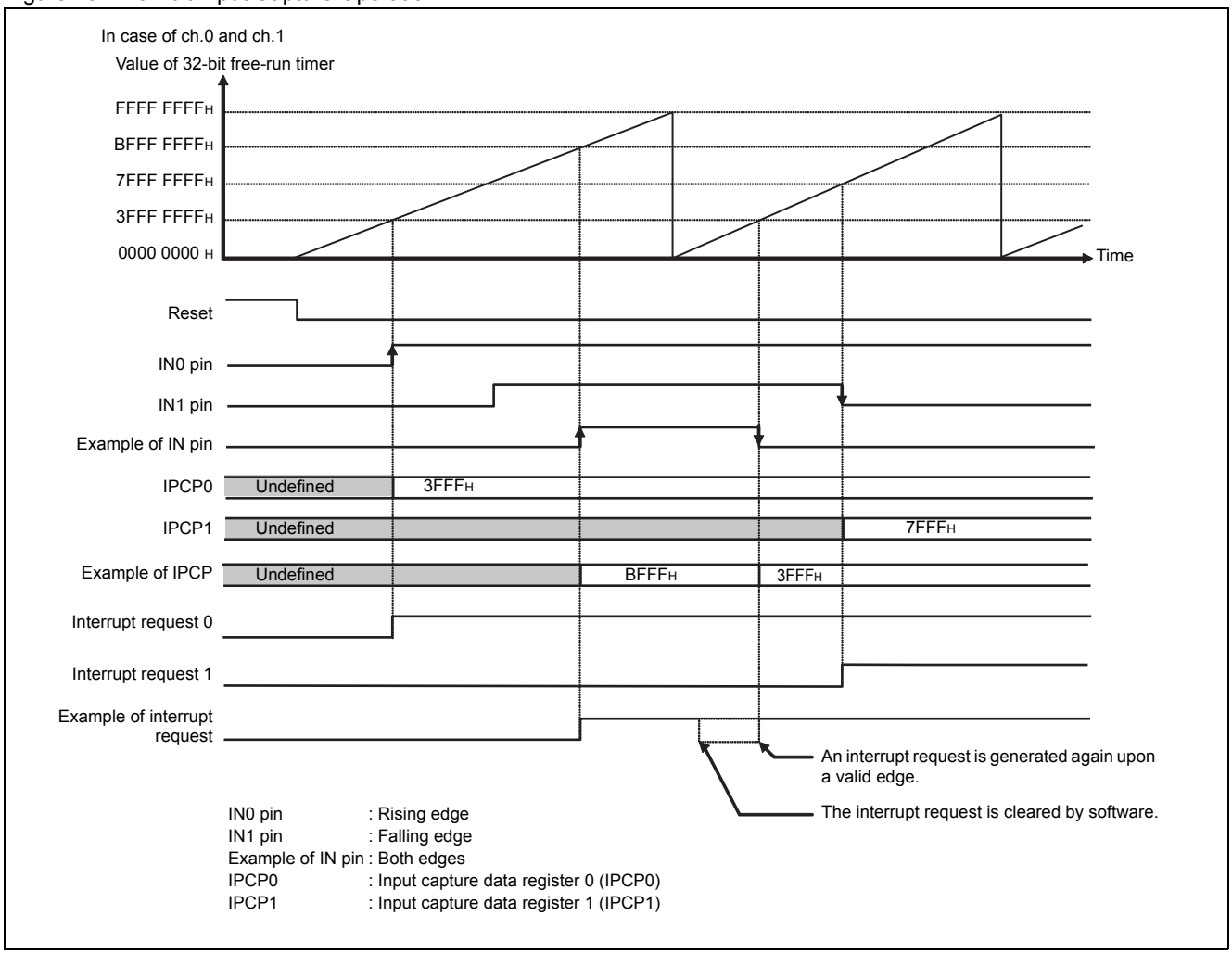
Selecting a valid edge with the following bits of the input capture status control register (ICS01, ICS23) enables 32-bit input capture operation.

- Selecting valid edge of odd-numbered channel/enabling operation: EG1m, EG0m
- Selecting valid edge of even-numbered channel/enabling operation: EG1n, EG0n

When a valid edge is detected at pins IN0 to IN3 while 32-bit input capture operation is enabled, the value of the 32-bit free-run timer at the time is saved to the input capture data register (IPCP0 to IPCP3). If interrupt request generation has been enabled, an edge detection interrupt request is generated.

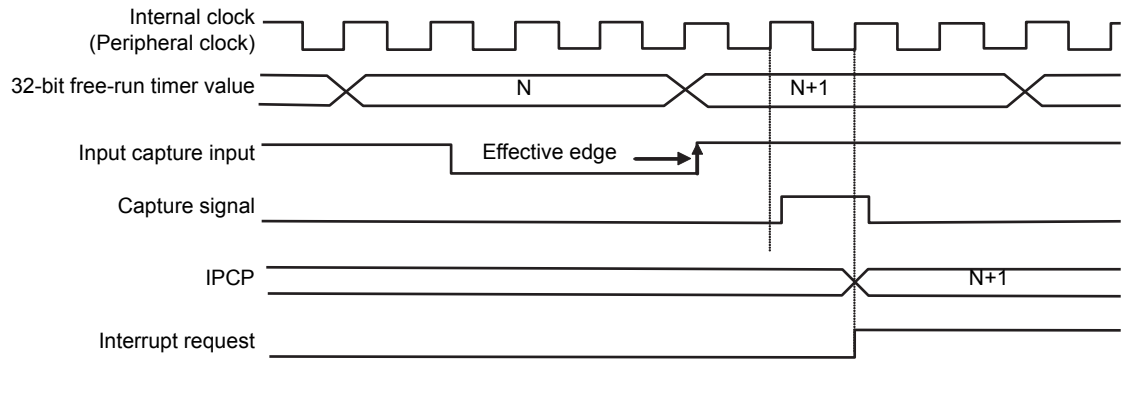
Figure 18-4 shows the 32-bit input capture operation.

Figure 18-4. 32-bit Input Capture Operation



When a valid edge is detected, a capture signal is generated to synchronize with the internal clock (peripheral clock). The generation of interrupt requests and the saving of 32-bit free-run timer values are performed based on the capture signals. Figure 18-5 shows an example of capture signal timing.

Figure 18-5. Example of capture signal timing



IPCP: Input capture data register (IPCP0 to IPCP3)

19. 32-bit Output Compare

This chapter explains the functions and operations of the 32-bit output compare.

19.1 Overview

19.2 Configuration

19.3 Pins

19.4 Registers

19.5 Interrupts

19.6 An Explanation of Operations and Setting Procedure Examples

19.1 Overview

After 32-bit free-run timer counts up to the preset value, the 32-bit output compare function inverts the level of output from a pin or generates an interrupt request.

This series microcontroller has 4 built-in channels for the 32-bit output compare.

Overview

The 32-bit output compare is part of the compare timer. The compare timer comprises the following three functions:

- 32-bit free-run timer (1 channel)

  See "17. 32-bit Free-Run Timer".
  - 32-bit output compare (4 channels)
  - 32-bit input capture (4 channels)

  See "18. 32-bit Input Capture".

This chapter explains the 32-bit output compare.

- 2 channels of the 32-bit output compare can be used either independently of each other or as a pair.

  If the 2 channels of the 32-bit output compare are used as a pair, comparison can be performed by 2 channels at one time and thus the CPU load can be reduced.

  The combinations of channels that can be used as pairs are as follows:
  - ch.0 and ch.1
  - ch.2 and ch.3
- The output levels at the OUT0 to OUT3 pins at the time of activation of the 32-bit output compare can be set.
- An interrupt request can be generated when the count value of the 32-bit free-run timer matches the preset value (compare value).

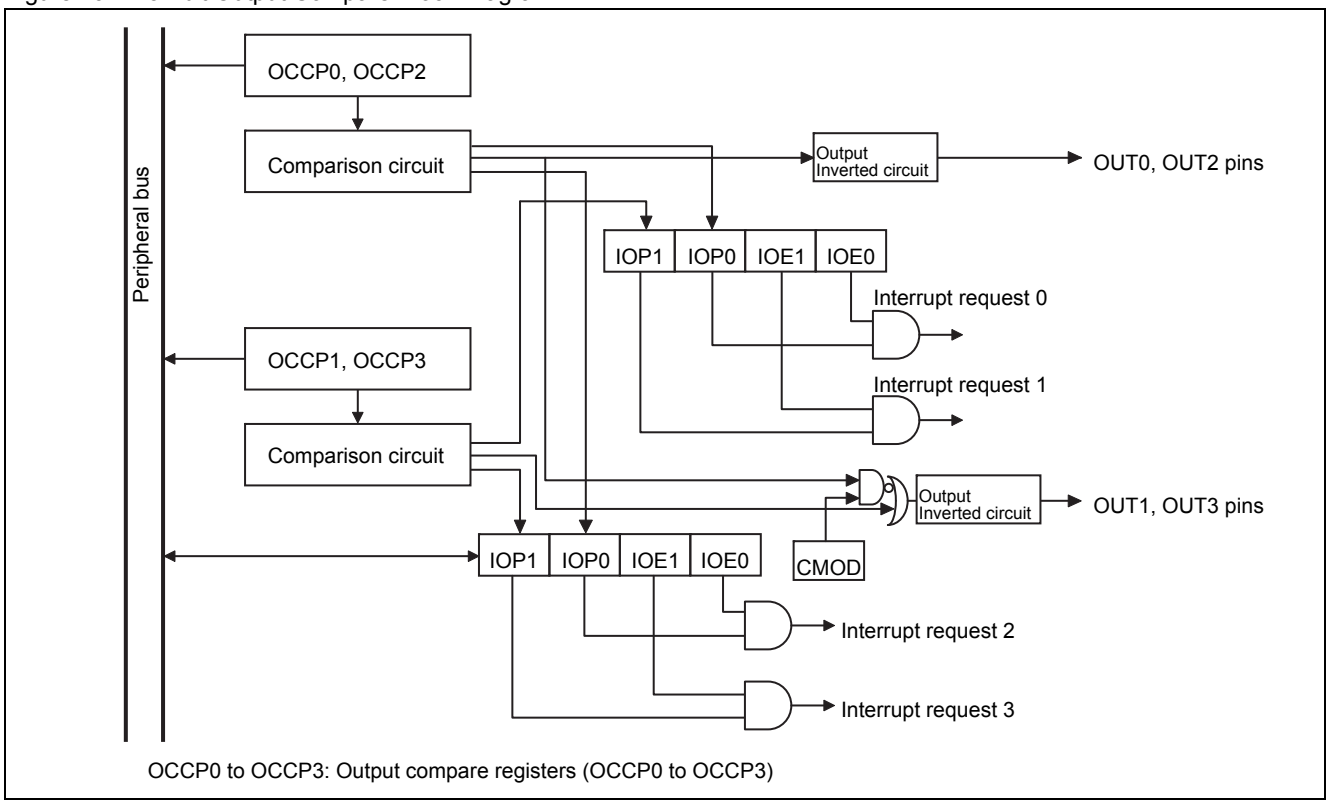
19.2 Configuration

This section explains the configuration of the 32-bit output compare.

32-bit output compare block diagram

Figure 19-1 is a block diagram of the 32-bit output compare.

Figure 19-1. 32-bit Output Compare Block Diagram



- Output compare register (OCCP0 to OCCP3)

  This register sets the value (compare value) to be compared with the count value of the 32-bit free-run timer.
- Compare control register

  This register controls the operation of the 32-bit output compare. This register is divided into the following two registers:
  - Compare control register upper (OCSH1, OCSH3)
  - Compare control register lower (OCSL0, OCSL2)
- Comparison circuit

  This circuit compares the count value of the 32-bit free-run timer and the compare value that is set in the output compare register (OCCP0 to OCCP3).

Note: For details of the compare timer block diagram, see "Compare timer block diagram" in "17. 32-bit Free-Run Timer".

Clocks

Table 19-1 lists the clock used for the 32-bit output compare.

Table 19-1. Clock used for 32-bit Output Compare

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

19.3 Pins

This section explains the pins used by the 32-bit output compare.

Overview

- OUT0 to OUT3 pins

  These are the output pins of the 32-bit output compare. These pins are multiplexed pins.

  For details of how to use these pins as the OUT0 to OUT3 pins of the 32-bit output compare, see "2.4 Setting Method for Pins".

Relationship between pins and channels

Table 19-2 lists the relationship between channels and pins.

Table 19-2. Relationship Between Channels and Pins

Channel	Output Pin
0	OUT0
1	OUT1
2	OUT2
3	OUT3

19.4 Registers

This section explains the configuration and functions of the registers used by the 32-bit output compare.

32-bit output compare registers

Table 19-3 lists the registers of the 32-bit output compare.

Table 19-3. Registers of 32-bit Output Compare

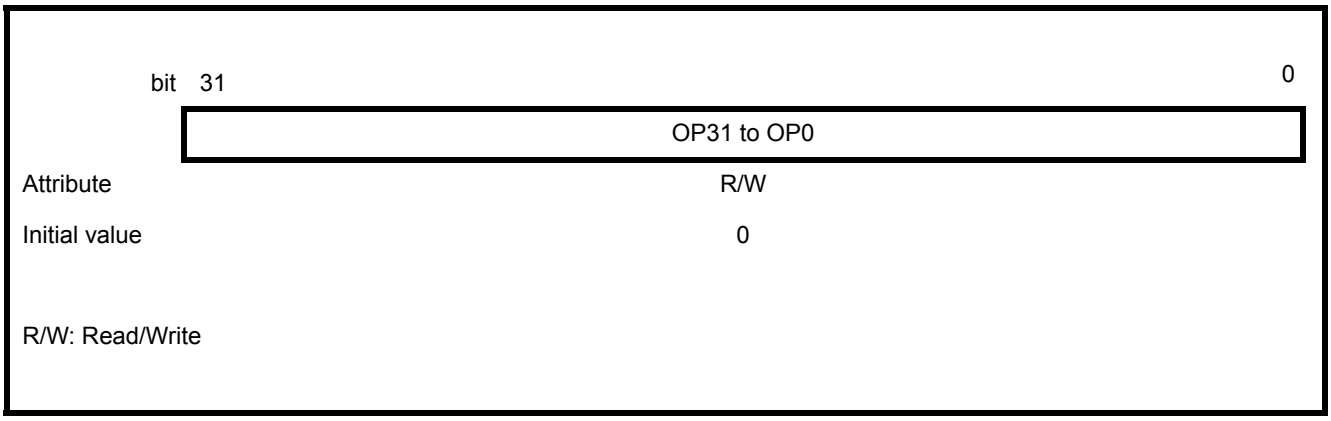
Channel	Abbreviated Register Name	Register Name	Reference
Common to 0 and 1	OCSH1	Compare control register upper1	19.4.2
	OCSL0	Compare control register lower 0	19.4.3
Common to 2 and 3	OCSH3	Compare control register upper 3	19.4.2
	OCSL2	Compare control register lower 2	19.4.3
0	OCCP0	Output compare register 0	19.4.1
1	OCCP1	Output compare register 1	19.4.1
2	OCCP2	Output compare register 2	19.4.1
3	OCCP3	Output compare register 3	19.4.1

19.4.1 Output Compare Register (OCCP0 to OCCP3)

This register sets the value (compare value) to be compared with the count value of the 32-bit free-run timer. Set the compare value in this register before activating the 32-bit free-run timer.

Figure 19-2 shows the bit configuration of the output compare register (OCCP0 to OCCP3).

Figure 19-2. Bit Configuration of Output Compare Register (OCCP0 to OCCP3)



Notes:

- This register can be rewritten even while the 32-bit free-run timer is active.
- The value written to this register is immediately used as a compare value. Therefore, if the compare value is rewritten from a small value to a large value during operation of the 32-bit free-run timer, an interrupt request is generated twice while the 32-bit free-run timer counts once.

  To prevent this problem, rewrite this register by using interrupt processing by the 32-bit free-run timer.
- Be sure to access this register in units of words (32 bits).

19.4.2 Compare Control Register Upper (OCSH1, OCSH3)

This register is used to specify whether to use the 2 channels of the 32-bit output compare independently of each other or as a pair. The register is also used to set the level of signals output through the OUT0 to OUT3 pins when the 32-bit output compare function is activated.

Figure 19-3 shows the bit configuration of the compare control register upper (OCSH1, OCSH3).

Figure 19-3. Bit Configuration of Compare Control Register Upper (OCSH1, OCSH3)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	CMOD	Undefined	Undefined	OTD1	OTD0
Attribute	-	-	-	R/W	-	-	R/W	R/W
Initial value	X	X	X	0	X	X	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit15 to bit13]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit12]: CMOD (output level invert mode bit)

This bit is used to specify whether to use the 2 channels of the 32-bit output compare independently of each other or as a pair. The invert mode of wave forms output from pins changes depending on this setting.

Written Value	Explanation
0	2 channels of the 32-bit output compare are used independently of each other. When the compare value of the output compare register (OCCP0 to OCCP3) matches the count value of the 32-bit free-run timer, the output level from the corresponding pin is inverted.
1	2 channels of the 32-bit output compare are used as a pair. When the compare value of the output compare register (OCCP0 to OCCP3) matches the value of the 32-bit free-run timer, the invert mode is inverted as shown below: When the count value matches the compare value of the even-numbered channel output compare register (OCCP0, OCCP2): the output levels from the following pins are inverted: ■ Output level from the pin corresponding to the channel ■ Output level from the pin corresponding to the odd-numbered channel used as a pair. When the count value matches the compare value of the odd-numbered channel output compare register (OCCP1, OCCP3): the output level from the following pin is inverted: ■ Output level from the pin corresponding to the channel

Table 19-4 summarizes the invert timings for output levels from OUT0 to OUT3 pins when "1" is set to this bit.

Table 19-4. Output Level Invert Timing

Register Whose Compare Value Matches the Value of the 32-bit Free-run Timer	Pin Whose Output Level Inverts
Output compare register 0 (OCCP0)	OUT0 pin, OUT1 pin
Output compare register 1 (OCCP1)	OUT1 pin
Output compare register 2 (OCCP2)	OUT2 pin, OUT3 pin
Output compare register 3 (OCCP3)	OUT3 pin

Notes:

- If the same compare value is set for the even-numbered and odd-numbered channels of the 32-bit output compare, the operation is the same as when the 2 channels of the 32-bit output compare are used independently of each other, even when "1" is set to this bit.
- Be sure to set "1" to this bit when the 2 channels of the 32-bit output compare are used as a pair.

[bit11, bit10]: Reserved bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit9]: OTD1 (output level bit)

This bit sets the signal level output from pins (OUT1, OUT3) when the odd-numbered channel of the 32-bit output compare is activated.

OTD1	In Case of Writing	In Case of Reading
0	The "L" level is output.	The output level is read.
1	The "H" level is output.	

Note: Do not rewrite this bit during 32-bit output compare operation.

[bit8]: OTD0 (output level bit)

This bit sets the signal level output from pins (OUT0, OUT2) when the even-numbered channels of the 32-bit output compare are activated.

OTD0	In Case of Writing	In Case of Reading
0	The "L" level is output.	The output level is read.
1	The "H" level is output.	

Note: Do not rewrite this bit during 32-bit output compare operation.

19.4.3 Compare Control Register Lower (OCSL0, OCSL2)

This register enables or disables 32-bit output compare operation or controls interrupt requests.

Figure 19-4 shows the bit configuration of the compare control register lower (OCSL0, OCSL2).

Figure 19-4. Bit Configuration of Compare Control Register Lower (OCSL0, OCSL2)

bit	7	6	5	4	3	2	1	0
	IOP1	IOP0	IOE1	IOE0	Undefined	Undefined	CST1	CST0
Attribute	R/W	R/W	R/W	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	X	X	0	0

R/W: Read/Write

–: Undefined

X: Undefined

[bit7]: IOP1 (odd-numbered channel compare match interrupt request flag bit)

This bit indicates that the compare value of the odd-numbered channel output compare register (OCCP1, OCCP3) matches the count value of the 32-bit free-run timer.

If "1" is set to the IOE1 bit when this bit is "1", a compare match interrupt request is generated.

IOP1	In Case of Reading	In Case of Writing
0	A comparison result indicates no match.	This bit is cleared to "0".
1	A comparison result indicates a match.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit6]: IOP0 (even-numbered channel compare match interrupt request flag bit)

This bit indicates that the compare value of the even-numbered channel output compare register (OCCP0, OCCP2) matches the count value of the 32-bit free-run timer.

If "1" is set to the IOE0 bit when this bit is "1", a compare match interrupt request is generated.

IOP0	In Case of Reading	In Case of Writing
0	A comparison result indicates no match.	This bit is cleared to "0".
1	A comparison result indicates a match.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit5]: IOE1 (odd-numbered channel compare match interrupt enable bit)

This bit specifies whether to generate a compare match interrupt request when the value of the odd-numbered channel output compare register (OCCP1, OCCP3) matches the count value of the 32-bit free-run timer (IOP1=1).

Written Value	Explanation
0	Disables generation of compare match interrupt requests.
1	Enables generation of compare match interrupt requests.

[bit4]: IOE0 (even-numbered channel compare match interrupt enable bit)

This bit specifies whether to generate a compare match interrupt request when the value of the even-numbered channel output compare register (OCCP0, OCCP2) matches the count value of the 32-bit free-run timer (IOP0=1).

Written Value	Explanation
0	Disables generation of compare match interrupt requests.
1	Enables generation of compare match interrupt requests.

[bit3, bit2]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit1]: CST1 (odd-numbered channel compare enable bit)

This bit enables or disables the comparison between odd-numbered channel 32-bit output compare and the count value of the 32-bit free-run timer.

Written Value	Explanation
0	Disables comparison.
1	Enables comparison.

Note: When the 32-bit free-run timer is stopped, the comparison of 32-bit output compare is also stopped.

[bit0]: CST0 (even-numbered channel compare enable bit)

This bit enables or disables the comparison between even-numbered channel 32-bit output compare and the count value of the 32-bit free-run timer.

Written Value	Explanation
0	Disables comparison.
1	Enables comparison.

Note: When the 32-bit free-run timer is stopped, the comparison of 32-bit output compare is also stopped.

19.5 Interrupts

An interrupt request (compare match interrupt request) is generated when the count value of the 32-bit free-run timer matches the value set in the output compare register (OCCP0 to OCCP3).

Table 19-5 outlines the interrupts that can be used with the 32bit output compare.

Table 19-5. Interrupts of the 32-bit Output Compare

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Compare result match interrupt request	Even-numbered channel: IOP0=1 for OCSL Odd-numbered channel: IOP1=1 for OCSL	Even-numbered channel: IOE0=1 for OCSL Odd-numbered channel: IOE1=1 for OCSL	Write "0" to the next bit Even-numbered channel: IOP0 bit for OCSL Odd-numbered channel: IOP1 bit for OCSL

OCSL: compare control register lower (OCSL0, OCSL2)

Notes:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use an interrupt control register (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "10. Interrupt Controller".

19.6 An Explanation of Operations and Setting Procedure Examples

This section explains the 32-bit output compare operation. Also, examples of procedures for setting the operating state are shown.

Overview

2 channels of the 32-bit output compare can be used either independently of each other or as a pair.

19.6.1 When the 2 Channels Are Used Independently of Each Other

This section explains the 32-bit output compare operation when the 2 channels are used independently of each other.

Overview

When the CMOD bit of the compare control register upper (OCSH1, OCSH3) is set to "0", the 2 channels of the 32-bit output compare operate independently of each other.

The output level of the pin corresponding to the channel is inverted when the count value of the 32-bit free-run timer matches the compare value of the output compare register (OCCP0 to OCCP3).

Operation

Writing "1" to the following bit enables the 32-bit output compare operation.

- Enabling even-numbered channel operation: CST0 bit of compare control register lower (OCSL0, OCSL2)
- Enabling odd-numbered channel operation: CST1 bit of compare control register lower (OCSL0, OCSL2)

When the count value of the 32-bit free-run timer matches the compare value of the output compare register (OCCP0 to OCCP3) while the 32-bit output compare is enabled, the following bits are set to "1":

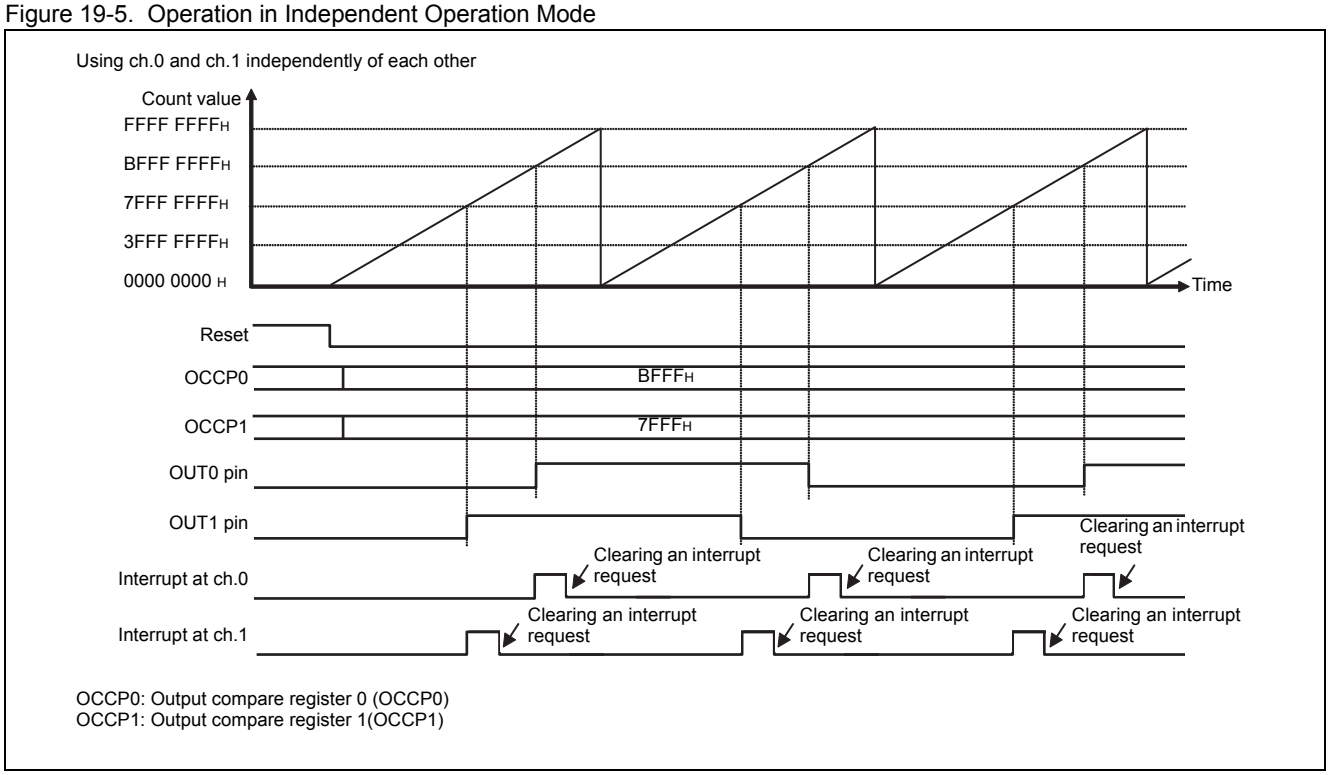
- Even-numbered channel: IOP0 bit of compare control register lower (OCSL0, OCSL2)
- Odd-numbered channel: IOP1 bit of compare control register lower (OCSL0, OCSL2)

If interrupt request generation has been enabled, a compare match interrupt request is generated.

Also, the output levels from the OUT0 to OUT3 pins are inverted.

Figure 19-5 shows the operation in independent operation mode.

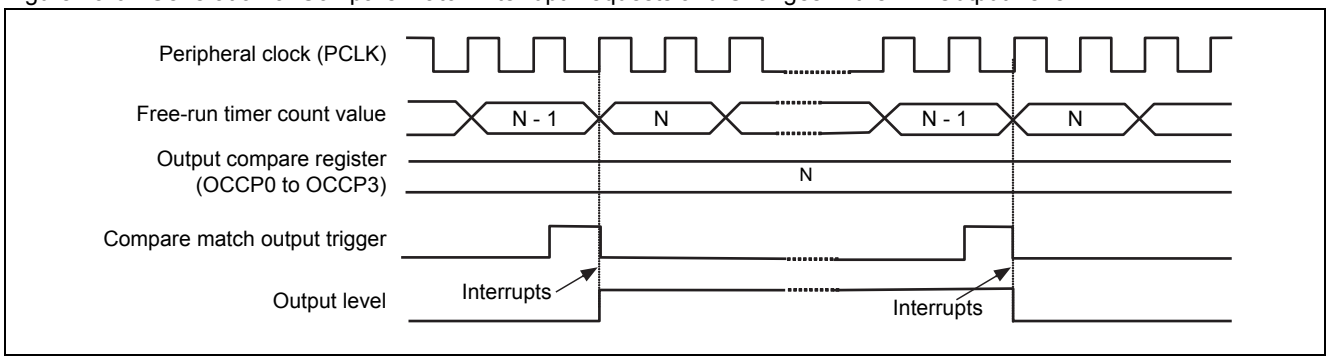
Figure 19-5. Operation in Independent Operation Mode



A compare match interrupt request or a change in the pin output level occurs upon detection of a compare match.

Figure 19-6 shows the generation of compare match interrupt requests and changes in the pin output level.

Figure 19-6. Generation of Compare Match Interrupt Requests and Changes in the Pin Output Level



Note: When using 2 channels of the 32-bit output compare independently of each other, be sure to write "0" to the CMOD bit of the compare control register upper (OCSH1, OCSH3).

19.6.2 When the 2 Channels Are Used as a Pair

This section explains the 32-bit output compare operation using the even-numbered and odd-numbered channels in pairs.

Overview

When the CMOD bit of the compare control register upper (OCSH1, OCSH3) is set to "1", the 2 channels of the 32-bit output compare operate in pairs.

By using the even-numbered and odd-numbered channels of the 32-bit output compare in pairs, compare values for 2 channels can be updated by 1 interrupt.

The combinations of even-numbered and odd-numbered channels that can be used in pairs are as follows:

- ch.0 and ch.1
- ch.2 and ch.3

Operation

Writing "1" to the following bit enables the 32-bit output compare operation.

- Enabling even-numbered channel operation: CST0 bit of compare control register lower (OCSL0, OCSL2)
- Enabling odd-numbered channel operation: CST1 bit of compare control register lower (OCSL0, OCSL2)

When the count value of the 32-bit free-run timer matches the compare value of the output compare register (OCCP0 to OCCP3) while the 32-bit output compare is enabled, the following bits are set to "1":

- Even-numbered channel: IOP0 bit of compare control register lower (OCSL0, OCSL2)
- Odd-numbered channel: IOP1 bit of compare control register lower (OCSL0, OCSL2)

If interrupt request generation has been enabled, a compare match interrupt request is generated.

Also, the output levels from the OUT0 to OUT3 pins are inverted. The pin whose output level is inverted varies depending on the channel of the output compare register (OCCP0 to OCCP3) whose compare value matches the count value of the 32-bit free-run timer.

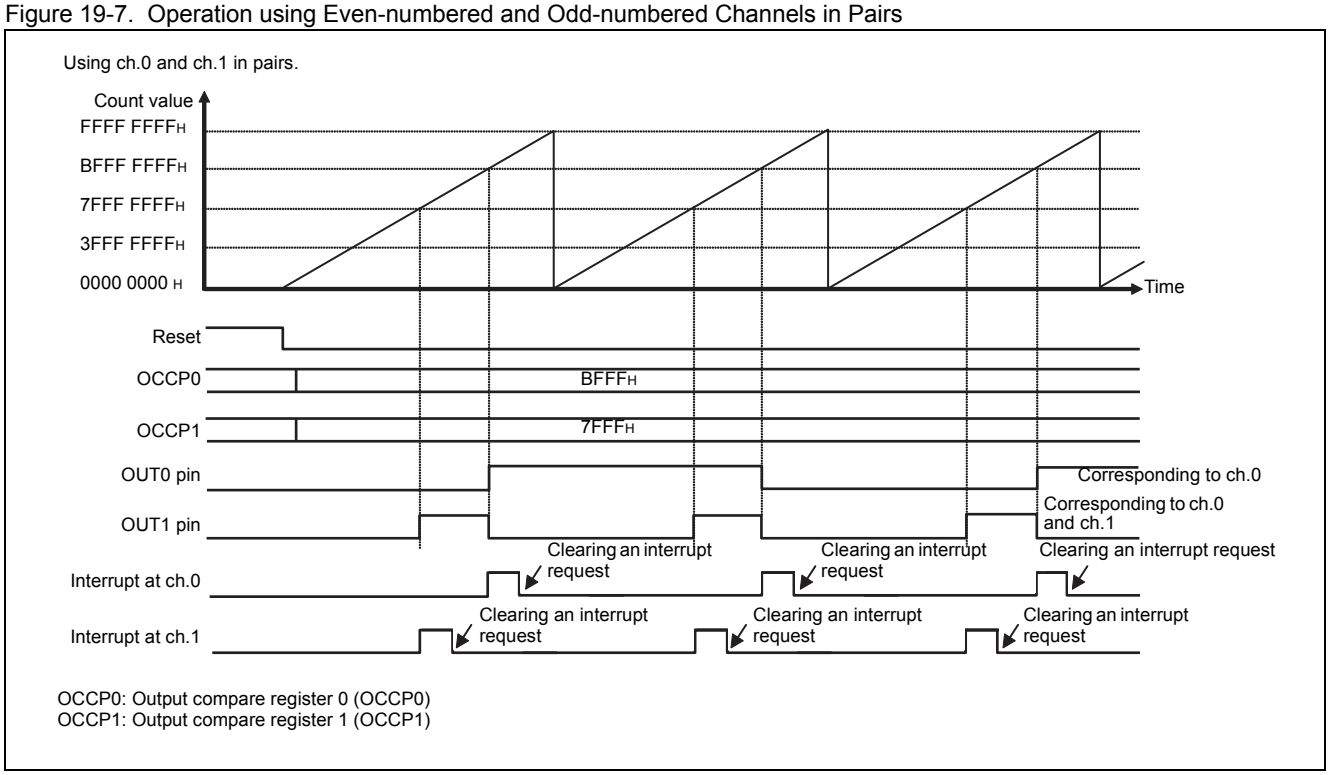
Table 19-6 shows the relationship between the channels for which compare values are set and the pins whose output levels are inverted.

Table 19-6. Relationship Between the Channels for which Compare Values are Set and the Pins whose Output Levels are Inverted

Register Whose Compare Value Matches the Value of the 32-bit Free-run Timer	Pin Whose Output Level Inverts
Output compare register 0 (OCCP0)	OUT0 pin, OUT1 pin
Output compare register 1 (OCCP1)	OUT1 pin
Output compare register 2 (OCCP2)	OUT2 pin, OUT3 pin
Output compare register 3 (OCCP3)	OUT3 pin

Figure 19-7 shows the operation using even-numbered and odd-numbered channels in pairs.

Figure 19-7. Operation using Even-numbered and Odd-numbered Channels in Pairs



A compare match interrupt request or a change in the pin output level occurs upon detection of a compare match.

See "19.6.1 When the 2 Channels Are Used Independently of Each Other" for details of the generation of compare match interrupt requests and changes in the pin output level.

Note: When using even-numbered and odd-numbered channels of the 32-bit output compare in pairs, be sure to write "1" to the CMOD bit of the compare control register upper (OCSH1, OCSH3).

20. 16-bit Reload Timer

This chapter explains the functions and operations of the 16-bit reload timer.

20.1 Overview

20.2 Configuration

20.3 Pins

20.4 Registers

20.5 Interrupts

20.6 An Explanation of Operations and Setting Procedure Examples

20.7 Notes on Use

20.1 Overview

The 16-bit reload timer is a down counter that performs a countdown from a preset value. This timer can be used as an interval timer that counts down synchronously with an internal clock (peripheral clock), and it can also be used as an event counter that counts external events.

This series has 3 built-in channels of the 16-bit reload timer.

Overview

- Timer mode: Internal timer mode and event counter mode are available.
  - Interval timer mode

    It counts down synchronously with an internal clock (peripheral clock). The internal clock (peripheral clock) is selected from 6 clock types, which are peripheral clocks (PCLK) divided by 2, 4, 8, 16, 32, and 64.
  - Event counter mode

    It detects and counts the edges (rising edge/falling edge/both edges) of the external clock.

    Cascade mode that counts ch.0 outputs with ch.1 and ch.1 outputs with ch.2 is also available.
- Operation mode: One of the following two modes can be selected.
  - Reload mode

    In this mode, the reload value is reloaded, and counting is repeated when the down counter enters an underflow condition.
  - One shot mode

    In this mode, counting stops when the down counter enters an underflow condition.
- Input pin function: In interval timer mode, the trigger input function or gate input function can be selected for the input pin function.
  - Trigger input function

    When it detects a valid edge (rising edge/falling edge/both edges) from the input pin, it starts counting.
  - Gate input function

    It continues counting as long as the input pin maintains its effective level of input.
- Interrupt request: It can generate an interrupt request when the down counter enters an underflow condition.

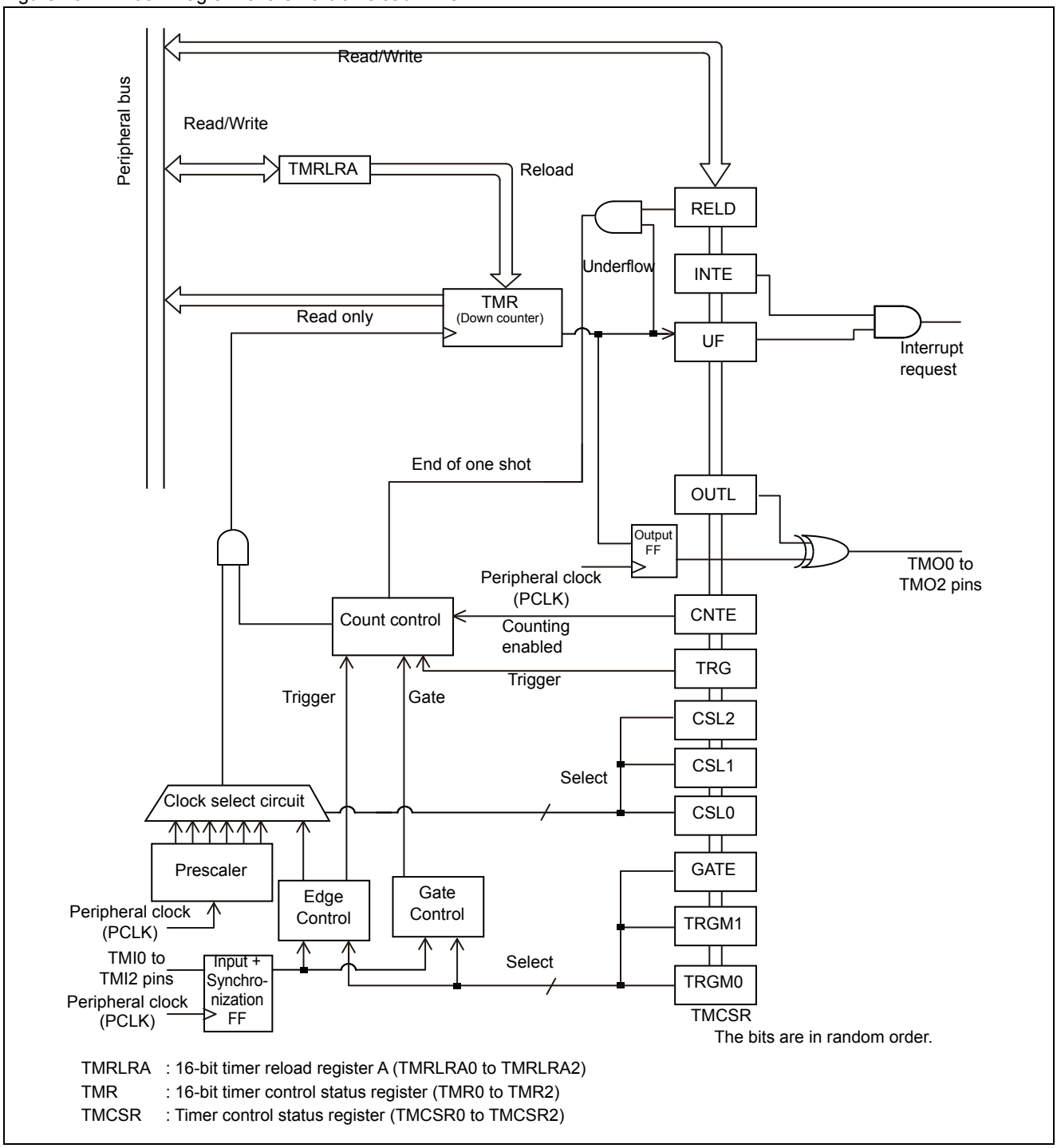
20.2 Configuration

This section explains the 16-bit reload timer configuration.

Block diagram of the 16-bit reload timer

Figure 20-1 is a block diagram of the 16-bit reload timer.

Figure 20-1. Block Diagram of the 16-bit Reload Timer



- Timer control status register (TMCSR0 to TMCSR2)
  This register controls the operations of the 16-bit reload timer.
- 16-bit timer reload register A (TMRLRA0 to TMRLRA2)
  This register sets the reload values.
- 16-bit timer register (TMR0 to TMR2)
  This register operates as a down counter. When this register is read, the down counter value can be read.
- Prescaler
  When the interval timer mode is selected, the prescaler divides the peripheral clock (PCLK).
- Clock select circuit
  The clock select circuit selects a count clock.
- Edge controller
  The edge controller controls the detection edges of signals when the TMI0 to TMI2 pins are used as trigger input pins.
- Gate controller
  The gate controller controls the signal levels of the signals input from the pins when the TMI0 to TMI2 pins are used as gate input pins.
- Count controller
  The count controller controls the counts of the 16-bit reload timer.

Clocks

Table 20-1 shows the clock used for the 16-bit reload timer.

Table 20-1. Clock used for the 16-bit Reload Timer

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Internal clock (peripheral clock)	Created through division of the peripheral clock (PCLK).
	External clock	Input from TMI0 to TMI2 pins

20.3 Pins

This section explains the pins of the 16-bit reload timer.

Overview

There are two types of 16-bit reload timer as follows.

- TMO0 to TMO2 pins

  16-bit reload timer wave form output pin

  These pins are multiplexed pins. For information on using as the wave form output pin of the 16-bit reload timer, see "2.4 Setting Method for Pins".

- TMI0 to TMI2 pins

  16-bit reload timer input pin This inputs count clock, clock, trigger, or gate depending on its setting.

  These pins are multiplexed pins. For information on using as the input pin of the 16-bit reload timer, see "2.4 Setting Method for Pins".

Relationship between pins and channels

Table 20-2 outlines the relationship between channels and pins.

Table 20-2. Relationship Between Channels and Pins

Channel	Wave Form Output Pin	Input Pin
0	TMO0	TMI0
1	TMO1	TMI1
2	TMO2	TMI2

20.4 Registers

This section explains the configuration and functions of registers used by the 16-bit reload timer.

Registers of 16-bit reload timer

Table 20-3 lists the registers of the 16-bit reload timer.

Table 20-3. Registers of 16-bit Reload Timer

Channel	Abbreviated Register Name	Register Name	Reference
0	TMCSR0	Timer control status register 0	20.4.1
	TMRLRA0	16-bit timer reload register A0	20.4.2
	TMR0	16-bit timer register 0	20.4.3
1	TMCSR1	Timer control status register 1	20.4.1
	TMRLRA1	16-bit timer reload register A1	20.4.2
	TMR1	16-bit timer register 1	20.4.3
2	TMCSR2	Timer control status register 2	20.4.1
	TMRLRA2	16-bit timer reload register A2	20.4.2
	TMR2	16-bit timer register 2	20.4.3

20.4.1 Timer Control Status Register (TMCSR0 to TMCSR2)

This register controls the operations of the 16-bit reload timer.

Figure 20-2 shows the bit configuration of the timer control status registers (TMCSR0 to TMCSR2).

Figure 20-2. Bit Configuration of the tImer Control Status Registers (TMCSR0 to TMCSR2)

	bit	15	14	13	12	11	10	9	8
		Reserved	Reserved	TRGM1	TRGM0	CSL2	CSL1	CSL0	GATE
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
		Undefined	Undefined	OUTL	RELD	INTE	UF	CNTE	TRG
Attribute		-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		X	X	0	0	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit15, bit14]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit13, bit12]: TRGM1, TRGM0 (Input pin operation selection bit)

This bit selects the operation of TMI0 to TMI2 pins of the 16-bit reload timer.The meaning of this bit varies depending whether the 16-bit reload timer is used in interval timer mode, or in event counter mode.

- Interval timer mode (CSL2 to CSL0 = 000 to 101)
  - Select the trigger input function with TMI0 to TMI2 pins (GATE = 0).

    Select an effective edge.

    When the edge set with this bit is detected in the signal input from the TMI0 to TMI2 pins, the down counter starts counting down.
  - Select the gate function with TMI0 to TMI2 pins (GATE = 1).

    Select an effective level.

    The down counter counts down only while the signal of the level that is set with this bit is input from the TMI0 to TMI2 pins.

TRGM1	TRGM0	When the Trigger Input Is Selected [1] (GATE =0)	When the Gate Function Is Selected (GATE =1)
0	0	Edge detection disabled	"L" level
0	1	Rising edge	"H" level
1	0	Falling edge	"L" level
1	1	Both edges	"H" level

[1]: When "1" is written in the TRG bit, the down counter starts counting down regardless of the setting of this bit.

- In event counter mode (CSL2 to CSL0 = 110, 111)

  Select an effective edge.

  When the edge set with this bit is detected in the signal input from the TMI0 to TMI2 pins, the down counter starts counting down.

TRGM1	TRGM0	Explanation
0	0	Setting prohibited
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Note: Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).
If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.

[bit11 to bit9]: CSL2 to CSL0 (Count source selection bits)

This bit selects the timer mode of the 16-bit reload timer. In interval timer mode, it also selects the division rate of the peripheral clock (PCLK), and in event counter mode, it also selects whether to use cascade mode and whether to use the external clock.

CSL2	CSL1	CSL0	Explanation	
0	0	0	Interval timer mode	Peripheral clock (PCLK) divided by 2 (= 2^1)
0	0	1		Peripheral clock (PCLK) divided by 4 (= 2^2)
0	1	0		Peripheral clock (PCLK) divided by 8 (= 2^3)
0	1	1		Peripheral clock (PCLK) divided by 16 (= 2^4)
1	0	0		Peripheral clock (PCLK) divided by 32 (= 2^5)
1	0	1		Peripheral clock (PCLK) divided by 64 (= 2^6)
1	1	0	Event counter mode	Cascade mode[1]
1	1	1		External clock

[1]: For information on the operation when cascade mode is selected, see "20.6.3 Operation in Cascade Mode".

Notes:

- Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).

  If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.
- To use the 2-channel 16-bit reload timer connected in cascade, set this bit as shown below.
  - Channel with smaller number: Select interval timer mode or an external clock.
  - Channel with larger number: Specify cascade mode.
- When event counter mode is selected for this bit, the setting of the GATE bit is ignored.

[bit8]: GATE (Gate input enable bit)

When the timer mode is set to interval timer mode, this bit selects the functions to be assigned to the TMI0 to TMI2 pins.

- Trigger input function: When an effective edge is input from TMI0 to TMI2 pins, a countdown starts.
- Gate function: A countdown is performed only while the effective level signal is input from TMI0 to TMI2 pins.

Written Value	Explanation
0	Trigger input function
1	Gate function

Notes:

- Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).

  If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.
- If event counter mode is selected with CSL2 to CSL0 bits (CSL2 to CSL0 = 110/111), this bit setting is ignored.

[bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit5]: OUTL (Output polarity setting bit)

When the 16-bit reload timer is activated, this bit sets the signal level of the signals to be output from TMO0 to TMO2 pins.

Written Value	Explanation
0	Normal polarity ("L" level)
1	Inverted polarity ("H" level)

Note: Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).
If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.

[bit4]: RELD (Reload operation enable bit)

This bit selects any of the following operation modes for the 16-bit reload timer.

- One shot mode

  When the down counter enters an underflow condition, counting stops in this mode until the next activation trigger is input.
- Reload mode

  When the down counter enters an underflow condition in this mode, the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter so that it continues counting.

Written Value	Explanation
0	One shot mode
1	Reload mode

Note: Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).
If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.

[bit3]: INTE (Interrupt request enable bit)

This bit sets whether to generate the underflow interrupt request when the down counter underflows (UF bit = 1).

Written Value	Explanation
0	Disables generation of underflow interrupt requests.
1	Enables generation of underflow interrupt requests.

[bit2]: UF (Underflow interrupt request flag bit)

This bit indicates that the down counter enters an underflow condition.

If the INTE is set to "1" when this bit is "1", an underflow interrupt request is generated.

UF	In Case of Reading	In Case of Writing
0	The down counter has not entered an underflow condition.	This bit is cleared to "0".
1	The down counter has entered an underflow condition.	Ignored

[bit1]: CNTE (Count operation enable bit)

This bit enables/disables the operation of the down counter.

Written Value	Explanation
0	Stops the count operation.
1	Enables the count operation (activation trigger wait).

Note: If "0" is written to this bit during a down counter operation, the down counter stops.

[bit0]: TRG (Software trigger bit)

This bit activates the 16-bit reload timer through software. When "1" is written to this bit, the down counter loads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and starts counting.

TRG	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Activates the 16-bit reload timer.	

Notes:

- The down counter does not operate while the CNTE bit is "0" even if "1" is written to this bit.
- When the 16-bit reload timer operation is enabled (CNTE=1), if "1" is written to this bit, the down counter starts regardless of the setting of TRGM1 or TRGM0 bit.

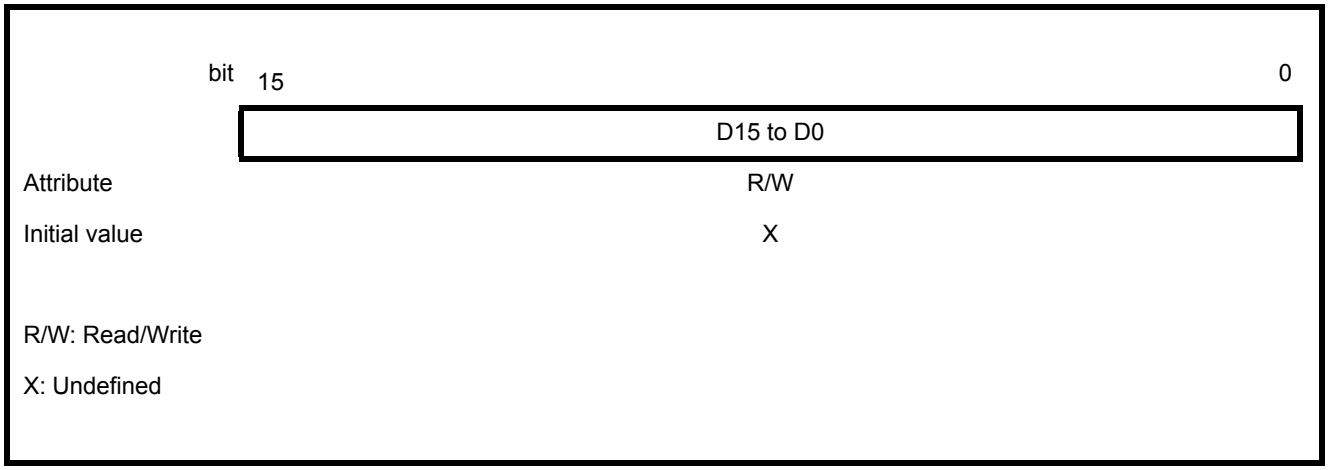
20.4.2 16-bit Timer Reload Register A (TMRLRA0 to TMRLRA2)

This register sets the initial value of the down counter.

In reload mode, if an underflow occurs, the value of this register is reloaded to the down counter.

Figure 20-3 shows the bit configuration of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2).

Figure 20-3. Bit Configuration of 16-bit Timer Reload Register A (TMRLRA0 to TMRLRA2)



When the counter completes counting the value set to this register + 1, an underflow occurs.The signal level of the signals output from TMO0 to TMO2 pins is inverted.

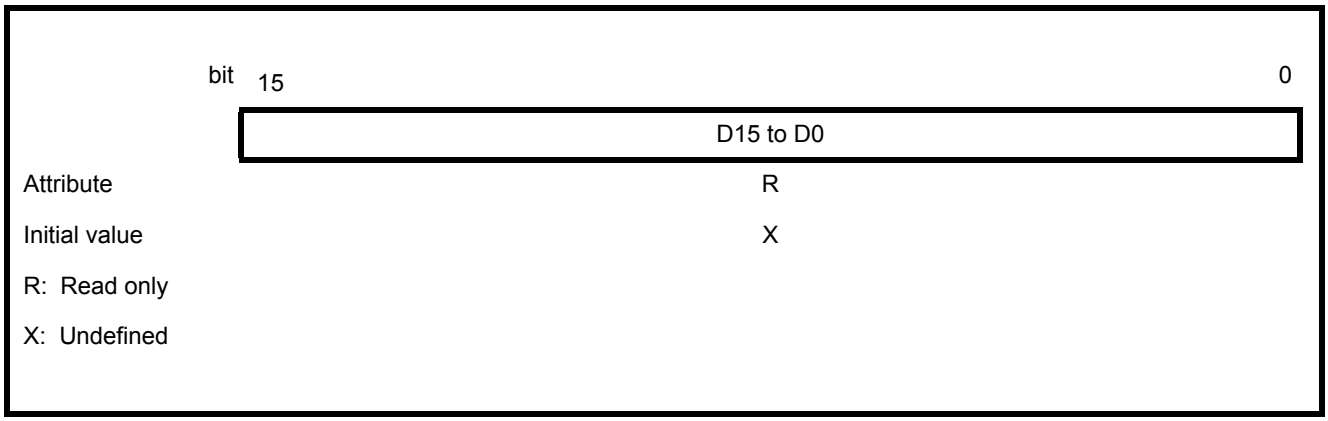
Note: Be sure to access this register in units of half words.

20.4.3 16-bit Timer Register (TMR0 to TMR2)

When this register is read, the down counter value can be read.

shows the bit configuration of the 16-bit timer registers (TMR0 to TMR2).

Figure 20-4. Bit Configuration of 16-bit Timer Register (TMR0 to TMR2)



Note: Be sure to read this register in units of half words.

20.5 Interrupts

An underflow interrupt request is generated when the down counter enters an underflow condition.

Overview

Table 20-4 outlines the interrupts that can be used with the 16-bit reload timer

Table 20-4. Interrupts of the 16-bit Reload Timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Underflow interrupt request	UF=1 for TMCSR	INTE=1 for TMCSR	Write "0" to the UF bit for TMCSR

TMCSR: Timer Control Status Register (TMCSR0 to TMCSR2)

Notes:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For information on the settings of the interrupt levels, see "10. Interrupt Controller".

20.6 An Explanation of Operations and Setting Procedure Examples

This chapter explains the operations of the 16-bit reload timer. Also, examples of procedures for setting the operating state are shown.

Overview

The 16-bit reload timer is a down counter that counts down from a preset value.One of the following timer modes can be selected using the CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2).

- Interval timer mode (CSL2 to CSL0 = 000 to 101)
  It operates with the count clock, which is the divided peripheral clock (PCLK).
- Event counter mode (CSL2 to CSL0 = 110, 111)
  In this mode, the counter counts every time an effective edge is input from TMI0 to TMI2 pins.
  Cascade mode that counts ch.0 outputs with ch.1 and ch.1 outputs with ch.2 is also available.

How to set the signal level of the signals output from TMO0 to TMO2 pins.

The signal level of the signals output from TMO0 to TMO2 pins varies with the settings of OUTL bit of the timer control status register (TMCSR0 to TMCSR2).

- In reload mode
  Table 20-5 shows the signal level of the signals output from TMO0 to TMO2 pins in reload mode.

Table 20-5. Signal Level in Reload Mode

	Normal polarity (OUTL = 0)	Inverted polarity (OUTL = 1)
When the 16-bit reload timer is activated	"L" level	"H" level
Subsequent	The output level is inverted every time an underflow occurs.	

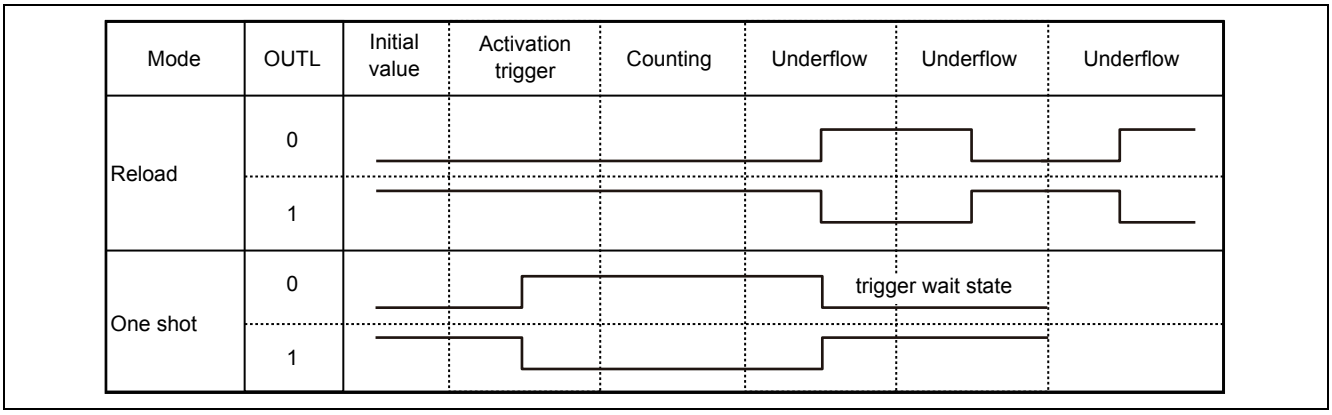
- In one shot mode
  Table 20-6 shows the signal level of the signals output from TMO0 to TMO2 pins in one shot mode.

Table 20-6. Signal Level in One Shot Mode

	Normal polarity (OUTL = 0)	Inverted polarity (OUTL = 1)
When the 16-bit reload timer is activated	"L" level	"H" level
When an activation trigger is input	"H" level	"L" level
When an underflow occurs	"L" level	"H" level

Figure 20-5 shows the OUTL bits of the timer control status register (TMCSR0 to TMCSR2) and their output wave forms.

Figure 20-5. OUTL Bits of the Timer Control Status Registers (TMCSR0 to TMCSR2) and their output wave forms



20.6.1 Operation in Interval Timer Mode

This section explains the operation for using the 16-bit reload timer that counts synchronously with the internal clock (peripheral clock) in interval timer mode.

The count clock is generated by dividing the peripheral clock (PCLK).

Setting

This section also explains the settings required for using the 16-bit reload timer in interval timer mode.

- Interval timer mode settings

  To use the 16-bit reload timer in interval timer mode, make any of the following settings for the CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2), and select the division rate of the peripheral clock (PCLK).

CSL2	CSL1	CSL0	Timer Mode	Division Rate of Peripheral Clock
0	0	0	Interval timer mode	Divided by 2 (= 2^1)
0	0	1		Divided by 4 (= 2^2)
0	1	0		Divided by 8 (= 2^3)
0	1	1		Divided by 16 (= 2^4)
1	0	0		Divided by 32 (= 2^5)
1	0	1		Divided by 64 (= 2^6)

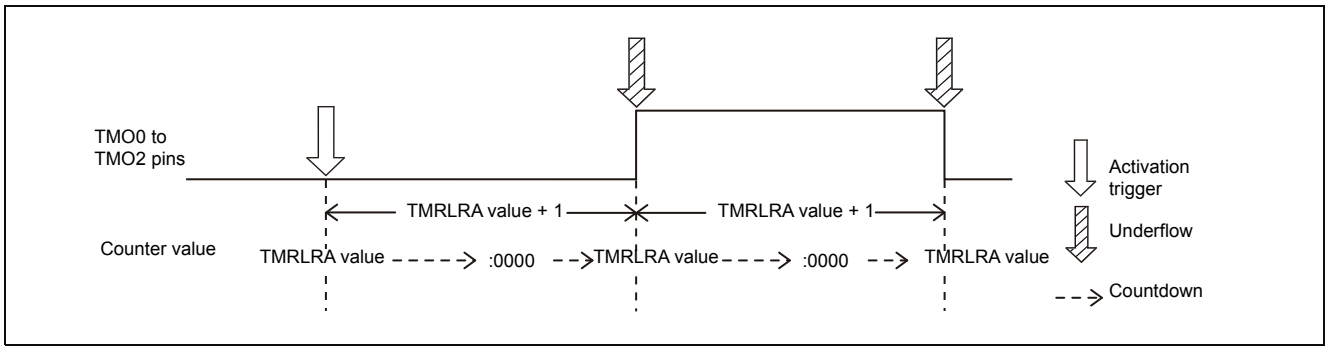
- Operation mode settings

  In interval timer mode, one of the following operation modes can be selected using the RELD bits of the timer control status register (TMCSR0 to TMCSR2).

  - Reload mode (RELD = 1)

    When the down counter enters an underflow condition, it reloads the value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and repeats counting in this mode. Figure 20-6 shows the basic operation in reload mode.

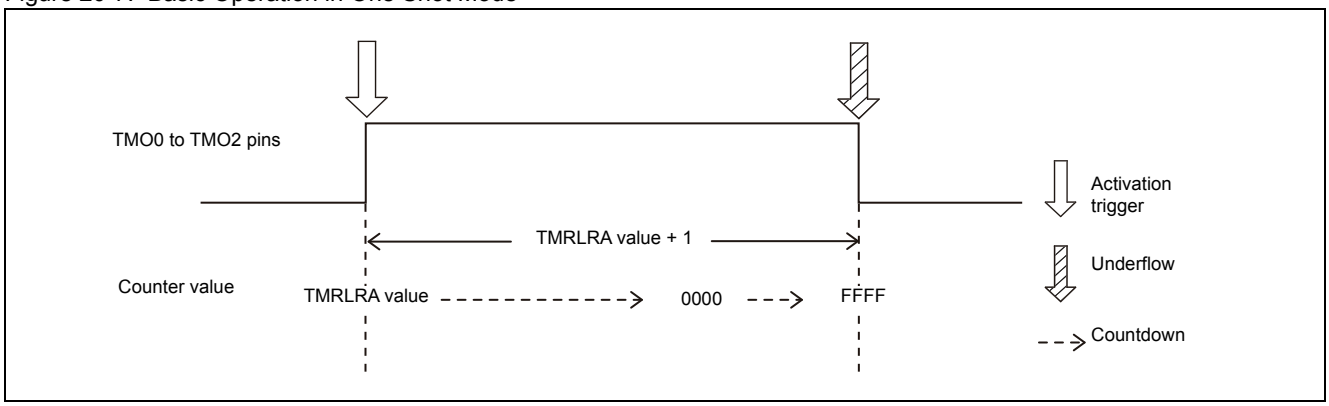
Figure 20-6. Basic Operation in Reload Mode



- One shot mode (RELD = 0)

  In this mode, counting stops when the down counter enters an underflow condition. Figure 20-7 shows the basic operation in one shot mode.

Figure 20-7. Basic Operation in One Shot Mode



- TMI0 to TMI2 pin function settings

  Using TRGM1 and TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) and the GATE bit, the function of TMI0 to TMI2 pins can be selected from the following list.

  shows the combination of bits.

Table 20-7. Combination of Bits

TRGM1, TRGM0	GATE	Pin Function
00	0	TMI0 to TMI2 pins do not work.
01	0	TMI0 to TMI2 pins operate as the trigger input function. The effective edge is a rising edge.
10	0	TMI0 to TMI2 pins operate as the trigger input function. The effective edge is a falling edge.
11	0	TMI0 to TMI2 pins operate as the trigger input function. The effective edge is both edges.
00/10	1	TMI0 to TMI2 pins operate as the gate input function. The effective level is "L".
01/11	1	TMI0 to TMI2 pins operate as the gate input function. The effective level is "H".

Pulse width calculation

How to calculate the pulse width of the signals output from TMO0 to TMO2 pins in interval timer mode is explained below.

Pulse width = T × (L + 1)

L Value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

T Cycles of the count clock

How to calculate underflow cycles

If the down counter attempts to count further from the value of "0000_H", an underflow occurs. A cycle from when the down counter starts counting to when an underflow occurs is set in the 16-bit timer reload register (TMRLRA0 to TMRLRA2).

The following shows how to calculate the underflow cycles.

T × (L + 1)

T Cycles of the count clock

L Value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

Operations in reload mode (TMI0 to TMI2 pins = trigger input)

In this mode, TMI0 to TMI2 pins are used for trigger input, and the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded each time underflow occurs to continue a countdown.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

1. One of the TRGM1, TRGM0 bits = 01 to 11
2. GATE bit = 0
3. RELD bit = 1

- Activate

  Use the following procedure for activating.

  1. Use the CNTE bit of the timer control status register (TMCSR0 to TMCSR2) to enable the operation of the 16-bit reload timer (CNTE = 1).

     The 16-bit reload timer enters the activation trigger wait state.

  2. The activation trigger is input in either of the following ways:

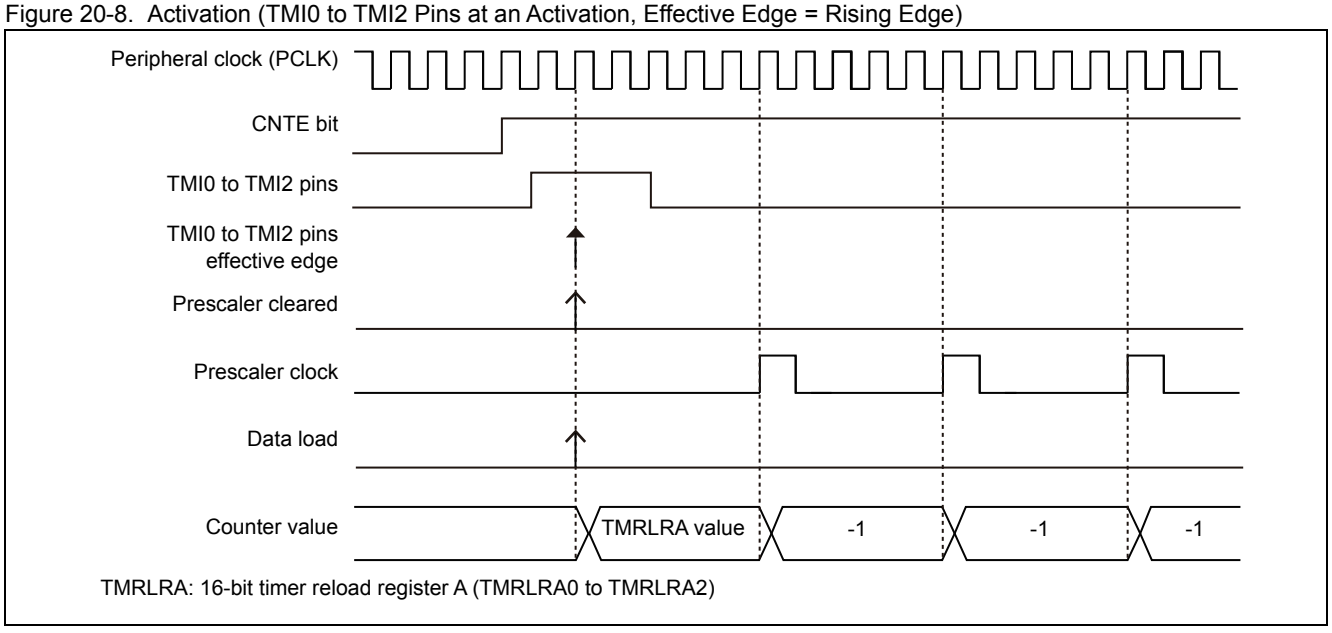
     a. Input the edge set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TMI0 to TMI2 pins.

     b. Write "1" to the TRG bit of the timer control status register (TMCSR0 to TMCSR2).

     The prescaler is cleared. The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is loaded to the down counter, and counting starts.

  Figure 20-8 shows an activation.

Figure 20-8. Activation (TMI0 to TMI2 Pins at an Activation, Effective Edge = Rising Edge)



Note: Be sure that the pulse width of the activation trigger input from TMI0 to TMI2 pins never falls below 2T (T: cycle of the peripheral clock (PCLK)).

- Count operation

  The down counter starts a countdown synchronously with the count clock from the value of 16-bit timer reload register A (TMRLRA0 to TMRLRA2).

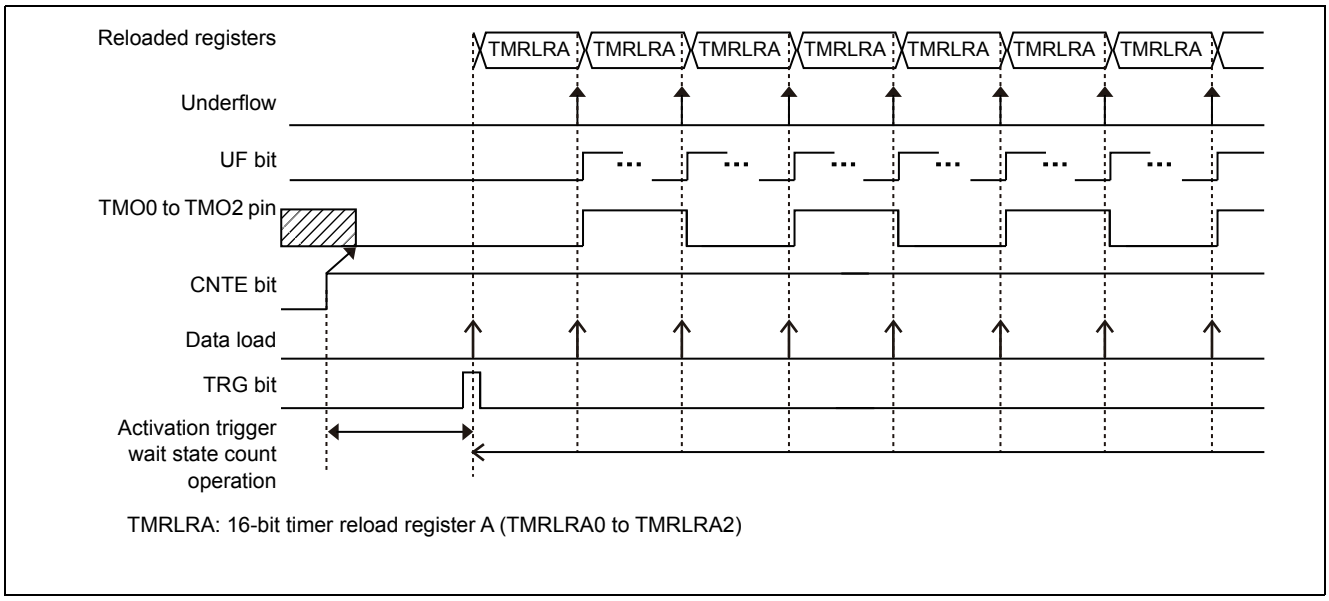
  If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

  - The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
  - The signal level of the signals output from TMO0 to TMO2 pins is inverted.
  - The timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

  As described, every time an underflow occurs, the timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

  Figure 20-9 shows the count operation.

Figure 20-9. Count Operation (Activation Through Software, Output Polarity = Normal Polarity)



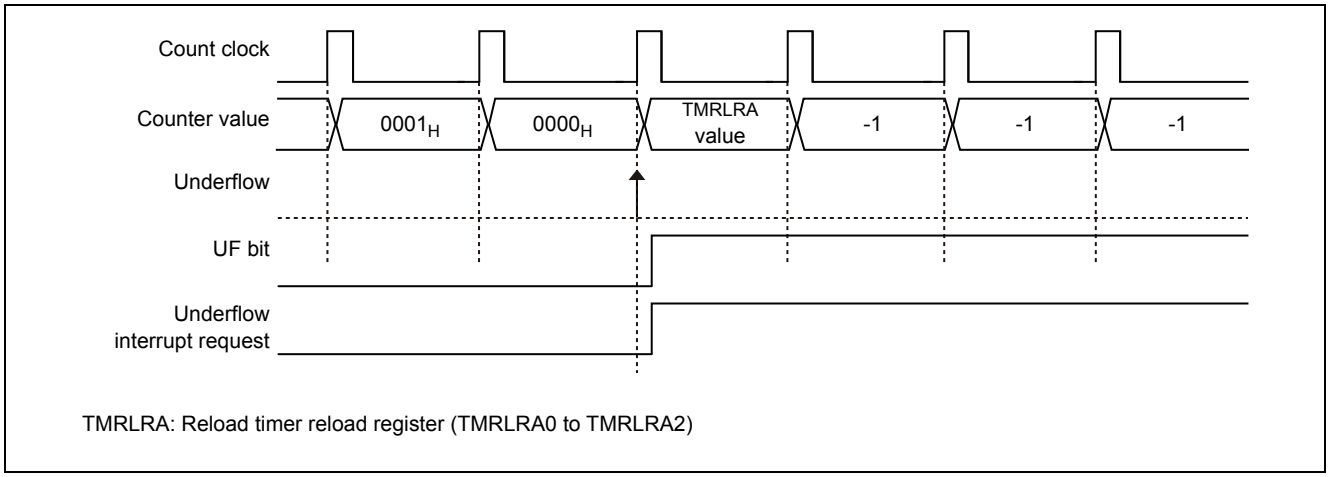
- Operation of interrupt processing

  If the down counter enters an underflow condition, the UF bit of the timer control status register (TMCSR0 to TMCSR2) changes to "1".

  In this case, if the INTE bit of the timer control status register (TMCSR0 to TMCSR2) is set to "1", an underflow interrupt request is generated.

  Figure 20-10 shows the underflow interrupt request generation timing.

Figure 20-10. Underflow Interrupt Request Generation Timing



When "0" is written to the UF bit of the timer control status register (TMCSR0 to TMCSR2), the underflow interrupt request can be cleared.

Note: If an underflow interrupt request is generated at the same time the other underflow interrupt request is cleared, the clearing operation is ignored, and the underflow interrupt request remains generated.

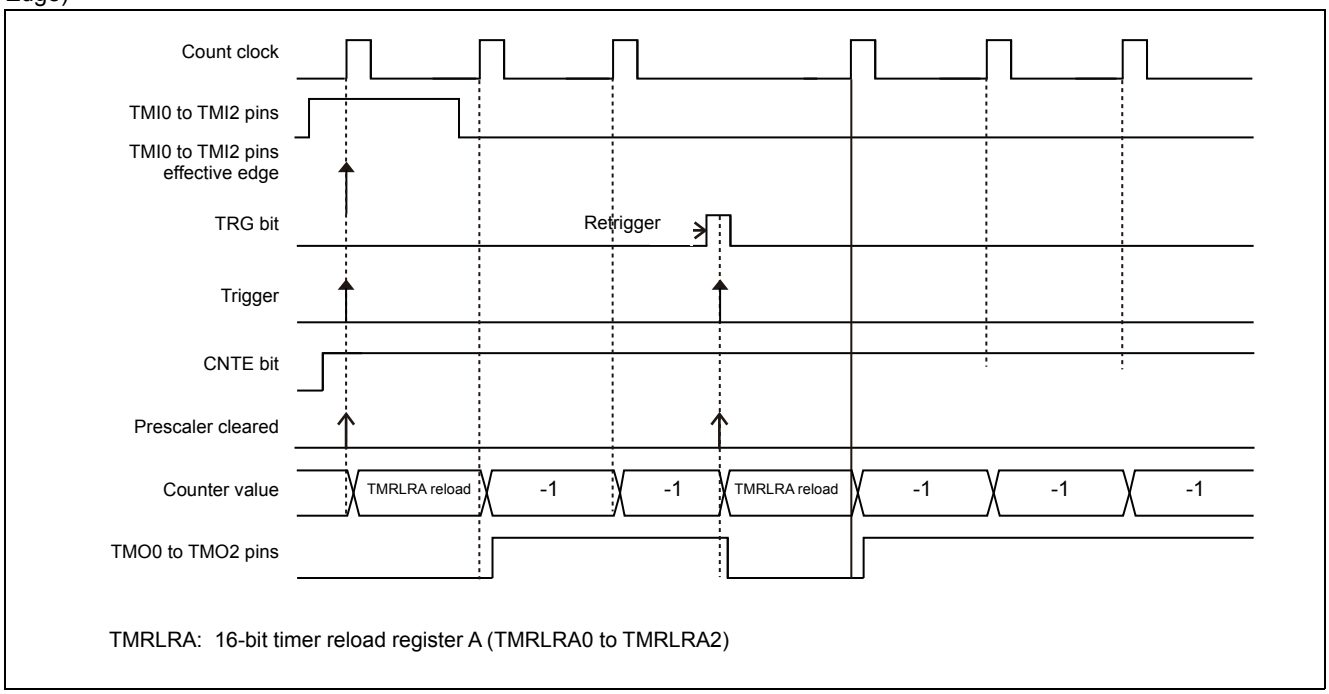
- Retrigger operation

  If an activation trigger of the 16-bit reload timer is detected during a count operation, the retrigger is generated, and the following operations are performed.

  - The signal level of TMI0 to TMI2 pins is initialized.
  - The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.
  - The prescaler is cleared.
  - Count operation starts.

  Figure 20-11 shows the operation when a retrigger is generated.

Figure 20-11. Operation when a Retrigger is Generated (Retrigger Generated on TMI0 to TMI2 Pins, Effective Edge = Rising Edge)



Note: When the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is rewritten, if a retrigger occurs at the same time the reload value is changed, the down counter loads the value before the change.
The value after change is loaded at the next reloading.

Operations in reload mode (TMI0 to TMI2 pins = at a gate input)

In this mode, TMI0 to TMI2 pins are used for gate input, and the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded each time underflow occurs to continue a countdown.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

1. TRGM0 bit = 0/1
2. GATE bit = 1
3. RELD bit = 1

- Activate

  Use the following procedure for activating.

  1. Use the CNTE bit of the timer control status register (TMCSR0 to TMCSR2) to enable the operation of the 16-bit reload timer (CNTE = 1).

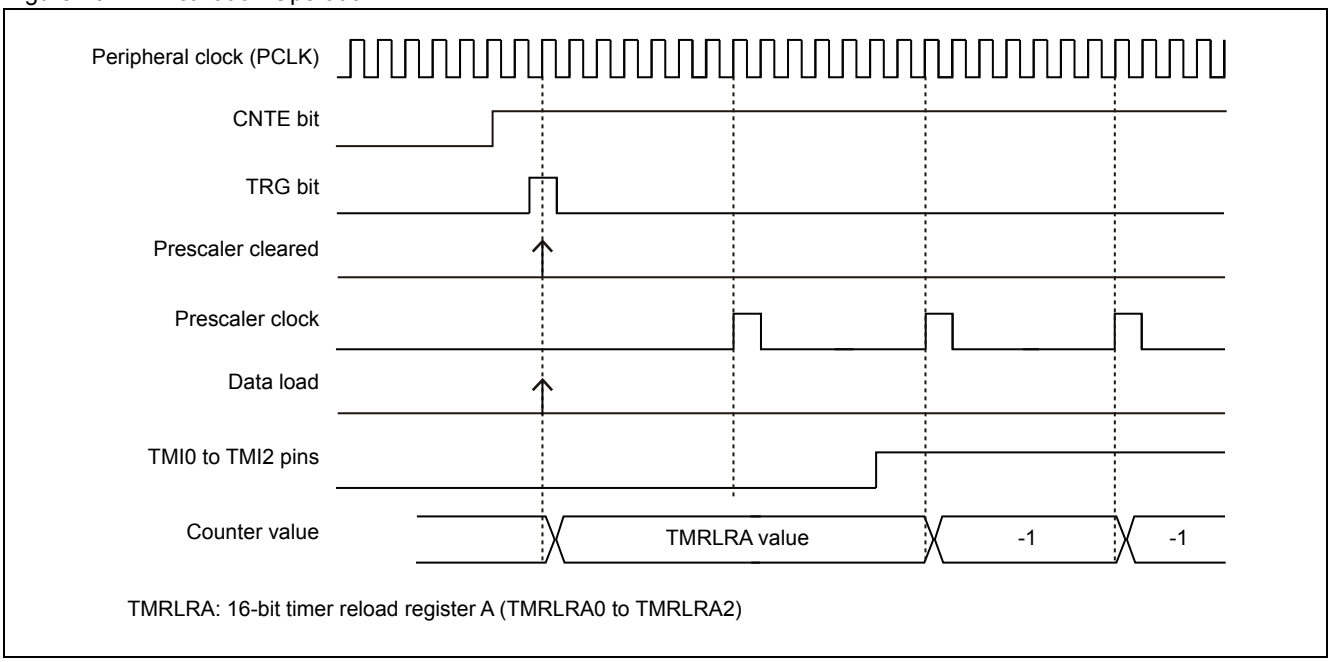
     The 16-bit reload timer enters the activation trigger wait state.
  2. Input an activation trigger using the TRG bit of the timer control status register (TMCSR0 to TMCSR2). (TRG = 1)

     The prescaler is cleared. The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is loaded to the down counter, and the 16-bit reload timer enters the effective input polarity (from TMI0 to TMI2 pins) wait state.
  3. Input the signal with the level set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TMI0 to TMI2 pins.

     The counter starts counting.

  Figure 20-12 shows an activation operation.

Figure 20-12. Activation Operation



Note: Be sure that the effective level input from TMI0 to TMI2 pins never falls below 2T (T: cycle of the peripheral clock (PCLK)).

- Count operation

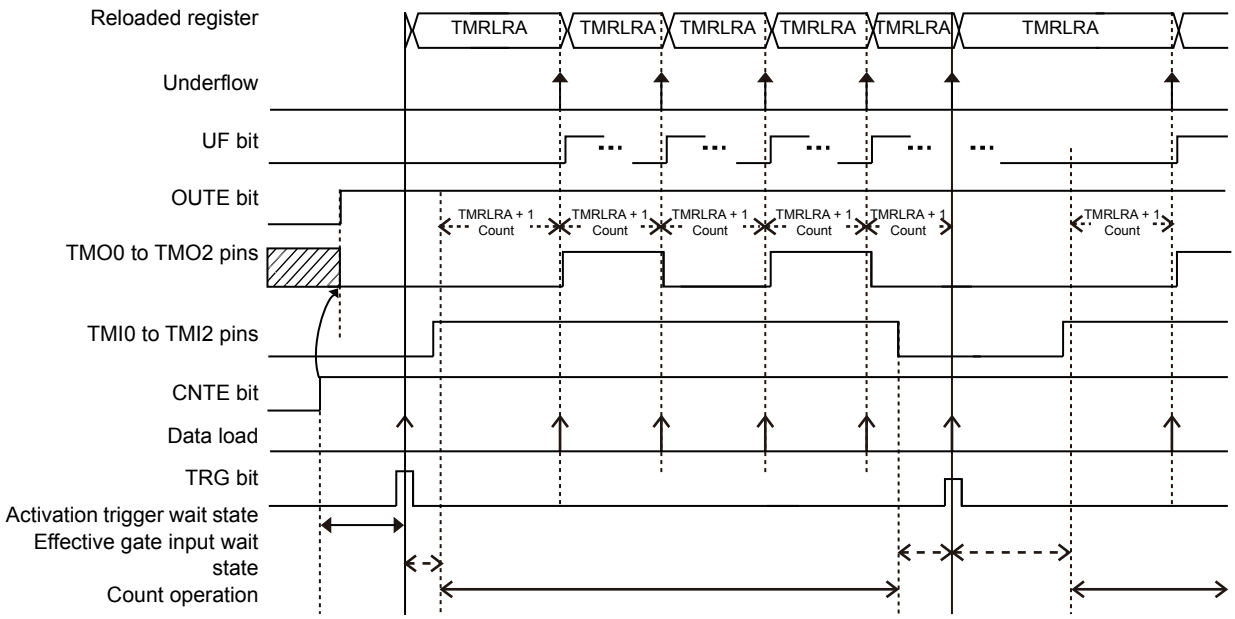
  Only while the effective level signal is input from TMI0 to TMI2 pins does the down counter perform a countdown from the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) synchronously with the count clock.

  If the effective level signal is not input from TMI0 to TMI2 pins, the down counter stops counting. If the effective level signal is input while the down counter is stopped, the counter starts counting from the value where it stopped.

  The subsequent operations are the same as those when TMI0 to TMI2 pins = trigger input function is set. See "Operations in reload mode (TMI0 to TMI2 pins = trigger input)"

  Figure 20-13 shows the count operation.

Figure 20-13. Count Operation (Effective Level = "H" Level, Output Polarity = Normal Polarity)



TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

- Operation of interrupt processing

  The operation is the same as in reload mode. See "Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

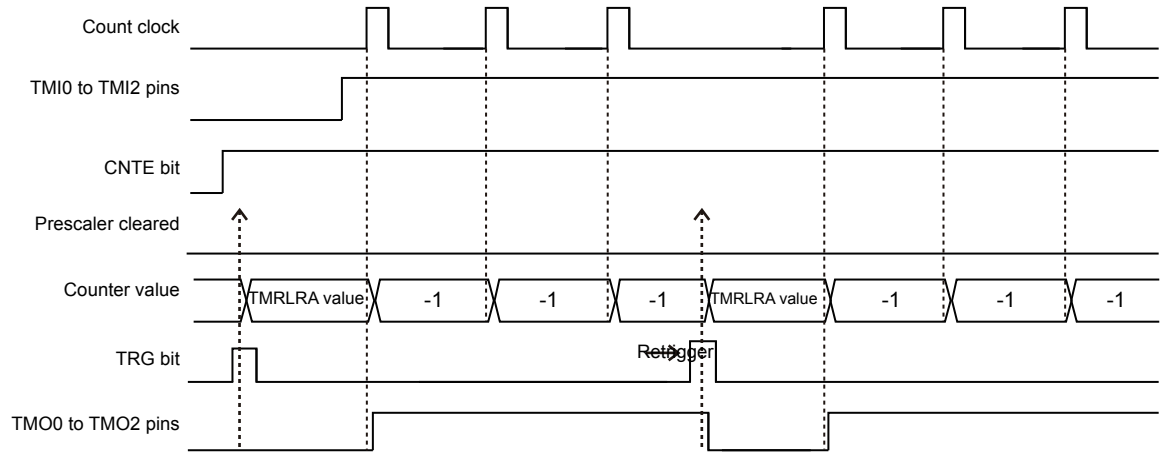
- Retrigger operation

  If an activation trigger of the 16-bit reload timer is detected during a count operation, the retrigger is generated, and the following operations are performed.

  - The signal level of TMI0 to TMI2 pins is initialized.
  - The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.
  - The prescaler is cleared.

  When an effective level signal is input from TMI0 to TMI2 pin in such condition, counting starts. Figure 20-14 shows the operation when a retrigger is generated.

Figure 20-14. Operation when a retrigger is generated (effective level = "H" level)



TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

Operations in one shot mode (TMI0 to TMI2 pins = trigger input)

When TMI0 to TMI2 pins are used for trigger input, if an underflow occurs, this mode stops counting until the next activation trigger is input.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

1. One of the TRGM1, TRGM0 bits = 01 to 11
2. GATE bit = 0
3. RELD bit = 0

- Activate

  The operation is the same as in reload mode. See "Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

  However, if an activation trigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

- Count operation

  The down counter starts a countdown synchronously with the count clock from the value of 16-bit timer reload register A (TMRLRA0 to TMRLRA2).

  If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

  - The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
  - The signal level of the signals output from TMO0 to TMO2 pins is initialized.
  - Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "$FFFF_H$").

  Figure 20-15 shows the count operation when TMI0 to TMI2 pins are used for activation.

Figure 20-15. Count Operation (Effective Edge = Rising Edge, Output Polarity = Normal Polarity)

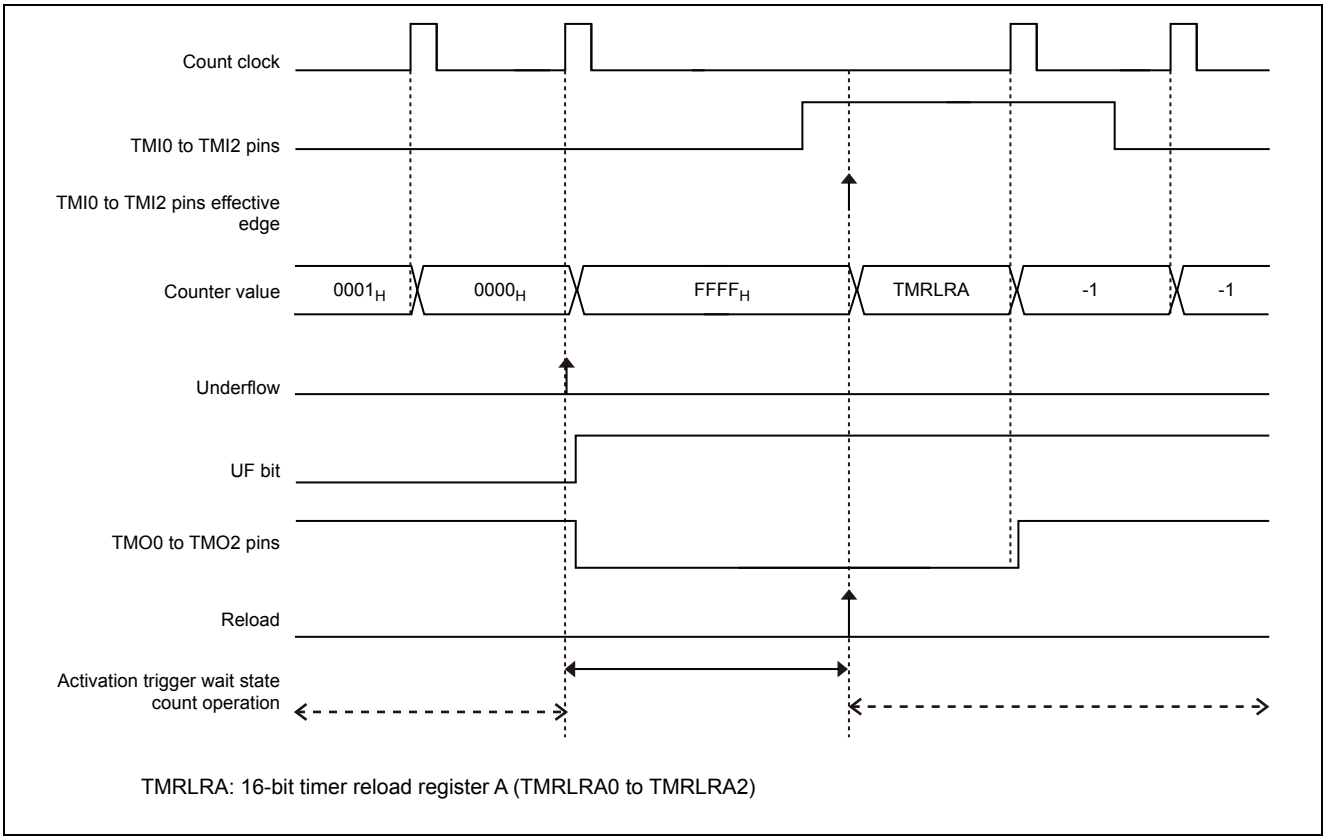
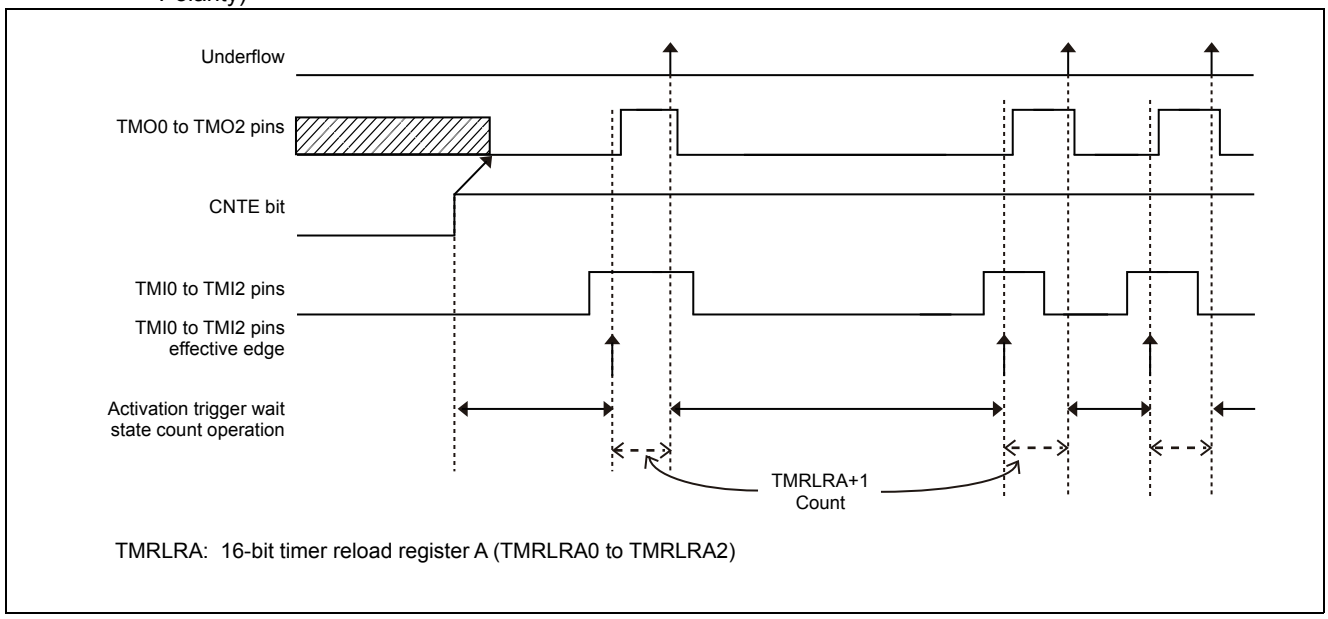


Figure 20-16 shows the detailed operation when an underflow occurs.

Figure 20-16. Detailed Operation when an Underflow Occurs (Effective Edge = Rising Edge, Output Polarity = Normal Polarity)



- Operation of interrupt processing

  The operation is the same as in reload mode. See "Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

- Retrigger operation

  The operation is the same as in reload mode. See "Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

  However, if a retrigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

Operations in one shot mode (TMI0 to TMI2 pins = gate input)

When TMI0 to TMI2 pins are used for gate input, if an underflow occurs, this mode stops counting until the next activation trigger is input.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

1. TRGM0 bit = 0/1
2. GATE bit = 1
3. RELD bit = 0

- Activate

  The operation is the same as in reload mode. See "Operations in reload mode (TMI0 to TMI2 pins = at a gate input)".

  However, if an activation trigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

- Count operation

  Only while the effective level signal is input from TMI0 to TMI2 pins does the down counter counts down from the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) synchronously with count clock.

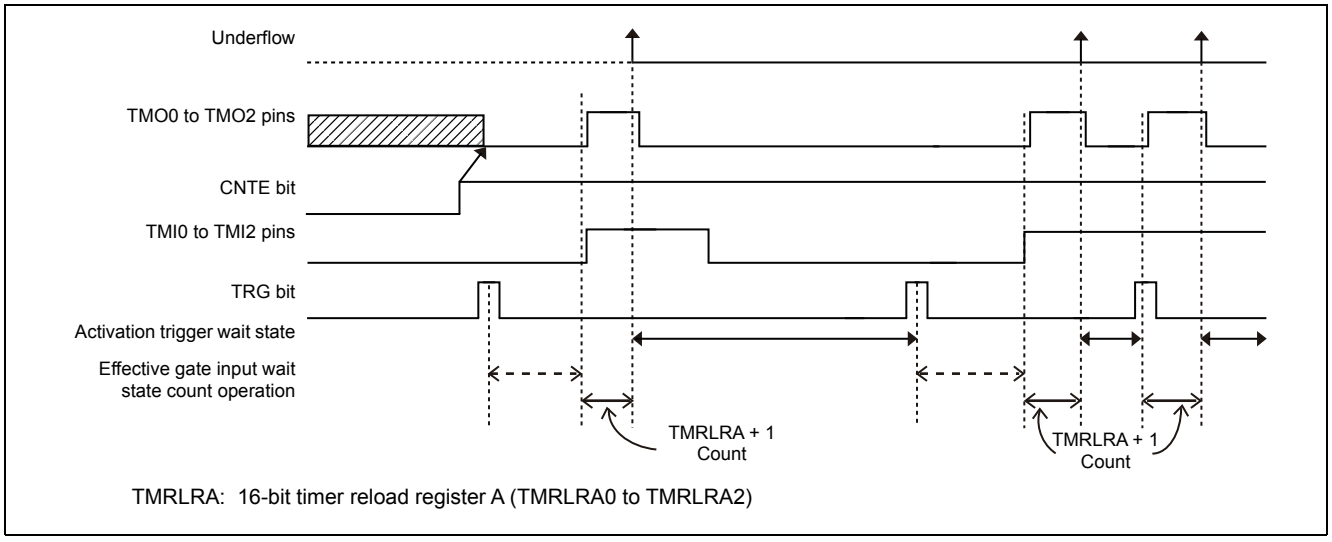
  If the effective level signal is not input from TMI0 to TMI2 pins, the down counter stops counting. If the effective level signal is input while the down counter is stopped, the counter starts counting from the value where it stopped.

  If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

  - The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
  - The signal level of the signals output from TMO0 to TMO2 pins is initialized.
  - Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "$FFFF_H$").

  Figure 20-17 shows the count operation.

Figure 20-17. Count Operation (Effective Level = "H" Level, Output Polarity = Normal Polarity)



- Operation of interrupt processing

  The operation is the same as in reload mode. See "Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

- Retrigger operation

  The operation is the same as in reload mode. See "Operations in reload mode (TMI0 to TMI2 pins = at a gate input)".

  However, if a retrigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

20.6.2 Operations in Event Counter Mode

This section explains the operations for using 16-bit reload timer as an event counter. This section explains the operation for counting external events.

Overview

In event counter mode, external events input from TMI0 to TMI2 pins are counted. It performs a countdown every time an effective edge is input from TMI0 to TMI2 pins.

For information on cascade mode, see "20.6.3 Operation in Cascade Mode".

Setting

- Event counter mode settings

  To use the 16-bit reload timer in event counter mode, set CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2) as shown below.

CSL2	CSL1	CSL0	Mode	Count Clock
1	1	1	Event counter mode	External clock

- Operation mode settings

  In event counter mode, one of the following operation modes can be selected using the RELD bits of the timer control status register (TMCSR0 to TMCSR2).

  - Reload mode (RELD = 1)

    When the down counter enters an underflow condition, it reloads the value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and repeats counting in this mode.

  - One shot mode (RELD = 0)

    In this mode, counting stops when the down counter enters an underflow condition.

- Effective edge settings

  The 16-bit reload timer performs a count down every time an effective edge is input from TMI0 to TMI2 pins.

  The effective edge can be selected from the following settings of TRGM1 and TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2).

TRGM1, TRGM0	Pin Function
00	TMI0 to TMI2 pins do not work.
01	Rising edge
10	Falling edge
11	Both edges

Operation in reload mode

In this mode, every time an underflow occurs, the timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

1. One of the TRGM1, TRGM0 bits = 01 to 11
2. RELD bit = 1

- Activate

  Use the following procedure for activating.

  1. Use the CNTE bit of the timer control status register (TMCSR0 to TMCSR2) to enable the operation of the 16-bit reload timer (CNTE = 1).
     The 16-bit reload timer enters the activation trigger wait state.
  2. Input an activation trigger using the TRG bit of the timer control status register (TMCSR0 to TMCSR2). (TRG = 1)
     The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is loaded to the down counter, and the 16-bit reload timer enters the effective edge detection (of the signal output from TMI0 to TMI2 pins) wait state.
  3. Input the effective edge set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TMI0 to TMI2 pins.
     The counter starts counting.

- Count operation

  Every time an effective edge is detected in the input signal from TMI0 to TMI2 pins, it performs a countdown.

  Figure 20-18 to Figure 20-20 show the count timing.

Figure 20-18. Count Timing (Effective Edge = Rising Edge)

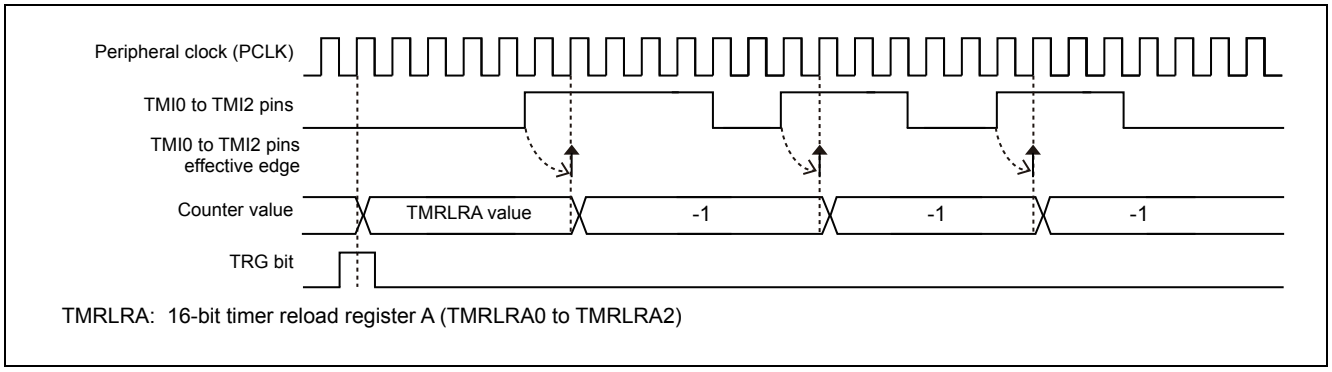


Figure 20-19. Count Timing (Effective Edge = Falling Edge)

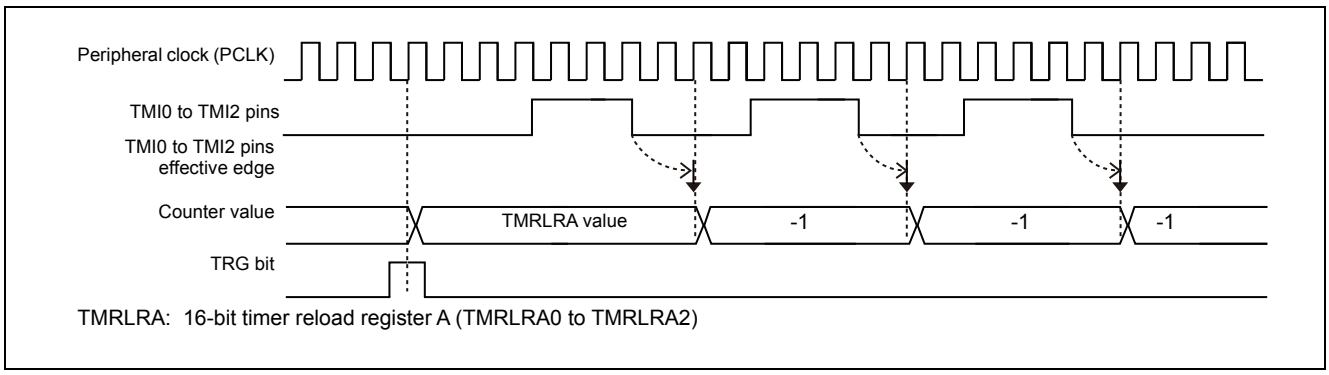
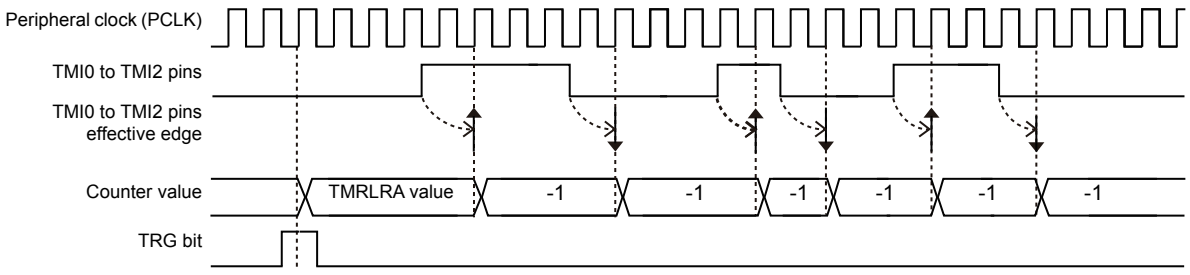


Figure 20-20. Count Timing (Effective Edge = Both Edges)



TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

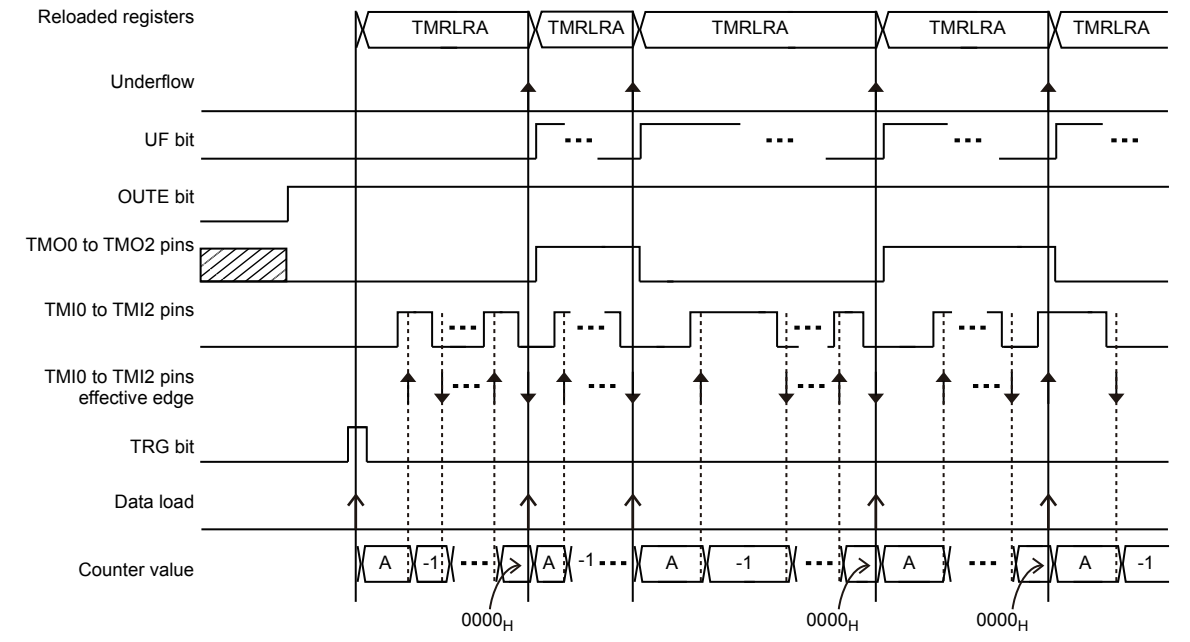
- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TMO0 to TMO2 pins is inverted.
- The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.
- The counter continues counting when an effective level signal is input from TMI0 to TMI2 pins.

As described, every time an underflow occurs, the timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

After an underflow occurs, counting does not start until an effective edge of the signal input from TMI0 to TMI2 pins is detected.

Figure 20-21 shows the count operation.

Figure 20-21. Count Operation (Detection Edge = Both Edges, Output Polarity = Normal Polarity)



TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

- Operation of interrupt processing

  The operation is the same as in interval timer mode. See "Operations in reload mode (TMI0 to TMI2 pins = trigger input)" in "20.6.1 Operation in Interval Timer Mode".

- Retrigger operation

  If an activation trigger of the 16-bit reload timer is detected during a count operation, the retrigger is generated, and the following operations are performed.

  - The signal level of the signals output from TMO0 to TMO2 pins is initialized to the level set in the OUTL bit of the timer control status register (TMCSR0 to TMCSR2).
  - The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.

  When an effective edge is input from TMI0 to TMI2 pin in such condition, counting starts.

Operation in one shot mode

When an underflow occurs, counting stops in this mode until the next activation trigger is input.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

1. One of the TRGM1, TRGM0 bits = 01 to 11
2. RELD bit = 0

- Activate

  The operation is the same as in reload mode. See "Operation in reload mode".

- Count operation

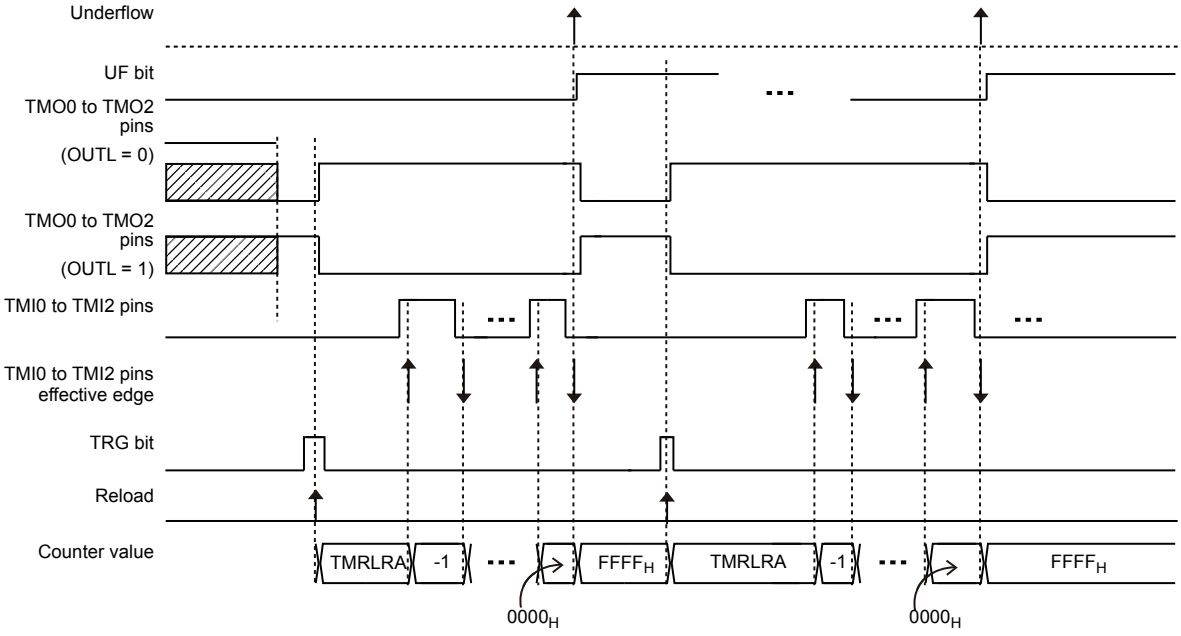
  Every time an effective edge is detected from TMI0 to TMI2 pins, the counter counts down.

  If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

  - The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
  - The signal level of the signals output from TMO0 to TMO2 pins is initialized.
  - Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "$FFFF_H$").

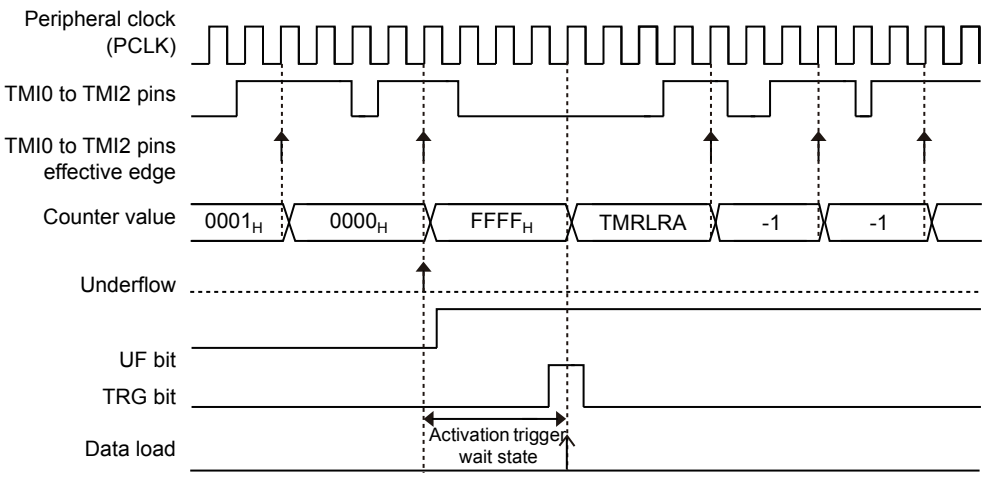
  Figure 20-22 and Figure 20-23 show the count operations.

Figure 20-22. Count Operation (Detection Edge = Both Edges)



TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

Figure 20-23. Count Operation (Detection Edge = Rising Edges)



TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

- Operation of interrupt processing
  The operation is the same as in reload mode. See "Operation in reload mode".
- Retrigger operation
  The operation is the same as in reload mode. See "Operation in reload mode".

20.6.3 Operation in Cascade Mode

In cascade mode, ch.1 can count the outputs from ch.0 of the 16-bit reload timer, and ch.2 can count the outputs from ch.1. This section explains the operations in cascade mode.

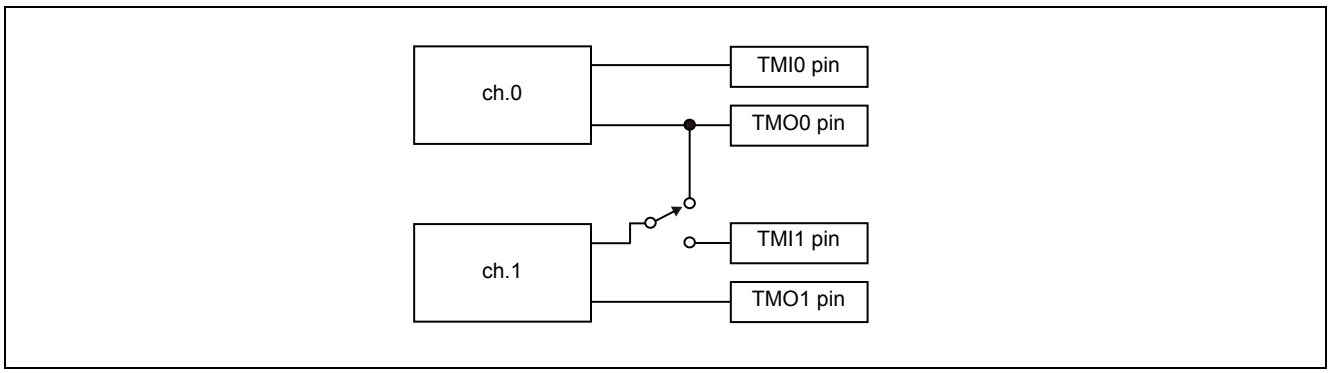
Operation

The following shows the count operation when cascade mode is selected with the CSL2 to CSL0 bits (CSL2 to CSL0 = 110) of the timer control status register (TMCSR0 to TMCSR2).

- When ch.1 is connected in cascade mode

  It counts the outputs from ch.0. Figure 20-24 shows the I/O operation when ch.1 is used in cascade mode.

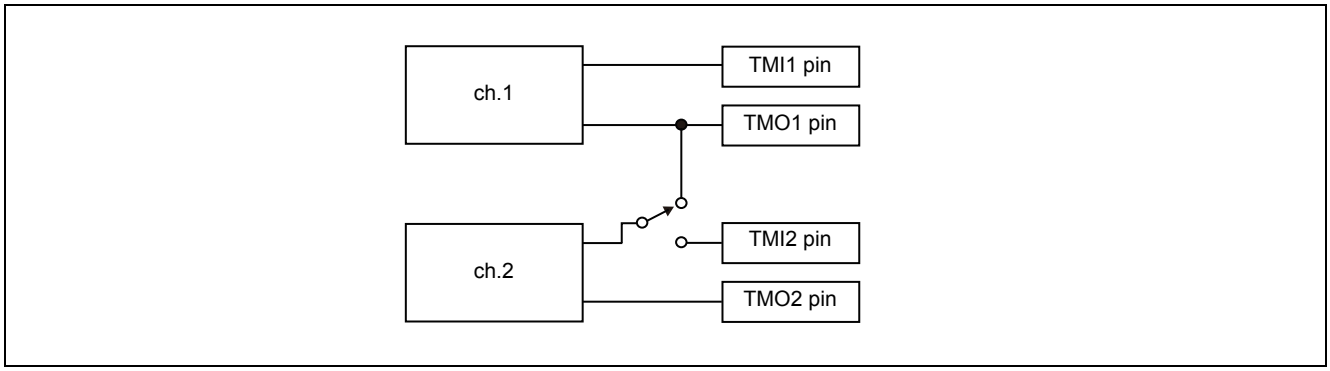
Figure 20-24. I/O operation when ch.1 is used in cascade mode



- When ch.2 is connected in cascade mode

  It counts the outputs from ch.1. Figure 20-25 shows the I/O operation when ch.2 is used in cascade mode.

Figure 20-25. I/O operation when ch.2 is used in cascade mode



Note:

In cascade mode, use the CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2) to set the timer mode as shown below.

- Lower number channel

  Select interval timer mode or external clock (CSL2 to CSL0 = other than 110)

- Higher number channel

  Set cascade mode (CSL2 to CSL0 = 110)

Underflow cycle

This section explains the calculation of the underflow cycles of ch.1 and ch.2.

- When ch.1 is connected in cascade mode

  T × (TMRLRA0 value + 1) × (TMRLRA1 value + 1)

  T: Cycle of the count clock for ch.0
  TMRLRA0: 16-bit timer reload register A0 (TMRLRA0)
  TMRLRA1: 16-bit timer reload register A1 (TMRLRA1)

- When ch.2 is connected in cascade mode

  T × (TMRLRA1 value + 1) × (TMRLRA2 value + 1)

  T: Cycle of the count clock for ch.1
  TMRLRA1: 16-bit timer reload register A1 (TMRLRA1)
  TMRLRA2: 16-bit timer reload register A2 (TMRLRA2)

20.7 Notes on Use

Note the following points on using the 16-bit reload timer.

Notes on interrupts

If an underflow interrupt request flag is cleared at the same time that it is set to "1", the clearing of the underflow interrupt request flag is ignored and the underflow interrupt request flag remains "1".

Operations for simultaneous activations

If more than one of the events used to determine the operating state of the 16-bit reload timer occur simultaneously, the priority order of these events is shown below.

1. Register reading
2. Trigger input
3. Underflow
4. Clock count input

21. Base Timer I/O Select Function

This chapter explains the I/O select function of the base timer.

21.1 Overview

21.2 Configuration

21.3 Pins

21.4 Registers

21.5 I/O Mode

21.1 Overview

The I/O select function of the base timer determines the I/O method of the signals (external clock/external activation trigger/ wave form) to/from the base timer by setting the I/O mode.

In addition, the base timer can be used separately by channel as either of the following timers by switching the timer function.

- 16-bit PWM timer
- 16-bit PPG timer
- 16/32-bit reload timer
- 16/32-bit PWC timer

Be sure to use the base timer after reading both this chapter and the chapter on the timer function to be used.

Overview

The I/O mode can be selected from among the 9 types of modes for each 2 channels.

- I/O mode 0: 16-bit timer standard mode

  This mode operates the base timer individually, one channel at a time.
- I/O mode 1: Timer full mode

  In this mode, signals of the even-numbered channel of the base timer are allocated to the external pins separately to operate the timer.
- I/O mode 2: External trigger shared mode

  In this mode, the external activation trigger can be input to the 2 channels of base timers at the same time. This mode enables activating 2 channels of base timers at the same time.
- I/O mode 3: Other channel trigger shared mode

  In this mode, the external signal from other channels is input as an external activation trigger to activate the timer. This mode cannot be set for ch.0 and ch.1.
- I/O mode 4: Timer activation/stop mode

  This mode controls activation/stop of the odd-numbered channel by using the even-numbered channel.
  The odd-numbered channel is activated at the rising edge of the output signal from the even-numbered channel and stops at the falling edge.
- I/O mode 5: Same time software activation mode

  This mode activates multiple channels at the same time using the software.
- I/O mode 6: Software activation timer activation/stop mode

  This mode controls activation/stop of the odd-numbered channel by using the even-numbered channel.
  The even-numbered channel is activated through software. The odd-numbered channel is activated at the rising edge of the output signal from the even-numbered channel and stops at the falling edge.
- I/O mode 7: Timer activation mode

  This mode controls activation of the odd-numbered channel by using the even-numbered channel. The odd-numbered channel is activated at the rising edge of the output signal from the even-numbered channel.
- I/O mode 8: Other channel trigger shared timer activation/stop mode

  In this mode, the external signal from other channels is input as an external activation trigger to activate the timer. This mode cannot be set for ch.0 and ch.1.

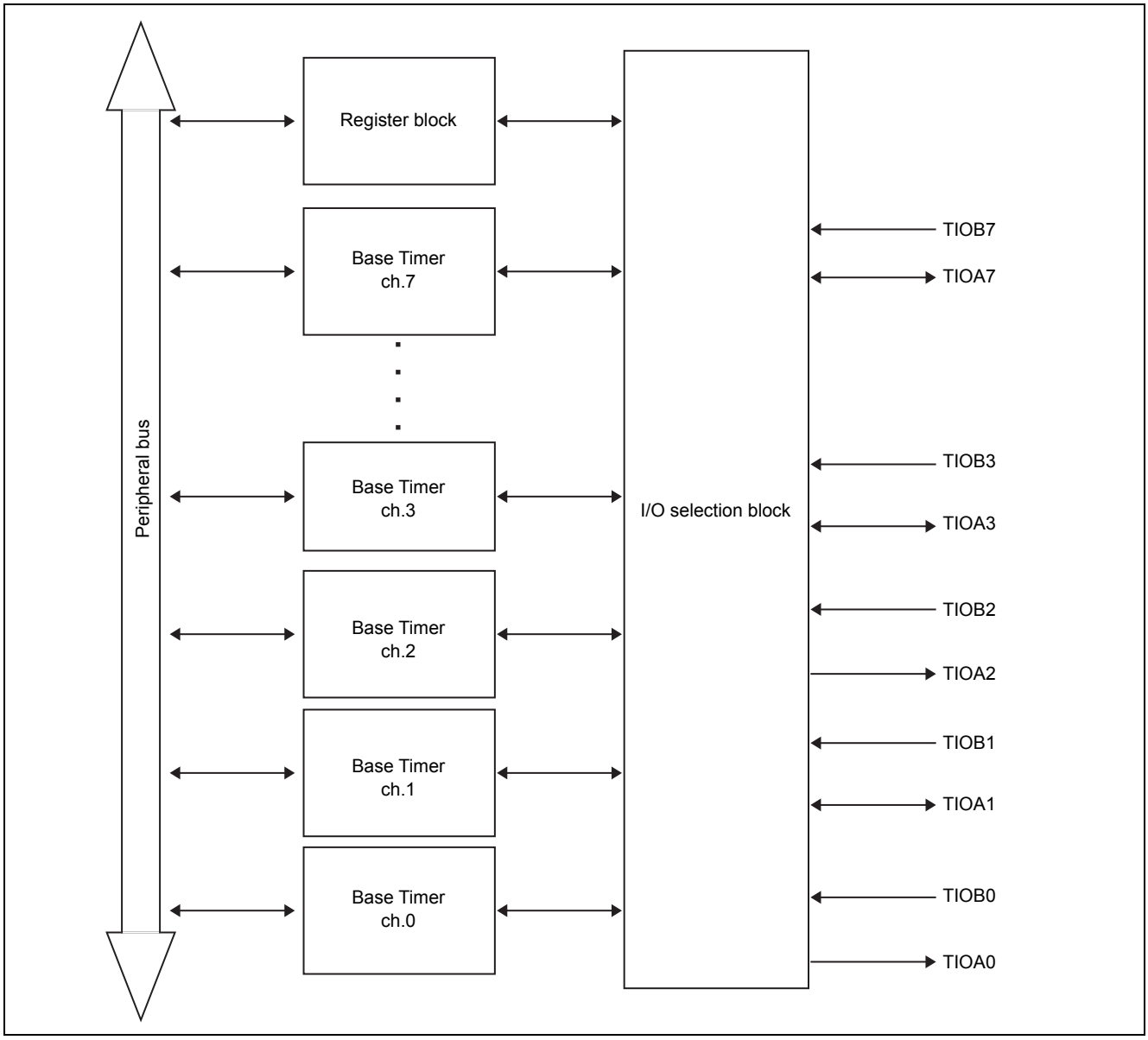
21.2 Configuration

The base timer I/O select function consists of the following blocks:

Block diagram of the base timer I/O select function

Figure 21-1 is a block diagram of the base timer I/O select function.

Figure 21-1. Block Diagram of Base Timer I/O Select Function



- I/O selection block
  This circuit selects the I/O mode of the base timer for each channel.
- Base timer (ch.0 to ch.7)
  ch.0 to ch.7 of the base timer.

21.3 Pins

This section explains the pins for setting the I/O mode using the base timer I/O select function.

Overview

The base timer has 2 types of external pins and 5 types of internal signals for each channel.

By connecting the external pins and internal signals, signals that correspond to the connection destination (external clock (ECK signal)/external activation trigger (TGIN signal)/wave form (TIN signal)) are input to or output from the base timer.

The external pins and internal signals are connected by setting the I/O mode of the base timer. The pins that are used and the signals to be input/output vary depending on the I/O mode.

- External pin
  - TIOA0 to TIOA7 pins

    These pins are used to output the wave form of the base timer (TOUT signal) or input the external activation trigger (TGIN signal).

    These pins are multiplexed pins. To use them as TIOA0 to TIOA7 pins of the base timer, see "2.4 Setting Method for Pins".
  - TIOB0 to TIOB7 pins

    These pins are used to input the external activation trigger (TGIN signal)/external clock (ECK signal)/wave form of another channel (TIN signal).

    These pins are multiplexed pins. To use them as TIOB0 to TIOB7 pins of the base timer, see "2.4 Setting Method for Pins".
- Internal signal

  By connecting these pins to the above mentioned external pins or by inputting the output signal from another channel, signals is input to or output from the base timer.
  - TOUT signal

    Output wave form of the base timer. (It is not used in the 16/32-bit PWC timer.)
  - ECK signal

    External clock of the base timer. (It is not used in the 16/32-bit PWC timer.)

    This signal is input when the external clock is selected for the count clock.
  - TGIN signal

    External activation trigger of the base timer. (It is not used in the 16/32-bit PWC timer.)

    When the effective edge of the external activation trigger is selected, the edge of this signal is detected to activate the base timer.
  - TIN signal

    The wave form to be measured. (It is used only in the 16/32-bit PWC timer.)
  - DTRG signal

    The base timer stops operation at the falling edge of this signal.
  - COUT signal

    Output signal to other channels.
  - CIN signal

    Signal that is input from other channels.

- Connection of the external pins and internal signals

  The external pins and internal signals are connected by setting the I/O mode of the base timer. Table 21-1 outlines the relationship between the I/O mode and pin connections.

Table 21-1. Relationship Between the I/O Mode and Pin Connections

I/O Mode	TIOAn (Even-numbered Channel)		TIOBn (Even-numbered Channel)		TIOAn+1 (Odd-numbered Channel)		TIOBn+1 (Odd-numbered Channel)	
	Connection Destination	I/O	Connection Destination	I/O	Connection Destination	I/O	Connection Destination	I/O
0	ch.n's TOUT	Output	ch.n's ECK/TGIN/TIN	Input	ch.n+1's TOUT	Output	ch.n+1's ECK/TGIN/TIN	Input
1	ch.n's TOUT	Output	ch.n's ECK	Input	ch.n's TGIN	Input	ch.n's TIN	Input
2	ch.n's TOUT	Output	ch.n/ch.n+1's ECK/TGIN/TIN[1]	Input	ch.n+1's TOUT	Output	Not used	
3	ch.n's TOUT	Output	Not used		ch.n+1's TOUT	Output		
4	ch.n's TOUT	Output	ch.n's ECK/TGIN/TIN	Input	ch.n+1's TOUT	Output		
5	ch.n's TOUT	Output	Not used		ch.n+1's TOUT	Output		
6	ch.n's TOUT	Output			ch.n+1's TOUT	Output		
7	ch.n's TOUT	Output	ch.n's ECK/TGIN/TIN	Input	ch.n+1's TOUT	Output		
8	ch.n's TOUT	Output	Not used		ch.n+1's TOUT	Output		

ch.n even-numbered channel

ch.n+1 odd-numbered channel

n = 0, 2, 4, 6

[1]: Synchronize with the peripheral clock (PCLK)

21.4 Registers

This section explains the configuration and functions of registers used in the base timer I/O select function.

List of registers of the base timer I/O select function

Table 21-2 lists registers of the base timer I/O select function.

Table 21-2. Registers of the Base Timer I/O Select Function

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.3
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	21.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	21.4.2

21.4.1 Base Timer IO Select Register for Ch.0/1/2/3 (BTSEL0123)

This register sets the I/O mode of ch.0 to ch.3 of the base timer.

Figure 21-2 shows the bit configuration of the base timer I/O select register for ch.0/1/2/3 (BTSEL0123).

Figure 21-2. Bit Configuration of Base Timer I/O Select Register for ch.0/1/2/3 (BTSEL0123)

bit	7	6	5	4	3	2	1	0
	SEL23_3	SEL23_2	SEL23_1	SEL23_0	SEL01_3	SEL01_2	SEL01_1	SEL01_0
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

R/W: Read/Write

Note: Change this register after setting the base timer to reset mode in FMD2 to FMD0 bits (FMD2 to FMD0 = 000) of the base timer x timer control register (BTxTMCR).

[bit7 to bit4]: SEL23_3 to SEL23_0 (I/O select bit for ch.2/ch.3)

These bits set the I/O mode for ch.2 and ch.3 of the base timer.

SEL23_3	SEL23_2	SEL23_1	SEL23_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/stop mode)

Note: Setting the values other than above is prohibited.

[bit3 to bit0]: SEL01_3 to SEL01_0 (I/O select bit for ch.0/ch.1)

These bits set the I/O mode of ch.0 and ch.1 of the base timer.

ch.0 and ch.1 are the lowest channels of the base timer so that modes that use signals from the lower side channels cannot be used in these channels. Therefore, the setting of the following modes is prohibited.

- I/O mode 3 (other channel trigger shared mode)
- I/O mode 8 (other channel trigger shared timer activation/stop mode)

SEL01_3	SEL01_2	SEL01_1	SEL01_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	Setting prohibited
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	Setting prohibited

Note: Setting the values other than above is prohibited.

21.4.2 Base Timer IO Select Register for Ch.4/5/6/7 (BTSEL4567)

This register sets the I/O mode of ch.4 to ch.7 of the base timer.

Figure 21-3 shows the bit configuration of the base timer I/O select register for ch.4/5/6/7 (BTSEL4567).

Figure 21-3. Bit Configuration of Base Timer I/O Select Register for ch.4/5/6/7 (BTSEL4567)

bit	7	6	5	4	3	2	1	0
	SEL67_3	SEL67_2	SEL67_1	SEL67_0	SEL45_3	SEL45_2	SEL45_1	SEL45_0
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

R/W: Read/Write

Note: Change this register after setting the base timer to reset mode in FMD2 to FMD0 bits (FMD2 to FMD0 = 000) of the base timer x timer control register (BTxTMCR).

[bit7 to bit4]: SEL67_3 to SEL67_0 (I/O select bit for ch.6/ch.7)

These bits set the I/O mode of ch.6 and ch.7 of the base timer.

SEL67_3	SEL67_2	SEL67_1	SEL67_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/stop mode)

Note: Setting the values other than above is prohibited.

[bit3 to bit0]: SEL45_3 to SEL45_0 (I/O select bit for ch.4/ch.5)

These bits set the I/O mode of ch.4 and ch.5 of the base timer.

SEL45_3	SEL45_2	SEL45_1	SEL45_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/stop mode)

Note: Setting the values other than above is prohibited.

21.4.3 Base Timer Same Time Soft Start Register (BTSSSR)

This register simultaneously activates the base timers using the software.

Up to 8 channels corresponding to the bits in which "1" is written can be simultaneously activated.

Figure 21-4 shows the bit configuration of the base timer same time soft start register (BTSSSR).

Figure 21-4. Bit Configuration of Base Timer Same Time Soft Start Register (BTSSSR)

	bit	15	14	13	12	11	10	9	8
		Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Attribute		W	W	W	W	W	W	W	W
Initial value		X	X	X	X	X	X	X	X

	bit	7	6	5	4	3	2	1	0
		SSSR7	SSSR6	SSSR5	SSSR4	SSSR3	SSSR2	SSSR1	SSSR0
Attribute		W	W	W	W	W	W	W	W
Initial value		X	X	X	X	X	X	X	X

W: Write only

X: Undefined

Notes:

- Do not write to this register when the modes other than the following are set.
  - I/O mode 5 (same time software activation mode)
  - I/O mode 6 (software activation timer activation/stop mode) (only for even-numbered channels)
- For channels that are activated using this register, set the trigger input edge to the rising edge in the EGS1 and EGS0 bits (EGS1, EGS 0 = 01) of the base timer x timer control register (BTxTMCR).

[bit15 to bit8]: Undefined bits

In case of writing	Always write "0" to this (these) bit (bits).

[bit7]: SSSR7 (Same time software start bit for ch.7)

This bit activates the ch.7 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.7 of the base timer.[1]

[1]: Only when the I/O mode is set to "5" (same time software activation mode) in SEL67_3 to SEL67_0 bits of the base timer I/O select register for ch.4/5/6/7 (BTSEL4567) (SEL67_3 to SEL67_0 = 0101)

[bit6]: SSSR6 (Same time software start bit for ch.6)

This bit activates the ch.6 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.6 of the base timer.[1]

[1]: Only when the I/O mode is set to either of the following modes in the SEL67_3 to SEL67_0 bits of the base timer I/O select register for ch.4/5/6/7 (BTSEL4567)

- "5" (Same time software activation mode) (SEL67_3 to SEL67_0 = 0101)
- "6" (Software activation timer activation/stop mode) (SEL67_3 to SEL67_0 = 0110)

[bit5]: SSSR5 (Same time software start bit for ch.5)

This bit activates the ch.5 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.5 of the base timer.[1]

[1]: Only when the I/O mode is set to "5" (same time software activation mode) in SEL45_3 to SEL45_0 bits of the base timer I/O select register for ch.4/5/6/7 (BTSEL4567) (SEL45_3 to SEL45_0 = 0101)

[bit4]: SSSR4 (Same time software start bit for ch.4)

This bit activates the ch.4 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.4 of the base timer.[1]

[1]: Only when the I/O mode is set to either of the following modes in the SEL45_3 to SEL45_0 bits of the base timer I/O select register for ch.4/5/6/7 (BTSEL4567)

- "5" (Same time software activation mode) (SEL45_3 to SEL45_0 = 0101)
- "6" (Software activation timer activation/stop mode) (SEL45_3 to SEL45_0 = 0110)

[bit3]: SSSR3 (Same time software start bit for ch.3)

This bit activates the ch.3 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.3 of the base timer.[1]

[1]: Only when the I/O mode is set to "5" (same time software activation mode) in SEL23_3 to SEL23_0 bits of the base timer I/O select register for ch.0/1/2/3 (BTSEL0123) (SEL23_3 to SEL23_0 = 0101)

[bit2]: SSSR2 (Same time software start bit for ch.2)

This bit activates the ch.2 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.2 of the base timer.[1]

[1]: Only when the I/O mode is set to either of the following modes in the SEL23_3 to SEL23_0 bits of the base timer I/O select register for ch.0/1/2/3 (BTSEL0123)

- "5" (Same time software activation mode) (SEL23_3 to SEL23_0 = 0101)
- "6" (Software activation timer activation/stop mode) (SEL23_3 to SEL23_0 = 0110)

[bit1]: SSSR1 (Same time software start bit for ch.1)

This bit activates the ch.1 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.1 of the base timer.[1]

[1]: Only when the I/O mode is set to "5" (same time software activation mode) in SEL01_3 to SEL01_0 bits of the base timer I/O select register for ch.0/1/2/3 (BTSEL0123) (SEL01_3 to SEL01_0 = 0101)

[bit0]: SSSR0 (Same time software start bit for ch.0)

This bit activates the ch.0 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.0 of the base timer.[1]

[1]: Only when the I/O mode is set to either of the following modes in the SEL01_3 to SEL01_0 bits of the base timer I/O select register for ch.0/1/2/3 (BTSEL0123)

- "5" (Same time software activation mode) SEL01_3 to SEL01_0)
- "6" (Software activation timer activation/stop mode) (SEL01_3 to SEL01_0)

21.5 I/O Mode

Operations of the external pins and activation/stop timing of the base timer vary depending on the I/O mode set in the base timer I/O select register (BTSEL0123, BTSEL4567).

21.5.1 I/O Mode 0 (16-bit Timer Standard Mode)

In this mode, each channel of the base timer is used separately.

Table 21-3 lists the external pins used when this mode is set.

Table 21-3. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	1
Output pin	1	1

Table 21-4 lists the connection destinations of the external pins used and I/O signals.

Table 21-4. Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOA0 to TIOA7	Output	TOUT	Output wave form the base timer
TIOB0 to TIOB7	Input	ECK/TGIN/TIN[1]	Use the signals that have been input as one of the following: ■ External clock (ECK signal) ■ External activation trigger (TGIN signal) ■ Measured wave form (TIN signal)

[1]: Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21-5 is a block diagram of I/O mode 0 (16-bit timer standard mode), taking ch.0 as an example.

Figure 21-5. Block Diagram of I/O Mode 0 (16-bit Timer Standard Mode)

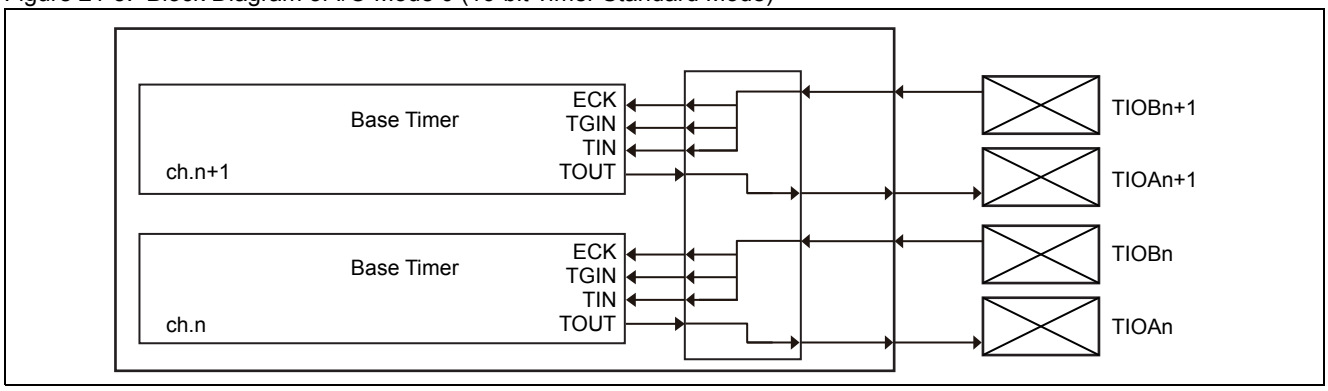


Table 21-5 lists the connections for I/O mode 0.

Table 21-5. Connections for I/O Mode 0

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
Input signal from the TIOBn pin	Input to ch.n as TIN/TGIN/ECK
TOUT signal of ch.n+1	Output from the TIOAn+1 pin
Input signal from the TIOBn+1 pin	Input to ch.n+1 as TIN/TGIN/ECK

n=0, 2, 4, 6

21.5.2 I/O Mode 1 (Timer Full Mode)

In this mode, signals from the even-numbered channels are allocated to all the external pins separately to operate the timer.

Table 21-6 lists the external pins used when this mode is set.

Table 21-6. External Pins Used

	Even-numbered Channel
Input pin	3
Output pin	1

Table 21-7 lists the connection destinations of the external pins used and I/O signals.

Table 21-7. Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOBn	Input	ECK of the even-numbered channel	Input the external clock (ECK signal) to the even-numbered channel
TIOAn+1	Input	TGIN of the even-numbered channel	Input the external activation trigger (TGIN signal) to the even-numbered channel
TIOBn+1	Input	TIN of the even-numbered channel	Input the measured wave form (TIN signal) in the even-numbered channel

n=0, 2, 4, 6

Figure 21-6 is a block diagram of I/O mode 1 (timer full mode).

Figure 21-6. Example of Block Diagram of I/O Mode 1 (Timer Full Mode)

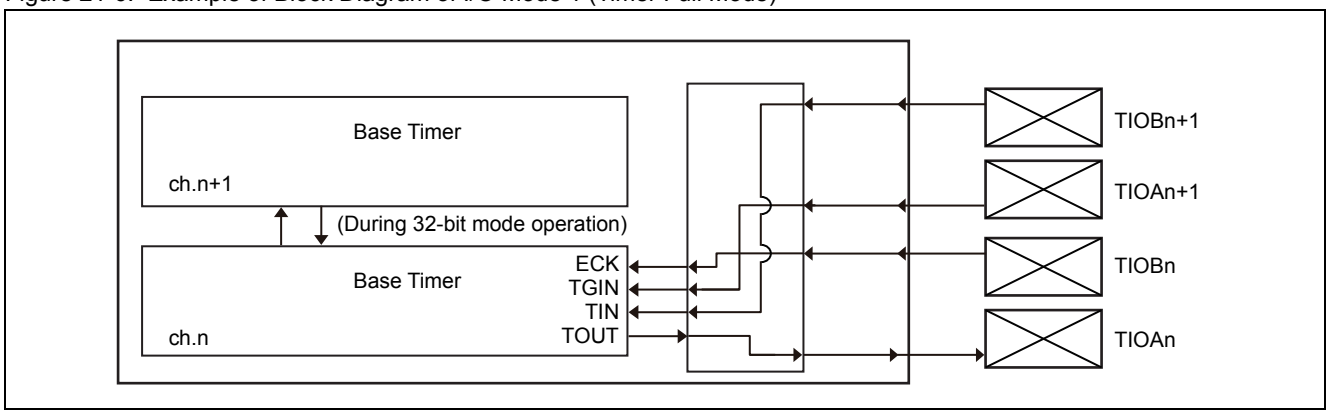


Table 21-8 lists the connections for I/O mode 1.

Table 21-8. Connections for I/O Mode 1

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
Input signal from the TIOBn pin	Input to ch.n as a ECK signal
Input signal from the TIOAn+1 pin	Input to ch.n as a TGIN signal
Input signal from the TIOBn+1 pin	Input to ch.n as an TIN signal

n=0, 2, 4, 6

Note: If this mode is set, set the TIOAn pins (TIOA1, TIOA3, TIOA5, TIOA7) corresponding to the odd-numbered channel to the port input mode in the port function register (PFR). For details of the setting of pins, see "2.4 Setting Method for Pins".

21.5.3 I/O Mode 2 (External Trigger Shared Mode)

In this mode, input signals to the base timer (ECK/TGIN/TIN) are shared by 2 channels.

Table 21-9 lists the external pins used when this mode is set.

Table 21-9. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1 (shared by 2 channels)	
Output pin	1	1

Table 21-10 lists the connection destinations of the external pins used and I/O signals.

Table 21-10. Connection Destinations of the External Pins and I/O Signals

External pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn	Input	ECK/TGIN/TIN of the even/odd-numbered channel[1]	Input to both of the even/odd-numbered channels (synchronized with the peripheral clock (PCLK)) and use it as one of the following: ■ External clock (ECK signal) ■ External activation trigger (TGIN signal) ■ Measured wave form (TIN signal)
TIOBn+1	-	-	Not used

n=0, 2, 4, 6
[1]: Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21-7 is a block diagram of I/O mode 2 (external trigger shared mode).

Figure 21-7. Block Diagram of I/O Mode 2 (External Trigger Shared Mode)

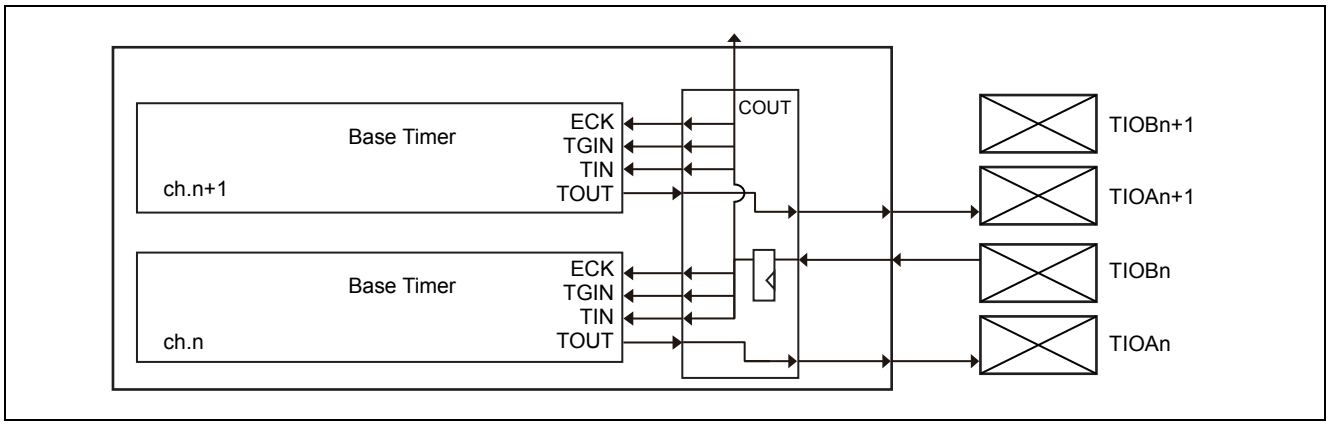


Table 21-11 lists the connections for I/O mode 2.

Table 21-11. Connections for I/O Mode 2

Connection Source	Connection Destination	Remarks
TOUT signal of ch.n	Output from the TIOAn pin	
Input signal from the TIOBn pin	■ Input to ch.n and ch.n+1 as TIN/TGIN/ECK signals ■ Output to another channel as the COUT signal	Synchronization with the peripheral clock (PCLK)
TOUT signal of ch.n+1	Output from the TIOAn+1 pin	

n=0, 2, 4, 6

Note: If the upper 2 channels (n + 2, n + 3) of those that have been set to this mode are set to I/O mode 3 (other channel trigger shared mode), the input signals (ECK/TGIN/TIN) can be input to 4 channels at the same time.
(Example: If this mode is set for ch.0 and ch.1 and I/O mode 3 is set for ch.2 and ch.3, the input signals (ECK/TGIN/TIN) can be input to all 4 channels of ch.0 to ch.3 at the same time.)

21.5.4 I/O Mode 3 (Other Channel Trigger Shared Mode)

In this mode, the COUT signal of the channel that is lower by 2 channels is input as a CIN signal to be used as the ECK/TGIN/TIN signal.

Table 21-12 lists the external pins used when this mode is set.

Table 21-12. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21-13 lists the connection destinations of the external pins used and I/O signals.

Table 21-13. Connection Destinations of the External Pins and I/O Signals

External pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=2, 4, 6

Figure 21-8 is a block diagram of I/O mode 3 (other channel trigger shared mode).

Figure 21-8. Block Diagram of I/O Mode 3 (Other Channel Trigger Shared Mode)

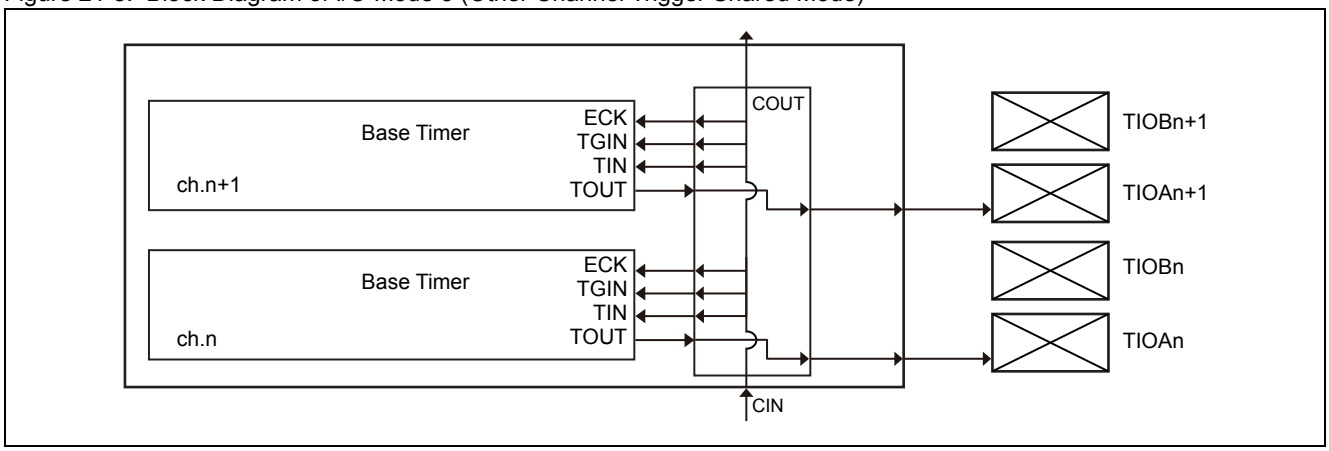


Table 21-14 lists the connections for I/O mode 3.

Table 21-14. Connections for I/O Mode 3

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
CIN signal[1]	■ Input to ch.n and ch.n+1 as the TIN/TGIN/ECK signal ■ Output to another channel as the COUT signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=2, 4, 6
[1]: Input the COUT signal of the other channel as the CIN signal.

The signals of ch.n-2/n-1 that can be input to ECK, TGIN and TIN of ch.n/n+1 are as below.

- The signal that synchronized TIOBn-2 input of input/output mode 2 with peripheral clock.
- The trigger signal input from ch.n-4/n-3 of input/output mode 3.
- TIONAn-2 output of input/output mode 4.
- TIONAn-2 output of input/output mode 6.
- TIONAn-2 output of input/output mode 7.
- The trigger signal input from ch.n-4/n-3 of input/output mode 8.

Notes:

- Set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0=01) of the base timer x timer control register (BTxTMCR).
- Channels that have been set to this mode use the COUT signal of the channels (n - 2, n - 1) that are lower by 2 channels, as the CIN signal input.
  (Example: If ch.2 and ch.3 are set to this mode, they use the COUT signal of ch.0 and ch.1.)
  Therefore, ch.0 and ch.1 cannot be set to this mode.

21.5.5 Operations in I/O Mode 4 (Timer Activation/Stop Mode)

This mode enables control of activation/stop of the odd-numbered channel by using the even-numbered channel.

The odd-numbered channel is activated at the rising edge of the output wave form (TOUT signal) of the even-numbered channel and stops at the falling edge.

Table 21-15 lists the external pins used when this mode is set.

Table 21-15. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	Not used
Output pin	1	1

Table 21-16 lists the functions of pins.

Table 21-16. Functions of Pins

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn	Input	ECK/TGIN/TIN of the even-numbered channel[1]	Input to the even-numbered channel and use as one of the following. ■ External clock (ECK signal) ■ External activation trigger (TGIN signal) ■ Measured wave form (TIN signal)
TIOBn+1	-	-	Not used

n=0, 2, 4, 6

[1]: Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21-9 is a block diagram of I/O mode 4 (timer activation/stop mode).

Figure 21-9. Block Diagram of I/O Mode 4 (Timer Activation/Stop Mode)

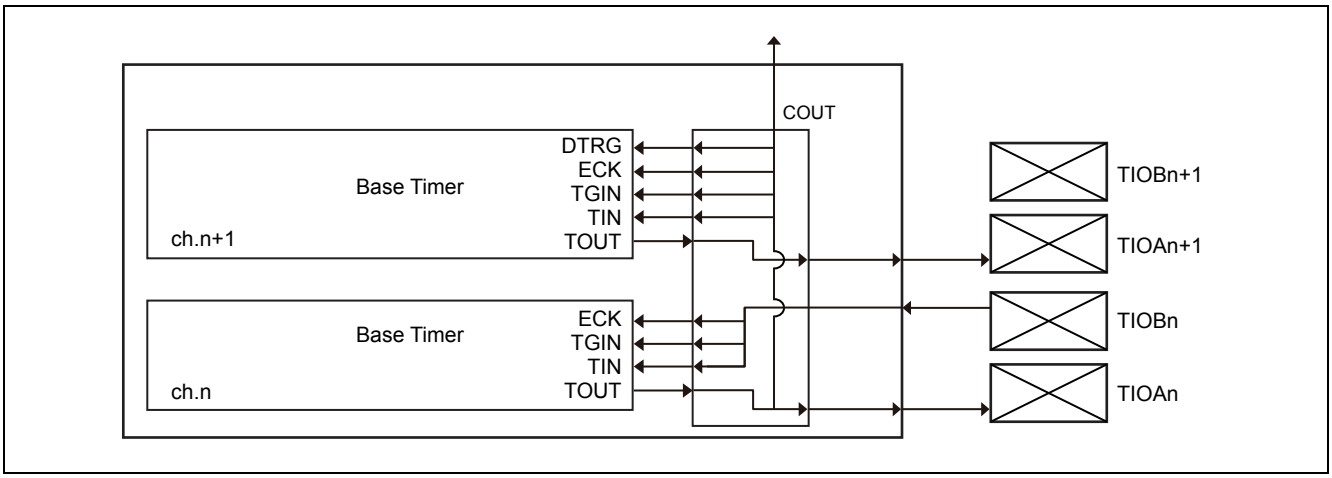


Table 21-17 lists the connections for I/O mode 4.

Table 21-17. Connections for I/O Mode 4

Connection Source	Connection Destination
TOUT signal of ch.n	■ Output from the TIOAn pin ■ Input to ch.n+1 as the TIN/TGIN/ECK signal and DTRG signal ■ Output to another channel as the COUT signal
Input signal from the TIOBn pin	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=0, 2, 4, 6

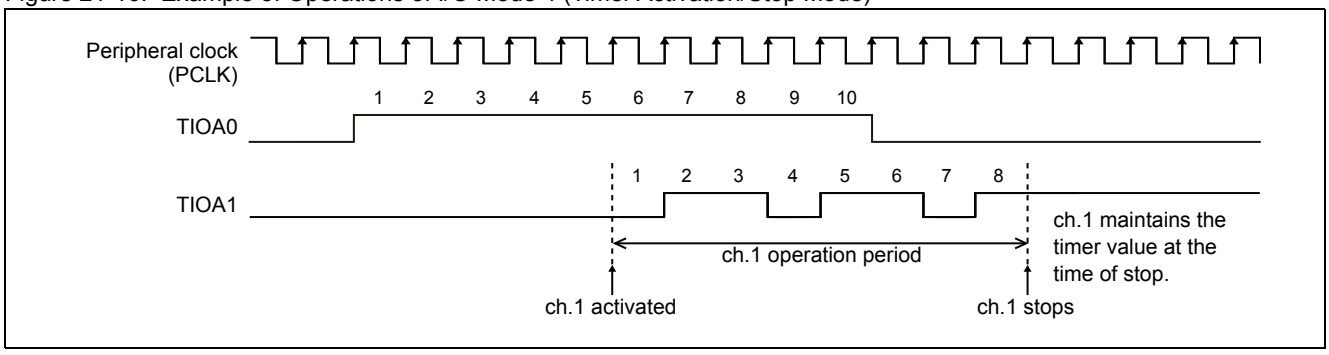
Notes:

- Set the trigger input edge of the odd-numbered channel to the rising edge in the EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).
- The odd-numbered channel stops operation when the falling edge is detected in the DTRG signal.

Figure 21-10 shows the operation when I/O mode 4 (timer activation/stop mode) is set, taking as an example the case where ch.0 and ch.1 are used as the PWM timer.

Register (ch.0)	Setting Value	Register (ch.1)	Setting Value
Base timer 0 cycle setting register (BT0PCSR)	0010_H	Base timer 1 cycle setting register (BT1PCSR)	0002_H
Base timer 0 duty setting register (BT0PDUT)	0009_H	Base timer 1 duty setting register (BT1PDUT)	0001_H
Base timer 0 timer control register (BT0TMCR)	0013_H	Base timer 1 timer control register (BT1TMCR)	0112_H

Figure 21-10. Example of Operations of I/O Mode 4 (Timer Activation/Stop Mode)



21.5.6 Operations in I/O Mode 5 (Same Time Software Activation Mode)

This mode enables activating multiple channels at the same time by using the base timer same time soft start register (BTSSSR).

All channels corresponding to the bits in which "1" is written in the base timer same time soft start register (BTSSSR) are activated at the same time.

Table 21-18 lists the external pins used when this mode is set.

Table 21-18. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21-19 lists the connection destinations of the external pins used and I/O signals.

Table 21-19. Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=0, 2, 4, 6

Figure 21-11 is a block diagram of I/O mode 5 (same time software activation mode).

Figure 21-11. Block Diagram of I/O Mode 5 (Same Time Software Activation Mode)

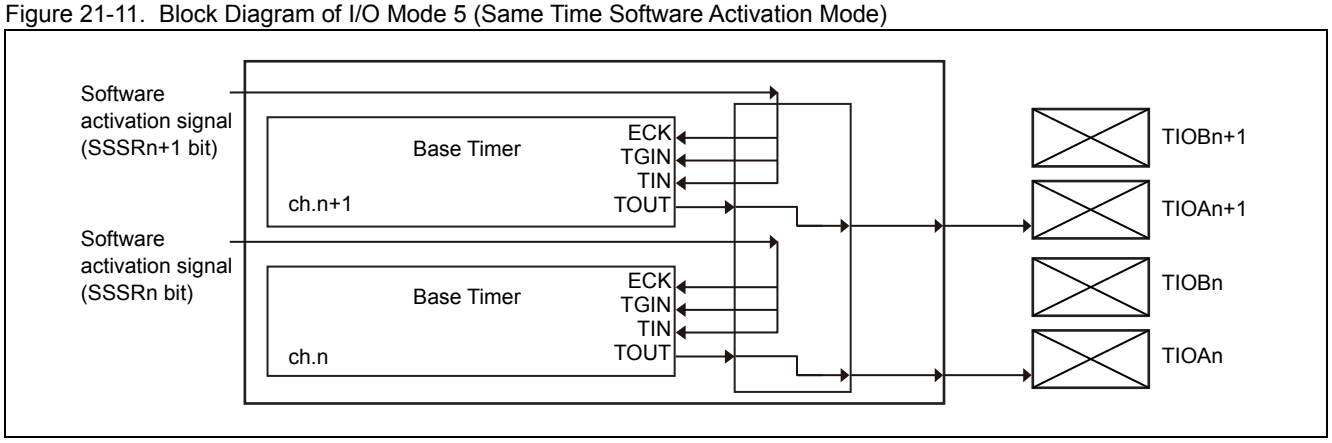


Table 21-20 lists the connections for I/O mode 5.

Table 21-20. Connections for I/O Mode 5

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
Software activation signal (Writing "1" in SSSRn bit of BTSSSR)	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin
Software activation signal (Writing "1" in SSSRn+1 bit of BTSSSR)	Input to ch.n+1 as the TIN/TGIN/ECK signal

n=0, 2, 4, 6
BTSSSR: Base timer same time soft start register (BTSSSR)

If "1" is written in the base timer same time soft start register (BTSSSR), the rising edge is input (ECK/TGIN/TIN signal) in the channels that correspond to the written bits.

Note: Set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).

21.5.7 Operations in I/O Mode 6 (Software Activation Timer Activation/Stop Mode)

This mode enables control of activation/stop of the odd-numbered channel by using the even-numbered channel.

The even-numbered channel is activated by writing "1" in the base timer same time soft start register (BTSSSR).

The odd-numbered channel is activated when the rising edge is detected in the output wave form (TOUT signal) of the even-numbered channel and stops when the falling edge is detected.

Table 21-21 lists the external pins used when this mode is set.

Table 21-21. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21-22 lists the connection destinations of the external pins used and I/O signals.

Table 21-22. Connection Destinations of the External Pins and I/O Signals

Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=0, 2, 4, 6

Figure 21-12 is a block diagram of I/O mode 6 (software activation timer activation/stop mode).

Figure 21-12. Block Diagram of I/O Mode 6 (Software Activation Timer Activation/Stop Mode)

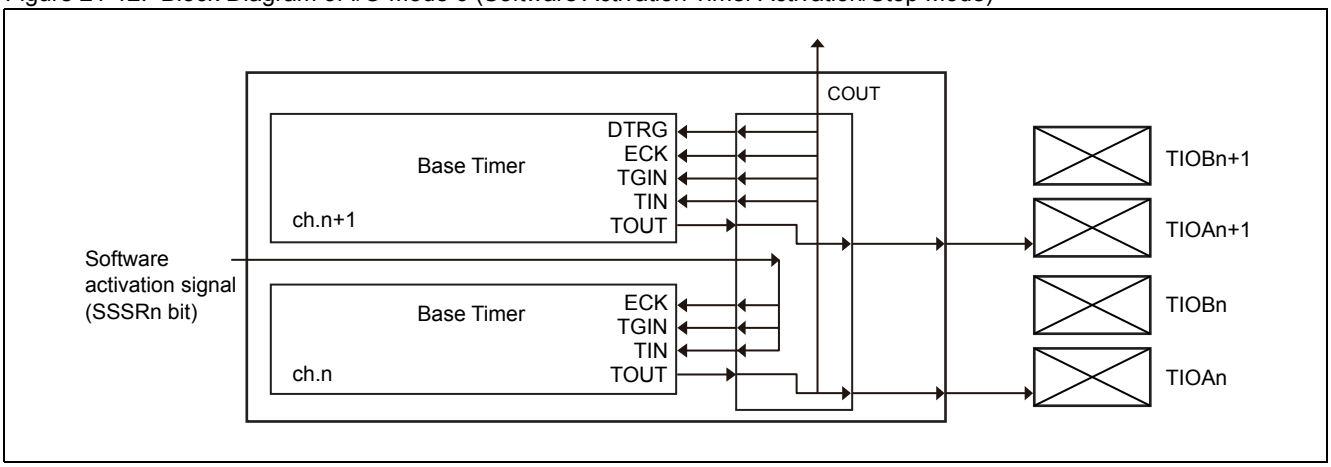


Table 21-23 lists the connections for I/O mode 6.

Table 21-23. Connections for I/O Mode 6

Connection Source	Connection Destination
TOUT signal of ch.n	■ Output from the TIOAn pin ■ Input to ch.n+1 as the TIN/TGIN/ECK/DTRG signal ■ Output to another channel as the COUT signal
Software activation signal (Writing "1" in SSSRn bit of BTSSSR)	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=0, 2, 4, 6
BTSSSR: Base timer same time soft start register (BTSSSR)

If "1" is written in the bits of the base timer same time soft start register (BTSSSR) that correspond to the even-numbered channels to be activated, the rising edge is input (ECK, TGIN, TIN signal) in the corresponding channels.

Start-up and stop timing of ch.n are same as input/output mode4.

Notes:

- Set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).
- The odd-numbered channel stops operation when the falling edge is detected in the DTRG signal.

21.5.8 Operations in I/O Mode 7 (Timer Activation Mode)

In this mode, the output wave form (TOUT signal) of the even-numbered channel is used as input signals (ECK/TGIN/TIN signal) of the odd-numbered channel.

Table 21-24 lists the external pins used when this mode is set.

Table 21-24. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	Not used
Output pin	1	1

Table 21-25 lists the connection destinations of the external pins used and I/O signals.

Table 21-25. Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn	Input	ECK/TGIN/TIN of the even-numbered channel[1]	Input to the even-numbered channel and use as one of the following. ■ External clock (ECK signal) ■ External activation trigger (TGIN signal) ■ Measured wave form (TIN signal)
TIOBn+1	-	-	Not used

n=0, 2, 4, 6
[1]: Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21-13 is a block diagram of I/O mode 7 (timer activation mode).

Figure 21-13. Block Diagram of I/O Mode 7 (Timer Activation Mode)

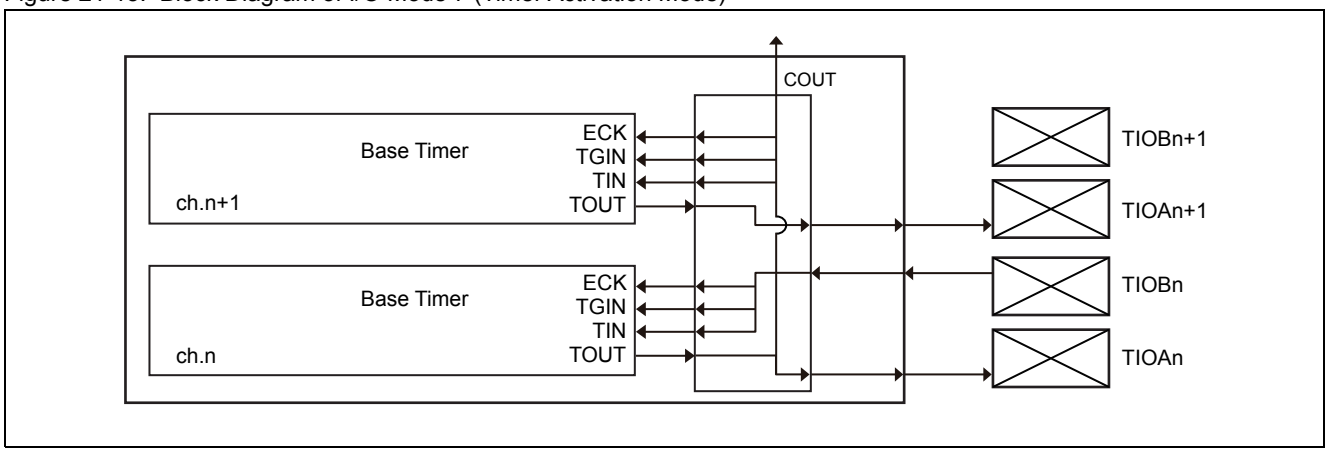


Table 21-26 lists the connection for I/O mode 7.

Table 21-26. Connection for I/O Mode 7

Connection Source	Connection Destination
TOUT signal of ch.n	■ Output from the TIOAn pin ■ Input to ch.n+1 as the TIN/TGIN/ECK signal ■ Output to another channel as the COUT signal
Input signal from the TIOBn pin	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=0, 2, 4, 6
Start-up timing of ch.n is same as input/output mode4.

21.5.9 Operations in I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode)

In this mode, the COUT signal of the channel that is lower by 2 channels is input as the CIN signal to be used as the external activation trigger (TGIN signal).

Table 21-27 lists the external pins used when this mode is set.

Table 21-27. External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21-28 lists the connection destinations of the external pins used and I/O signals.

Table 21-28. Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=2, 4, 6

Figure 21-14 is a block diagram of I/O mode 8 (other channel trigger shared timer activation/stop mode).

Figure 21-14. Block Diagram of I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode)

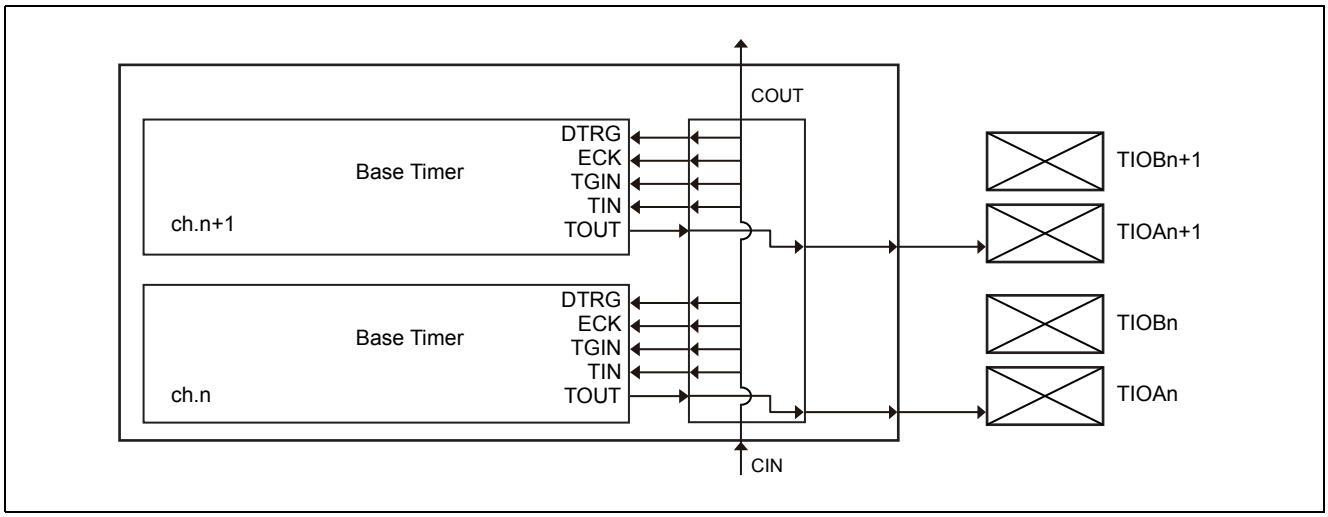


Table 21-29 lists the connections for I/O mode 8.

Table 21-29. Connections for I/O Mode 8

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
CIN signal[1]	■ Input to ch.n and ch.n+1 as the TIN/TGIN/ECK signal and DTRG signal ■ Output to another channel as the COUT signal

n=2, 4, 6
[1]: Input the COUT signal of the other channel as the CIN signal.

The signals of ch.n-2/n-1 that can be input to ECK, TGIN and TIN of ch.n/n+1 are as below.

- The signal that synchronized TIOBn-2 input of input/output mode 2 with peripheral clock.
- The trigger signal input from ch.n-4/n-3 of input/output mode 3.
- TIONAn-2 output of input/output mode 4.
- TIONAn-2 output of input/output mode 6.
- TIONAn-2 output of input/output mode 7.
- The trigger signal input from ch.n-4/n-3 of input/output mode 8.

Notes:

- Channels that have been set to this mode use the COUT signal of the channels (n - 2, n - 1) that are lower by 2 channels, as the CIN signal input.

  (Example: If ch.2 and ch.3 are set to this mode, they use the COUT signal of ch.0 and ch.1.)

  Therefore, ch.0 and ch.1 cannot be set to this mode.
- For the channels that are set to this mode, set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).

  However, the above setting does not apply to the case where the timer function is set to 16/32-bit PWC timer in the FMD2 to FMD0 bits (FMD2 to FMD0 = 100) of the base timer x timer control register (BTxTMCR).
- The odd-numbered channel stops operation when the falling edge is detected in the DTRG signal.

22. Base Timer

This chapter provides an overview of the base timer, summarizes its register configuration and functions, and describes its operations.

22.1 Overview of the Base Timer

22.2 Block Diagrams of the Base Timer

22.3 Base Timer's Registers

22.4 Operations of the Base Timer

22.5 32-bit Mode Operations

22.6 Notes of Using the Base Timer

22.7 Base Timer Interrupts

22.8 Base Timer Description by Function Mode

22.1 Overview of the Base Timer

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section outlines the base timer in each function mode available. This series is equipped with 8 channels.

Function mode bit settings and timer function modes assigned

FMD2/FMD1/FMD0 bit Settings	Timer Function Mode
000_B	Reset mode
001_B	16-bit PWM timer
010_B	16-bit PPG timer
011_B	16/32-bit reload timer
100_B	16/32-bit PWC timer

Reset mode

Placing the base timer in this mode resets its macro (with each register reset to the initial value). Place the base timer in this mode once before changing its function mode or T32 bit setting. After a reset, however, the base timer can set its function mode and the T32 bit without entering the reset mode in advance.

16-bit PWM timer

The 16-bit PWM timer mainly consists of a 16-bit down counter, a 16-bit data register buffered for period setting, a 16-bit compare register buffered for duty cycle setting, and a pin controller.

Period data and duty cycle data can be updated during timer operation as they are held in their buffered respective registers.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256) and three different external events (rising edge, falling edge and both edge detection).

The PWM timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the PWM timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

16-bit PPG timer

The 16-bit PPG timer mainly consists of a 16-bit down counter, a 16-bit data register for "H"-width setting, a 16-bit data register for "L"-width setting, and a pin controller.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256) and three different external events (rising edge, falling edge and both edge detection).

The PPG timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the PPG timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

16/32-bit reload timer

The 16/32-bit reload timer mainly consists of a 16-bit down counter, a 16-bit reload register, and a pin controller.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256) and three different external events (rising edge, falling edge and both edge detection).

The reload timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the reload timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

16/32-bit PWC timer

The 16/32-bit PWC timer mainly consists of a 16-bit up counter, a measurement input pin, and control registers.

The PWC timer measures the time between arbitrary events based on the pulse input from an external source.

The reference count clock can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256).

Measurement modes: "H" pulse width (↑ to ↓) / "L" pulse width (↓ to ↑)
Rising period (↑ to ↑) / Falling period (↓ to ↓)
Inter-edge measurement (↑ or ↓ to ↓ or ↑)

The PWC timer can generate an interrupt request upon completion of measurement.

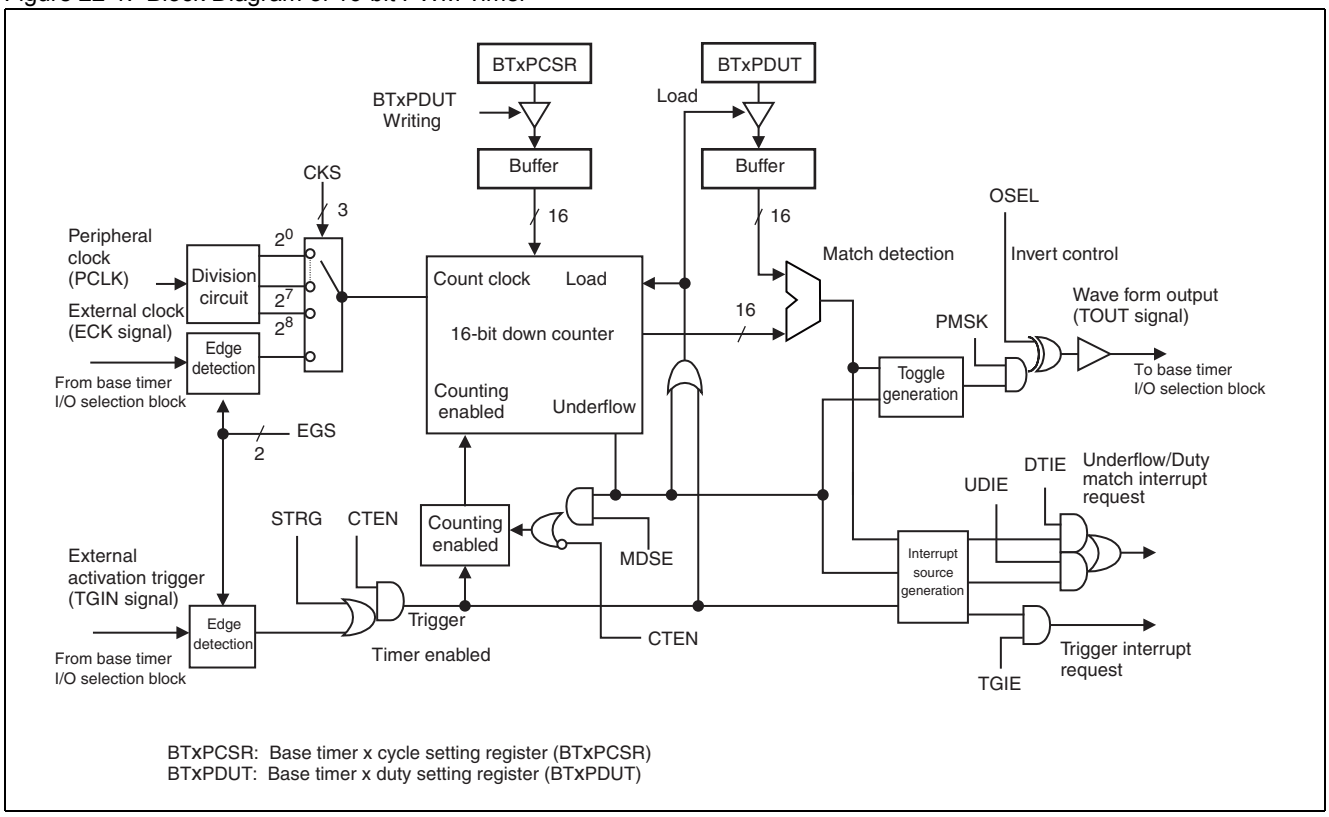
The PWC timer can select one-shot measurement or continuous measurement.

22.2 Block Diagrams of the Base Timer

This section provides a block diagram of the base timer in each function mode.

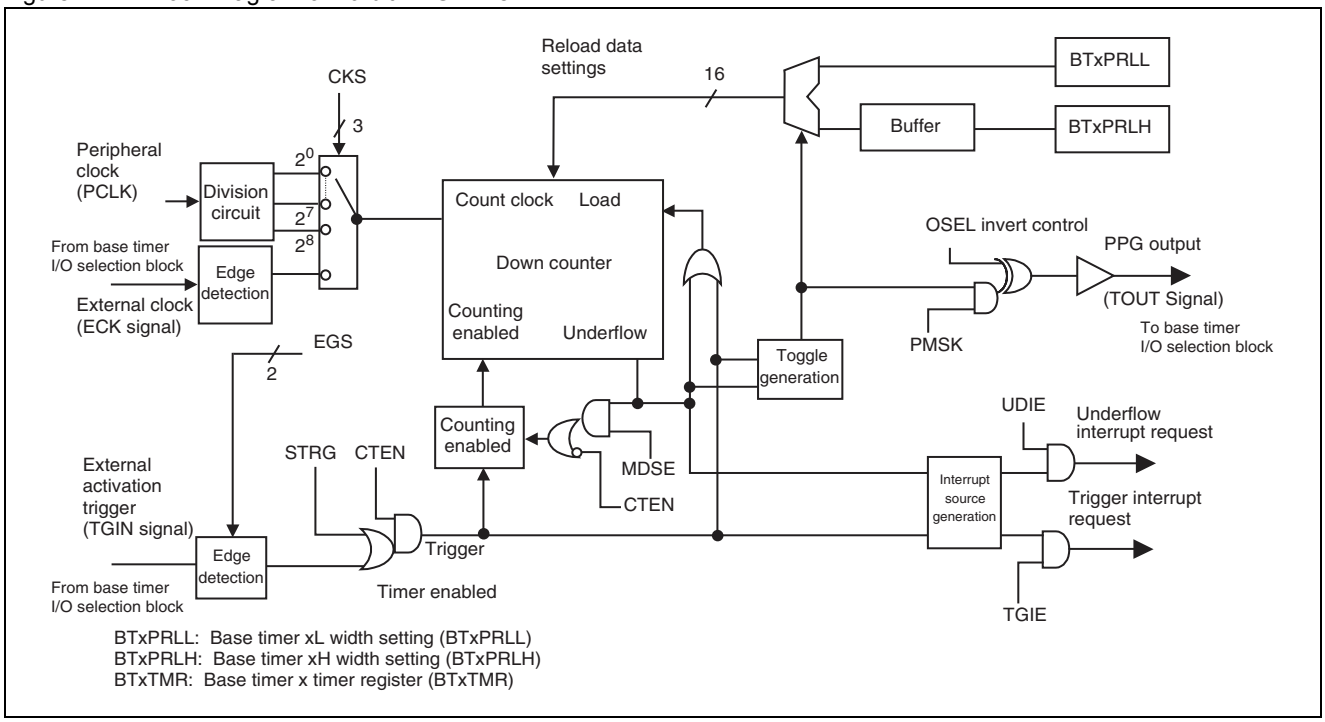
Block diagram of 16-bit PWM timer

Figure 22-1. Block Diagram of 16-bit PWM Timer



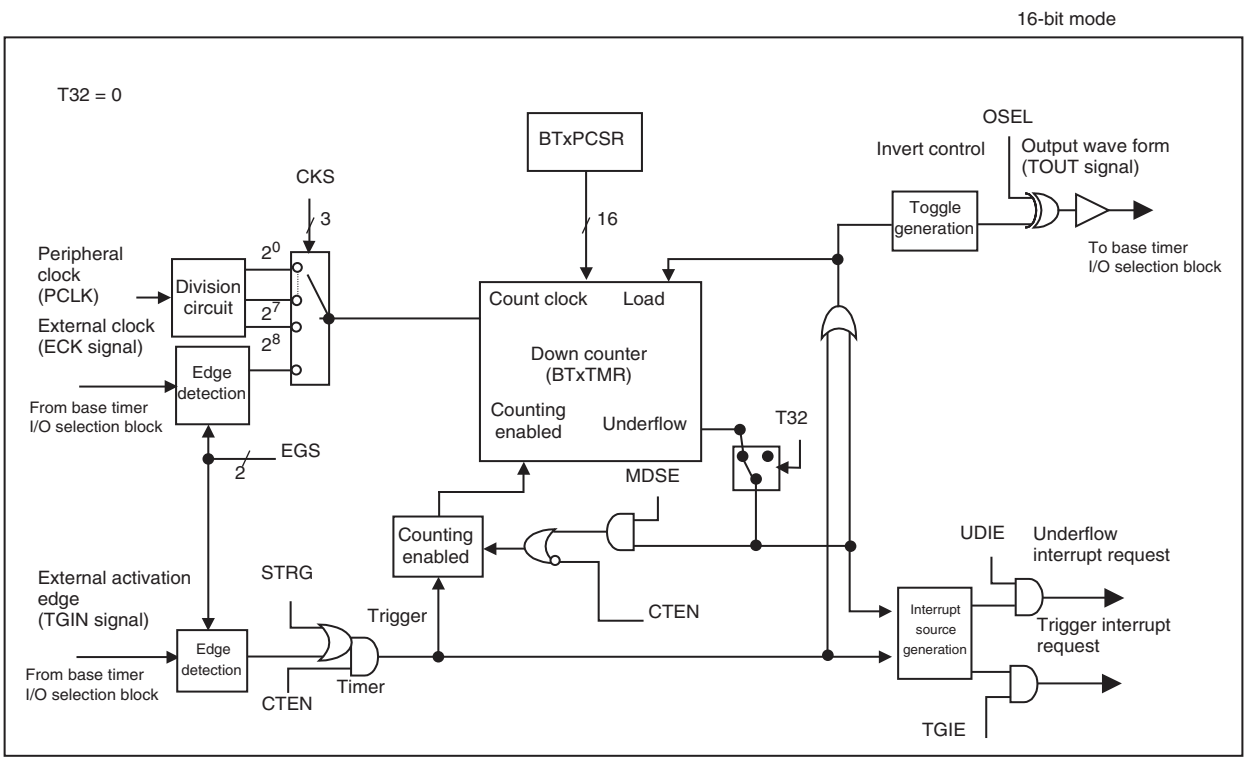
Block diagram of 16-bit PPG timer

Figure 22-2. Block Diagram of 16-bit PPG Timer



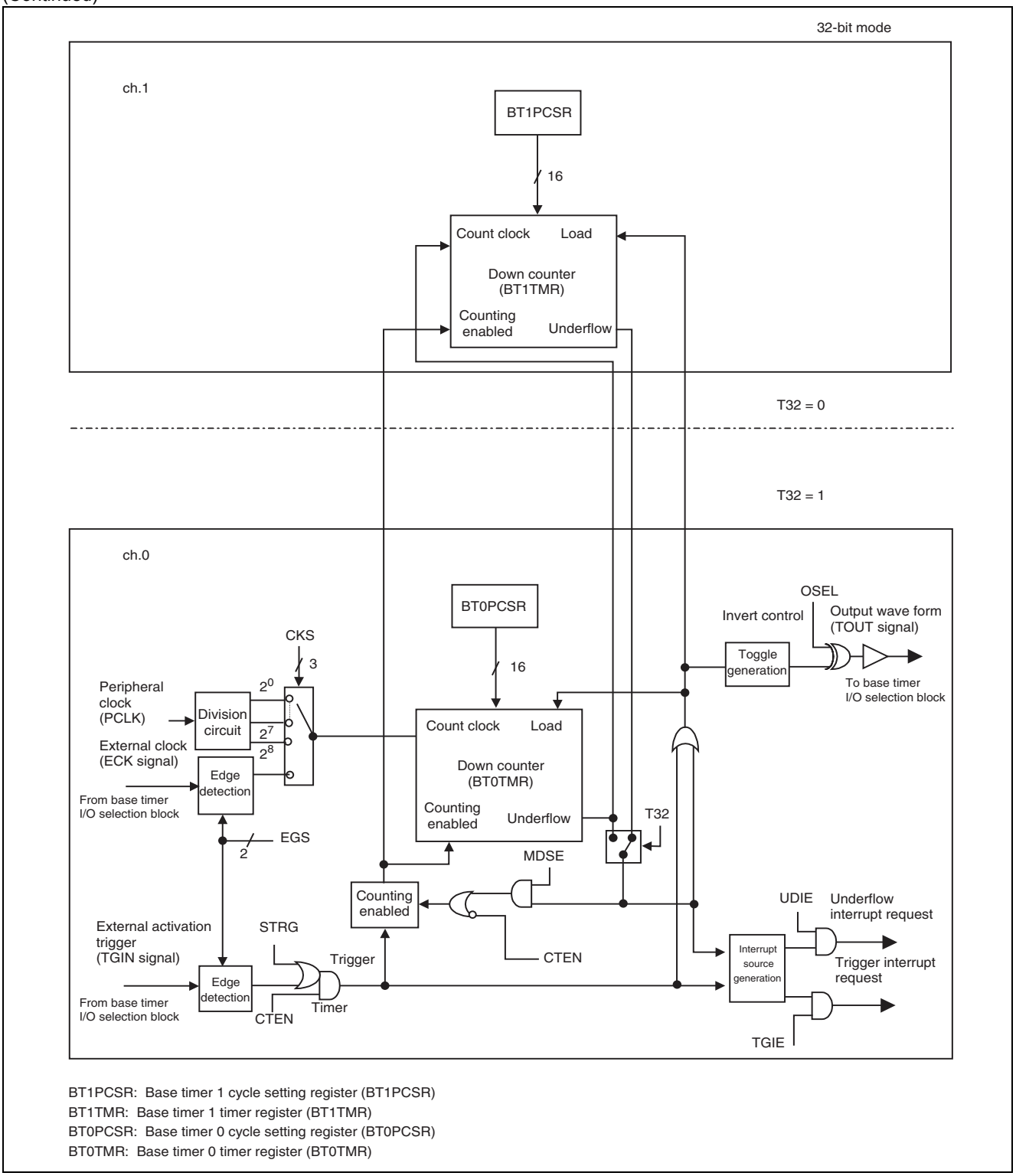
Block diagram of 16/32-bit reload timer (ch.1, ch.0)

Figure 22-3. Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)



(Continued)

(Continued)



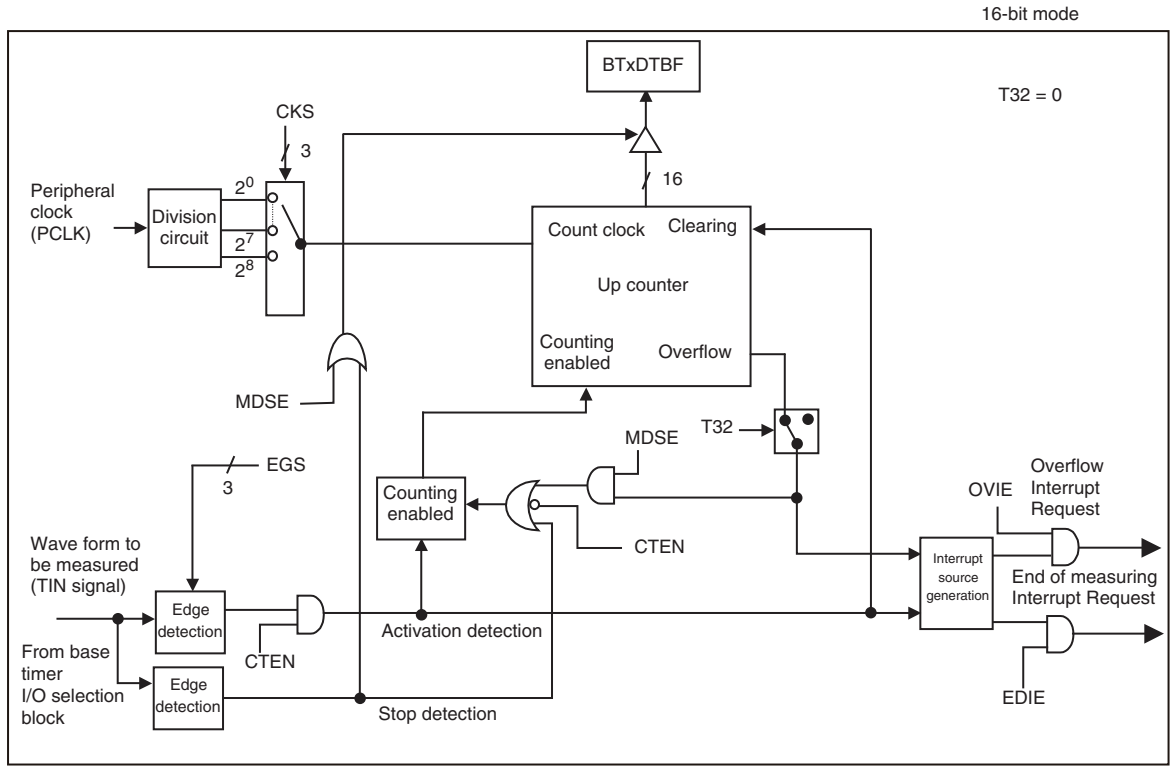
BT1PCSR: Base timer 1 cycle setting register (BT1PCSR)
BT1TMR: Base timer 1 timer register (BT1TMR)
BT0PCSR: Base timer 0 cycle setting register (BT0PCSR)
BT0TMR: Base timer 0 timer register (BT0TMR)

Notes:

- The reload timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, and between ch.6 and ch.7. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "21. Base Timer I/O Select Function".

Block diagram of 16/32-bit PWC timer (ch.1, ch.0)

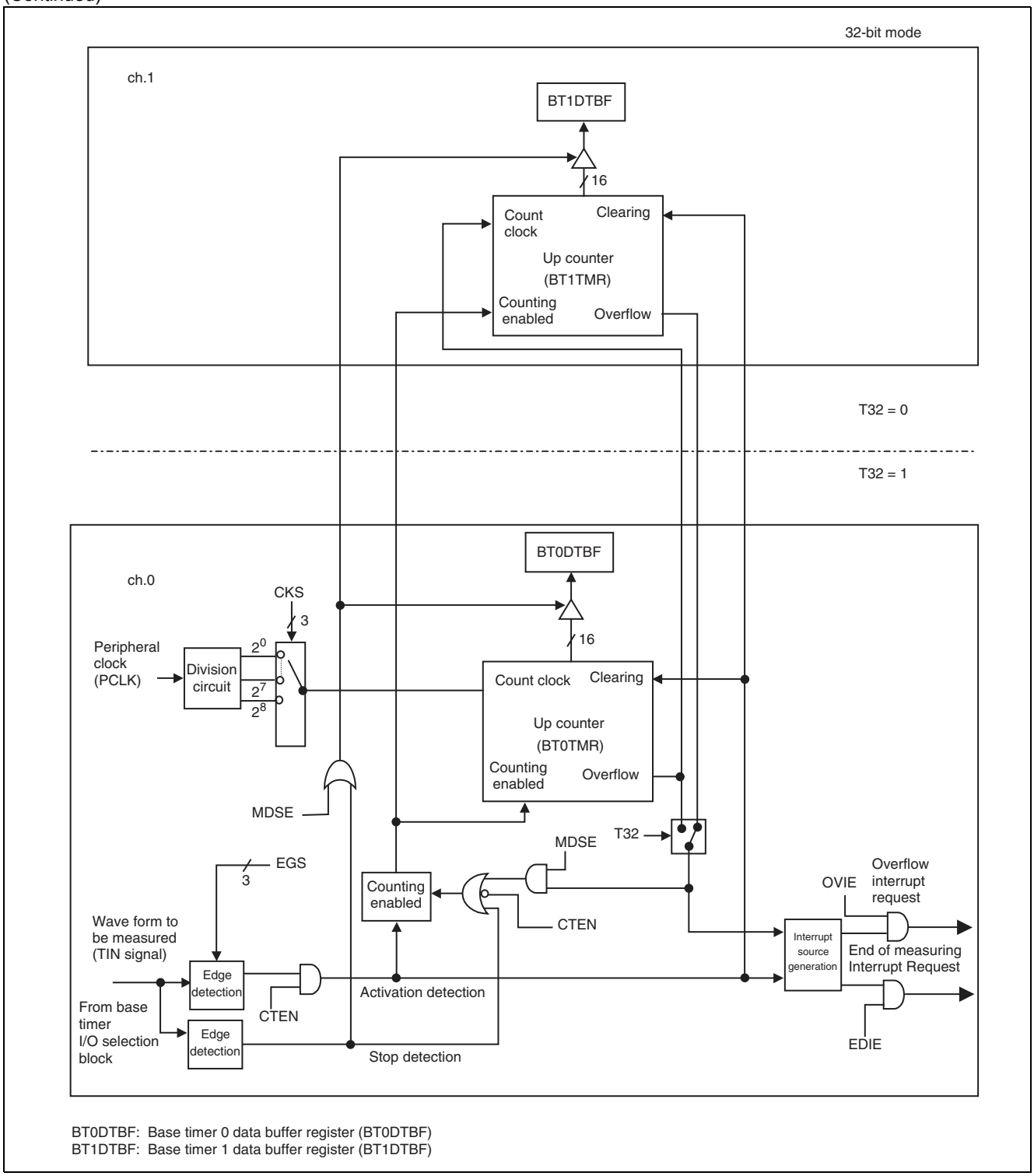
Figure 22-4. Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)



BTxDTBF: Base timer x data buffer register (BTxDTBF)

(Continued)

(Continued)



BT0DTBF: Base timer 0 data buffer register (BT0DTBF)
BT1DTBF: Base timer 1 data buffer register (BT1DTBF)

Notes:

- The PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, and between ch.6 and ch.7. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "21. Base Timer I/O Select Function".

22.3 Base Timer's Registers

This section lists the registers used for the base timer and their bit configurations in each timer function mode.

List of base timer's registers

Table 22-1. Registers used for 16-bit PWM Timer (Sheet 1 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.3
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	21.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	21.4.2
0	BT0TMCR	Base timer 0 timer control register	22.8.1.1
	BT0STC	Base timer 0 status control register	22.8.1.1
	BT0PCSR	Base timer 0 cycle setting register	22.8.1.2
	BT0PDUT	Base timer 0 duty setting register	22.8.1.3
	BT0TMR	Base timer 0 timer register	22.8.1.4
1	BT1TMCR	Base timer 1 timer control register	22.8.1.1
	BT1STC	Base timer 1 status control register	22.8.1.1
	BT1PCSR	Base timer 1 cycle setting register	22.8.1.2
	BT1PDUT	Base timer 1 duty setting register	22.8.1.3
	BT1TMR	Base timer 1 timer register	22.8.1.4
2	BT2TMCR	Base timer 2 timer control register	22.8.1.1
	BT2STC	Base timer 2 status control register	22.8.1.1
	BT2PCSR	Base timer 2 cycle setting register	22.8.1.2
	BT2PDUT	Base timer 2 duty setting register	22.8.1.3
	BT2TMR	Base timer 2 timer register	22.8.1.4
3	BT3TMCR	Base timer 3 timer control register	22.8.1.1
	BT3STC	Base timer 3 status control register	22.8.1.1
	BT3PCSR	Base timer 3 cycle setting register	22.8.1.2
	BT3PDUT	Base timer 3 duty setting register	22.8.1.3
	BT3TMR	Base timer 3 timer register	22.8.1.4
4	BT4TMCR	Base timer 4 timer control register	22.8.1.1
	BT4STC	Base timer 4 status control register	22.8.1.1
	BT4PCSR	Base timer 4 cycle setting register	22.8.1.2
	BT4PDUT	Base timer 4 duty setting register	22.8.1.3
	BT4TMR	Base timer 4 timer register	22.8.1.4
5	BT5TMCR	Base timer 5 timer control register	22.8.1.1
	BT5STC	Base timer 5 status control register	22.8.1.1
	BT5PCSR	Base timer 5 cycle setting register	22.8.1.2
	BT5PDUT	Base timer 5 duty setting register	22.8.1.3
	BT5TMR	Base timer 5 timer register	22.8.1.4
6	BT6TMCR	Base timer 6 timer control register	22.8.1.1
	BT6STC	Base timer 6 status control register	22.8.1.1
	BT6PCSR	Base timer 6 cycle setting register	22.8.1.2
	BT6PDUT	Base timer 6 duty setting register	22.8.1.3
	BT6TMR	Base timer 6 timer register	22.8.1.4

Table 22-1. Registers used for 16-bit PWM Timer (Sheet 2 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
7	BT7TMCR	Base timer 7 timer control register	22.8.1.1
	BT7STC	Base timer 7 status control register	22.8.1.1
	BT7PCSR	Base timer 7 cycle setting register	22.8.1.2
	BT7PDUT	Base timer 7 duty setting register	22.8.1.3
	BT7TMR	Base timer 7 timer register	22.8.1.4

Table 22-2. Registers for the 16-bit PPG Timer

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.3
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	21.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	21.4.2
0	BT0TMCR	Base timer 0 timer control register	22.8.2.1
	BT0STC	Base timer 0 status control register	22.8.2.1
	BT0PRLL	Base timer 0 L width setting register	22.8.2.2
	BT0PRLH	Base timer 0 H width setting register	22.8.2.3
	BT0TMR	Base timer 0 timer register	22.8.2.4
1	BT1TMCR	Base timer 1 timer control register	22.8.2.1
	BT1STC	Base timer 1 status control register	22.8.2.1
	BT1PRLL	Base timer 1 L width setting register	22.8.2.2
	BT1PRLH	Base timer 1 H width setting register	22.8.2.3
	BT1TMR	Base timer 1 timer register	22.8.2.4
2	BT2TMCR	Base timer 2 timer control register	22.8.2.1
	BT2STC	Base timer 2 status control register	22.8.2.1
	BT2PRLL	Base timer 2 L width setting register	22.8.2.2
	BT2PRLH	Base timer 2 H width setting register	22.8.2.3
	BT2TM	Base timer 2 timer register	22.8.2.4
3	BT3TMCR	Base timer 3 timer control register	22.8.2.1
	BT3STC	Base timer 3 status control register	22.8.2.1
	BT3PRLL	Base timer 3 L width setting register	22.8.2.2
	BT3PRLH	Base timer 3 H width setting register	22.8.2.3
	BT3TMR	Base timer 3 timer register	22.8.2.4
4	BT4TMCR	Base timer 4 timer control register	22.8.2.1
	BT4STC	Base timer 4 status control register	22.8.2.1
	BT4PRLL	Base timer 4 L width setting register	22.8.2.2
	BT4PRLH	Base timer 4 H width setting register	22.8.2.3
	BT4TMR	Base timer 4 timer register	22.8.2.4
5	BT5TMCR	Base timer 5 timer control register	22.8.2.1
	BT5STC	Base timer 5 status control register	22.8.2.1
	BT5PRLL	Base timer 5 L width setting register	22.8.2.2
	BT5PRLH	Base timer 5 H width setting register	22.8.2.3
	BT5TMR	Base timer 5 timer register	22.8.2.4
6	BT6TMCR	Base timer 6 timer control register	22.8.2.1
	BT6STC	Base timer 6 status control register	22.8.2.1
	BT6PRLL	Base timer 6 L width setting register	22.8.2.2
	BT6PRLH	Base timer 6 H width setting register	22.8.2.3
	BT6TMR	Base timer 6 timer register	22.8.2.4
7	BT7TMCR	Base timer 7 timer control register	22.8.2.1
	BT7STC	Base timer 7 status control register	22.8.2.1
	BT7PRLL	Base timer 7 L width setting register	22.8.2.2
	BT7PRLH	Base timer 7 H width setting register	22.8.2.3
	BT7TMR	Base timer 7 timer register	22.8.2.4

Table 22-3. Registers for the 16/32-bit Reload Timer

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.3
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	21.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	21.4.2
0	BT0TMCR	Base timer 0 timer control register	22.8.3.1
	BT0STC	Base timer 0 status control register	22.8.3.1
	BT0PCSR	Base timer 0 cycle setting register	22.8.3.2
	BT0TMR	Base timer 0 timer register	22.8.3.3
1	BT1TMCR	Base timer 1 timer control register	22.8.3.1
	BT1STC	Base timer 1 status control register	22.8.3.1
	BT1PCSR	Base timer 1 cycle setting register	22.8.3.2
	BT1TMR	Base timer 1 timer register	22.8.3.3
2	BT2TMCR	Base timer 2 timer control register	22.8.3.1
	BT2STC	Base timer 2 status control register	22.8.3.1
	BT2PCSR	Base timer 2 cycle setting register	22.8.3.2
	BT2TMR	Base timer 2 timer register	22.8.3.3
3	BT3TMCR	Base timer 3 timer control register	22.8.3.1
	BT3STC	Base timer 3 status control register	22.8.3.1
	BT3PCSR	Base timer 3 cycle setting register	22.8.3.2
	BT3TMR	Base timer 3 timer register	22.8.3.3
4	BT4TMCR	Base timer 4 timer control register	22.8.3.1
	BT4STC	Base timer 4 status control register	22.8.3.1
	BT4PCSR	Base timer 4 cycle setting register	22.8.3.2
	BT4TMR	Base timer 4 timer register	22.8.3.3
5	BT5TMCR	Base timer 5 timer control register	22.8.3.1
	BT5STC	Base timer 5 status control register	22.8.3.1
	BT5PCSR	Base timer 5 cycle setting register	22.8.3.2
	BT5TMR	Base timer 5 timer register	22.8.3.3
6	BT6TMCR	Base timer 6 timer control register	22.8.3.1
	BT6STC	Base timer 6 status control register	22.8.3.1
	BT6PCSR	Base timer 6 cycle setting register	22.8.3.2
	BT6TMR	Base timer 6 timer register	22.8.3.3
7	BT7TMCR	Base timer 7 timer control register	22.8.3.1
	BT7STC	Base timer 7 status control register	22.8.3.1
	BT7PCSR	Base timer 7 cycle setting register	22.8.3.2
	BT7TMR	Base timer 7 timer register	22.8.3.3

Table 22-4. List of Registers used for 16/32-bit PWC Timer

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.3
Common to 0 to 3	BTSEL0123	Base timer I/O select register for ch.0/1/2/3	21.4.1
Common to 4 to 7	BTSEL4567	Base timer I/O select register for ch.4/5/6/7	21.4.2
0	BT0TMCR	Base timer 0 timer control register	22.8.4.1
	BT0STC	Base timer 0 status control register	22.8.4.1
	BT0DTBF	Base timer 0 data buffer register	22.8.4.2
1	BT1TMCR	Base timer 1 timer control register	22.8.4.1
	BT1STC	Base timer 1 status control register	22.8.4.1
	BT1DTBF	Base timer 1 data buffer register	22.8.4.2
2	BT2TMCR	Base timer 2 timer control register	22.8.4.1
	BT2STC	Base timer 2 status control register	22.8.4.1
	BT2DTBF	Base timer 2 data buffer register	22.8.4.2
3	BT3TMCR	Base timer 3 timer control register	22.8.4.1
	BT3STC	Base timer 3 status control register	22.8.4.1
	BT3DTBF	Base timer 3 data buffer register	22.8.4.2
4	BT4TMCR	Base timer 4 timer control register	22.8.4.1
	BT4STC	Base timer 4 status control register	22.8.4.1
	BT4DTBF	Base timer 4 data buffer register	22.8.4.2
5	BT5TMCR	Base timer 5 timer control register	22.8.4.1
	BT5STC	Base timer 5 status control register	22.8.4.1
	BT5DTBF	Base timer 5 data buffer register	22.8.4.2
6	BT6TMCR	Base timer 6 timer control register	22.8.4.1
	BT6STC	Base timer 6 status control register	22.8.4.1
	BT6DTBF	Base timer 6 data buffer register	22.8.4.2
7	BT7TMCR	Base timer 7 timer control register	22.8.4.1
	BT7STC	Base timer 7 status control register	22.8.4.1
	BT7DTBF	Base timer 7 data buffer register	22.8.4.2

22.4 Operations of the Base Timer

This section introduces how the base timer operates in each timer function mode.

Operations of the base timer

- Reset mode

  Placing the base timer in this mode resets its macro (with each register reset to the initial value). Place the base timer in this mode once before changing its function mode or T32 bit setting. After a reset, however, the base timer can set its function mode and the T32 bit without entering the reset mode in advance. If you set this mode for even-numbered channels in 32-bit mode, odd-numbered channels are reset as well at the same time. Thus you do not have to set the reset mode for odd-numbered channels.

- 16-bit PWM timer

  The 16-bit PWM timer starts decrementing its counter by the value set as a period when triggered to start. The PWM timer then sets the output to the "L" level first and, if the 16-bit down counter value matches the value set in the duty setting register, inverts the output to the "H" level. Then it inverts the output back to the "L" level when the counter causes an underflow subsequently. This generates a waveform with an arbitrary period and duty cycle.

- 16-bit PPG timer

  The 16-bit PPG timer starts decrementing its counter by the value set in the "L"-width setting reload register when triggered to start. The PPG timer then sets the output to the "L" level first and inverts the output back to the "H" level when the counter causes an underflow. The PPG timer continuously decrements the counter by the value set in the "H"-width setting reload register and inverts the output level to "L" when the counter causes an underflow. This generates a waveform with arbitrary "L" and "H" widths.

- 16-bit reload timer

  The 16-bit reload timer starts decrementing its 16-bit down counter by the value set as a period when triggered to start. When the down counter causes an underflow, the interrupt flag is set. Depending on the MDSE bit setting, the output level either toggles, or is inverted, between "H" and "L" each time the counter causes an underflow or becomes "H" when the counter starts counting and "L" when it causes an underflow.

- 32-bit reload timer

  The 32-bit reload timer is the same in basic operation as the 16-bit reload timer, except that it works as a 32-bit version using a pair of even-numbered and odd-numbered channels. Although the even-numbered and odd-numbered channels then operate as the lower 16-bit and upper 16-bit timers, respectively, interrupt control and output wave control follow their respective settings for the even-numbered channel. To set the period, write the value to the upper register (odd-numbered channel) first and then to the lower register (even-numbered channel).

  To obtain the timer value, read the lower register (even-numbered channel) first and then the upper register (odd-numbered channel).

  Notes:

  - The reload timers can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, and between ch.6 and ch.7. No 32-bit operation is applicable to any other combination of channels.
  - This function supports simultaneous activation. For details, see "21. Base Timer I/O Select Function".

- 16-bit PWC timer

  The 16-bit PWC timer starts the 16-bit up counter upon input of a pre-set measurement start edge and stops the counter upon detection of a measurement stop edge. The count value between the two edges is written to the data buffer register as a pulse width.

- 32-bit PWC timer

  The 32-bit PWC timer is the same in basic operation as the 16-bit PWC timer, except that it works as a 32-bit version using a pair of even-numbered and odd-numbered channels. Although the even-numbered and odd-numbered channels then operate as the lower 16-bit and upper 16-bit counters, respectively, interrupt control follows the setting for the even-numbered channel. To obtain the measured value or count value, read the lower register (even-numbered channel) first and then the upper register (odd-numbered channel).

  Notes:

  - The PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, and between ch.6 and ch.7. No 32-bit operation is applicable to any other combination of channels.
  - This function supports simultaneous activation. For details, see "21. Base Timer I/O Select Function".

22.5 32-bit Mode Operations

The reload timer and PWC timer can operate in 32-bit mode using a pair of channels. This section describes the basic functions and operations of 32-bit mode.

Functions of 32-bit mode

The 32-bit mode combines two channels of base timer into a 32-bit data reload timer or PWC timer. Either 32-bit timer allows the timer/counter value to be read even during operation as it takes the upper 16-bit timer/counter value of the odd-numbered channel also when reading the lower 16-bit timer/counter value of the even-numbered channel.

Setting the 32-bit mode

First, set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register for the even-numbered channel to "000_B" to reset in reset mode. Then, select the reload timer or PWC timer and set its operations in the same way as in 16-bit mode. At this time, write "1" to the T32 bit in the BTxTMCR register to enter the 32-bit operation mode. The T32 bit for the odd-numbered channel must be left containing "0". Neither the reset mode setting is required for the odd-numbered channel. To use the base timer as the reload timer, set the period setting register for the odd-numbered channel to the upper 16-bit reload value among 32 bits and set the period setting register for the even-numbered channel to the lower 16-bit reload value.

As the transition to 32-bit operation mode takes place the moment is written to the T32 bit, the setting must be changed with counting halted on both of the channels.

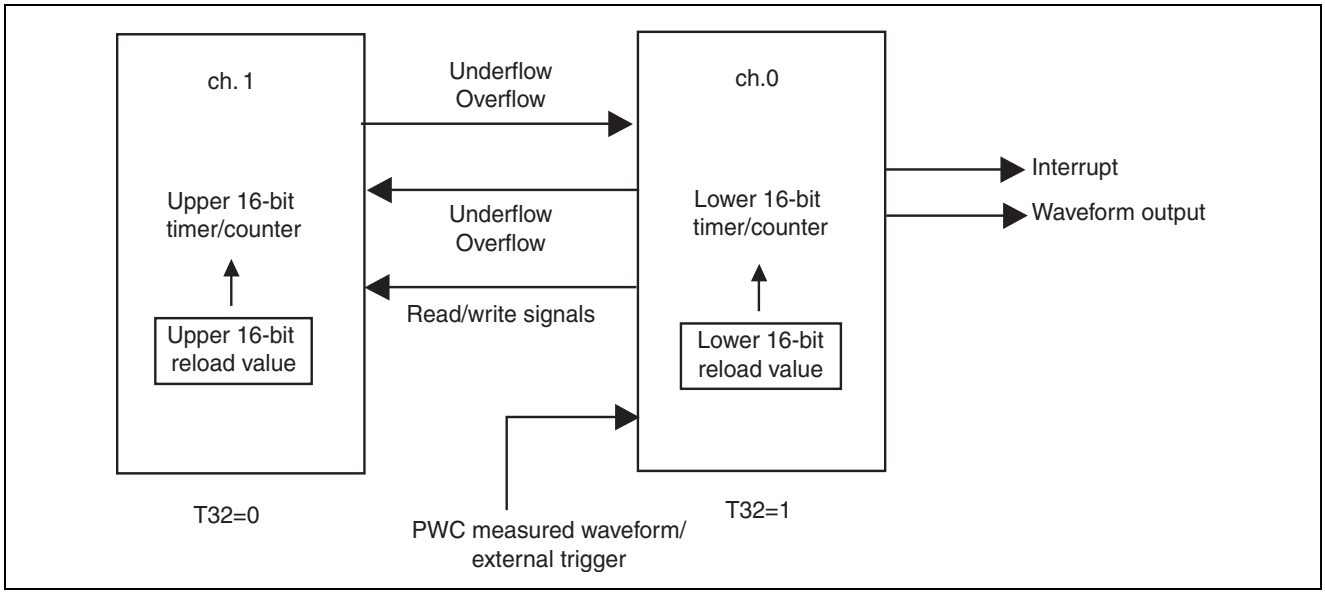
To switch from 32-bit mode to 16-bit mode, set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register for the even-numbered channel to "000_B" to reset the states of both of the even-numbered and odd-numbered channels in reset mode. Then set each channel for operation in 16-bit mode.

Operations in 32-bit mode

When the reload timer or PWC timer is started in 32-bit mode under control of the even-numbered channel, the timer/counter of the even-numbered channel operates as the lower 16-bit timer/counter and the timer/counter of the odd-numbered channel operates as the upper 16-bit one.

In 32-bit mode, the base timer follows the settings for the even-numbered channel while ignoring those for the odd-numbered channel (except the period setting register when serving as the reload timer). Even for the timer start, waveform output, and interrupt signal settings, the even-numbered channel overrides the odd-numbered channel (odd-numbered channel is always masked at "L").

The following example shows a PWC configuration using ch.0 and ch.1.



Notes:

- The reload timer or PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, and between ch.6 and ch.7. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "21. Base Timer I/O Select Function".

22.6 Notes of Using the Base Timer

This section summarizes the notes on using the base timer.

Common notes on using each type of timer

- Notes on setting through programming
  - The following bits in the BTxTMCR register must not be updated during operation. Be sure to update them before starting the base timer or after stopping it.

    [bit14, bit13, bit12] CKS2, CKS1, CKS0 : Clock select bits
    [bit10, bit9, bit8] EGS2, EGS1, EGS0 : Measurement edge select bits
    [bit7] T32 : 32-bit timer select bit
    (Used with the reload timer or PWC timer selected)
    [bit6, bit5,bit4] FMD2, FMD1, FMD0 : Timer function mode select bits
    [bit2] MDSE : Measurement mode (one-shot/continuous) select bit
  - If you set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register to "000_B" to enter the reset mode, all the registers of the base timer are initialized and thus they must be set all over again.
  - If you set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register to "000_B" to enter the reset mode, the other bits in the BTxTMCR register are initialized with their settings ignored.

Notes on using the 16-bit PWM/PPG/reload timer

- Notes on setting through programming
  - When the interrupt request flag is attempted to be set and cleared at the same timing, the flag set action overrides the flag clear action.
  - When the down counter is attempted to load and count at the same timing, the load action overrides the count action.
  - Set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register to select the timer function mode before setting the period, duty cycle, "H" width, and "L" width.
  - If a restart is detected when counting is completed in one-shot mode, the counter is restarted with the count value reloaded.

Notes on using the PWC timer

- Notes on setting through programming
  - Writing "1" to the counting enable bit (CTEN) clears the counter, nullifying the data existing in the counter before counting is enabled.
  - If you set the PWC mode (FMD = 100_B) after a system reset or in reset mode and enables measurement (CTEN = 1) at the same time, the timer may operate according to the immediately preceding measurement signal.
  - If a measurement start edge is detected the moment a restart is set in continuous measurement mode, the timer immediately starts counting from "0001_H".
  - An attempt to restart the timer after starting counting can result as follows, depending on that timing:
  - If the attempt is made at a measurement end edge in one-shot pulse width measurement mode:
    Although the timer is restarted and waits for an measurement start edge, the measurement end flag (EDIR) is set.
  - If the attempt is made at a measurement end edge in continuous pulse width measurement mode:
    Although the timer is restarted and waits for a measurement start edge, the measurement end flag (EDIR) is set and the current measurement result is transferred to the BTxDTBF register.

  When restarting the timer during operation, control interrupts while paying attention to the behaviors of flags.

22.7 Base Timer Interrupts

This section lists the interrupt request flags, interrupt enable bits, and interrupt factors for the base timer in each timer function mode.

Interrupt control bits and interrupt factors by timer function mode

Table 22-5 lists the interrupt control bits and interrupt factors for the base timer in each timer function mode.

Table 22-5. Interrupt Control Bits and Interrupt Factors in Each Timer Function Mode

	Status control register (BTxSTC)			
	Interrupt request flag bits	**Interrupt request enable bits**	**Interrupt factors**	**IRQ**
PWM timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	DTIR: bit1	DTIE: bit5	Duty match detection	
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
PPG timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
Reload timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
PWC timer function	OVIR: bit0	OVIE: bit4	Overflow detection	IRQ0
	EDIR: bit2	EDIE: bit6	Measurement end detection	IRQ1

22.8 Base Timer Description by Function Mode

This section describes each function of the base timer.

Base timer function

- PWM function
- PPG function
- Reload timer function
- PWC function

22.8.1 PWM Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PWM timer.

- Timer Control Register (BTxTMCR) for PWM Timer
- PWM Period Setting Register (BTxPCSR)
- PWM Duty Setting Register (BTxPDUT)
- Timer Register (BTxTMR)
- 16-bit PWM Timer Operation
- One-shot Operation
- Interrupt Factors and Timing Chart
- Output Waveforms

22.8.1.1 Timer Control Register (BTxTMCR) for PWM Timer

The timer control register (BTxTMCR) controls the PWM timer. Keep in mind that the register contains bits which cannot be updated with the PWM timer operating.

Timer control register (BTxTMCR upper byte)

Figure 22-5. Timer Control Register (BTxTMCR Upper Byte)

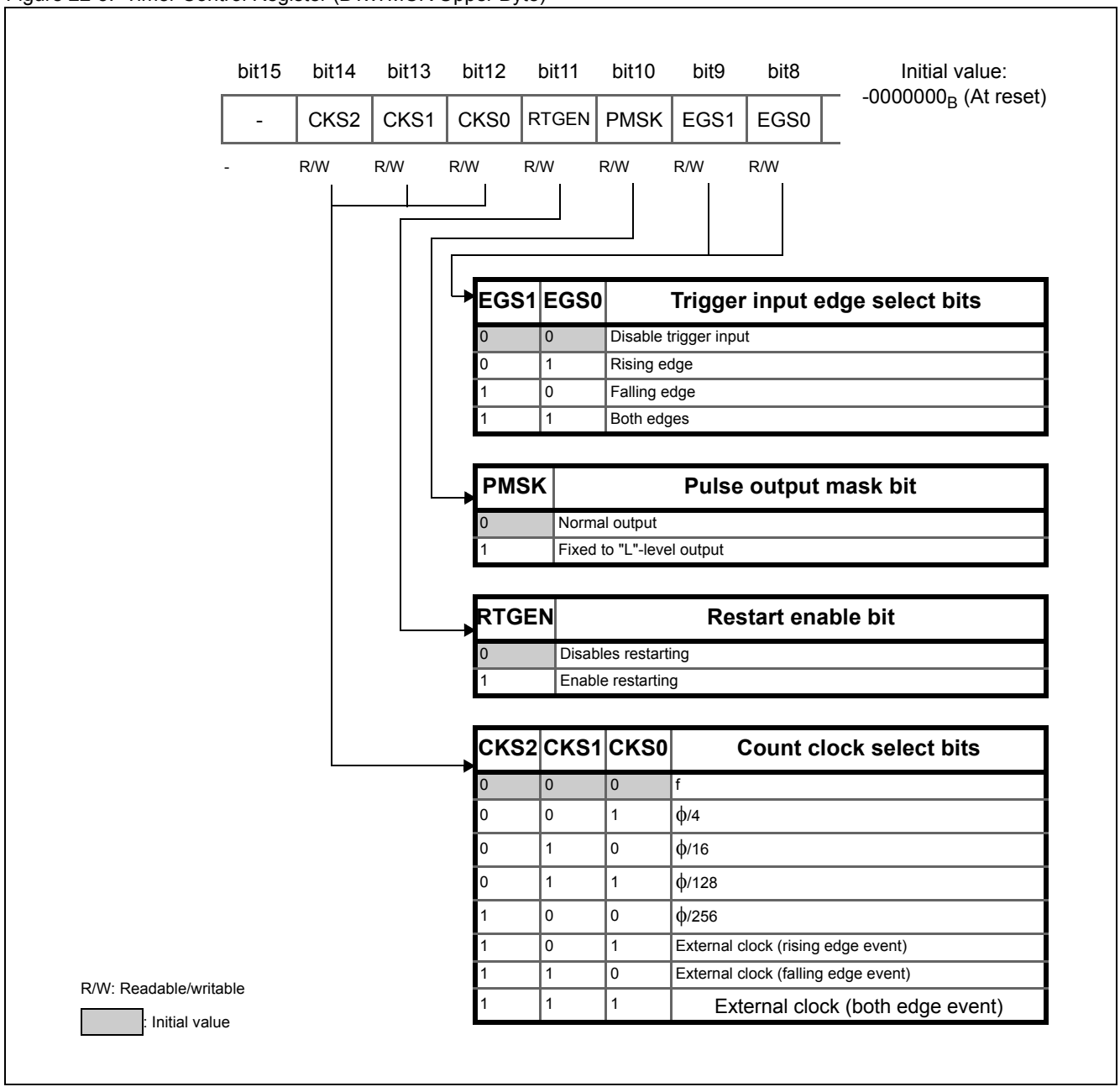


Table 22-6. Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	■ The read value of this bit is undefined. ■ Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	■ Select the count clock for the 16-bit down counter. ■ The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	RTGEN: Restart enable bit	Enables restarting with a software trigger or trigger input.
bit10	PMSK: Pulse output mask bit	■ Controls the PWM output waveform level. ■ When this bit is "0", the PWM waveform is output as it is. ■ When the bit is "1", the PWM output is masked to the "L" level irrespective of the period and duty cycle. **Note:** Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PWM output to the "H" level.
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	■ Select the effective edge of the input waveform as an external trigger to set the trigger condition. ■ When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. **Note:** Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0. ■ EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

Timer control register (BTxTMCR lower byte)

Figure 22-6. Timer Control Register (BTxTMCR Lower Byte)

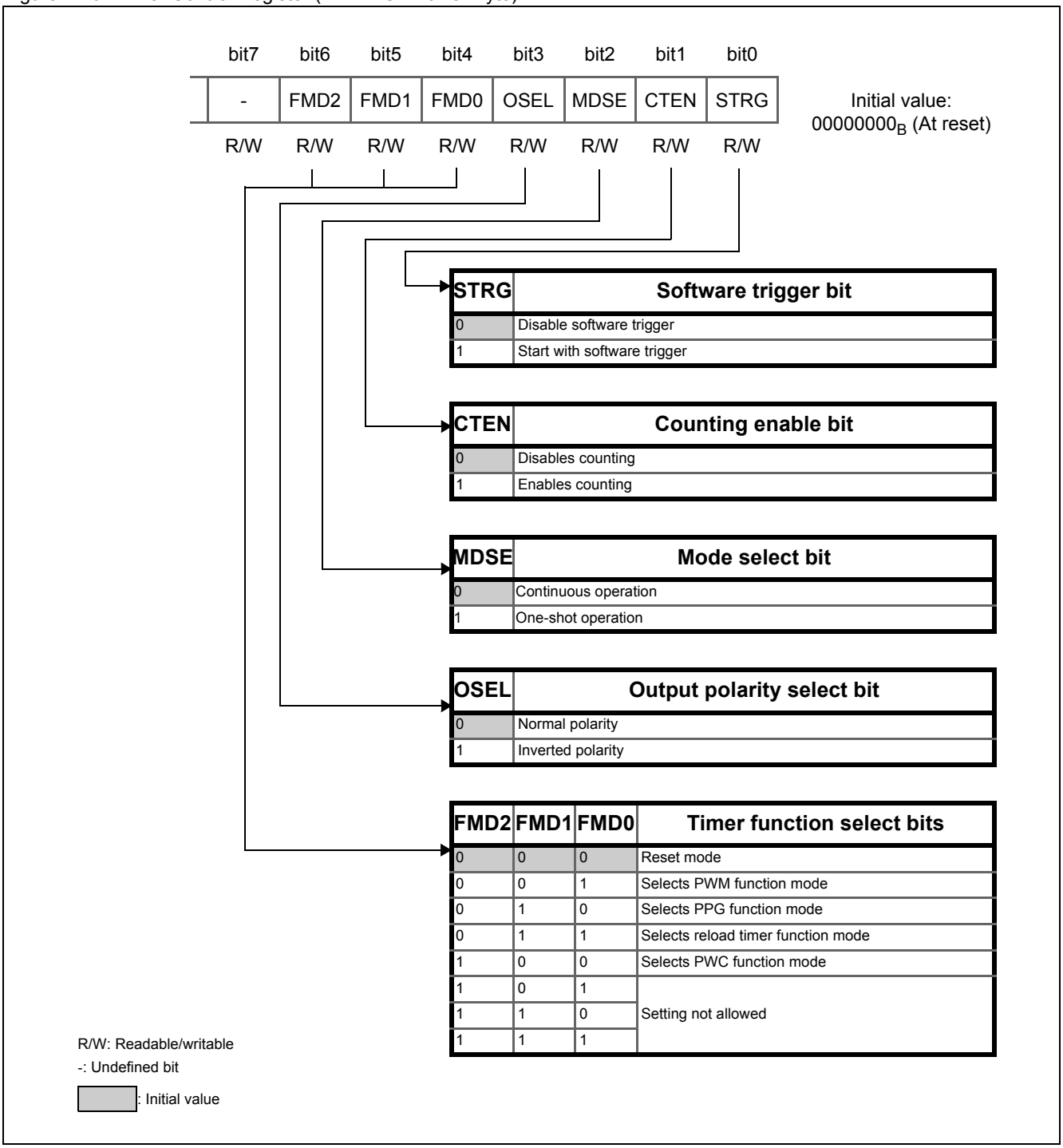


Table 22-7. Timer Control Register (BTxTMCR Lower Byte)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>Undefined bit</td><td>■ The value read is "0"
■ When writing to this bit, write "0".</td></tr>
<tr><td>bit6
to
bit4</td><td>FMD2, FMD1, FMD0:
Timer function select bits</td><td>■ These bits select the timer function mode.
■ Setting the FMD2, FMD1, and FMD0 bits to "001_B" selects the PWM function mode.
■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit3</td><td>OSEL:
Output polarity select bit</td><td>Selects the polarity of PWM output.
<table>
<tr><th>Polarity</th><th>After reset</th><th>Duty match</th><th>Underflow</th></tr>
<tr><td>Normal</td><td>"L" output</td><td></td><td></td></tr>
<tr><td>Inverted</td><td>"H" output</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>bit2</td><td>MDSE:
Mode select bit</td><td>■ Selects continuous pulse output or one-shot pulse output.
■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit1</td><td>CTEN:
Counting enable bit</td><td>■ This bit enables the down counter.
■ Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.</td></tr>
<tr><td>bit0</td><td>STRG:
Software trigger bit</td><td>■ Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger.
Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.
■ The value read from the STRG bit is always "0".
Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</td></tr>
</table>

Status control register (BTxSTC)

Figure 22-7. Status Control Register (BTxSTC)

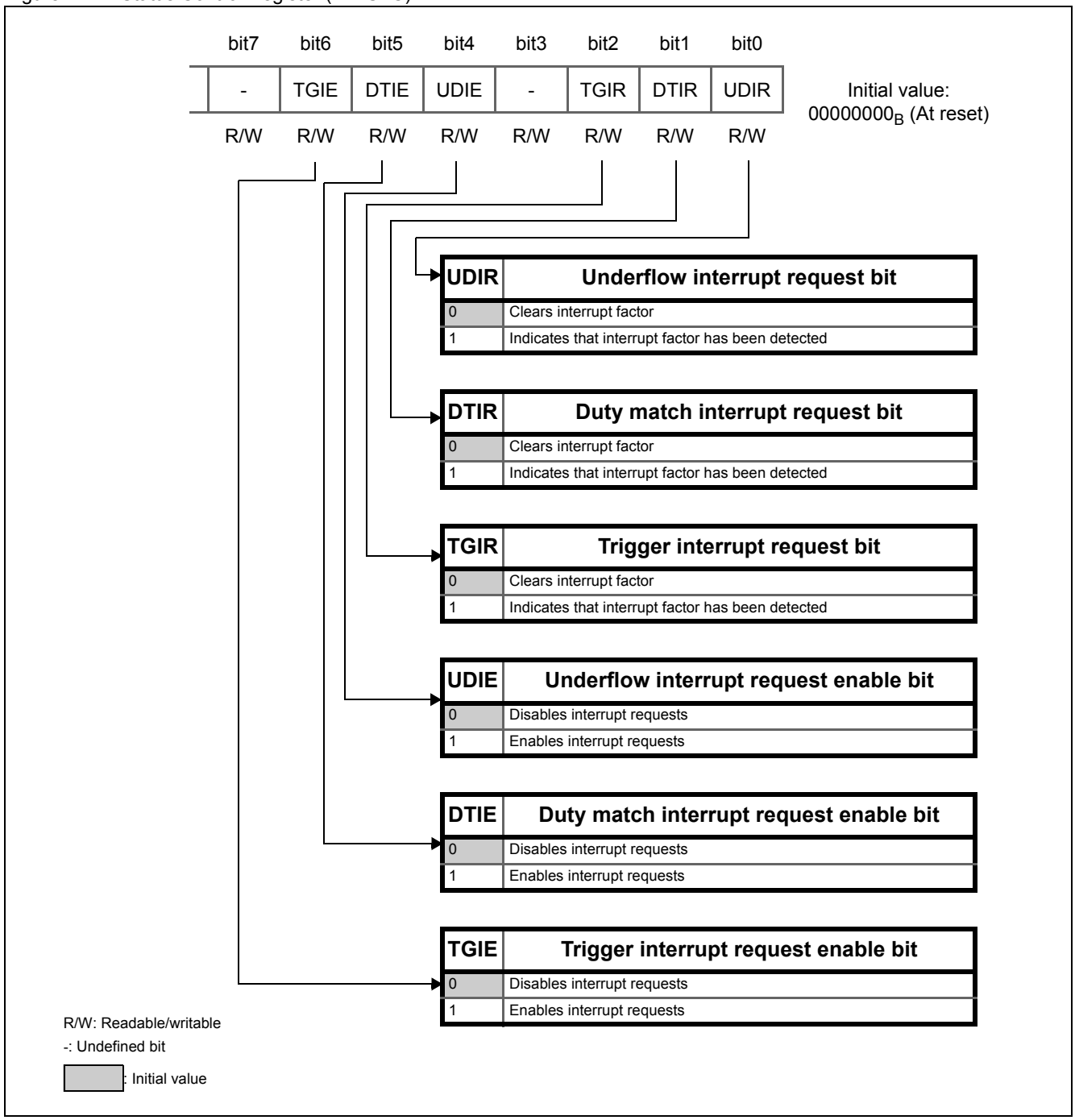


Table 22-8. Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	■ Controls bit2: TGIR interrupt requests. ■ Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	DTIE: Duty match interrupt request enable bit	■ Controls bit1: DTIR interrupt requests. ■ Setting the DTIR bit (bit1) with the DTIE bit enabling duty match interrupt requests generates an interrupt request to the CPU.
bit4	UDIE: Underflow interrupt request enable bit	■ Controls bit0: UDIR interrupt requests. ■ Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	■ The TGIR bit is set to "1" upon detection of a software trigger or trigger input. ■ Writing "0" to the TGIR bit clears it. ■ Writing "1" to the TGIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	DTIR: Duty match interrupt request bit	■ The DTIR bit is set to "1" when the count value matches the duty cycle setting. ■ Writing "0" to the DTIR bit clears it. ■ Writing "1" to the DTIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit0	UDIR: Underflow interrupt request bit	■ The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to $FFFF_H$. ■ Writing "0" to the UDIR bit clears it. ■ Writing "1" to the UDIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

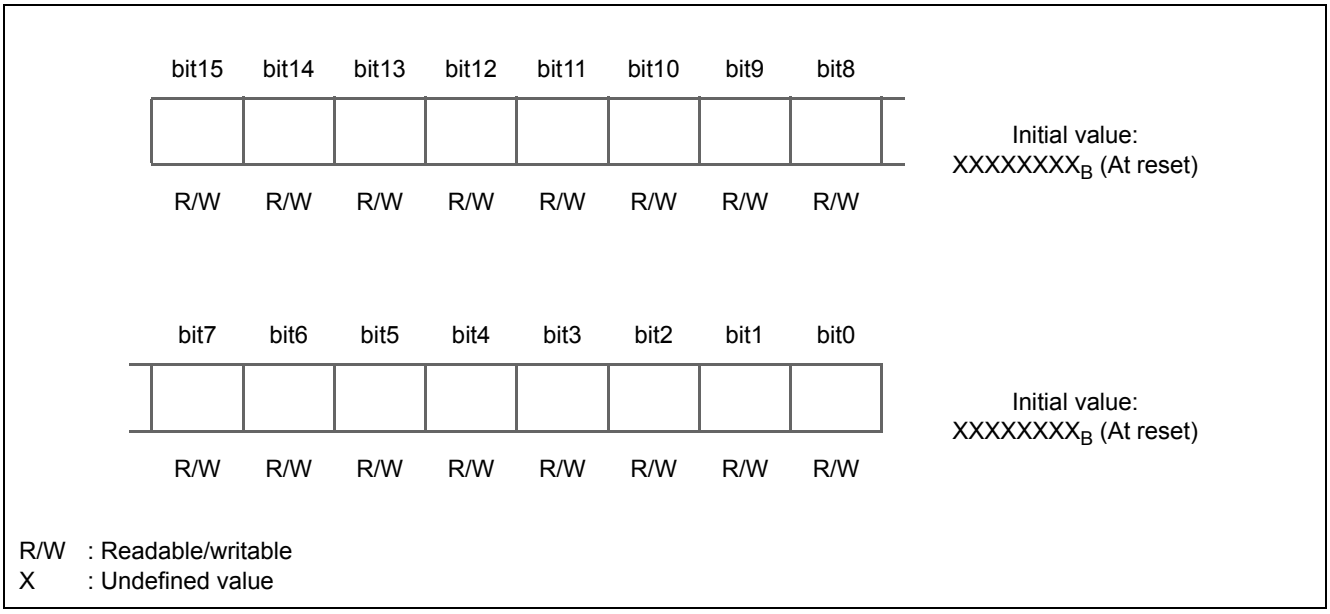
22.8.1.2 PWM Period Setting Register (BTxPCSR)

The PWM period setting register (BTxPCSR) is a buffered register for setting the PWM period. Transfer to the timer register takes place when the counter is started and when it causes an underflow.

Bit configuration of the PWM period setting register (BTxPCSR)

Figure 22-8 shows the bit configuration of the PWM period setting register (BTxPCSR).

Figure 22-8. Bit Configuration of the PWM Period Setting Register (BTxPCSR)



The BTxPCSR register is a buffered register for setting the PWM period. Transfer to the timer register takes place when the counter is started and when it causes an underflow.

After writing to the period setting register to initially set or update it, be sure to write to the duty setting register.

- Access the BTxPCSR register using 16-bit data.
- Set the PWM period using the BTxPCSR register after selecting the PWM function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

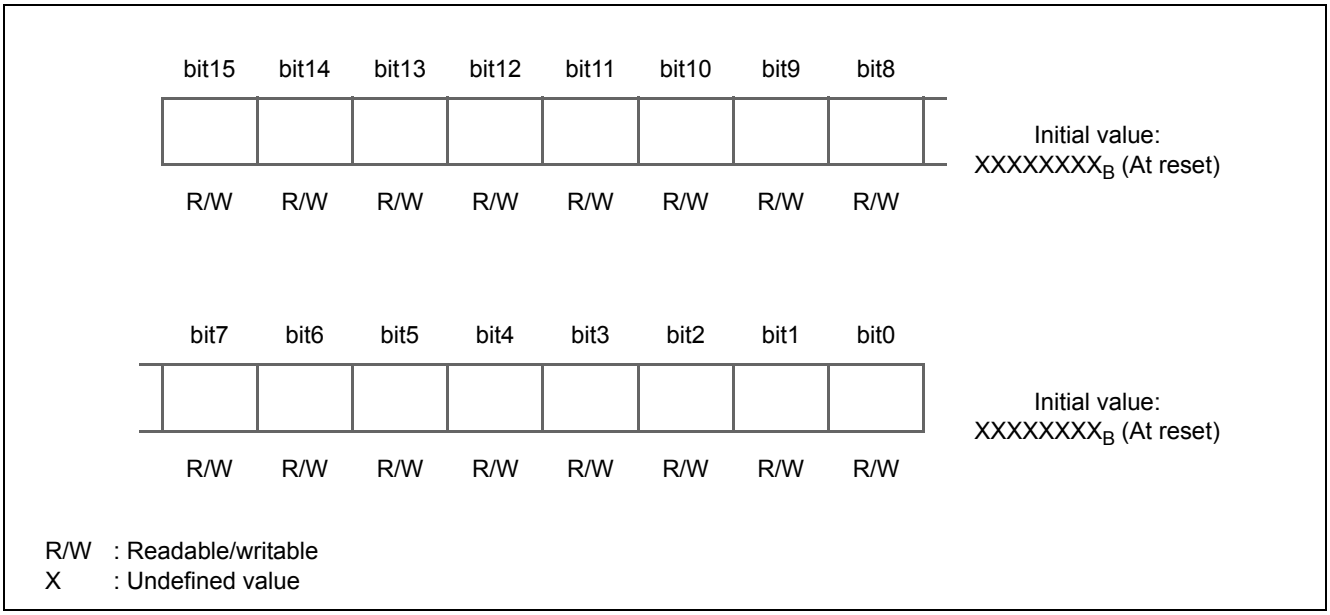
22.8.1.3 PWM Duty Setting Register (BTxPDUT)

The PWM duty setting register (BTxPDUT) is a buffered register for setting the PWM duty cycle. Transfer from the buffer takes place when an underflow occurs.

Bit configuration of the PWM duty setting register (BTxPDUT)

Figure 22-9 shows the bit configuration of the PWM duty setting register (BTxPDUT).

Figure 22-9. Bit Configuration of the PWM Duty Setting Register (BTxPDUT)



The BTxPDUT register is a buffered register for setting the PWM duty cycle. Transfer from the buffer takes place when an underflow occurs.

If you set the period setting and duty setting registers to the same value, the output level is all "H" in normal polarity or all "L" in inverted polarity.

Do not set the BTxPDUT register to a value greater than the value of the BTxPSCR register, or PWM output will be undefined.

- Access the BTxPDUT register using 16-bit data.
- Set the PWM duty cycle using the BTxPDUT register after selecting the PWM function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

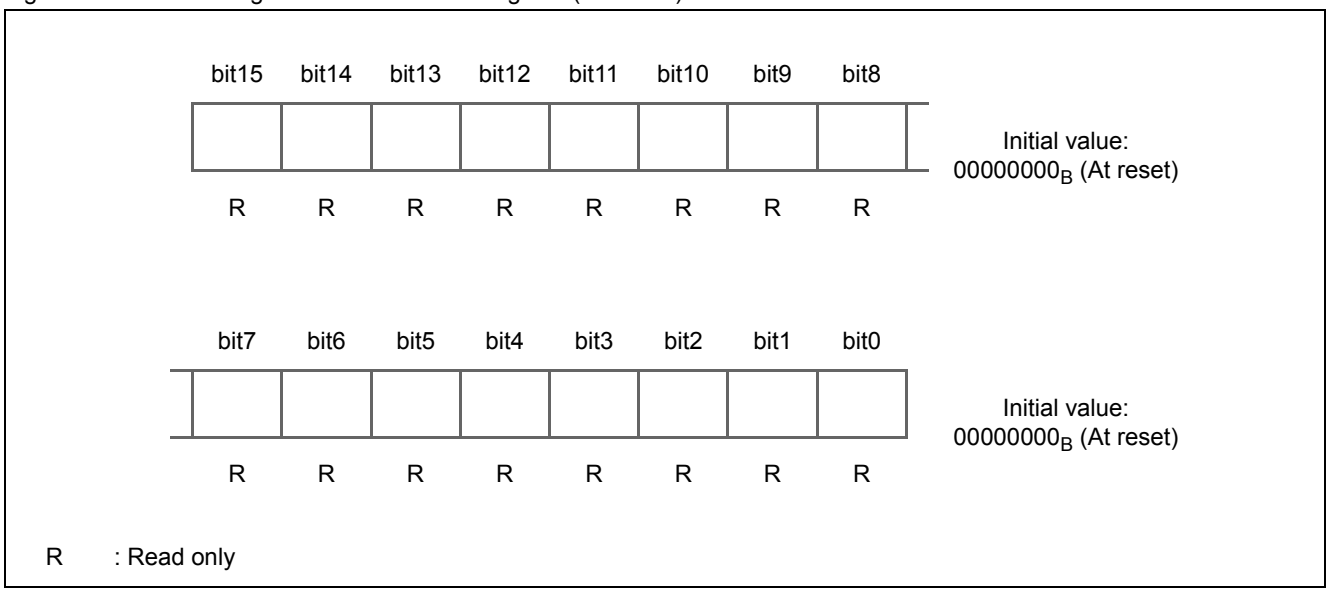
22.8.1.4 Timer Register (BTxTMR)

The timer register (BTxTMR) allows the value of the 16-bit down counter to be read from.

Bit configuration of the timer register (BTxTMR)

Figure 22-10 shows the bit configuration of the PWM timer register (BTxTMR).

Figure 22-10. Bit Configuration of the Timer Register (BTxTMR)



The BTxTMR register allows the value of the 16-bit down counter to be read from.

Note: Access the BTxTMR register using 16-bit data.

22.8.1.5 16-bit PWM Timer Operation

In PWM timer mode, a waveform having a specified period can be output either in single shots or continuously after detection of a trigger.

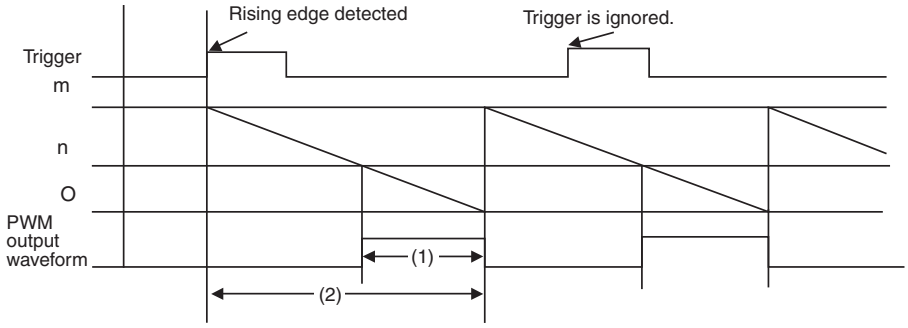
The period of output pulses can be controlled by changing the BTxPCSR value.

The duty ratio can be controlled by changing the BTxPDUT value. After writing data to the BTxPCSR register, be sure to write to the BTxPDUT register as well.

Continuous operation

- When restarting is disabled (RTGEN = 0)

Figure 22-11. PWM Operation Timing Chart (Restarting Disabled)

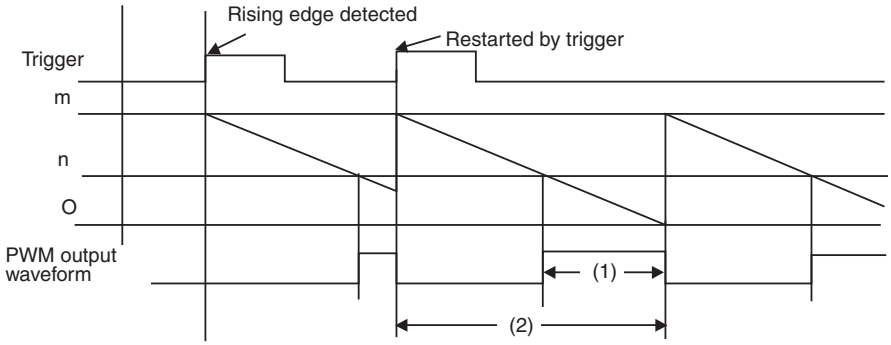


(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

- When restarting is enabled (RTGEN = 1)

Figure 22-12. PWM Operation Timing Chart (Restarting Enabled)



(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

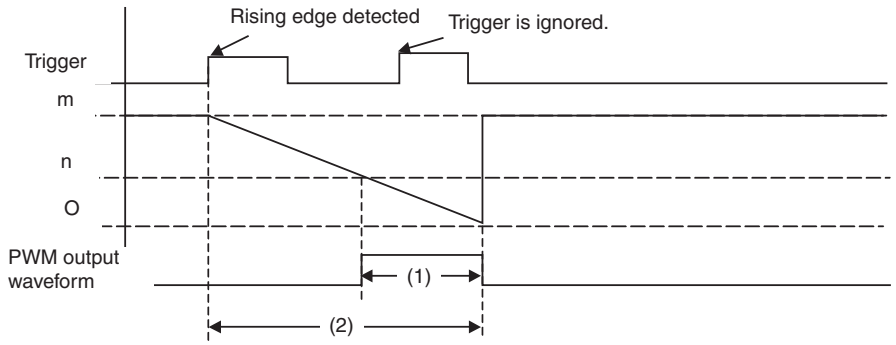
22.8.1.6 One-shot Operation

In one-shot operation mode, single pulses with an arbitrary width can be output by trigger. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

One-shot operation

- When restarting is disabled (RTGEN = 0)

Figure 22-13. One-shot Operation Timing Chart (Trigger Restarting Disabled)

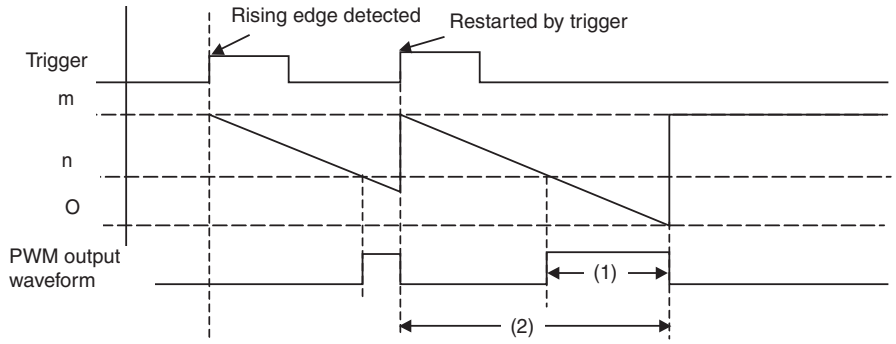


(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

- When restarting is enabled (RTGEN = 1)

Figure 22-14. One-shot Operation Timing Chart (Trigger Restarting Enabled)



(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

22.8.1.7 Interrupt Factors and Timing Chart

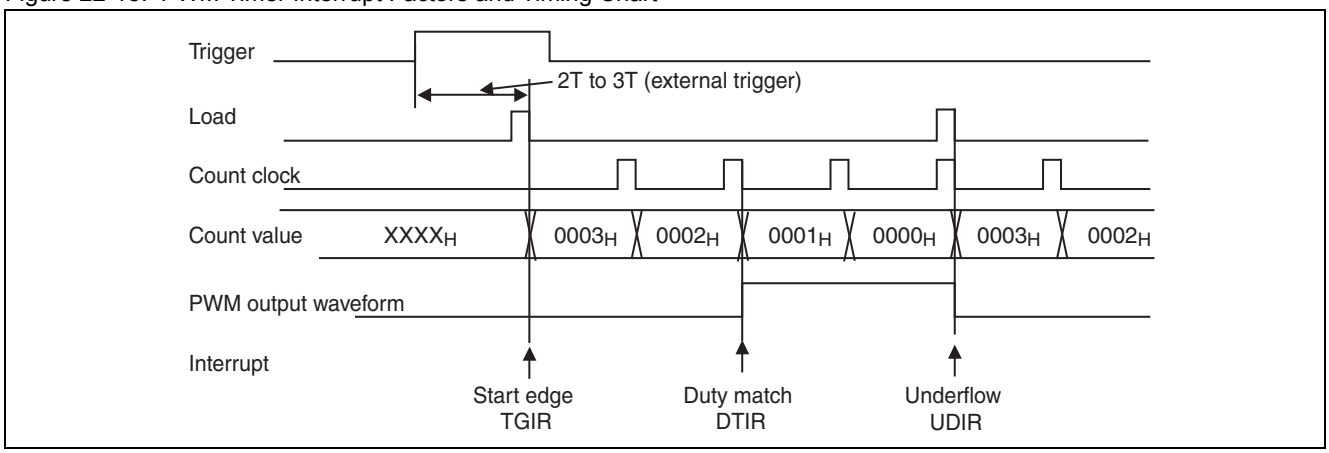
This section provides the interrupt factors and timing chart.

Interrupt factors and timing chart (PWM output: Normal polarity)

A software trigger requires T and an external trigger requires 2T to 3T (T: peripheral clock (PCLK) cycle) until the counter value is loaded after the input of the trigger.

Figure 22-15 shows the interrupt factors and timing chart, assuming "period setting" = 3 and "duty value" = 1.

Figure 22-15. PWM Timer Interrupt Factors and Timing Chart



22.8.1.8 Output Waveforms

This section illustrates PWM output.

PWM output at all "L" or all "H" level

Figure 22-16 and Figure 22-17. illustrate how to provide PWM output at all "L" and all "H" levels, respectively.

Figure 22-16. Example of PWM Output at All "L" Level

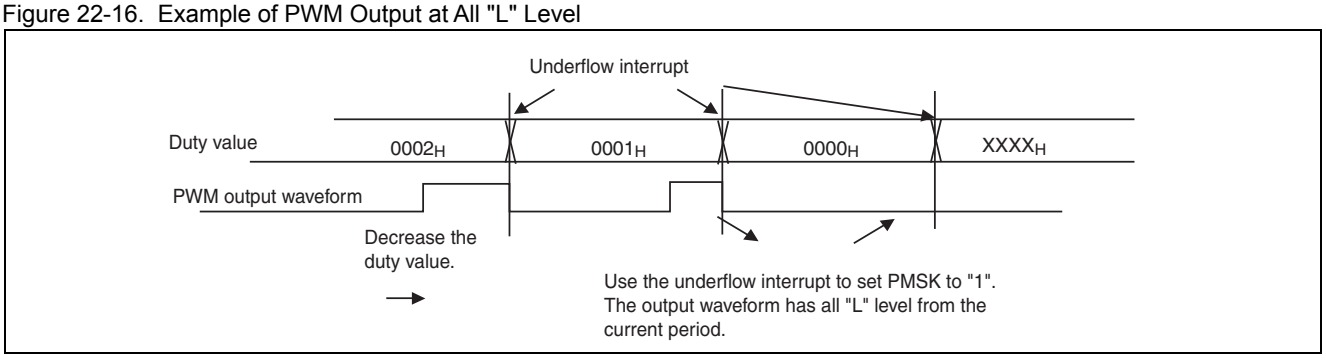
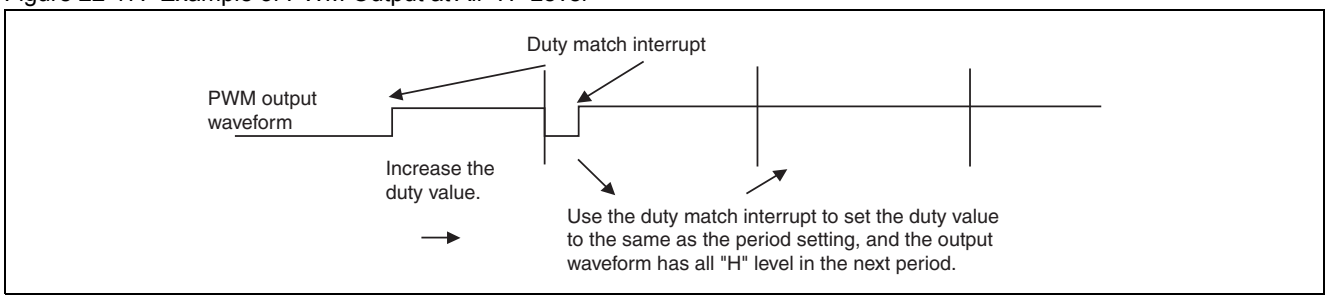


Figure 22-17. Example of PWM Output at All "H" Level



22.8.2 PPG Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PPG timer.

- Timer Control Register (BTxTMCR) for PPG Timer
- "L"-width Setting Reload Register (BTxPRLL)
- "H"-width Setting Reload Register (BTxPRLH)
- Timer Register (BTxTMR)
- 16-bit PPG Timer Operation
- Continuous Operation
- One-shot Operation
- Interrupt Factors and Timing Chart

22.8.2.1 Timer Control Register (BTxTMCR) for PPG Timer

The timer control register (BTxTMCR) controls the PPG timer. Keep in mind that the register contains bits which cannot be updated with the PPG timer operating.

Timer control register (BTxTMCR upper byte)

Figure 22-18. Timer Control Register (BTxTMCR Upper Byte)

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	CKS2	CKS1	CKS0	RTGEN	PMSK	EGS1	EGS0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: -0000000_B (At reset)

EGS1	EGS0	Trigger input edge select bits
0	0	Disable trigger input
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

PMSK	Pulse output mask bit
0	Normal output
1	Fixed to "L"-level output

RTGEN	Restart enable bit
0	Disables restarting
1	Enable restarting

CKS2	CKS1	CKS0	Count clock select bits
0	0	0	f
0	0	1	ϕ/4
0	1	0	ϕ/16
0	1	1	ϕ/128
1	0	0	ϕ/256
1	0	1	External clock (rising edge event)
1	1	0	External clock (falling edge event)
1	1	1	External clock (both edge event)

R/W: Readable/writable

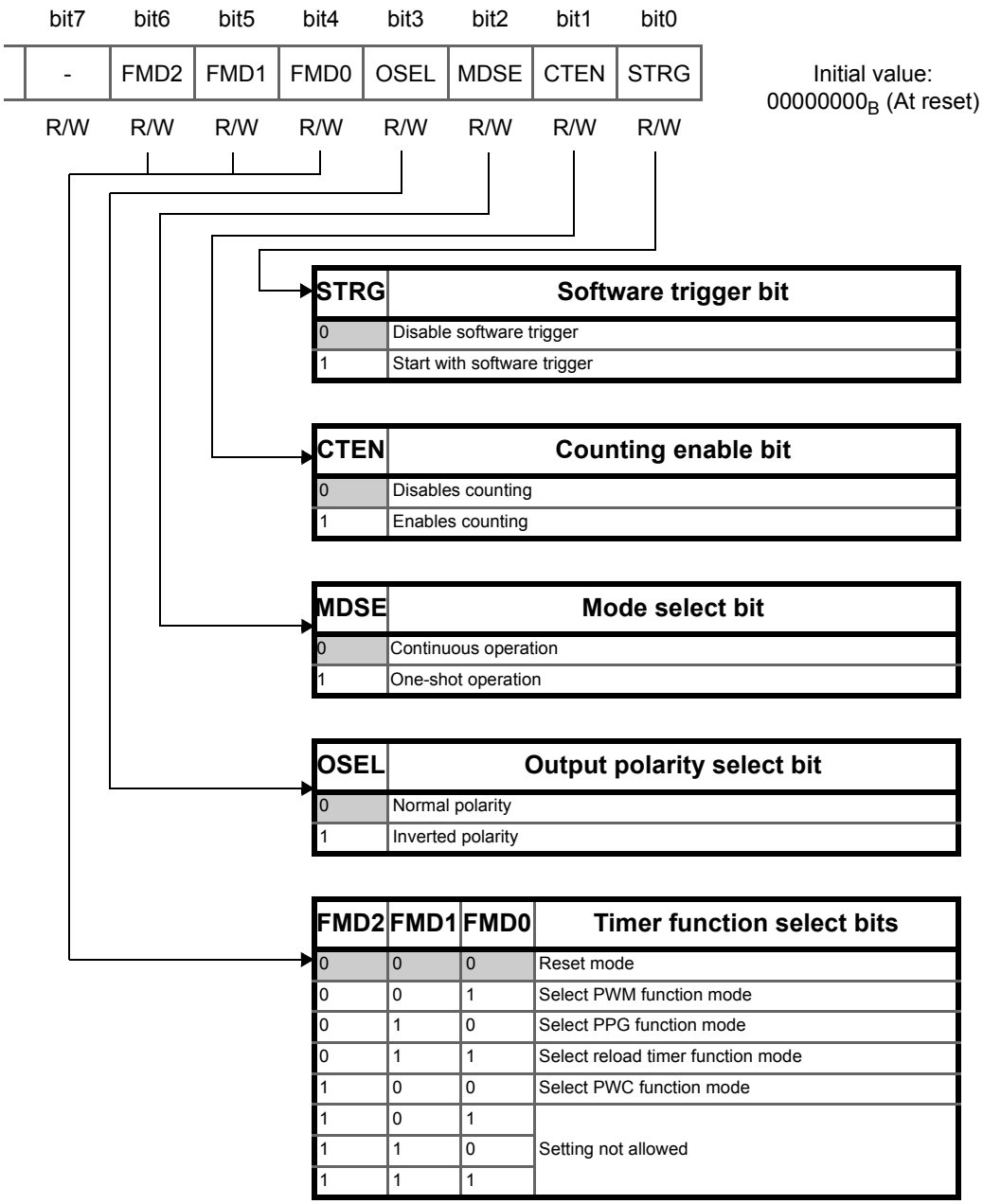
: Initial value

Table 22-9. Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	■ The read value of this bit is undefined. ■ Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	■ Select the count clock for the 16-bit down counter. ■ The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	RTGEN: Restart enable bit	This bit enables restarting with a software trigger or trigger input.
bit10	PMSK: Pulse output mask bit	■ Controls the PPG output waveform level. ■ When this bit is "0", the PPG waveform is output as it is. ■ When the bit is "1", the PPG output is masked to the "L" level irrespective of the "H" and "L" width settings. **Note:** Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PPG output to the "H" level.
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	■ Select the effective edge of the input waveform as an external trigger to set the trigger condition. ■ When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. **Note:** Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0. ■ EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

Timer control register (BTxTMCR lower byte)

Figure 22-19. Timer Control Register (BTxTMCR Lower Byte)



STRG	Software trigger bit
0	Disable software trigger
1	Start with software trigger

CTEN	Counting enable bit
0	Disables counting
1	Enables counting

MDSE	Mode select bit
0	Continuous operation
1	One-shot operation

OSEL	Output polarity select bit
0	Normal polarity
1	Inverted polarity

FMD2	FMD1	FMD0	Timer function select bits
0	0	0	Reset mode
0	0	1	Select PWM function mode
0	1	0	Select PPG function mode
0	1	1	Select reload timer function mode
1	0	0	Select PWC function mode
1	0	1	Setting not allowed
1	1	0	
1	1	1	

R/W: Readable/writable

-: Undefined bit

: Initial value

Table 22-10. Timer Control Register (BTxTMCR Lower Byte)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>Undefined bit</td><td>■ The value read is "0"
■ When writing to this bit, write "0".</td></tr>
<tr><td>bit6
to
bit4</td><td>FMD2, FMD1,
FMD0:
Timer function select bits</td><td>■ These bits select the timer function mode.
■ Setting the FMD2, FMD1, and FMD0 bits to "010_B" selects the PPG function mode.
■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit3</td><td>OSEL:
Output polarity select bit</td><td>■ Selects the polarity of PPG output.
<table>
<tr><th>Polarity</th><th>After reset</th><th>End of "L"-width counting</th><th>End of "H"-width counting</th></tr>
<tr><td>Normal</td><td>"L" output</td><td></td><td></td></tr>
<tr><td>Inverted</td><td>"H" output</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>bit2</td><td>MDSE:
Mode select bit</td><td>■ Selects continuous pulse output or one-shot pulse output.
■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit1</td><td>CTEN:
Counting enable bit</td><td>■ This bit enables the down counter.
■ Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.</td></tr>
<tr><td>bit0</td><td>STRG:
Software trigger bit</td><td>■ Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger.
Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.
■ The value read from the STRG bit is always "0".
Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</td></tr>
</table>

Status control register (BTxSTC)

Figure 22-20. Status Control Register (BTxSTC)

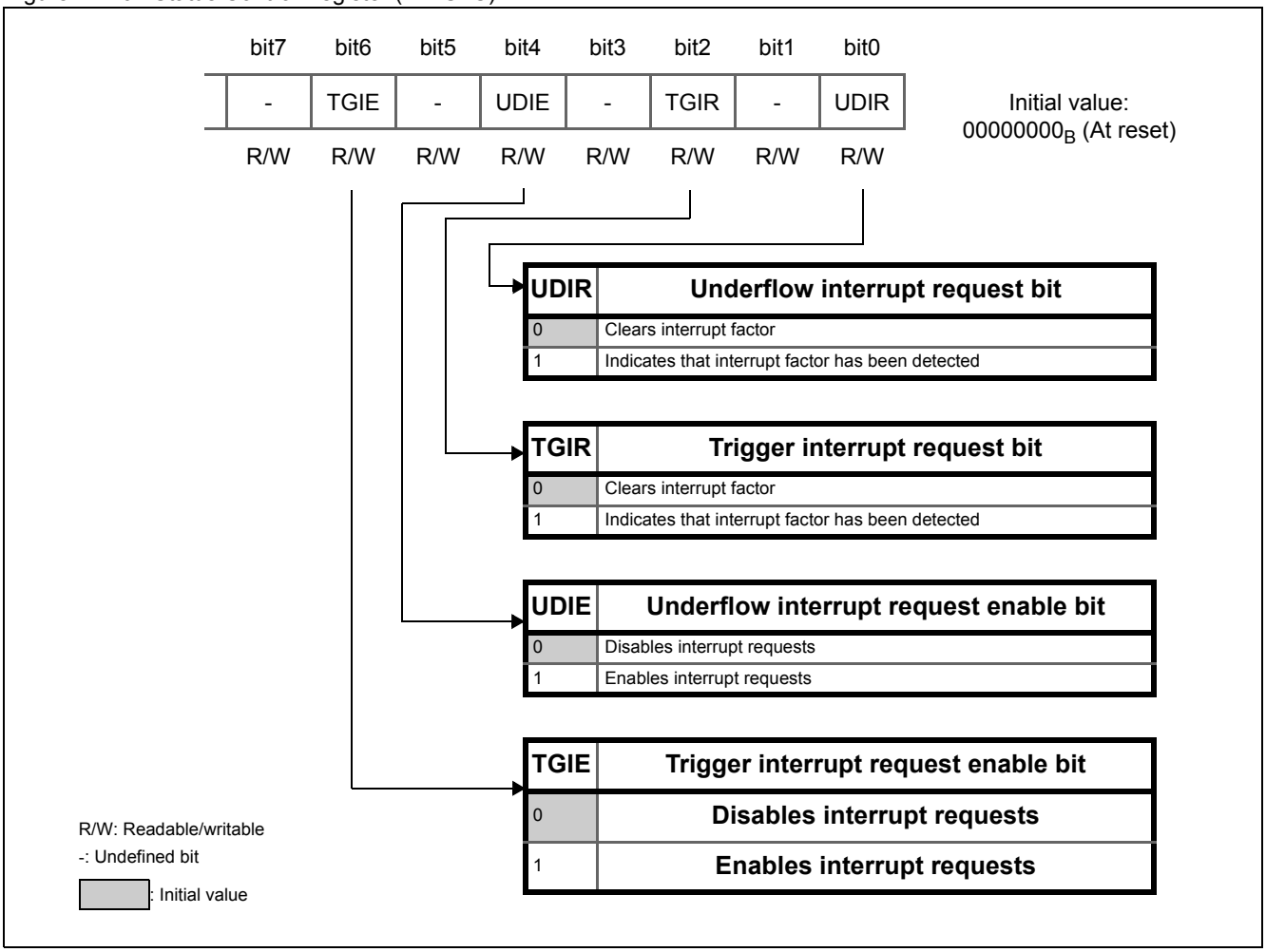


Table 22-11. Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	■ Controls bit2: TGIR interrupt requests. ■ Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	■ Controls bit0: UDIR interrupt requests. ■ Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	■ The TGIR bit is set to "1" upon detection of a software trigger or trigger input. ■ Writing "0" to the TGIR bit clears it. ■ Writing "1" to the TGIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	■ The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to $FFFF_H$ during counting from the value set as the "H" width. ■ Writing "0" to the UDIR bit clears it. ■ Writing "1" to the UDIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

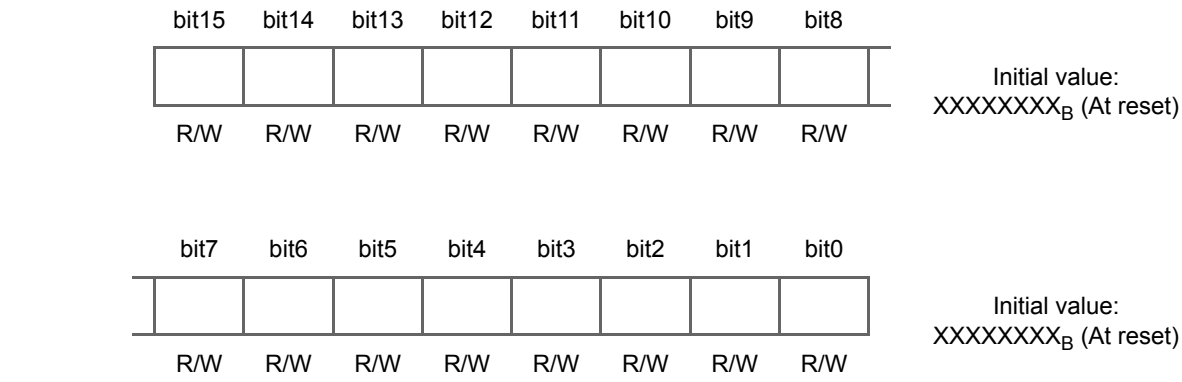
22.8.2.2 "L"-width Setting Reload Register (BTxPRLL)

The "L"-width setting reload register (BTxPRLL) is used to set the "L" width of PPG output waveforms. Transfer to the timer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting.

Bit configuration of the "L"-width setting reload register (BTxPRLL)

Figure 22-21 shows the bit configuration of the "L"-width setting reload register (BTxPRLL).

Figure 22-21. Bit Configuration of the "L"-width Setting Reload Register (BTxPRLL)



R/W : Readable/writable
X : Undefined value

The BTxPRLL register is used to set the "L" width of PPG output waveforms. Transfer to the timer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting.

- Access the BTxPRLL register using 16-bit data.
- Set the "L" width using the BTxPRLL register after selecting the PPG function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

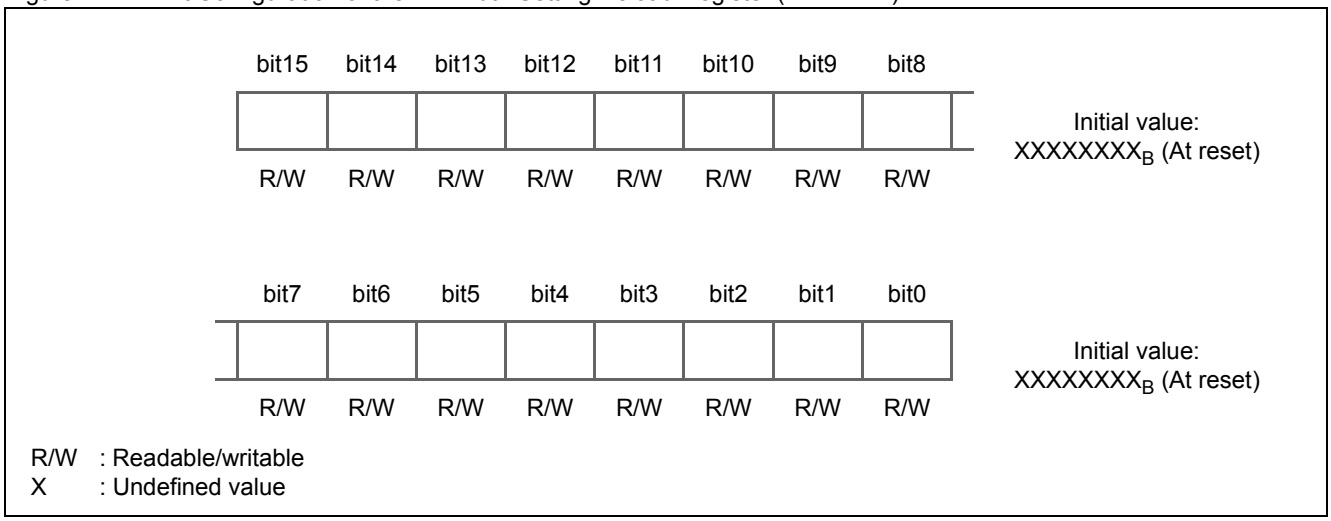
22.8.2.3 "H"-width Setting Reload Register (BTxPRLH)

The "H"-width setting reload register (BTxPRLH) is a buffered register for setting the "H" width of PPG output waveforms. Transfer from the BTxPRLH register to the buffer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting. Transfer from the buffer register to the timer register takes place when an underflow occurs at the end of "L" width counting.

Bit configuration of the "H"-width setting reload register (BTxPRLH)

Figure 22-22 shows the bit configuration of the "H"-width setting reload register (BTxPRLH).

Figure 22-22. Bit Configuration of the "H"-width Setting Reload Register (BTxPRLH)



The BTxPRLH register is used to set the "H" width of PPG output waveforms. Transfer from the BTxPRLH register to the buffer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting. Transfer from the buffer register to the timer register takes place when an underflow occurs at the end of "L" width counting.

- Access the BTxPRLH register using 16-bit data.
- Set the "H" width using the BTxPRLH register after selecting the PPG function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

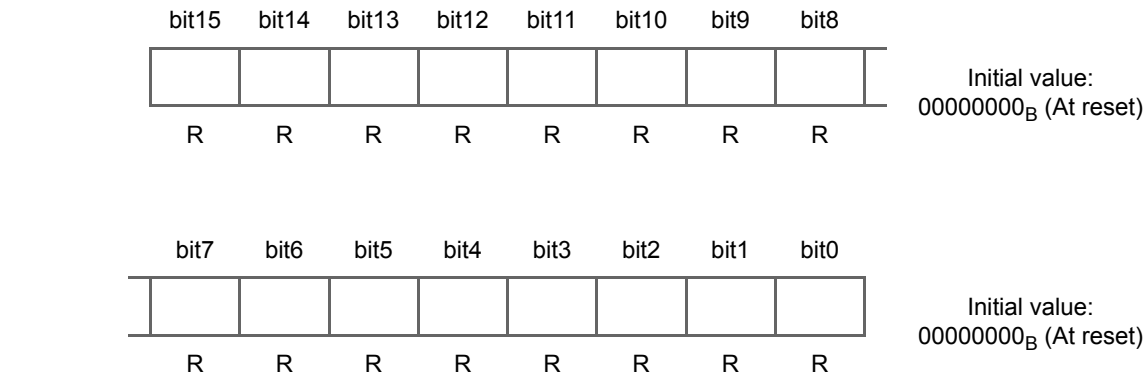
22.8.2.4 Timer Register (BTxTMR)

The timer register (BTxTMR) allows the value of the 16-bit down counter to be read from.

Bit configuration of the timer register (BTxTMR)

Figure 22-23 shows the bit configuration of the PPG timer register (BTxTMR).

Figure 22-23. Bit Configuration of the Timer Register (BTxTMR)



The BTxTMR register allows the value of the 16-bit down counter to be read from.

Note: Access the BTxTMR register using 16-bit data.

22.8.2.5 16-bit PPG Timer Operation

In PPG timer mode, an arbitrary output pulse can be controlled by setting its "L" and "H" widths in their respective reload registers.

Principles of operation

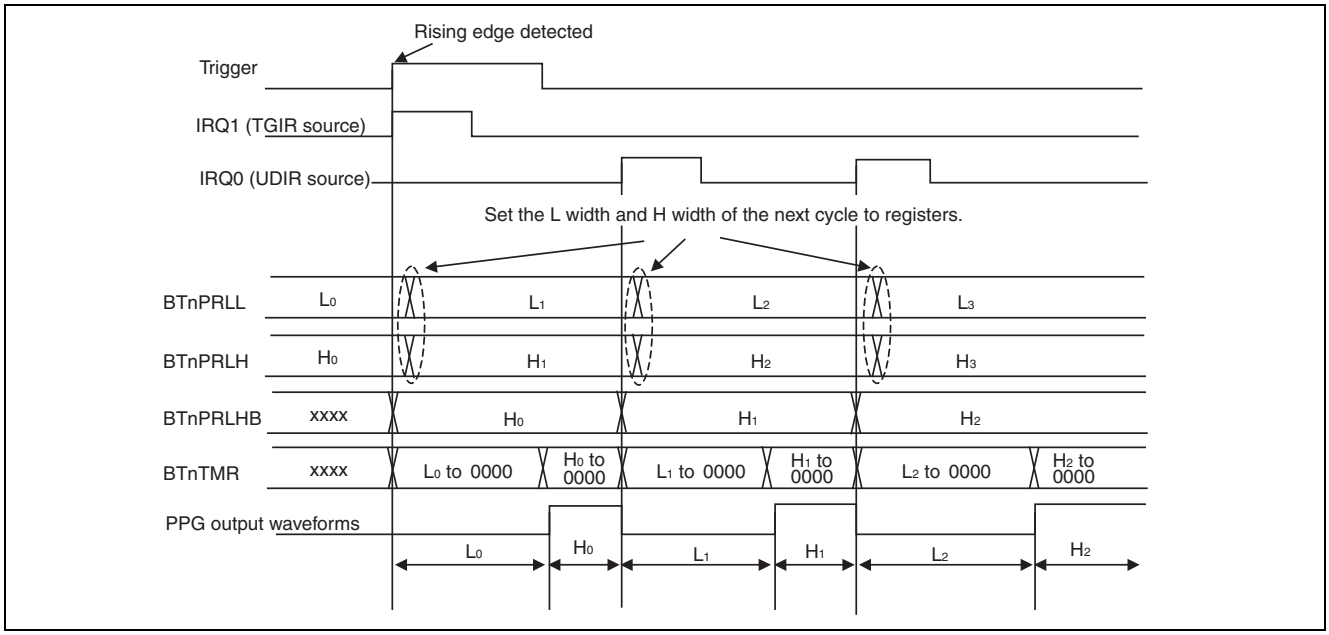
The PPG timer has two 16-bit reload registers for setting the "L" and "H" widths respectively and one "H" width setting buffer (BTxPRLL, BTxPRLH, BTxPRLHB).

In response to the start trigger, the 16-bit down counter loads the BTxPRLL value and the BTxPRLH value is transferred to the BTxPRLHB buffer at the same time. The counter is decremented every count clock with the PPG output at the "L" level. When an underflow is detected, the counter reloads the BTxPRLHB value and is decremented with the PPG output waveform inverted. When an underflow is detected again, the PPG output waveform is inverted, the counter reloads the BTxPRLL set value, and the BTxPRLH set value is transferred to the BTxPRLHB buffer.

Through these steps, the output waveform becomes the pulse output with the "L" and "H" widths corresponding to their respective reload register values.

Reload register write timing

Data is written to the BTxPRLL and BTxPRLH reload registers upon detection of a start trigger and between when the underflow interrupt request bit (UDIR) is set and when the next period begins. The data set then becomes the setting for the next period. The BTxPRLL and BTxPRLH settings are automatically transferred to the BTxTMR and BTxPRLHB, respectively, upon detection of a start trigger and when an underflow occurs at the end of "H" width counting. The data transferred to the BTxPRLHB is automatically reloaded to the BTxTMR when an underflow occurs at the end of "L" width counting.



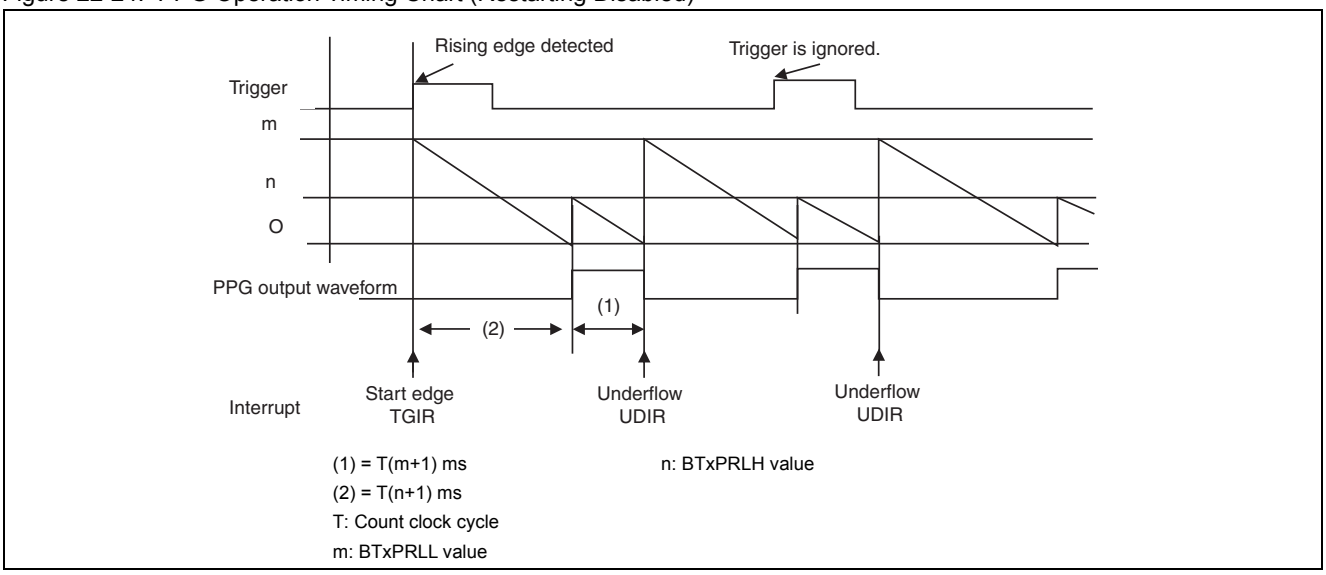
22.8.2.6 Continuous Operation

In continuous operation mode, an arbitrary pulse can be output continuously by updating the "L" and "H" widths at the set timing of each interrupt. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

Continuous operation

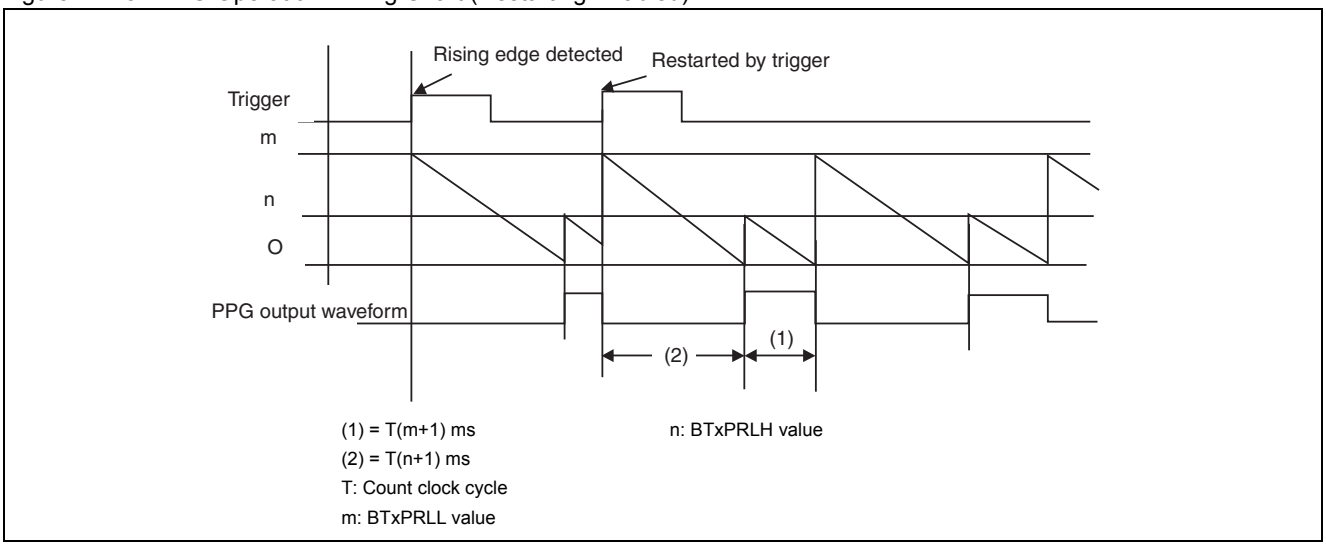
- When restarting is disabled (RTGEN = 0)

Figure 22-24. PPG Operation Timing Chart (Restarting Disabled)



- When restarting is enabled (RTGEN = 1)

Figure 22-25. PPG Operation Timing Chart (Restarting Enabled)



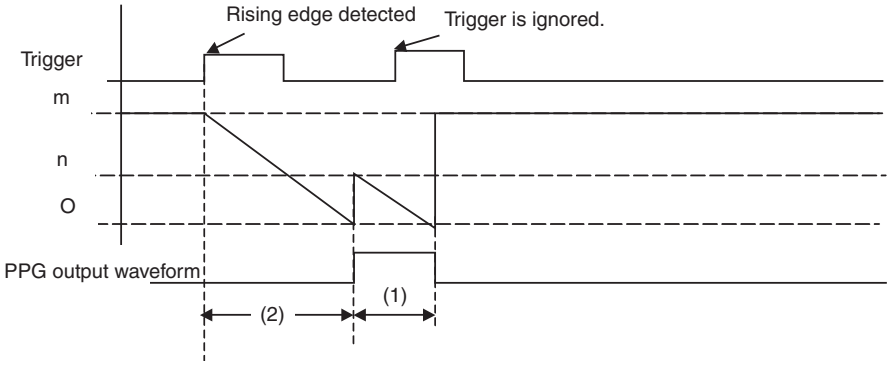
22.8.2.7 One-shot Operation

In one-shot operation mode, single pulses with an arbitrary width can be output by trigger. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

One-shot operation

- When restarting is disabled (RTGEN = 0)

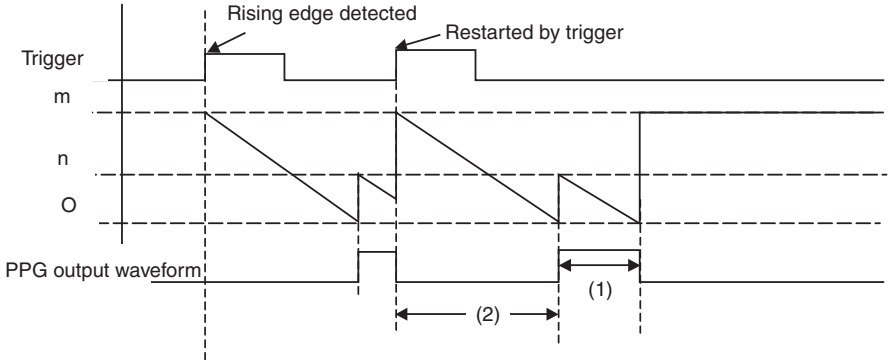
Figure 22-26. One-shot Operation Timing Chart (Trigger Restarting Disabled)



(1) = T(m+1) ms
(2) = T(n+1) ms
T: Count clock cycle
m: BTxPRLL value
n: BTxPRLH value

- When restarting is enabled (RTGEN = 1)

Figure 22-27. One-shot Operation Timing Chart (Trigger Restarting Enabled)



(1) = T(m+1) ms
(2) = T(n+1) ms
T: Count clock cycle
m: BTxPRLL value
n: BTxPRLH value

Relationship between Reload Value and Pulse Width

The output pulse width is obtained by adding 1 to the value written in the 16-bit reload register and multiplying the result by the count clock cycle. When the reload register value is 0000_H, therefore, the output has a pulse width of one count clock cycle. When the reload register value is $FFFF_H$, the output has a pulse width of 65536 count clock cycles. The pulse width is calculated from the following equation.

PL = T × (L+1)
PH = T × (H+1)

PL : "L" pulse width
PH : "H" pulse width
T : Count clock cycle
L : BTxPRLL value
H : BTxPRLH value

22.8.2.8 Interrupt Factors and Timing Chart

This section provides the interrupt factors and timing chart.

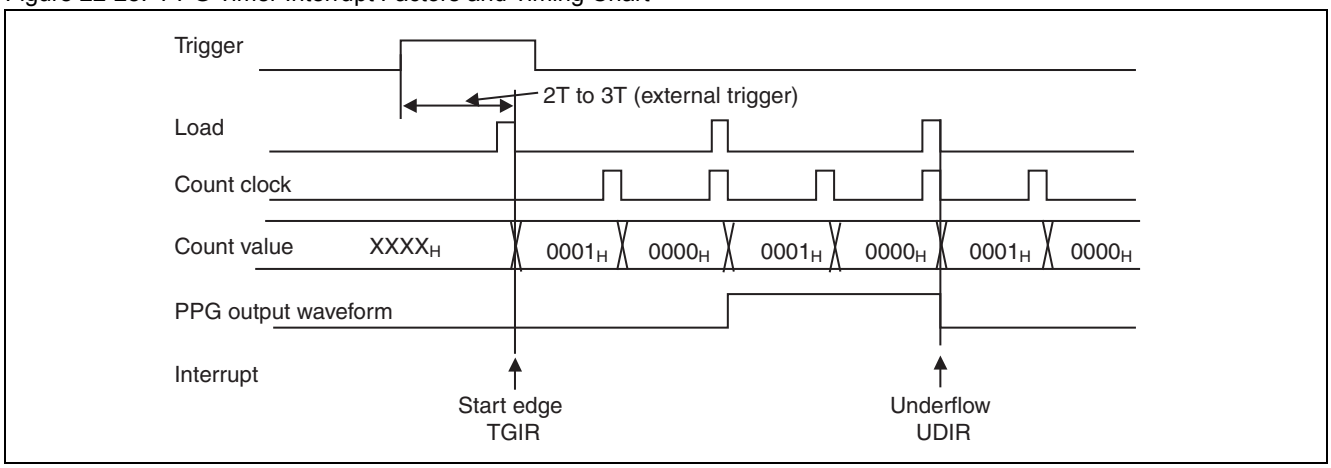
Interrupt factors and timing chart (PPG output: Normal polarity)

A software trigger requires T and an external trigger requires 2T to 3T (T: peripheral clock (PCLK) cycle) until the counter value is loaded after the trigger is generated.

Interrupt factors are set when the PPG start trigger is detected and when an underflow is detected during "H" level output.

Figure 22-28 shows the interrupt factors and timing chart, assuming "L" width setting = 1 and "H" width setting = 1.

Figure 22-28. PPG Timer Interrupt Factors and Timing Chart



22.8.3 Reload Timer Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the reload timer.

- Timer Control Register (BTxTMCR) for Reload Timer
- Period Setting Register (BTxPCSR)
- Timer Register (BTxTMR)
- 16-bit Reload Timer Operation

22.8.3.1 Timer Control Register (BTxTMCR) for Reload Timer

The timer control register (BTxTMCR) controls the reload timer.

Timer control register (BTxTMCR upper byte)

Figure 22-29. Timer Control Register (BTxTMCR Upper Byte)

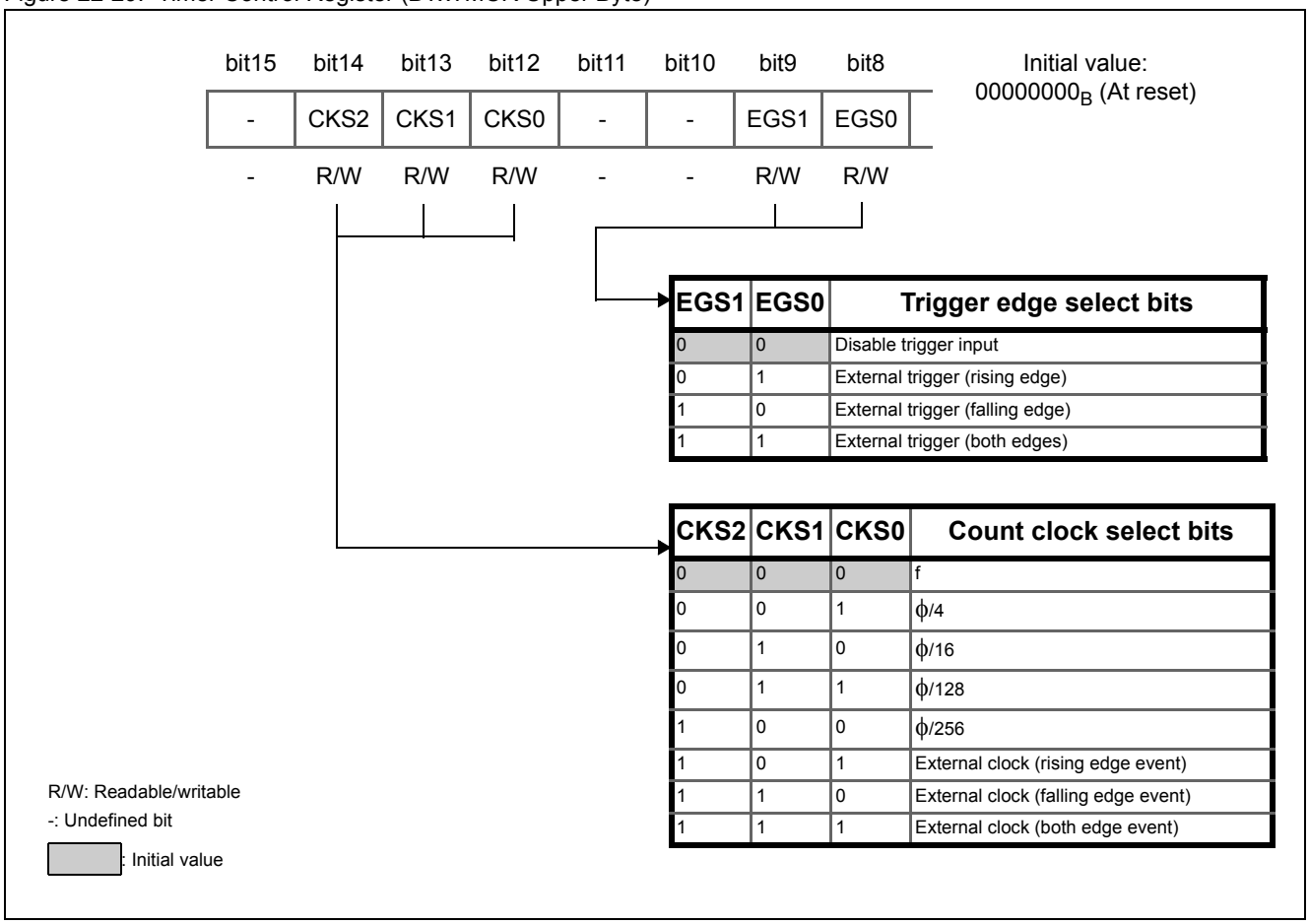


Table 22-12. Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	■ The read value of this bit is undefined. ■ Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	■ Select the count clock for the 16-bit down counter. ■ The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11, bit10	Undefined bits	■ The value read is "0" ■ When writing to these bits, write "0".
bit9, bit8	EGS1, EGS0: Trigger edge select bits	■ Select the effective edge of the input waveform as an external trigger to set the trigger condition. ■ When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. **Note:** Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0. ■ EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

Timer control register (BTxTMCR Lower Byte)

Figure 22-30. Timer Control Register (BTxTMCR Lower Byte)

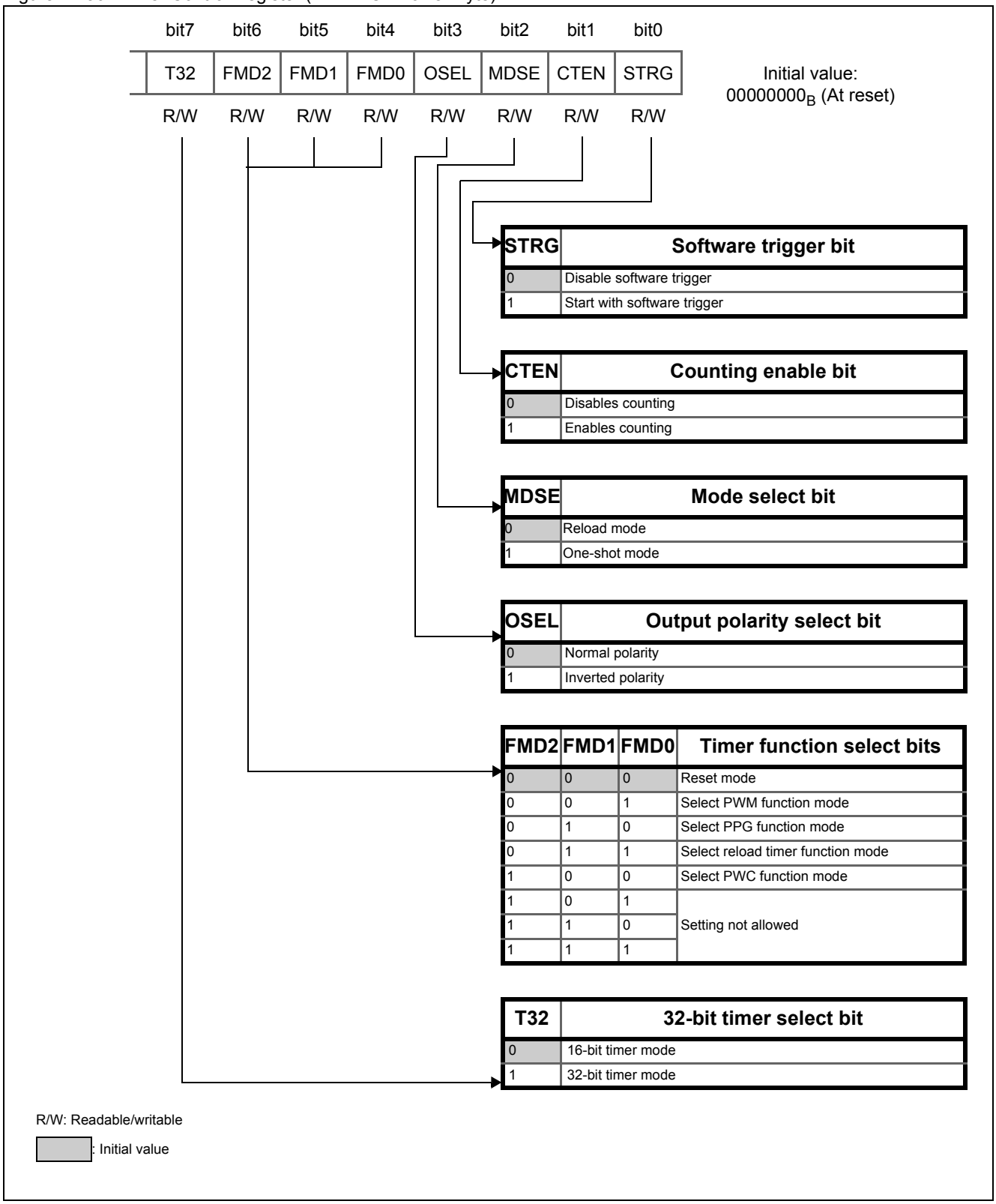


Table 22-13. Timer Control Register (BTxTMCR Lower Byte)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>T32:
32-bit timer select bit</td><td>■ This bit selects the 32-bit timer mode.
■ When the FMD2, FMD1, and FMD0 bits contain "011_B" to select the reload timer, setting the T32 bit to "1" places the timer in 32-bit timer mode.
■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
→ See Section "22.5 32-bit Mode Operations".</td></tr>
<tr><td>bit6
to
bit4</td><td>FMD2, FMD1,
FMD0:
Timer function select bits</td><td>■ These bits select the timer function mode.
■ Setting the FMD2, FMD1, and FMD0 bits to "011_B" selects the reload timer function mode.
■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit3</td><td>OSEL:
Output polarity select bit</td><td>■ Selects the timer output at normal level or inverted level.
■ The output waveform is generated as follows depending on the combination with the MDSE bit (bit2):
<table>
<tr><th>MDSE</th><th>OSEL</th><th>Output Waveforms</th></tr>
<tr><td>0</td><td>0</td><td>Toggle output of "L" at the count start</td></tr>
<tr><td>0</td><td>1</td><td>Toggle output of "H" at the count start</td></tr>
<tr><td>1</td><td>0</td><td>Rectangular wave of "H" during count</td></tr>
<tr><td>1</td><td>1</td><td>Rectangular wave of "L" during count</td></tr>
</table></td></tr>
<tr><td>bit2</td><td>MDSE:
Mode select bit</td><td>■ Setting the MDSE bit to "0" selects reload mode, in which the counter loads the reload register value to continue counting the moment a count value underflow occurs from 0000_H to $FFFF_H$.
■ Setting the MDSE bit to "1" selects one-shot mode, in which the counter stops operation the moment a count value underflow occurs from 0000_H to $FFFF_H$.
■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit1</td><td>CTEN:
Counting enable bit</td><td>■ This bit enables the down counter.
■ Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.</td></tr>
<tr><td>bit0</td><td>STRG:
Software trigger bit</td><td>■ Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger.
Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.
■ The value read from the STRG bit is always "0".
Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</td></tr>
</table>

Status control register (BTxSTC)

Figure 22-31. Status Control Register (BTxSTC)

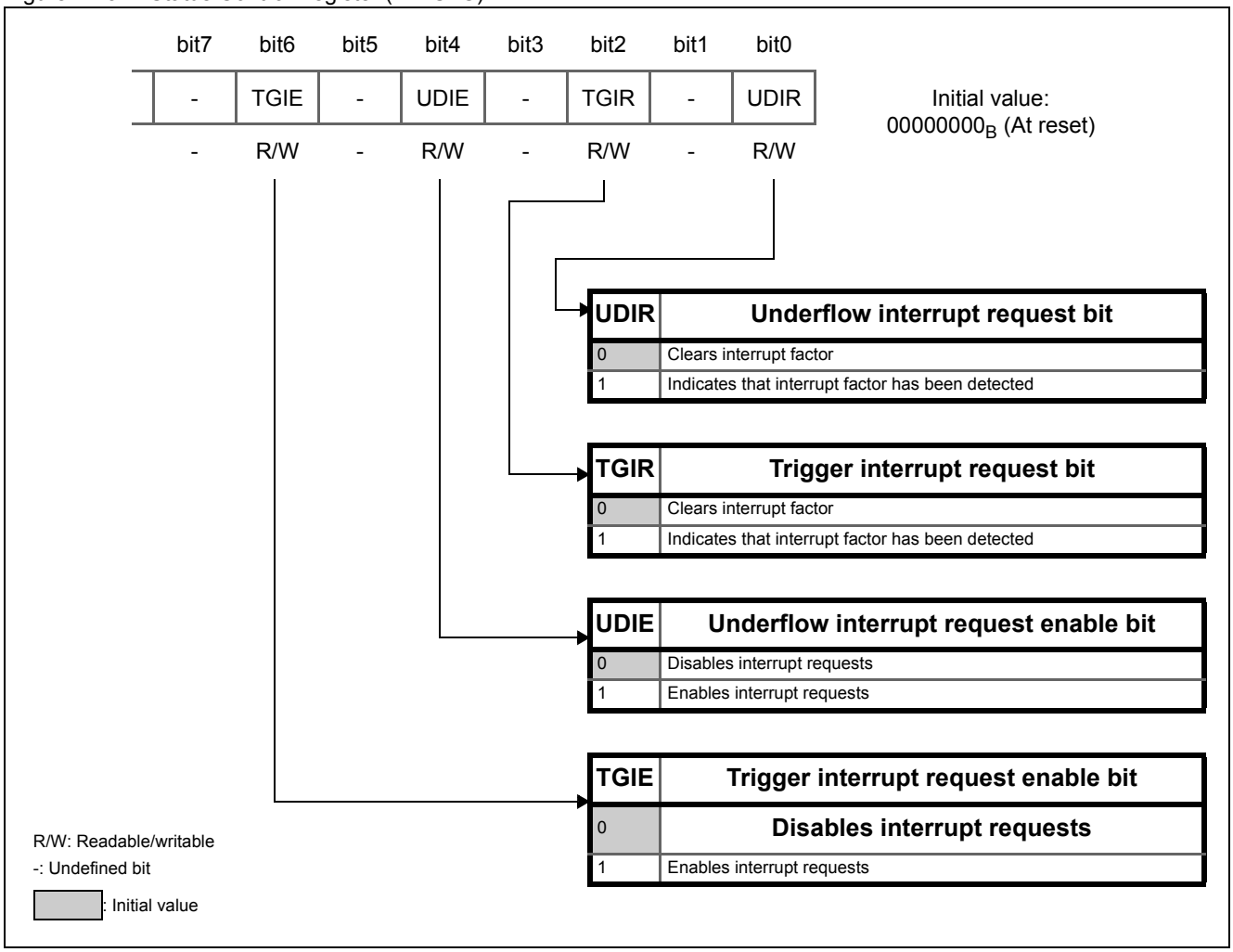


Table 22-14. Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	■ Controls bit2:TGIR interrupt requests. ■ Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	■ Controls bit0:UDIR interrupt requests. ■ Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	■ The TGIR bit is set to "1" upon detection of a software trigger or trigger input. ■ Writing "0" to the TGIR bit clears it. ■ Writing "1" to the TGIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	■ The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to $FFFF_H$. ■ Writing "0" to the UDIR bit clears it. ■ Writing "1" to the UDIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

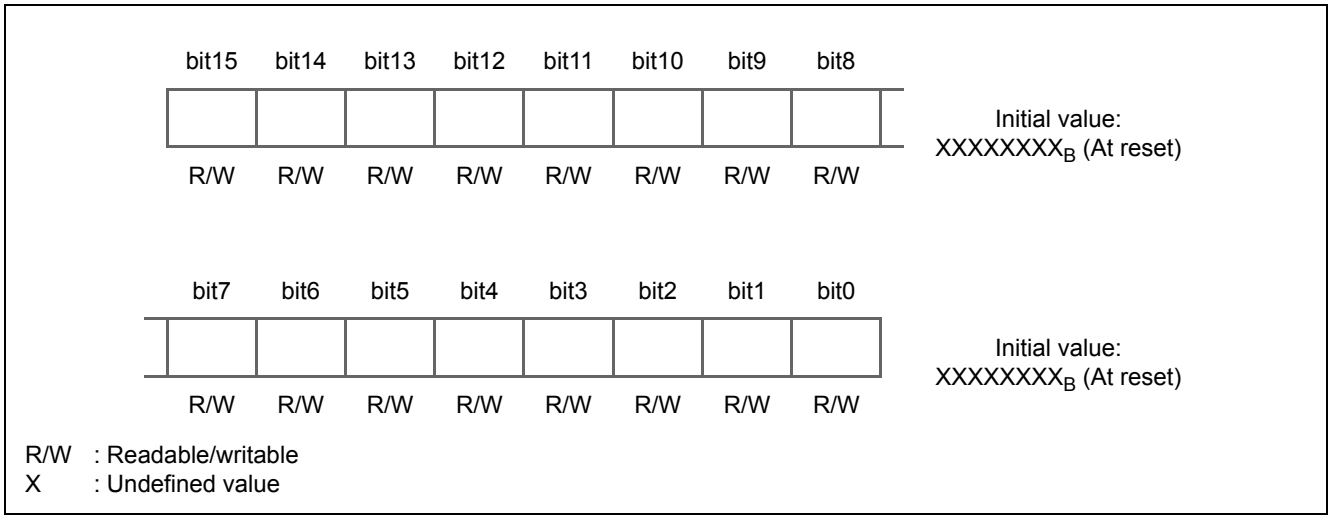
22.8.3.2 Period Setting Register (BTxPCSR)

The period setting register (BTxPCSR) holds the initial count value. In 32-bit mode, the register holds the initial count value of the lower 16 bits for the even-numbered channel or the initial count value of the upper 16 bits for the odd-numbered channel. The initial value immediately after a reset is undefined. To access this register, be sure to use a 16-bit data transfer instruction.

Bit configuration of the period setting register (BTxPCSR)

Figure 22-32 shows the bit configuration of the period setting register (BTxPCSR).

Figure 22-32. Bit Configuration of the Period Setting Register (BTxPCSR)



The BTxPCSR register is used to set the period. Transfer to the timer register takes place when an underflow occurs.

- Access the BTxPCSR register using 16-bit data.
- Set the period using the BTxPCSR register after selecting the reload timer function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.
- To write data to the BTxPCSR register in 32-bit mode, access its upper 16-bit data (data for the odd-numbered channel) first and then the lower 16-bit data (data for the even-numbered channel).

22.8.3.3 Timer Register (BTxTMR)

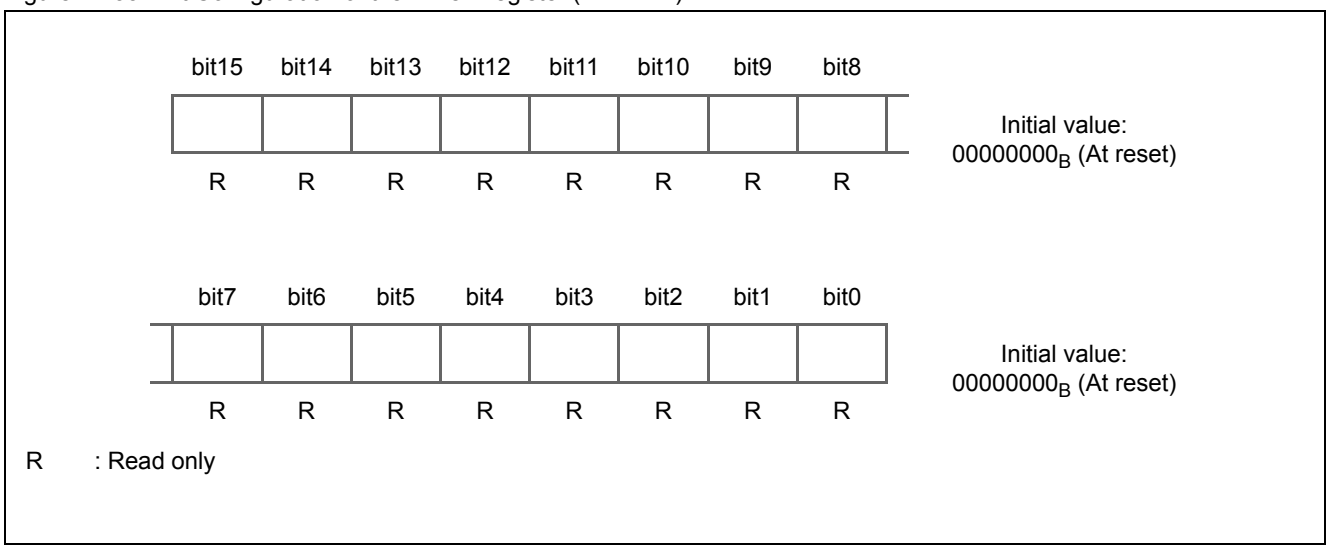
The timer register (BTxTMR) allows the count value of the timer to be read from. In 32-bit mode, the register holds the count value of the lower 16 bits for the even-numbered channel or the count value for the upper 16 bits for the odd-numbered channel. The initial value is undefined.

To read this register, be sure to use a 16-bit data transfer instruction.

Bit configuration of the timer register (BTxTMR)

Figure 22-33 shows the bit configuration of the timer register (BTxTMR).

Figure 22-33. Bit Configuration of the Timer Register (BTxTMR)



The BTxTMR register allows the value of the 16-bit down counter to be read from.

Notes:

- Access the BTxTMR register using 16-bit data.
- To read data from the BTxTMR register in 32-bit mode, access its lower 16-bit data (data for the even-numbered channel) first and then the upper 16-bit data (data for the odd-numbered channel).

22.8.3.4 16-bit Reload Timer Operation

In reload timer mode, the timer decrements the counter from the value set in the period setting register in synchronization with the count clock, and finishes counting when the count value reaches "0" or continues operation with the period setting loaded automatically until the counter stops being decremented.

Counting with the internal clock selected

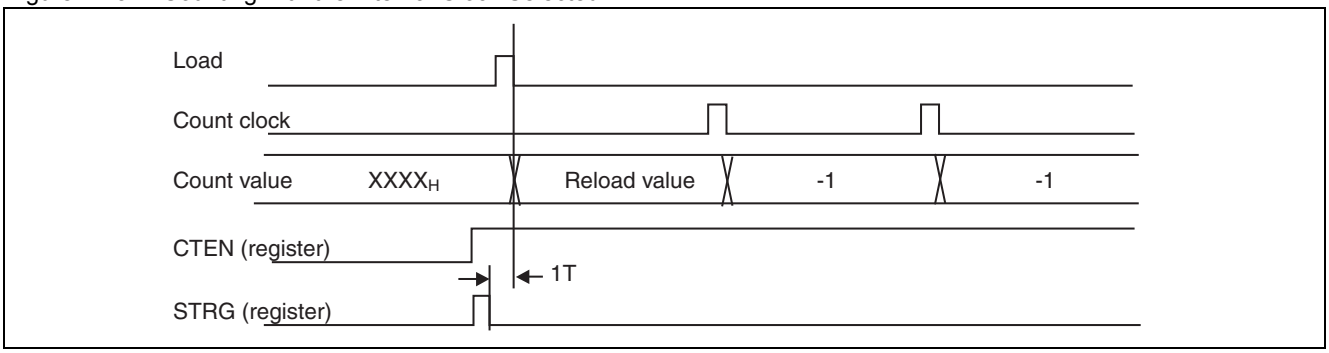
To start counting the moment counting is enabled, write "1" to both of the CTEN and STRG bits in the timer control register. The STRG bit maintains the trigger input always enabled irrespective of the operation mode as long as the timer is active (CNTE = 1).

Enable counting and start the timer using a software trigger or external trigger, and the timer loads the period setting register value to the counter to start decrementing the counter.

It takes 1T (T: peripheral clock (PCLK) cycle) for data in the period setting register to be loaded into the counter after the counter start trigger is set.

Figure 22-34 illustrates how the counter is started by the software trigger and operates.

Figure 22-34. Counting with the Internal Clock Selected



Underflow operation

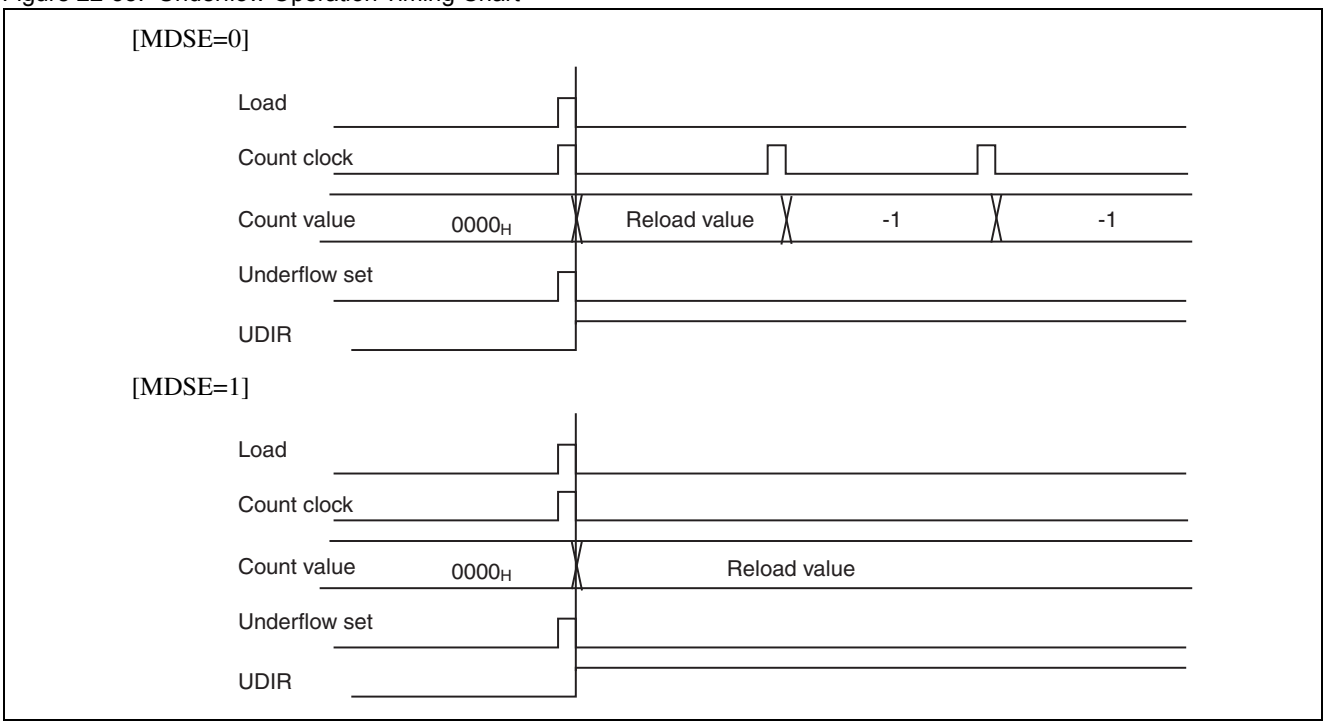
When the counter value changes from "0000_H" to "$FFFF_H$", the transition is detected as an underflow. When the counter counts [period setting register value + 1], therefore, an underflow occurs.

When an underflow occurs, the content of the period setting register (BTxPCSR) is loaded into the counter, and the counter continues counting if the MDSE bit in the timer control register (BTxTMCR) is "0". If the MDSE bit is "1", the counter stops operation with the loaded counter value left unchanged.

When an underflow occurs, the UDIR bit in the status control register (BTxSTC) is set and an interrupt request occurs if the UDIE bit is "1".

Figure 22-35 is a timing chart of underflow operation.

Figure 22-35. Underflow Operation Timing Chart

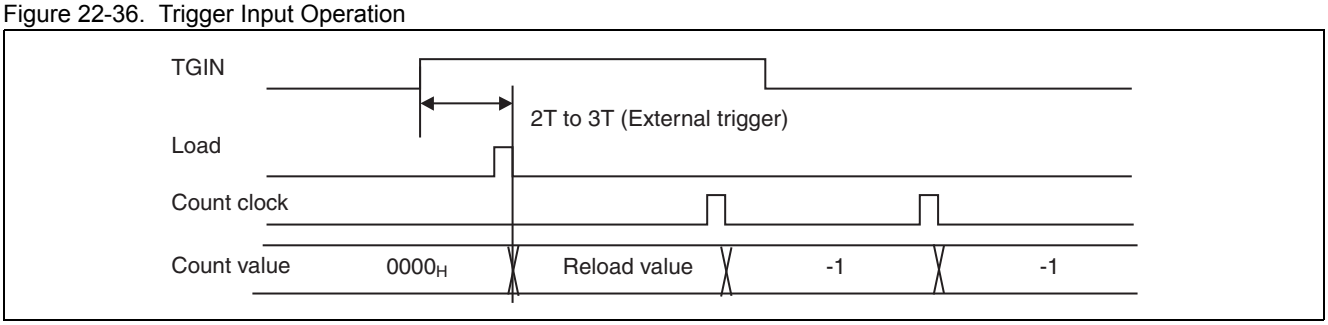


Input pin operation

The TGIN pin can be used as a trigger input. When the effective edge is input to the TGIN pin, the counter loads the content of the period setting register and starts counting. It takes 2T or 3T (T: peripheral clock (PCLK) cycle) for the counter value to be loaded after the trigger is applied.

Figure 22-36 illustrates the trigger input operation with the rising edge selected as the effective edge.

Figure 22-36. Trigger Input Operation

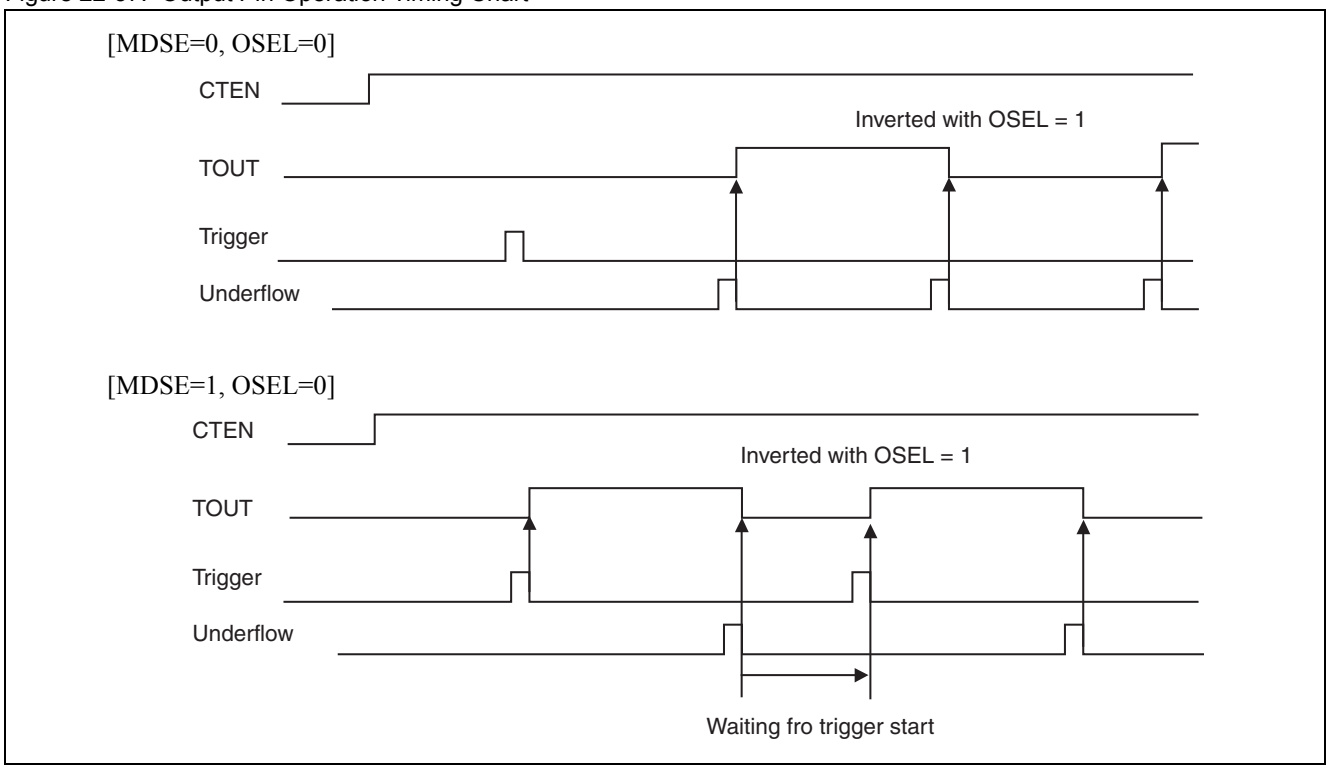


Output pin operation

The TOUT pin functions as a toggle output to be inverted at each underflow in reload mode and as a pulse output to indicate that counting is in process in one-shot mode. The output polarity can be set by the OSEL bit in the timer control register (BTxTMCR). When the OSEL bit is "0", the initial value of the toggle output is "0" and that of the one-shot pulse output is "1" (indicating that counting is in process). Setting the OSEL bit to "1" inverts the output waveform.

Figure 22-37 is a timing chart of output pin operation.

Figure 22-37. Output Pin Operation Timing Chart



22.8.4 PWC Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PWC timer.

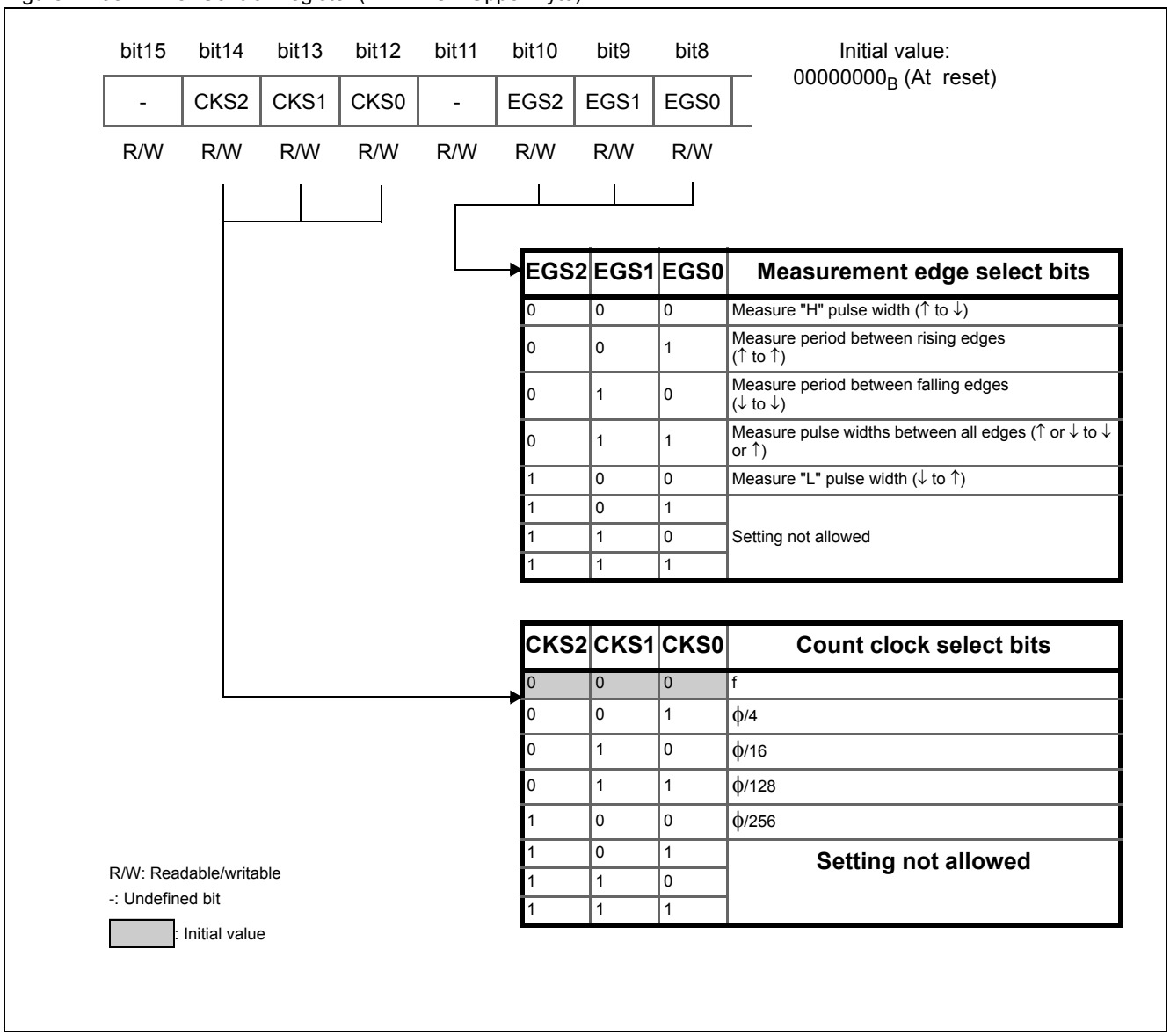
- Timer Control Register (BTxTMCR) for PWC Timer
- Data Buffer Register (BTxDTBF)
- PWC Operation

22.8.4.1 Timer Control Register (BTxTMCR) for PWC Timer

The timer control register (BTxTMCR) controls the PWC timer.

Timer control register (BTxTMCR upper byte)

Figure 22-38. Timer Control Register (BTxTMCR Upper Byte)



EGS2	EGS1	EGS0	Measurement edge select bits
0	0	0	Measure "H" pulse width (↑ to ↓)
0	0	1	Measure period between rising edges (↑ to ↑)
0	1	0	Measure period between falling edges (↓ to ↓)
0	1	1	Measure pulse widths between all edges (↑ or ↓ to ↓ or ↑)
1	0	0	Measure "L" pulse width (↓ to ↑)
1	0	1	Setting not allowed
1	1	0	
1	1	1	

CKS2	CKS1	CKS0	Count clock select bits
0	0	0	f
0	0	1	ϕ/4
0	1	0	ϕ/16
0	1	1	ϕ/128
1	0	0	ϕ/256
1	0	1	Setting not allowed
1	1	0	
1	1	1	

Figure 22-39. Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	■ Select the count clock for the 16-bit up counter. ■ The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit10 to bit8	EGS2, EGS1, EGS0: Measurement edge select bits	■ Set the measurement edge condition. ■ EGS2, EGS1, and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

Timer control register (BTxTMCR lower byte)

Figure 22-40. Timer Control Register (BTxTMCR Lower Byte)

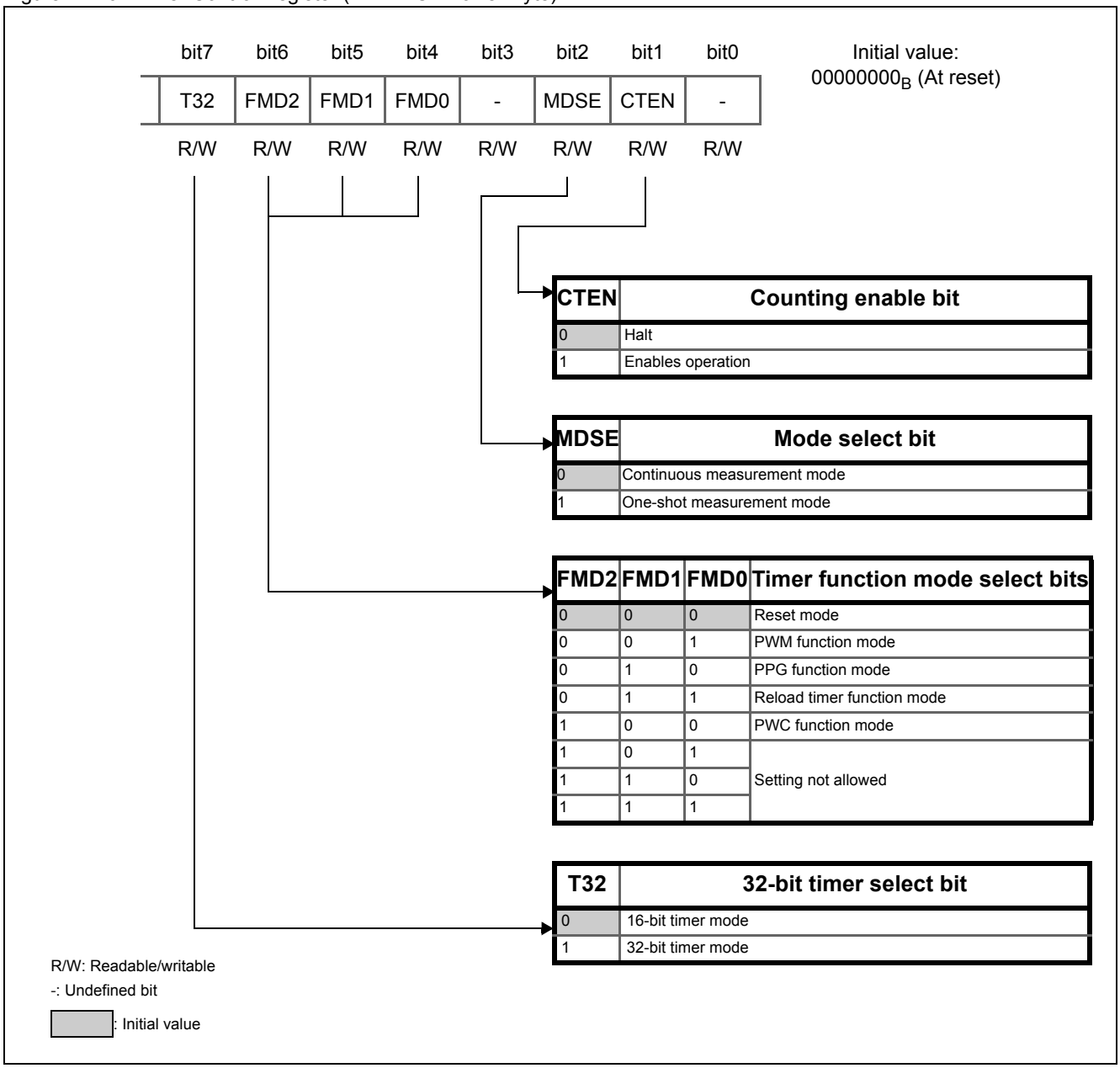


Table 22-15. Timer Control Register (BTxTMCR Lower Byte)

Bit name		Function
bit7	T32: 32-bit timer select bit	■ This bit selects the 32-bit timer mode. ■ When the FMD2, FMD1, and FMD0 bits contain "100_B" to select the PWC timer, setting the T32 bit to "1" places the timer in 32-bit PWC mode. ■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. → See Section "22.5 32-bit Mode Operations".
bit6 to bit4	FMD2, FMD1, FMD0: Timer function mode select bits	■ These bits select the timer function mode. ■ Setting the FMD2, FMD1, and FMD0 bits to "100_B" selects the PWC timer function mode. ■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
bit3	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit2	MDSE: Mode select bit	■ Selects measurement mode as follows. **MDSE / Mode / Operation** 0 / Continuous measurement / Continuous measurement: buffer register enabled 1 / One-shot measurement / Halts after each measurement ■ The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
bit1	CTEN: Counting enable bit	■ This bit enables the starting or restarting of the up counter. ■ Writing "1" to this bit with the counter enabled for operation (CTEN bit = 1) causes a restart, resulting in the counter cleared and waiting for the measurement start edge. ■ Writing "0" to the bit with the counter enabled for operation (CTEN bit = 1 stops the counter.
bit0	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".

Status control register (BTxSTC)

Figure 22-41. Status Control Register (BTxSTC)

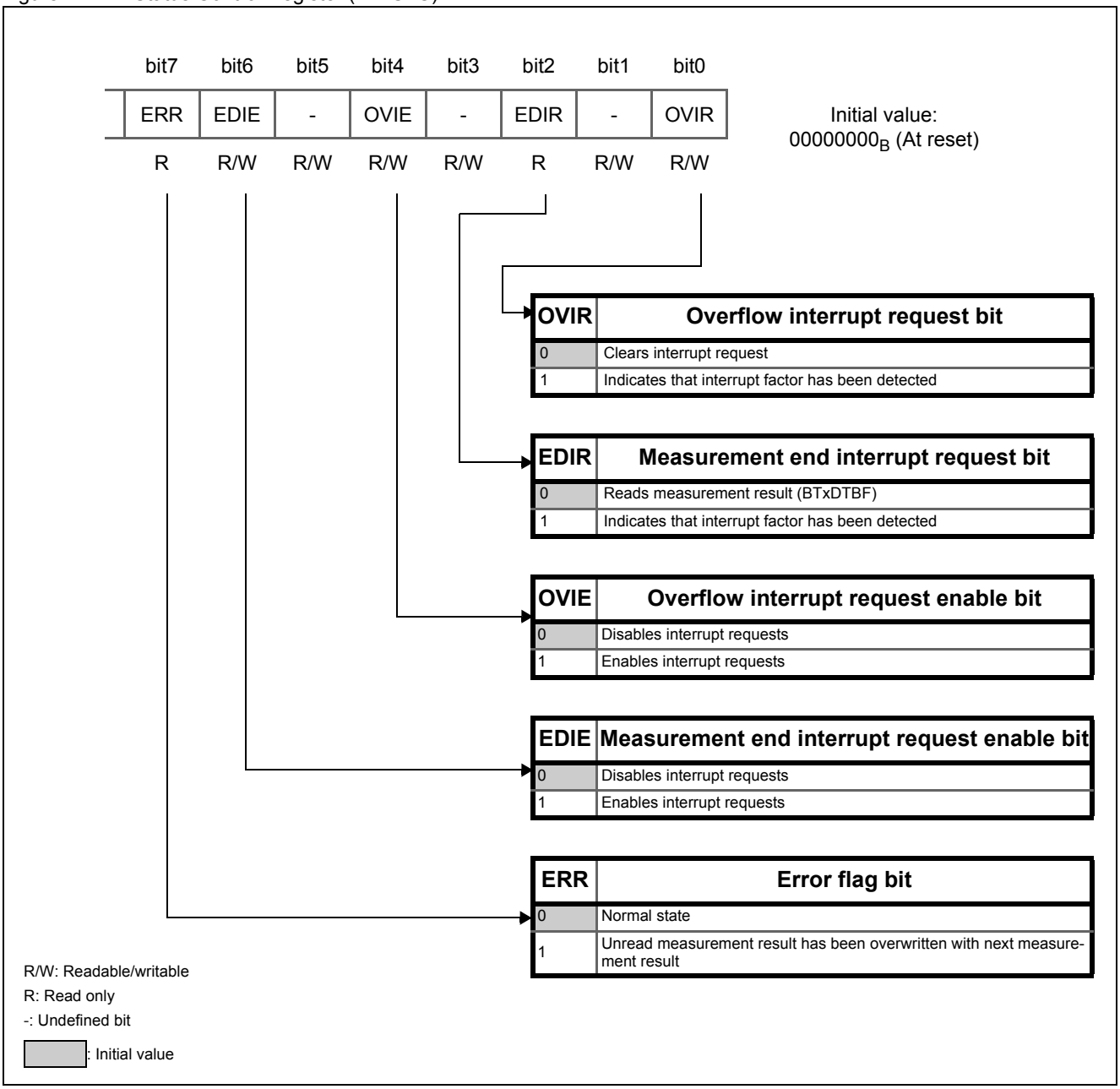


Table 22-16. Status Control Register (BTxSTC)

Bit name		Function
bit7	ERR: Error flag bit	■ This flag indicates that the next measurement has been completed before reading the current measurement result from the BTxDTBF register in continuous measurement mode. In this case, the BTxDTBF register is updated with the new measurement result, discarding the preceding measurement result. ■ Measurement continues irrespective of the ERR bit value. ■ The ERR bit can only be read; an attempt to write to it has no effect on the bit value. ■ The ERR bit is cleared by reading the measurement result (BTxDTBF).
bit6	EDIE: Measurement end interrupt request enable bit	■ Controls bit2: EDIR interrupt requests. ■ Setting the EDIR bit (bit2) with the EDIE bit enabling measurement end interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit4	OVIE: Overflow interrupt request enable bit	■ Controls bit0: OVIR interrupt requests. ■ Setting the OVIR bit (bit0) with the OVIE bit enabling overflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit2	EDIR: Measurement end interrupt request bit	■ Indicates that measurement has been completed. The flag is set to "1" upon completion. ■ The EDIR bit is cleared by reading the measurement result (BTxDTBF). ■ The EDIR bit can only be read; an attempt to write to it has no effect on the bit value.
bit1	Undefined bit	■ The value read is "0" ■ When writing to this bit, write "0".
bit0	OVIR: Overflow interrupt request bit	■ The flag is set to "1" when a count value overflow occurs from $FFFF_H$ to 0000_H. ■ Writing "0" to the OVIR bit clears it. ■ Writing "1" to the OVIR bit has no effect on the bit value. ■ When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

22.8.4.2 Data Buffer Register (BTxDTBF)

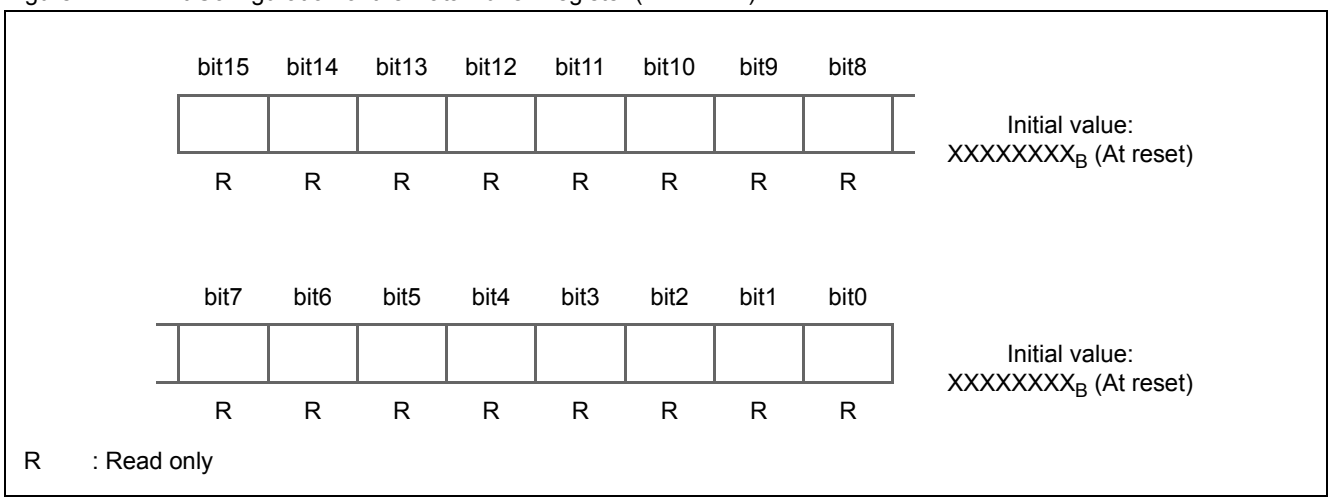
The data buffer register (BTxDTBF) allows the measured value or count value of the PWC timer to be read from. In 32-bit mode, the register holds the value of the lower 16 bits for the even-numbered channel or the value of the upper 16 bits for the odd-numbered channel.

To read this register, be sure to use a 16-bit data transfer instruction.

Bit configuration of the data buffer register (BTxDTBF)

Figure 22-42 shows the bit configuration of the data buffer register (BTxDTBF).

Figure 22-42. Bit Configuration of the Data Buffer Register (BTxDTBF)



- The BTxDTBF register can only be read in both of the continuous and one-shot measurement modes. An attempt to write to the register makes no change to the register value.
- In continuous measurement mode (BTxTMCR: bit3 MDSE = 1), the BTxDTBF register serves as a buffer register holding the preceding measurement result.
- In one-shot measurement mode (BTxTMCR: bit3 MDSE = 0), the BTxDTBF register directly accesses the up counter. Even during counting, the count value can be read from this register. When the measurement is completed, the register preserved the measurement result as it is.
- Access the BTxDTBF register using 16-bit data.

22.8.4.3 PWC Operation

The PWC timer has a pulse width measurement feature, capable of selecting the count clock from among five types and measuring the time between arbitrary events of the input pulse and their cycle. The following outlines the basic functions and operations of the pulse width measurement feature.

Pulse width measurement feature

When started, the timer clears the counter to "0000_H" but does not perform counting until the pre-set measurement start edge is input. Upon detection of the measurement start edge, the timer increments the counter from "0001_H". Upon detection of the measurement end edge, the timer stops the counter. The timer saves the count value between the two events as the pulse width to the register.

An interrupt request can be generated upon completion of measurement or when an overflow occurs.

After measurement, the timer acts as follows depending on the measurement mode:

- In one-shot measurement mode: The timer stops operation.
- In continuous measurement mode: The timer transfers the counter value to the buffer register and stops counting until the measurement start edge is input again.

Figure 22-43. Pulse Width Measurement Operation (One-shot Measurement Mode/"H" Width Measurement)

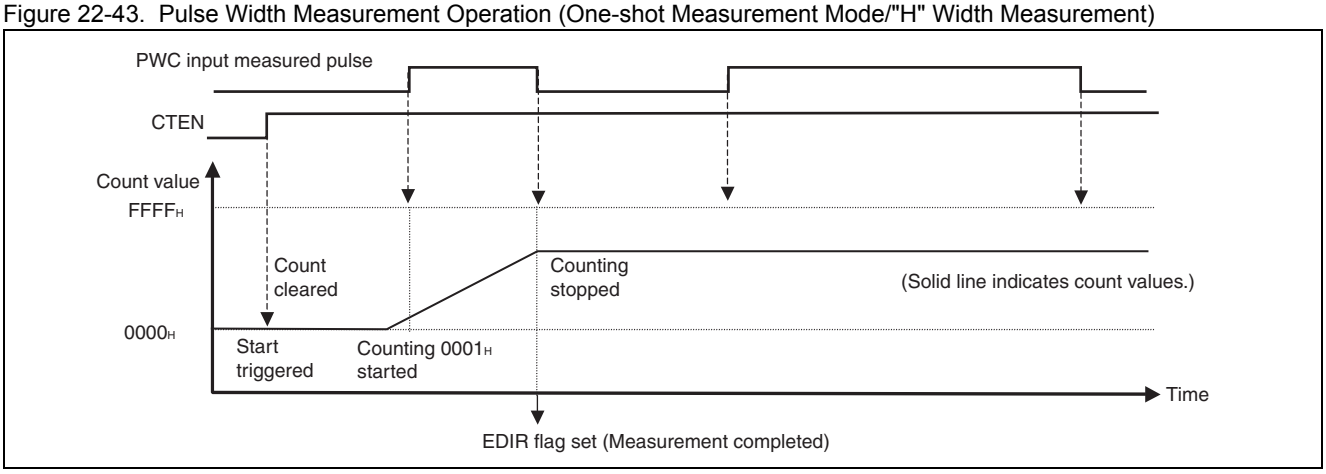
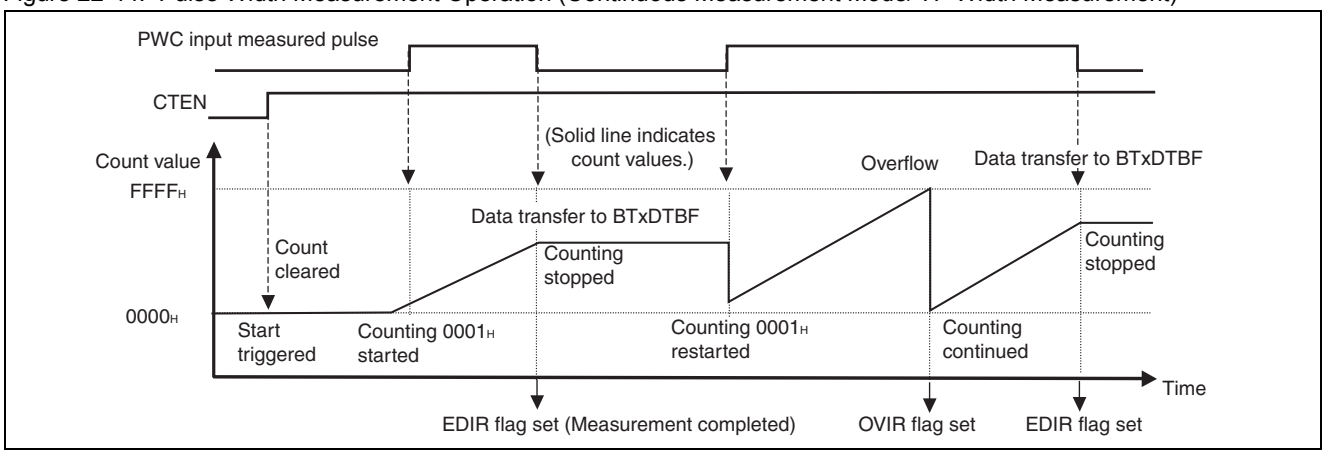


Figure 22-44. Pulse Width Measurement Operation (Continuous Measurement Mode/"H" Width Measurement)



Selecting the count clock

The count clock for the counter can be selected from among five types, depending on the settings of the CKS2 (bit6), CKS1 (bit5), and CKS0 (bit4) in the BTxTMCR registers.

The following count clocks can be selected:

BTxTMCR Register CKS2, CKS1, CKS0 bits	Internal count clock selected
000_B	Peripheral clock (PCLK) [Initial value]
001_B	Peripheral clock (PCLK) divided by 4
010_B	Peripheral clock (PCLK) divided by 16
011_B	Peripheral clock (PCLK) divided by 128
100_B	Peripheral clock (PCLK) divided by 256
101_B	Setting not allowed
110_B	
111_B	

The initial value immediately after a reset selects the peripheral clock (PCLK).

Note: Be sure to select the count clock before starting the counter.

Selecting the operation mode

Operation and measurement modes are selected depending on their settings in the BTxTMCR register.

Operation mode setting: BTxTMCR bit10 to bit8: EGS2, EGS1. EGS0
(Selecting the measurement edge)

Measurement mode setting: BTxTMCR bit2: MDSE
(Selecting one-shot/continuous measurement)

Listed below are the selectable operation modes and their respective bit settings..

Operation mode		MDSE	EGS2	EGS1	EGS0
↑ to ↓ "H" pulse width measurement	Continuous measurement: Buffer enabled	0	0	0	0
	One-shot measurement: Buffer disabled	1	0	0	0
↑ to ↑ measurement of period between rising edges	Continuous measurement: Buffer enabled	0	0	0	1
	One-shot measurement: Buffer disabled	1	0	0	1
↓ to ↓ measurement of period between falling edges	Continuous measurement: Buffer enabled	0	0	1	0
	One-shot measurement: Buffer disabled	1	0	1	0
↑ or ↓ to ↓ or ↑ measurement between all edges	Continuous measurement: Buffer enabled	0	0	1	1
	One-shot measurement: Buffer disabled	1	0	1	1
↓ to ↑ "L" pulse width measurement	Continuous measurement: Buffer enabled	0	1	0	0
	One-shot measurement: Buffer disabled	1	1	0	0
Setting not allowed		0	1	0	1
		1	1	0	1
		0	1	1	0
		1	1	1	0
		0	1	1	1
		1	1	1	1

The initial value immediately after a reset selects "H" pulse width/one-shot measurement mode.

Be sure to select the operation mode before starting the counter.

Starting and stopping pulse width measurement

Each type of measurement can be started, restarted, and aborted by the CTEN bit (bit1) in the BTxTMCR register.

You can start/restart pulse width measurement by writing "1" to the CTEN bit. You can abort it by writing "0" to the CTEN bit.

CTEN	Function
1	Starts/restarts pulse width measurement
0	Aborts pulse width measurement

Operation after being started

The timer operation after the pulse width measurement mode has been started does not start counting until the measurement start edge is input. Upon detection of the measurement start edge, the 16-bit up counter starts counting from "0001_H".

Restarting

Restarting the timer means starting the timer during operation again while it has already been started (by writing "1" again to the CTEN bit already containing "1"). When restarted, the timer behaves as follows:

- If restarted the timer waiting for the measurement start edge: No effect on its operation.
- If restarted during measurement: The timer clears the counter to "0000_H" and waits for the measurement start edge again.
  If the restart and measurement end edge detection occur at the same time, the measurement end flag (EDIR) is set. In continuous measurement mode, the measurement result is transferred to the BTxDTBF register.

Stopping

In one-shot measurement mode, the timer stops counting automatically when the counter causes an overflow or when measurement is completed, requiring no special attention. To stop the timer either in continuous measurement mode or before it stops automatically, you have to abort it.

Clearing the counters and their initial values

The 16-bit up counter is cleared to "0000_H" when:

- a reset occurs
- "1" is written to the CTEN bit (bit1) in the BTxTMCR register (including the case of restarting).

The 16-bit up counter is initialized to "0001_H" when measurement start edge is detected.

Details of pulse width measurement operation

- One-shot measurement and continuous measurement

  There are two modes of pulse width measurement: one is to perform measurement only once and the other is to perform measurement continuously. Each mode is selected by using the MDSE bit in the BTxTMCR register (see "Selecting the operation mode" in "22.8.4.3 PWC Operation"). The two modes have the following differences:

  - One-shot measurement mode:

    When the measurement end edge is input once, the counter stops counting and the measurement end flag (EDIR) in the BTxSTC register is set, finishing the current measurement session. If the counter is restarted at the same time, however, it waits for the measurement start edge.

  - Continuous measurement mode:

    When the measurement end edge is input, the counter stops counting, the measurement end flag (EDIR) in the BTxSTC register is set, and the counter remains idle until the measurement start edge is input again. Next time the measurement start edge is input, the counter is initialized to "0001_H" to start measurement. Upon completion of measurement, the measurement result in the counter is transferred to the BTxDTBF register.

    Be sure to select or change the measurement mode with the counter stopped.

- Measurement result data

  The one-shot measurement and continuous measurement modes are different in the handling of measurement results and counter values and the BTxDTBF function. The differences in measurement results between the two modes are as follows:

  - One-shot measurement mode:

    When the BTxDTBF register is read during operation, the count value being measured can be obtained.

    When the BTxDTBF register is read after measurement is completed, measurement result data is obtained.

  - Continuous measurement mode:

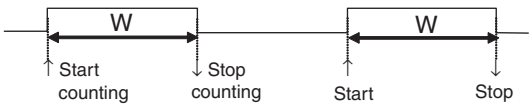
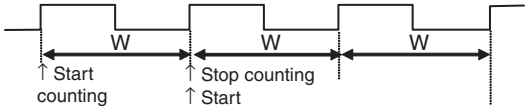
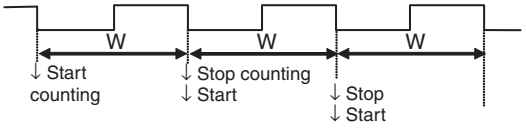
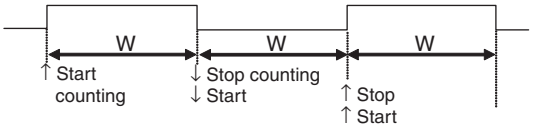
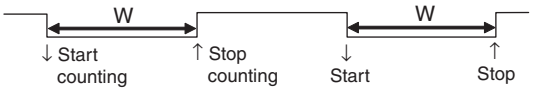
    When measurement is completed, the measurement result in the counter is transferred to the BTxDTBF register.

    When the BTxDTBF register is read, the last measurement result is obtained. During measurement operation, the BTxDTBF register holds the result of preceding measurement. The count value being measured cannot be read.

    If the current measurement is completed before the preceding measurement result is read in continuous measurement mode, the preceding measurement result is overwritten by the new measurement result. In this case, the error flag (ERR) in the BTxSTC register is set. The error flag (ERR) is cleared automatically when the BTxDTBF register is read.

Measurement mode and counting

Measurement mode can be selected from among five types, depending on what part of the input pulse is measured. The following table summarizes each measurement mode and its target.

Measurement mode	EGS2, EGS1, EGS0	Measurement target (W: Pulse width to be measured)
"H" pulse width measurement	000_B	 Measure the width of "H" period. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of falling edge
Measurement of period between rising edges	001_B	 Measure the period between rising edges. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of rising edge
Measurement of period between falling edges	010_B	 Measure the period between falling edges. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of falling edge
Measurement of pulse widths between all edges	011_B	 Measure the width between continuously input edges. Start counting (measurement) : upon detection of edge Stop counting (measurement) : upon detection of edge
Measurement of "L" pulse width	100_B	 Measure the width of the "L" period. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of rising edge

In any measurement mode, the counter started for measurement is cleared to "0000_H" and remains idle without counting until the measurement start edge is input. When the measurement start edge is input, the counter is incremented every count clock until the measurement end edge is input.

When measurement of pulse widths between all edges or period measurement is performed in continuous measurement mode, the end edge becomes the next measurement start edge.

- Pulse width/period calculation method

  The following equation can be used to calculate the measured pulse width/period from measurement result data obtained from the BTxDTBF register after measurement is completed:

$T_W = n \times t$ [ms]	T_W : Measured pulse width/period [ms] n : Measurement result data in BTxDTBF t : Count clock cycle [ms]

- Generating interrupt requests

  Interrupt requests can be generated in two ways.

  - Interrupt request in response to counter overflow

    When the counter is incremented to cause an overflow during measurement, the overflow flag (OVIR) is set and generates an interrupt request if overflow interrupt requests have been enabled.

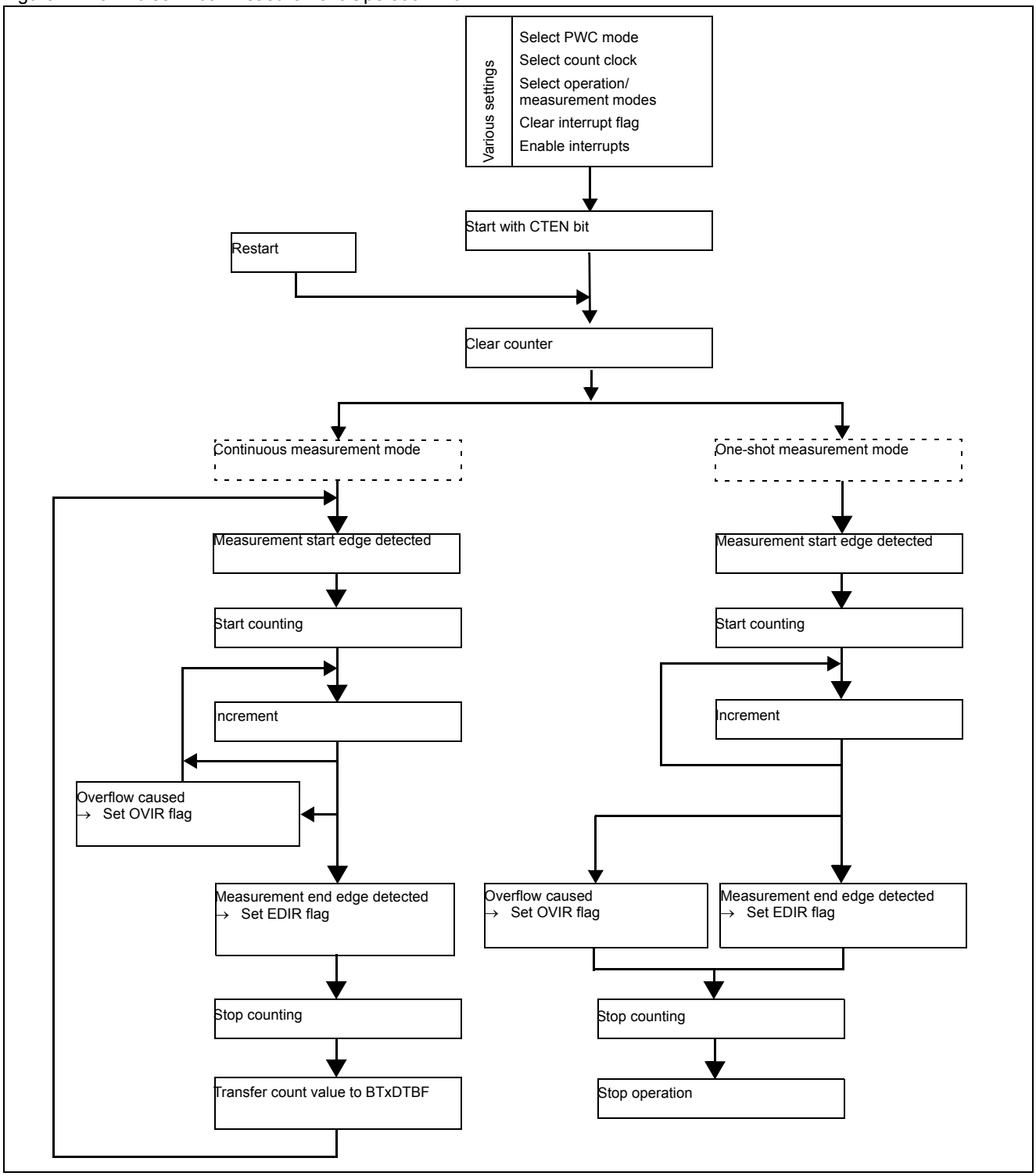
  - Interrupt request upon completion of measurement

    When the measurement end edge is detected, the measurement end flag (EDIR) in the BTxSTC register is set and generates an interrupt request if measurement end interrupt requests have been enabled.

    The measurement end flag (EDIR) is cleared automatically when the measurement result is read from the BTxDTBF register.

Pulse width measurement operation flow

Figure 22-45. Pulse Width Measurement Operation Flow



23. 10-Bit A/D Converter

This chapter explains the functions and operations of the 10-bit A/D converter.

23.1 Overview

23.2 Configuration

23.3 Pins

23.4 Registers

23.5 Interrupts

23.6 Explanation of Operations and Setting Procedure Examples

23.1 Overview

The 10-bit A/D converter is a device for converting analog signals to 10-bit digital signals.

This series microcontroller has 10-bit A/D converters, and 8 analog input channels can be assigned for conversion.

Overview

- Conversion time: 1.2 μs per channel, minimum (33 MHz peripheral clock (PCLK))
- Comparison/conversion method: RC-type successive comparison and conversion with sample-and-hold circuits
- Conversion mode: The modes that can be used are categorized into the following two types:
  - A/D scan conversion

    An optional conversion channel is selected from 8 channels and made subject to conversion.

    Two conversion modes are available: single conversion mode and repeat conversion mode. In single conversion mode, signals from the selected channel are converted only once. In repeat conversion mode, signals from the selected channel are converted repeatedly.
  - A/D priority conversion

    Once an activation trigger for high-priority A/D conversion is generated, that conversion is performed soon afterward by stopping A/D scan conversion in progress. There are two priority levels.
- Activation trigger: Activation triggers vary depending on the A/D conversion mode:
  - A/D scan conversion

    Conversion is activated by software or at detection of a rising edge of the TOUT signal of base timer ch.0.
  - A/D priority conversion (priority 1)

    Conversion is triggered by input of a falling edge from an external trigger input pin.
  - A/D priority conversion (priority 2)

    Conversion is activated by software or at detection of a rising edge of the TOUT signal of base timer ch.2.
- FIFO functionality: There 16 FIFO levels for A/D scan conversion and 4 FIFO levels for A/D priority conversion.
- Conversion result compare function: A/D conversion results can be compared.
- Independent control of channels: One of two kinds of sampling time can be set for each channel.
- Conversion results: A/D conversion results can specified to be stored left-justified (MSB side) or right-justified (LSB side).
- Interrupt request: Can be issued in the following cases:
  - Data has been stored in the predetermined number of stages in the FIFO used during A/D scan conversion.
  - Data has been stored in the predetermined number of stages in the FIFO used during A/D priority conversion.
  - A FIFO overrun occurred.
  - The comparison function is used to determine whether conversion results satisfy the interrupt request generation conditions.
- DMA transfer activation: Generation of an interrupt request can be used for DMA transfer of conversion results.

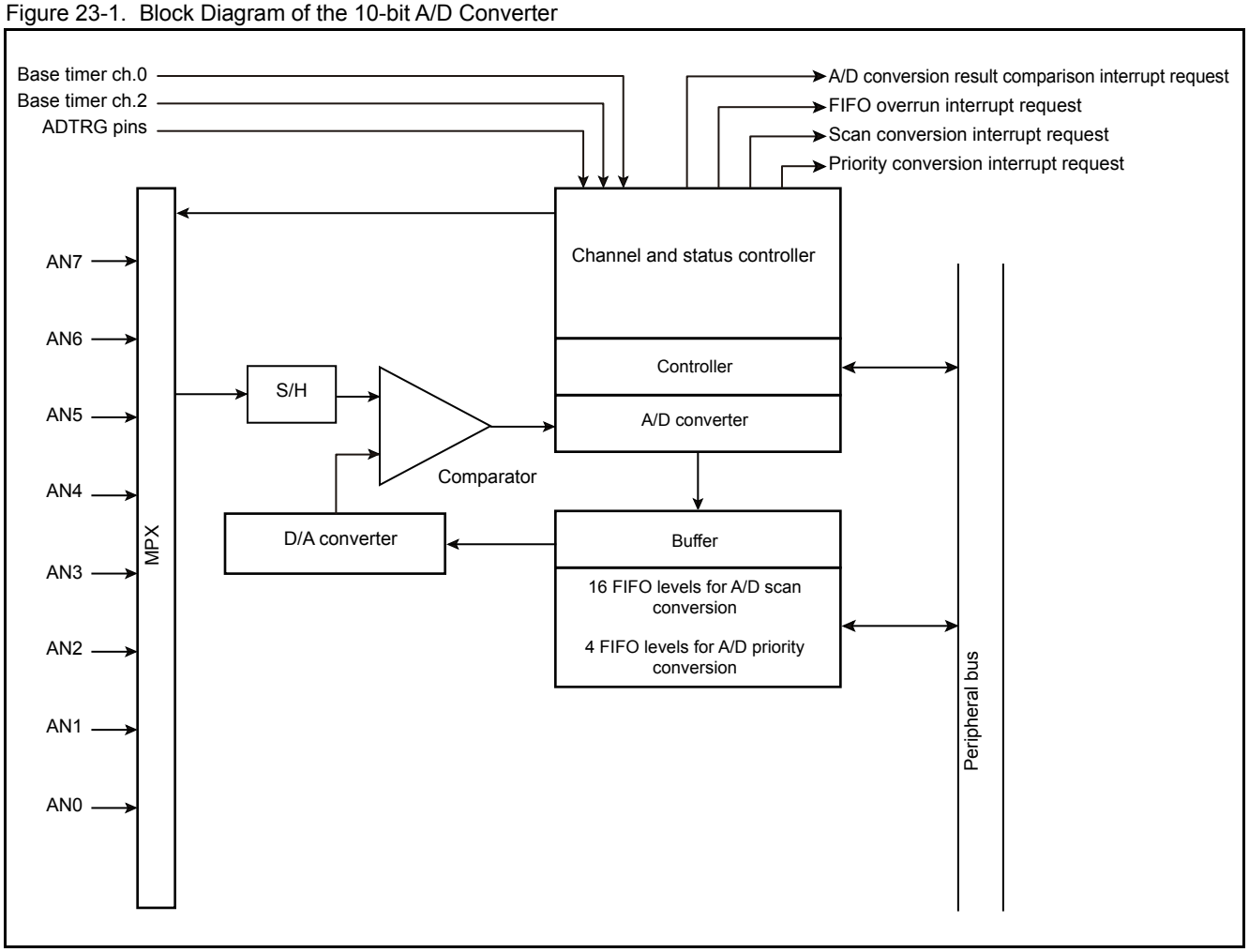
23.2 Configuration

This section explains the configuration of the 10-bit A/D converter.

Block diagram of the 10-bit A/D converter

Figure 23-1 shows a block diagram of the 10-bit A/D converter.

Figure 23-1. Block Diagram of the 10-bit A/D Converter



- A/D scan conversion FIFO
  This is the FIFO for A/D scan conversion. There are 16 FIFO levels.
- A/D priority conversion FIFO
  This is the FIFO for A/D priority conversion. There are four FIFO levels.
- Controller
  This controller controls conversion operations.
- Channel and status controller
  This controller controls the channels and status of the 10-bit A/D converter.
- MPX (analog multiplexer)
  The MPX selects (switches to), from multiple analog input signals, the analog signal to be converted.

Clocks

Table 23-1 lists the clocks used for the 10-bit A/D converter.

Table 23-1. Clock used for the 10-bit A/D Converter

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

23.3 Pins

This section explains the pins used for the 10-bit A/D converter.

Overview

The 10-bit A/D converter has the following pins:

- AV_{CC} pin

  10-bit A/D converter analog power input pin
- AVRH pin

  10-bit A/D converter reference voltage input pin
- AV_{SS} pin

  10-bit A/D converter GND pin
- AN0 to AN7 pins

  10-bit A/D converter analog input pins

  These pins are multiplexed pins. For details of using these pins as the AN0 to AN7 pins of the 10-bit A/D converter, see "13.4.6 A/D Channel Enable Register (ADCHE)".
- ADTRG pins

  10-bit A/D converter external trigger input pins

  These pins are multiplexed pins. For details of using these pins as the ADTRG pins of the 10-bit A/D converter, see "2.4 Setting Method for Pins".

Relationship between pins and channels

Table 23-2 shows the relationship between channels and pins.

Table 23-2. Relationship Between Channels and Pins

Channel	Analog Power Input Pin	Reference Voltage Input Pin	GND Pin	Analog Input Pin	External Trigger Input Pin
0	AV_{CC}	AVRH	AV_{SS}	AN0	ADTRG
1				AN1	
2				AN2	
3				AN3	
4				AN4	
5				AN5	
6				AN6	
7				AN7	

23.4 Registers

This section explains the configurations and functions of the registers used for the 10-bit A/D converter.

List of registers for the 10-bit A/D converter

Table 23-3 lists the registers used for the 10-bit A/D converter.

Table 23-3. Registers for the 10-bit A/D converter

Abbreviated Register Name	Register Name	Reference
ADCHE	A/D channel enable register	13.4.6
ADCR0	A/DC control register 0	23.4.1
ADSR0	A/DC status register 0	23.4.2
SCCR0	Scan conversion control register 0	23.4.3
SFNS0	Scan conversion FIFO number setting register 0	23.4.4
SCIS00	Scan conversion input select register 00	23.4.6
SCFD0	Scan conversion FIFO data register 0	23.4.5
PCCR0	Priority conversion control register 0	23.4.7
PFNS0	Priority conversion FIFO number setting register 0	23.4.8
PCIS0	Priority conversion input select register 0	23.4.10
PCFD0	Priority conversion FIFO data register 0	23.4.9
CMPD0	A/D comparison data setting register 0	23.4.11
CMPCR0	A/D comparison control register 0	23.4.12
ADSS00	Sampling time select register 00	23.4.14
ADST00	Sampling time setting register 00	23.4.13
ADST10	Sampling time setting register 10	23.4.13
ADCT0	Compare time setting register 0	23.4.15

23.4.1 A/DC Control Registers (ADCR0)

These registers control interrupt requests.

Figure 23-2 shows the bit configuration of the A/DC control registers (ADCR0).

Figure 23-2. Bit configuration of the A/DC control registers (ADCR0)

bit	7	6	5	4	3	2	1	0
	SCIF	PCIF	CMPIF	Undefined	SCIE	PCIE	CMPIE	OVRIE
Attribute	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	X	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit7]: SCIF (Scan conversion interrupt request flag bit)

This bit indicates that A/D scan conversion results have been stored in the number of stages in the FIFO as specified by the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).

If the SCIE bit is set to "1" when this bit is "1", a scan conversion interrupt request is generated.

SCIF	In Case of Reading	In Case of Writing
0	The number of stages storing conversion results has not reached the specified number of stages.	This bit is cleared to "0".
1	The number of stages storing conversion results has reached the specified number of stages.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit6]: PCIF (Priority conversion interrupt request flag bit)

This bit indicates that A/D priority conversion results have been stored up to the number of stages in the FIFO as specified by the PFS1 and PFS0 bits in a priority conversion FIFO number setting register (PFNS0).

If the PCIE bit is set to "1" when this bit is "1", a priority conversion interrupt request is generated.

PCIF	In Case of Reading	In Case of Writing
0	The number of stages storing conversion results has not reached the specified number of stages.	This bit is cleared to "0".
1	The number of stages storing conversion results has reached the specified number of stages.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit5]: CMPIF (Conversion result comparison interrupt request flag bit)

The A/D conversion result compare function is used when conversion results are compared with the data in the A/D comparison data setting registers (CMPD0).

This bit indicates that a conversion result satisfies the requirements set in an A/D comparison data setting register (CMPD0) and an A/D comparison control register (CMPCR0).

If the CMPIE bit is set to "1" when this bit is "1", a conversion result comparison interrupt request is generated.

CMPIF	In Case of Reading	In Case of Writing
0	The requirements are not satisfied.	This bit is cleared to "0".
1	The requirements are satisfied.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit4]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit3]: SCIE (Scan conversion interrupt enable bit)

This bit specifies whether to generate a scan conversion interrupt request when the number of stages storing A/D scan conversion results reaches the number of FIFO stages (SCIF bit = 1) specified in the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).

Written Value	Explanation
0	Disables generation of scan conversion interrupt requests.
1	Enables generation of scan conversion interrupt requests.

[bit2]: PCIE (Priority conversion interrupt enable bit)

This bit specifies whether to generate a priority conversion interrupt request when the number of stages storing A/D priority conversion results reaches the number of FIFO stages (PCIF bit = 1) specified in the PFS1 and PFS0 bits in a priority conversion FIFO number setting register (PFNS0).

Written Value	Explanation
0	Disables generation of priority conversion interrupt requests.
1	Enables generation of priority conversion interrupt requests.

[bit1]: CMPIE (Conversion result comparison interrupt enable bit)

The A/D conversion result compare function is used when conversion results are compared with the data in the A/D comparison data setting registers (CMPD0).

This bit specifies whether to generate a conversion result comparison interrupt request when a conversion result satisfies the requirements (CMPIF bit = 1) set in an A/D comparison control register (CMPCR0).

Written Value	Explanation
0	Disables generation of conversion result comparison interrupt requests.
1	Enables generation of conversion result comparison interrupt requests.

[bit0]: OVRIE (FIFO overrun interrupt enable bit)

This bit specifies whether to generate a FIFO overrun interrupt request when the SOVR bit in a scan conversion control register (SCCR0) or the POVR bit in a priority conversion control register (PCCR0) changes to "1".

If an attempt is made to write to a full FIFO, the SOVR bit in the scan conversion control register (SCCR0) or the POVR bit in the priority conversion control register (PCCR0) changes to "1".

Written Value	Explanation
0	Disables generation of FIFO overrun interrupt requests.
1	Enables generation of FIFO overrun interrupt requests.

23.4.2 A/DC Status Registers (ADSR0)

These registers indicate the A/D conversion status.

Figure 23-3 shows the bit configuration of the A/DC status registers (ADSR0).

Figure 23-3. Bit Configuration of the A/DC Status Registers (ADSR0)

bit	7	6	5	4	3	2	1	0
	ADSTP	FDAS	Undefined	Undefined	Undefined	PCNS	PCS	SCS
Attribute	R/W	R/W	-	-	-	R	R	R
Initial value	0	0	X	X	X	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

[bit7]: ADSTP (A/D conversion abort bit)

This bit forcibly stops A/D conversion.

ADSTP	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Forcibly stops A/D conversion.	

Notes:

- Writing "1" to this bit stops A/D conversion in either A/D scan conversion mode or A/D priority conversion mode.
- Writing "1" to this bit to forcibly stop A/D conversion clears the PCNS, PCS, and SCS bits to "0". However, it does not affect other registers.

[bit6]: FDAS (FIFO data allocation select bit)

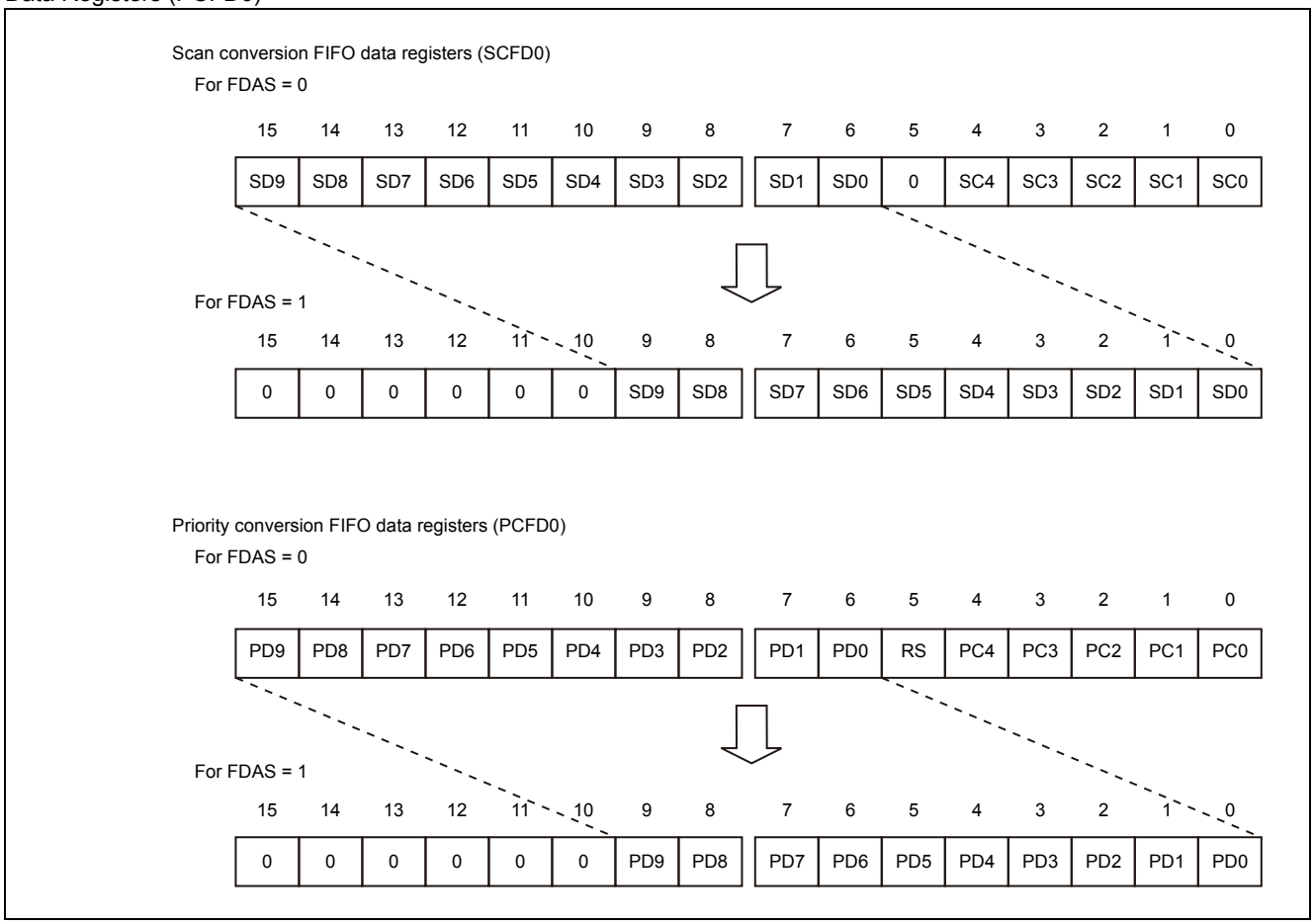
This bit specifies the mode of bit allocation to the scan conversion FIFO data registers (SCFD0) and priority conversion FIFO data registers (PCFD0).

- Left-justify: Conversion results (with channel information, with priority A/D activation trigger information (priority conversion only)) are left-justified.
- Right-justify: Conversion results (without channel information, without priority A/D activation trigger information (priority conversion only)) are shifted 6 bits to the LSB side to right-justify the results. Conversion results are allocated to bit9 to bit0.

Written Value	Explanation
0	Allocates conversion results left-justified.
1	Allocates conversion results right-justified.

Figure 23-4 shows the relationship between this bit and the scan conversion FIFO data registers (SCFD0) and the relationship between this bit and the priority conversion FIFO data registers (PCFD0).

Figure 23-4. Relationship Between FDAS and the Scan Conversion FIFO Data Registers (SCFD0)/Priority Conversion FIFO Data Registers (PCFD0)



Notes:

- If "1" is written to this bit to select right-justification, conversion results are shifted six bits to the LSB side, which consequently leads to a loss of converted information on channels (the SC4 to SC0 bits/PC4 to PC0 bits in). Right-justification is used only when channel information is not required in conversion results, such as when conversion involves only 1 channel.
- If "1" is written to this bit to select right-justification in A/D priority conversion mode, (the RS bit in Figure 23-4.) activation trigger information on A/D priority conversion is lost. Right-justification is used only when either priority 1 or 2 of A/D priority conversion mode is used.

[bit5 to bit3]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit2]: PCNS (Priority conversion pending flag bit)

This bit indicates that A/D priority conversion of priority 2 is pending.

If A/D priority conversion of priority 2 is activated during execution of A/D priority conversion of priority 1 or vice versa, the bit is changed to "1".

Read Value	Explanation
0	A/D priority conversion of priority 2 is not pending.
1	A/D priority conversion of priority 2 is pending.

[bit1]: PCS (Priority conversion status flag bit)

This bit indicates that A/D priority conversion of priority 1 or 2 is in progress.

Read Value	Explanation
0	A/D priority conversion is stopped.
1	A/D priority conversion is in progress.

[bit0]: SCS (Scan conversion status flag bit)

This bit indicates that A/D scan conversion is in progress.

Read Value	Explanation
0	A/D scan conversion is stopped.
1	A/D scan conversion is in progress.

23.4.3 Scan Conversion Control Registers (SCCR0)

These registers are used to control the operation of A/D scan conversion.

Figure 23-5 shows the bit configuration of the scan conversion control registers (SCCR0).

Figure 23-5. Bit Configuration of the Scan Conversion Control Registers (SCCR0)

bit	7	6	5	4	3	2	1	0
	SEMP	SFUL	SOVR	SFCLR	Undefined	RPT	SHEN	SSTR
Attribute	R	R	R/W	R/W	-	R/W	R/W	R/W
Initial value	1	0	0	0	X	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

Note: Do not perform word access to these registers.
The scan conversion FIFO data register (SCFD0) needs to be read when the SEMP bit is "0".

[bit7]: SEMP (Scan conversion FIFO empty flag bit)

This bit indicates that the FIFO for A/D scan conversion is empty.

Read Value	Explanation
0	The FIFO for A/D scan conversion contains data.
1	The FIFO for A/D scan conversion is empty.

This bit is cleared to "0" when data is stored in a scan conversion FIFO data register (SCFD0).

[bit6]: SFUL (Scan conversion FIFO full bit)

This bit indicates that the FIFO for A/D scan conversion is full.

Read Value	Explanation
0	The FIFO for A/D scan conversion has free space.
1	The FIFO for A/D scan conversion is full.

This bit is cleared to "0" when "1" is written to the SFCLR bit or a scan conversion FIFO data register (SCFD0) is read.

[bit5]: SOVR (Scan conversion overrun flag bit)

The bit indicates that an attempt has been made to write to a full A/D scan conversion FIFO (an overrun has occurred).

If the OVRIE bit in an A/DC control register (ADCR0) is set to "1" when this bit is "1", a FIFO overrun interrupt request is generated.

SOVR	In Case of Reading	In Case of Writing
0	No overrun occurred.	This bit is cleared to "0".
1	An overrun occurred.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- When an attempt is made to write data to a full FIFO, the conversion data in the FIFO is not overwritten.

[bit4]: SFCLR (Scan conversion FIFO clear bit)

This bit is used to clear the A/D scan conversion FIFO.

SFCLR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Clears the A/D scan conversion FIFO.	

Note: Writing "1" to this bit empties the FIFO for A/D scan conversion. Accordingly, the SEMP bit changes to "1".

[bit3]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit2]: RPT (Scan conversion repeat bit)

This bit specifies the A/D scan conversion mode.

- Single conversion mode: Signals from the channel specified by a scan conversion input select register (SCIS00) are converted only once.
- Repeat conversion mode: Signals from the channel specified by a scan conversion input select register (SCIS00) are converted repeatedly.

Written Value	Explanation
0	Single conversion mode
1	Repeat conversion mode

Notes:

- If "0" is written to this bit during conversion in repeat conversion mode, the conversion operation is stopped after the signals from the channel specified by the scan conversion input select register (SCIS00) are converted.
- To enable repeat conversion mode, write "1" to this bit after checking the SCS bit in the A/DC status registers (ADSR0) and confirming that A/D scan conversion is stopped
  (SCS = 0).

  However, to start A/D scan conversion (with SSTR = 1) while simultaneously enabling repeat conversion mode, the SSTR bit can be written at the same time as this bit.

[bit1]: SHEN (Scan conversion timer activation enable bit)

This bit specifies whether to activate A/D scan conversion upon detection of the rising edge of a TOUT signal of base timer ch.0.

Written Value	Explanation
0	Disables A/D scan conversion activation based on a base timer (ch.0).
1	Enables A/D scan conversion activation based on a base timer (ch.0).

Notes:

- If "1" is written to the SSTR bit, A/D scan conversion is activated regardless of the setting of this bit.
- After "1" is written to this bit, "1" may be written to the SSTR bit at the same time that activation is triggered by a base timer (ch.0). In this event, activation by software is given priority, and activation triggered by the base timer is ignored.
- For details of the TOUT signal, see "22. Base Timer".

[bit0]: SSTR (Scan conversion start bit)

This bit is used to activate A/D scan conversion by software.

Writing "1" to the bit during conversion stops and restarts the conversion.

SSTR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Activates or reactivates A/D scan conversion.	

23.4.4 Scan Conversion FIFO Number Setting Register (SFNS0)

These registers specify the maximum number of stages in the A/D scan conversion FIFO. A scan conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D scan conversion.

Figure 23-6 shows the bit configuration of the scan conversion FIFO number setting registers (SFNS0).

Figure 23-6. Bit Configuration of the Scan Conversion FIFO Number Setting Registers (SFNS0)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	SFS3	SFS2	SFS1	SFS0
Attribute	-	-	-	-	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

Note: Do not perform word access to these registers.

[bit7 to bit4]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit3 to bit0]: SFS3 to SFS0 (Scan conversion FIFO number setting bits)

These bits specify the maximum number of stages in the A/D scan conversion FIFO. A scan conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D scan conversion.

The SCIF bit in an A/DC control register (ADCR0) changes to "1" when the number of stages storing such data reaches the maximum number of FIFO stages specified by these bits.

SFS3	SFS2	SFS1	SFS0	Explanation
0	0	0	0	First level
0	0	0	1	Second level
0	0	1	0	Third level
0	0	1	1	Fourth level
0	1	0	0	Fifth level
0	1	0	1	Sixth level
0	1	1	0	Seventh level
0	1	1	1	Eighth level
1	0	0	0	Ninth level
1	0	0	1	Tenth level
1	0	1	0	Eleventh level
1	0	1	1	Twelfth level
1	1	0	0	Thirteenth level
1	1	0	1	Fourteenth level
1	1	1	0	Fifteenth level
1	1	1	1	Sixteenth level

23.4.5 Scan Conversion FIFO Data Registers (SCFD0)

These registers store A/D scan conversion results. Each register consists of 16 FIFO stages.

FIFO data can be read sequentially from the registers.

The bit configuration of these registers varies depending on the setting of the FDAS bit in the A/DC status registers (ADSR0).

Notes:

- One of these registers must always be read after the SEMP bit in a scan conversion control register (SCCR0) is checked to determine whether data remains in the A/D scan conversion FIFO (SEMP = 0).

  If this register is read when the A/D scan conversion FIFO is empty (SEMP = 1), it is impossible to determine whether the read data is valid. For details, see "Operation of A/D scan conversion" in "23.6.3 FIFO Operations".
- Do not perform word access to these registers.
- In byte access to these registers, the low-order byte (bit7 to bit0) must be accessed before the high-order byte (bit15 to bit8). FIFO data is shifted after the high-order bytes is read.

Left-justify (FDAS = 0)

Figure 23-7 shows the bit configuration of the scan conversion FIFO data registers (SCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies left-justification (FDAS = 0).

Figure 23-7. Bit Configuration of the Scan Conversion FIFO Data Registers (SCFD0)

	bit	15	14	13	12	11	10	9	8
		SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2
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
		SD1	SD0	Undefined	SC4	SC3	SC2	SC1	SC0
Attribute		R	R	-	R	R	R	R	R
Initial value		X	X	X	X	X	X	X	X

R: Read only

-: Undefined

X: Undefined

[bit15 to bit6]: SD9 to SD0 (A/D scan conversion result bits)

These bits store A/D scan conversion results.

[bit5]: Undefined bit

"0" is read.

[bit4 to bit0]: SC4 to SC0 (Conversion channel bits)

These bits indicate the channel from which analog input has been converted and stored in the SD9 to SD0 bits.

SC4	SC3	SC2	SC1	SC0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	Setting prohibited
0	1	0	0	1	
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	Setting prohibited
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

Right-justify (FDAS = 1)

Figure 23-8 shows the bit configuration of the scan conversion FIFO data registers (SCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies right-justification
(FDAS = 1).

Figure 23-8. Bit Configuration of the Scan Conversion FIFO Data Registers (SCFD0)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	SD9	SD8
Attribute	-	-	-	-	-	-	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

R: Read only

-: Undefined

X: Undefined

[bit15 to bit10]: Undefined bits

"0" is read.

[bit9 to bit0]: SD9 to SD0 (A/D scan conversion result bits)

These bits store A/D scan conversion results.

Note: Information on converted channels is not stored in right-justify mode. Right-justification is used only when channel information is not required in conversion results, such as when conversion involves only one channel.

23.4.6 Scan Conversion Input Select Registers (SCIS00)

These registers are used to select the channel to be subject to A/D scan conversion.

Figure 23-9 shows the bit configuration of the scan conversion input select registers (SCIS00).

Figure 23-9. Bit Configuration of the Scan Conversion Input Select Registers (SCIS00)

Scan conversion input select register 00 (SCIS00)

bit	7	6	5	4	3	2	1	0
	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
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

R/W: Read/Write

SCIS00: AN7 to AN0 (analog input select bits)

The channel corresponding to a bit that is set to "1" is made subject to conversion.

The AN7 bit corresponds to ch.7 (AN7 pin), the AN6 bit corresponds to ch.6 (AN6 pin), ... the AN1 bit corresponds to ch.1 (AN1 pin), and the AN0 bit corresponds to ch.0 (AN0 pin).

If multiple channels are selected with these registers, they are made subject to conversion sequentially in ascending order of channel number. For example, if "1" is written to the AN0, AN4, AN5, and AN7 bits, the corresponding channels are made subject to conversion in the following sequence:

ch.0 → ch.4 → ch.5 → ch.7

Note: Write to these registers while A/D conversion is stopped.

23.4.7 Priority Conversion Control Registers (PCCR0)

These registers are used to control the operation of A/D priority conversion. Two priority levels can be selected.

Figure 23-10 shows the bit configuration of the priority conversion control registers (PCCR0).

Figure 23-10. Bit Configuration of the Priority Conversion Control Registers (PCCR0)

bit	7	6	5	4	3	2	1	0
	PEMP	PFUL	POVR	PFCLR	Reserved	PEEN	PHEN	PSTR
Attribute	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

Note: Do not perform word access to these registers.
The priority conversion FIFO data register (PCFD0) needs to be read when the SEMP bit is "0".

[bit7]: PEMP (Priority conversion FIFO empty flag bit)

This bit indicates that the FIFO for A/D priority conversion is empty.

Read Value	Explanation
0	The FIFO for A/D priority conversion contains data.
1	The FIFO for A/D priority conversion is empty.

This bit is cleared to "0" when data is stored in the priority conversion FIFO data register (PCFD0).

[bit6]: PFUL (Priority conversion FIFO full bit)

This bit indicates that the FIFO for A/D priority conversion is full.

Read Value	Explanation
0	The A/D priority conversion FIFO has free space.
1	The A/D priority conversion FIFO is full.

This bit is cleared to "0" when "1" is written to the PFCLR bit or a priority conversion FIFO data register (PCFD0) is read.

[bit5]: POVR (Priority conversion overrun flag bit)

This bit indicates that an attempt has been made to write to a full A/D priority conversion FIFO (an overrun has been occurred).

If the OVRIE bit in an A/DC control register (ADCR0) is set to "1" when this bit is "1", a FIFO overrun interrupt request is generated.

POVR	In Case of Reading	In Case of Writing
0	Overrun has not been occurred	This bit is cleared to "0".
1	Overrun has been occurred	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- Even if an attempt is made to write data to a full FIFO, the conversion data in the FIFO is not overwritten.

[bit4]: PFCLR (Priority conversion FIFO clear bit)

This bit is used to clear the A/D priority conversion FIFO.

PFCLR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Clears the A/D priority conversion FIFO.	

Note: Writing "1" to this bit empties the FIFO for A/D priority conversion. Accordingly, the PEMP bit changes to "1".

[bit3]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2]: PEEN (Priority conversion external activation enable bit)

This bit specifies whether to activate A/D priority conversion of priority level 1 upon detection of a falling edge from the ADTRG pin. Priority 1 has the highest priority because priority 1 > priority 2.

Written Value	Explanation
0	Disables activation of A/D priority conversion of priority 1.
1	Enables activation of A/D priority conversion of priority 1.

[bit1]: PHEN (Priority conversion timer activation enable bit)

This bit specifies whether to activate A/D priority conversion of priority 2 upon detection of the rising edge of a TOUT signal of base timer ch.2. Priority 2 < priority 1.

Written Value	Explanation
0	Disables activation of A/D priority conversion of priority 2.
1	Enables activation of A/D priority conversion of priority 2.

Notes:

- If "1" is written to the PSTR bit, A/D priority conversion of priority 2 is activated regardless of the setting of this bit.
- For details of the TOUT signal, see "22. Base Timer".

[bit0]: PSTR (Priority conversion start bit)

This bit enables software to activate A/D priority conversion of priority 2. Priority 2 < priority 1.

PSTR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Activates A/D priority conversion of priority 2.	

Note: Even if "1" is written to this bit during A/D conversion, the A/D conversion cannot be reactivated.

23.4.8 Priority Conversion FIFO Number Setting Registers (PFNS0)

These registers specify the maximum number of stages in the A/D priority conversion FIFO. A priority conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D priority conversion.

Figure 23-11 shows the bit configuration of the priority conversion FIFO number setting registers (PFNS0).

Figure 23-11. Bit configuration of the Priority Conversion FIFO Number Setting Registers (PFNS0)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	PFS1	PFS0
Attribute	-	-	-	-	-	-	R/W	R/W
Initial value	X	X	X	X	X	X	0	0

R/W: Read/Write

-: Undefined

X: Undefined

Note: Do not perform word access to these registers.
The priority conversion FIFO data register (PCFD0) needs to be read when the PEMP bit is "0".

[bit7 to bit2]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit1, bit0]: PFS1, PFS0 (Priority conversion FIFO number setting bits)

These bits specify the maximum number of stages in the A/D priority conversion FIFO. A priority conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D priority conversion.

The PCIF bit in an A/DC control register (ADCR0) changes to "1" when the number of stages storing such data reaches the maximum number of FIFO stages specified by these bits.

PFS1	PFS0	Explanation
0	0	First level
0	1	Second level
1	0	Third level
1	1	Fourth level

23.4.9 Priority Conversion FIFO Data Registers (PCFD0)

These registers store A/D priority conversion results. Each register consists of 4 FIFO stages. FIFO data can be read sequentially from the registers.

The bit configuration of these registers varies depending on the setting of the FDAS bit in the A/DC status registers (ADSR0).

Notes:

- One of these registers must always be read after the PEMP bit in a priority conversion control register (PCCR0) is checked to determine whether data remains in the A/D priority conversion FIFO (PEMP = 0).

  If this register is read when the A/D priority conversion FIFO is empty (PEMP = 1), it is impossible to determine whether the read data is valid. For details, see "Operation of A/D priority conversion" in "23.6.3 FIFO Operations".
- Do not perform word access to these registers.
- In byte access to these registers, the low-order byte (bit7 to bit0) must be accessed before the high-order byte (bit15 to bit8). FIFO data is shifted after the high-order byte is read.

Left-justify (FDAS = 0)

Figure 23-12 shows the bit configuration of the priority conversion FIFO data registers (PCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies left-justification (FDAS = 0).

Figure 23-12. Bit configuration of the priority conversion FIFO data registers (PCFD0)

	bit	15	14	13	12	11	10	9	8
		PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2
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
		PD1	PD0	RS	PC4	PC3	PC2	PC1	PC0
Attribute		R	R	R	R	R	R	R	R
Initial value		X	X	X	X	X	X	X	X

R: Read only

X: Undefined

[bit15 to bit6]: PD9 to PD0 (A/D priority conversion result bits)

These bits store A/D priority conversion results.

[bit5]: RS (Priority A/D activation trigger bit)

This bit indicates whether the data stored in the PD9 to PD0 bits has been converted with priority 1 or 2 (activation trigger for A/D priority conversion).

Read Value	Explanation
0	Priority 2 (activation by software/base timer)
1	Priority 1 (activation by an external trigger)

Note: The activation trigger for A/D priority conversion of priority 2 cannot be distinguished as software or a base timer.

[bit4 to bit0]: PC4 to PC0 (Conversion channel bits)

These bits indicate the channel from which analog input has been converted and stored in the PD9 to PD0 bits.

PC4	PC3	PC2	PC1	PC0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	
0	1	0	0	1	
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	Setting prohibited
0	1	1	1	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	0	1	Setting prohibited
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

Right-justify (FDAS = 1)

Figure 23-13 shows the bit configuration of the priority conversion FIFO data registers (PCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies right-justification (FDAS = 1).

Figure 23-13. Bit Configuration of the Priority Conversion FIFO Data Registers (PCFD0)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	PD9	PD8
Attribute	-	-	-	-	-	-	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

R: Read only

-: Undefined

X: Undefined

[bit15 to bit10]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit9 to bit0]: PD9 to PD0 (A/D priority conversion result bits)

These bits store A/D priority conversion results.

Note: The activation trigger (priority) for A/D priority conversion and information on the converted channel are not stored in right-justify mode. Right-justification is used only when only either priority 1 or 2 of A/D priority conversion mode is used and when the converted result does not need the channel information, such as a conversion with 1 channel.

23.4.10 Priority Conversion Input Select Registers (PCIS0)

These registers are used to select the channel to be subject to A/D priority conversion.

Select 1 channel each from 8 channels; 1 channel subject to conversion with priority 2, and 1 channel subject to conversion with priority 1.

Figure 23-14 shows the bit configuration of the priority conversion input select registers (PCIS0).

Figure 23-14. Bit Configuration of the Priority Conversion Input Select Registers (PCIS0)

bit	7	6	5	4	3	2	1	0
	P2A4	P2A3	P2A2	P2A1	P2A0	P1A2	P1A1	P1A0
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

R/W: Read/Write

[bit7 to bit3]: P2A4 to P2A0 (Priority 2 analog input select bit)

These bits select the channel to be subject to A/D priority conversion of priority 2. Priority 2 < priority 1.

P2A4	P2A3	P2A2	P2A1	P2A0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	Setting prohibited
0	1	0	0	1	
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

[bit2 to bit0]: P1A2 to P1A0 (Priority 1 analog input select bit)

These bits select the channel to be subject to A/D priority conversion of priority 1. Priority 2 < priority 1.

P1A2	P1A1	P1A0	Explanation
0	0	0	ch.0 (AN0 pin)
0	0	1	ch.1 (AN1 pin)
0	1	0	ch.2 (AN2 pin)
0	1	1	ch.3 (AN3 pin)
1	0	0	ch.4 (AN4 pin)
1	0	1	ch.5 (AN5 pin)
1	1	0	ch.6 (AN6 pin)
1	1	1	ch.7 (AN7 pin)

23.4.11 A/D Comparison Data Setting Registers (CMPD0)

These registers are used to set the value that is compared with A/D conversion results when the comparison function is used. The eight high-order bits of the conversion results are compared with a value set in the registers. If the comparison result satisfies the requirements set in an A/D comparison control register (CMPCR0), the CMPIF bit in an A/DC control register (ADCR0) changes to "1".

Figure 23-15 shows the bit configuration of the A/D comparison data setting registers (CMPD0).

Figure 23-15. Bit Configuration of the A/D Comparison Data Setting Registers (CMPD0)

bit	7	6	5	4	3	2	1	0
	CMAD9	CMAD8	CMAD7	CMAD6	CMAD5	CMAD4	CMAD3	CMAD2
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

R/W: Read/Write

Note: A value that is set in these registers is compared with the eight high-order bits (bit9 to bit2) of A/D conversion results.The two bits (bit1, bit0) on the LSB side of A/D conversion results are not used for the comparison.

23.4.12 A/D Comparison Control Registers (CMPCR0)

These registers control the comparison function. The comparison function is used when an A/D conversion result is compared with the value set in an A/D comparison data setting register (CMPD0). If the comparison result satisfies the requirements set in that register, the CMPIF bit in an
A/DC control register (ADCR0) changes to "1".

Figure 23-16 shows the bit configuration of the A/D comparison control registers (CMPCR0).

Figure 23-16. Bit configuration of the A/D comparison control registers (CMPCR0)

bit	7	6	5	4	3	2	1	0
	CMPEN	CMD1	CMD0	CCH4	CCH3	CCH2	CCH1	CCH0
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

R/W: Read/Write

[bit7]: CMPEN (Comparison function operation enable bit)

This bit specifies whether to enable the comparison function.

Written Value	Explanation
0	Disables the comparison function.
1	Enables the comparison function.

[bit6]: CMD1 (Comparison mode 1 bit)

This bit sets the conversion interrupt request generation conditions.

Written Value	Explanation
0	A conversion result interrupt request is generated when the A/D conversion result is smaller than the value set in an A/D comparison data setting register (CMPD0).
1	A conversion result interrupt request is generated when the A/D conversion result is equal to or greater than the value set in an A/D comparison data setting register (CMPD0).

[bit5]: CMD0 (Comparison mode 0 bit)

This bit selects one of the following comparison modes:

- Comparing the conversion result of the channel specified by the CCH4 to CCH0 bits with the value set in an A/D comparison data setting register (CMPD0)
- Comparing the conversion results of all channels with the value set in an A/D comparison data setting register (CMPD0)

Written Value	Explanation
0	Compares the conversion result of the channel specified by the CCH4 to CCH0 bits.
1	Compares the conversion results of all channels.

Note: Writing "1" to this bit invalidates the settings of the CCH4 to CCH0 bits.

[bit4 to bit0]: CCH4 to CCH0 (Comparison target analog input channel bits)

These bits specify the channel to be compared with the value set in an A/D comparison data setting register (CMPD0) when the CMD0 bit is "0".

CCH4	CCH3	CCH2	CCH1	CCH0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	Setting prohibited
0	1	0	0	1	
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	Setting prohibited
1	1	1	1	1	

Note: If the CMD0 bit is set to "1" to compare the conversion results of all channels, the settings of these bits are ignored.

23.4.13 Sampling Time Setting Registers (ADST00, ADST10)

These registers specify the sampling time, that is, the period from the start of A/D conversion until the beginning of input voltage sampling when the sampled voltage is held in the sample-and-hold circuit. The A/D conversion time is the total of the sampling time and compare time.

2 ADST registers are provided to set the sampling time. After the sampling time is set in each register, the sampling time select registers (ADSS00) can be used to specify the register that has the set sampling time to be used for each channel.

Figure 23-17 shows the bit configuration of the sampling time setting registers (ADST00, ADST10).

Figure 23-17. Bit Configuration of the Sampling Time Setting Registers (ADST00, ADST10)

Sampling time setting registers 00 (ADST00)

bit	15	14	13	12	11	10	9	8
	STX01	STX00	ST05	ST04	ST03	ST02	ST01	ST00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	0	0

Sampling time setting registers 10 (ADST10)

bit	7	6	5	4	3	2	1	0
	STX11	STX10	ST15	ST14	ST13	ST12	ST11	ST10
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	0	0

R/W: Read/Write

Notes:

- Write to these registers while A/D conversion is stopped.
- For details of the sampling time, see "A/D conversion timeA/D conversion time" in "23.6 Explanation of Operations and Setting Procedure Examples".

- Sampling time setting registers 00 (ADST00)

  These registers set the first sampling time.

  [bit15, bit14]: STX01, STX00 (Sampling time Nx setting bit)

  These bits set a value (N) by which a value set in the ST05 to ST00 bits is multiplied.

STX01	STX00	Explanation
0	0	Multiplies the value by 1.
0	1	Multiplies the value by 4.
1	0	Multiplies the value by 8.
1	1	Multiplies the value by 16.

  [bit13 to bit8]: ST05 to ST00 (Sampling time setting bit)

  These bits set the value used to determine the sampling time.

  A value that is set in these bits is used to determine the sampling time based on the following formula:

  Sampling time = peripheral clock (PCLK) period × (ST + 1) × STX

  ST: Value set in the ST05 to ST00 bits

  STX: N (multiplier) set in the STX01 and STX00 bits

  Example: ST05 to ST00 = 9, STX01, STX00 = 01 (multiply by 4), peripheral clock (PCLK) = 20 MHz (50 ns)

  Sampling time = 50 ns × (9 + 1) × 4 = 2 μs

  Notes:

  - If "00" (multiply the setting value by 1) is set in the STX01 and STX00 bits, set "3" or a higher number in the ST05 to ST00 bits.
  - For details of the sampling time, see "A/D conversion timeA/D conversion time" in "23.6 Explanation of Operations and Setting Procedure Examples".
  - Sampling time setting registers 00 (ADST00) must be set such that the sampling time in the electrical characteristics is satisfied. For details of the electrical characteristics, see "Datasheet".

- Sampling time setting register 10 (ADST10)

  These registers set the second sampling time.

  [bit7, bit6]: STX11, STX10 (Sampling time Nx setting bit)

  These bits set a value (N) by which a value set in the ST15 to ST10 bits is multiplied.

STX11	STX10	Explanation
0	0	Multiplies the value by 1.
0	1	Multiplies the value by 4.
1	0	Multiplies the value by 8.
1	1	Multiplies the value by 16.

  [bit5 to bit0]: ST15 to ST10 (Sampling time setting bit)

  These bits set the value used to determine the sampling time.

  A value that is set in these bits is used to determine the sampling time based on the following formula:

  Sampling time = peripheral clock (PCLK) period × (ST + 1) × STX

  ST: Value set in the ST15 to ST10 bits

  STX: N (multiplier) set in the STX11 and STX10 bits

  Example: ST15 to ST10 = 9, STX11, STX10 = 01 (multiply by 4), peripheral clock (PCLK) = 20 MHz (50 ns)

  Sampling time = 50 ns × (9 + 1) × 4 = 2 μs

  Notes:

  - If "00" (multiply the setting value by 1) is set in the STX11 and STX10 bits, set "3" or a higher number in the ST15 to ST10 bits.
  - For details of the sampling time, see "A/D conversion timeA/D conversion time" in "23.6 Explanation of Operations and Setting Procedure Examples".
  - Sampling time setting registers 10 (ADST10) must be set such that the sampling time in the electrical characteristics is satisfied. For details of the electrical characteristics, see "Datasheet".

23.4.14 Sampling Time Select Registers (ADSS00)

These registers are used to select the A/D sampling time.

The sampling time to be used for each channel can be selected from that set in sampling time setting registers 00 (ADST00) or that set in sampling time setting registers 10 (ADST10).

Figure 23-18 shows the bit configuration of the sampling time select registers (ADSS00).

Figure 23-18. Bit configuration of the sampling time select registers (ADSS00)

Sampling time select registers 00 (ADSS00)

bit	7	6	5	4	3	2	1	0
	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
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

R/W: Read/Write

Note: Write to these registers while A/D conversion is stopped.

ADSS00: TS7 to TS0 (Sampling time selection bits)

These bits specify for each channel whether to use the sampling time that is set in one of sampling time setting registers 00 (ADST00) or sampling time setting registers 10 (ADST10).

Written Value	Explanation
0	Use a sampling time that is set in sampling time setting registers 00 (ADST00).
1	Use a sampling time that is set in sampling time setting registers 10 (ADST10).

The TS7 bit corresponds to ch.7 (AN7 pin), the TS6 bit corresponds to ch.6 (AN6 pin), ... the TS1 bit corresponds to ch.1 (AN1 pin), and the TS0 bit corresponds to ch.0 (AN0 pin).

23.4.15 Compare Time Setting Registers (ADCT0)

These registers set the compare time in the A/D conversion time. The A/D conversion time is the total of the sampling time and compare time.

Figure 23-19 shows the bit configuration of the compare time setting registers (ADCT0).

Figure 23-19. Bit configuration of the compare time setting registers (ADCT0)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	CT2	CT1	CT0
Attribute	-	-	-	-	-	R/W	R/W	R/W
Initial value	X	X	X	X	X	1	1	1

R/W: Read/Write

-: Undefined

X: Undefined

Note: Write to this register while A/D conversion is stopped.

[bit7 to bit3]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit2 to bit0]: CT2 to CT0 (Compare time setting bits)

These bits set the value used to determine the compare time.

A value that is set in these bits is used to determine the compare time based on the following formula:

Compare time = { (CT + 1) × 10 + 4} × peripheral clock (PCLK) period

CT: Value set in these bits

Example: CT = 1, peripheral clock (PCLK) = 20 MHz (50 ns)

Compare time = {(1 + 1) × 10 + 4} × 50 ns = 1.2 μs

Note: For details of the compare time, see "A/D conversion time" in "23.6 Explanation of Operations and Setting Procedure Examples",

23.5 Interrupts

An interrupt request can be generated in the following case(s):

- Data has been stored in the predetermined number of stages in the FIFO during A/D scan conversion. (Scan conversion interrupt request)
- Data has been stored in the predetermined number of stages in the FIFO during A/D priority conversion. (Priority conversion interrupt request)
- An attempt has been made to save the next conversion result to a full FIFO. (FIFO overrun interrupt request)
- The conversion result satisfies the interrupt request generation conditions when the comparison function is used. (Conversion result comparison interrupt request)

A/D scan conversion interrupt request

Table 23-4 outlines the interrupt requests of A/D scan conversion.

Table 23-4. Interrupt requests of A/D scan conversion

Interrupt Request	Interrupt Request Flag	Interrupt Request Enabled	Clearing of Interrupt Request
Scan conversion interrupt request	SCIF bit = 1 in an ADCR	SCIE bit = 1 in an ADCR	Write "0" to the SCIF bit in the ADCR.
FIFO overrun interrupt request	SOVR bit = 1 in an SCCR	OVRIE bit = 1 in an ADCR	Write "0" to the SOVR bit in the SCCR.
Conversion result comparison interrupt request	CMPIF bit = 1 in an ADCR	CMPIE bit = 1 in an ADCR	Write "0" to the CMPIF bit in the ADCR.

ADCR: A/DC control register (ADCR0)
SCCR: Scan conversion control register (SCCR0)

A/D priority conversion interrupt request

Table 23-5 outlines the interrupt requests of A/D priority conversion.

Table 23-5. Interrupt requests of A/D priority conversion

Interrupt Request	Interrupt Request Flag	Interrupt Request Enabled	Clearing of Interrupt Request
Priority conversion interrupt request	PCIF bit = 1 in an ADCR	PCIE bit = 1 in an ADCR	Write "0" to the PCIF bit in the ADCR.
FIFO overrun interrupt request	POVR bit = 1 in a PCCR	OVRIE bit = 1 in an ADCR	Write "0" to the POVR bit in the PCCR.
Conversion result comparison interrupt request	CMPIF bit = 1 in an ADCR	CMPIE bit = 1 in an ADCR	Write "0" to the CMPIF bit in the ADCR.

ADCR: A/DC control register (ADCR0)
PCCR: Priority conversion control register (PCCR0)

Notes:

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests:

  - Clear interrupt requests before enabling the generation of interrupt requests.
  - Clear interrupt requests simultaneously with interrupts enabled.
- For the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "10. Interrupt Controller".

Activating DMA transfer upon an interrupt

DMA transfer can be activated when one of the following interrupt requests is generated:

- Scan conversion interrupt request
- Priority conversion interrupt request

For details of DMA transfer, see "23.6.4 Activating the DMA Controller (DMAC)".

23.6 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the 10-bit A/D converter. Also, examples of procedures for setting the operating state are shown.

Overview

The 10-bit A/D converter enables A/D conversion by allowing analog signal input from the pin corresponding to each bit in the A/D channel enable register (ADCHE).

For details of the A/D channel enable register (ADCHE), see "13.4.6 A/D Channel Enable Register (ADCHE)".

The 10-bit A/D converter performs the following two types of conversion:

- A/D scan conversion

  Any selected channel is converted.

  Two conversion modes are available. One is single conversion mode in which the signals from the selected channel are converted only once, and the other is repeat conversion mode in which the signals from the selected channel are converted repeatedly.

- A/D priority conversion

  High-priority A/D conversion is performed soon after an activation trigger for the conversion is generated, because the trigger stops A/D scan conversion. The two priority levels are priority 1 and priority 2. Priority 1 > priority 2.

Table 23-6 summarizes the differences between A/D scan conversion and A/D priority conversion.

Table 23-6. Differences between A/D scan conversion and A/D priority conversion

	A/D Scan Conversion	A/D Priority Conversion	
		Priority 1	Priority 2
Supported channels	Up to 8 channels are selected arbitrarily from all 8 channels.	1 channel is selected from the 8 channels	1 channel is selected from the 8 channels.
Conversion activation trigger	Software Detection of a rising edge of the TOUT signal of base timer ch.0	Detection of a falling edge at the ADTRG pin	Software Detection of a rising edge of the TOUT signal of base timer ch.2
Restart	Enabled	Disabled	
FIFO	16 levels	4 levels	

Priority and state transition

Table 23-7 lists A/D conversion priorities.

Table 23-7. A/D conversion priority

Priority	A/D Conversion Type
1	A/D priority conversion of priority 1
2	A/D priority conversion of priority 2
3	A/D scan conversion

If A/D conversion of a different priority is activated while A/D conversion is already in progress, operations are performed as follows:

- The **A/D conversion** activated while **A/D conversion** is already in progress has a higher priority.

  The A/D conversion in progress is stopped and the A/D conversion of the higher priority is executed.

  After the higher-priority conversion is completed, the stopped A/D conversion is restarted.

  Example: An A/D priority conversion activation trigger is generated during A/D scan conversion

  The A/D scan conversion is interrupted, and A/D priority conversion begins. After the A/D priority conversion is completed, the A/D scan conversion resumes from the channel at which it interrupted.

  Example: An activation trigger for A/D priority conversion of priority 1 is generated during A/D priority conversion of priority 2.

  The A/D priority conversion of priority 2 is interrupted, and the A/D priority conversion of priority 1 begins. After the A/D priority conversion of priority 1 is completed, the A/D priority conversion of priority 2 resumes.

- The **A/D conversion activated** while **A/D conversion** is already in progress has a lower priority.

  The activation trigger for the A/D conversion of a lower priority is held, and the A/D conversion in progress is executed continuously.

  After the A/D conversion in progress is completed, the A/D conversion whose activation trigger has been held begins automatically.

  Example: An activation trigger for A/D priority conversion of priority 2 is generated during A/D priority conversion of priority 1.

  The activation trigger for the A/D priority conversion of priority 2 is held, and the A/D priority conversion of priority 1 is executed continuously.

  After the A/D priority conversion of priority 1 is completed, the A/D priority conversion of priority 2 begins automatically.

  Example: An activation trigger for A/D scan conversion is generated during A/D priority conversion of priority 1.

  The activation trigger for the A/D scan conversion is held, and the A/D priority conversion of priority 1 is executed continuously.

  After the A/D priority conversion of priority 1 is completed, the A/D scan conversion begins automatically.

  Example: An activation trigger for A/D scan conversion is generated during A/D priority conversion of priority 2.

  The activation trigger for the A/D scan conversion is held, and the A/D priority conversion of priority 2 is executed continuously.

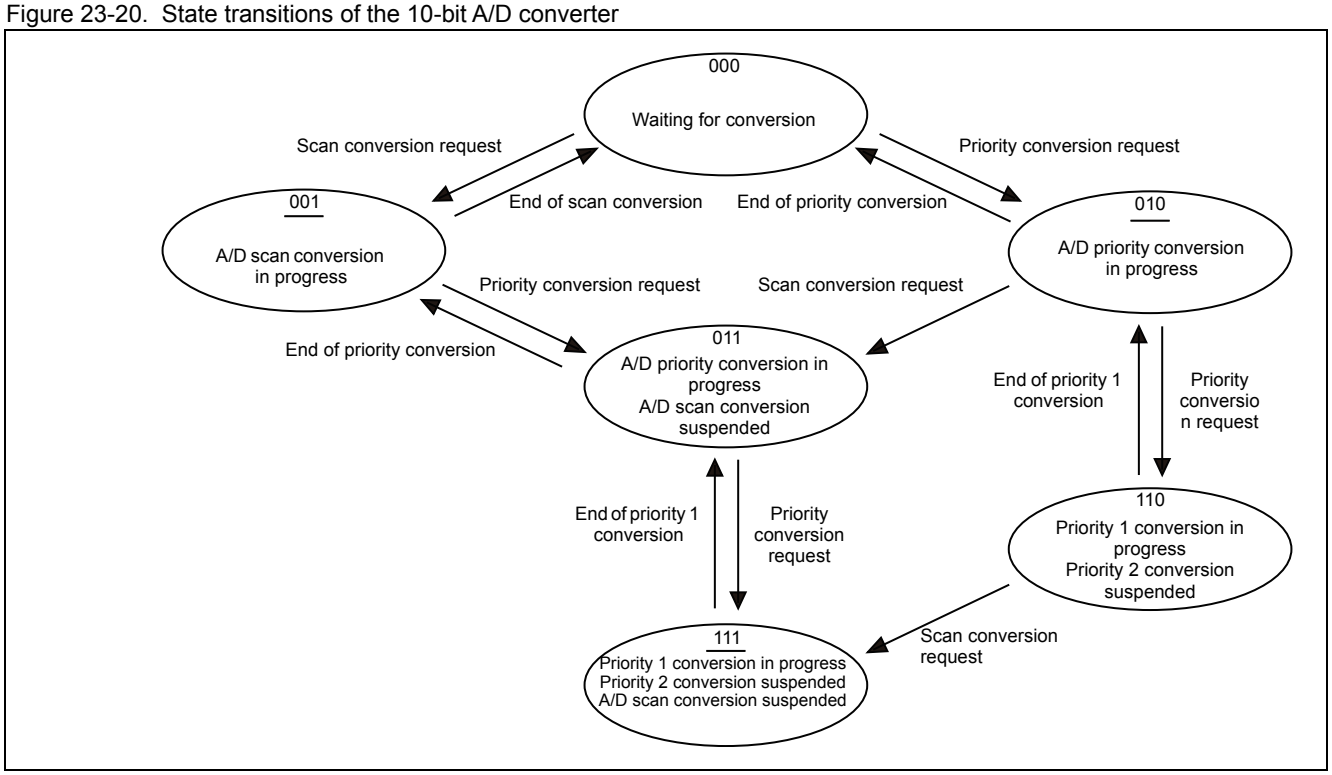
  After the A/D priority conversion of priority 2 is completed, the A/D scan conversion begins automatically.

- The **A/D conversion activated** during **A/D priority conversion** has the same priority.

  An activation trigger with the same priority is ignored. (The ignored activation trigger will not be reactivated.)

Figure 23-20 show state transitions of the 10-bit A/D converter.

Figure 23-20. State transitions of the 10-bit A/D converter



As shown in Figure 23-20, the states of the 10-bit A/D converter can be checked with the PCNS, PCS, and SCS bits in the A/DC status registers (ADSR0).

Table 23-8 shows the relationship between bits and operating states.

Table 23-8. Relationship between bits and operating states

PCNS	PCS	SCS	Explanation
0	0	0	Waiting for conversion
0	0	1	A/D scan conversion in progress
0	1	0	A/D priority conversion in progress
0	1	1	A/D priority conversion in progress, with A/D scan conversion suspended
1	1	0	Priority 1 A/D priority conversion in progress, with priority 2 conversion suspended
1	1	1	Priority 1 A/D priority conversion in progress, with priority 2 conversion and scan conversion suspended

Operation using the A/D comparison function

The A/D comparison function compares the eight high-order bits (bit9 to bit2) of A/D conversion results with a preset value in the A/D comparison data setting registers (CMPD0). If the comparison result satisfies the requirements set in an A/D comparison control register (CMPCR0), the function generates a conversion result comparison interrupt request.

The CMPEN bit must be set to "1" in the A/D comparison control register (CMPCR0) to enable the comparison function before conversion is started.

The comparison function can be used even with a full FIFO because a comparison is made before A/D conversion results are stored in the FIFO.

For details of the comparison function, see "23.4.11 A/D Comparison Data Setting Registers (CMPD0)", and "23.4.12 A/D Comparison Control Registers (CMPCR0)".

A/D conversion time

The A/D conversion time is the total of the sampling time and compare time.

To determine the A/D conversion time, add the sampling time and compare time.

- Sampling time

  The sampling time can be set in each of the sampling time setting registers (ADST00, ADST10).

  The sampling time select register (ADSS00) can be used to specify the register that has the set sampling time to be used for each channel. Therefore, the sampling time can be set individually for channels with different external impedances.

  The formula for calculating the sampling time is as follows:

  Sampling time = peripheral clock (PCLK) period $\times$ **(ST + 1)** $\times$ **STX**

  ST: Value that is set in the ST05 to ST00/ST15 to ST10 bits in a sampling time setting register (ADST00, ADST10)

  STX: Multiplier that is set in the STX01, STX00/STX11, and STX10 bits in a sampling time setting register (ADST00, ADST10)

  Notes:

  - If "00" (multiply the setting value by 1) is set in the STX01 and STX00 bits, set "3" or a higher number in the ST05 to ST00/ST15 to ST10 bits.
  - Sampling time setting registers 00 (ADST00) must be set so that the sampling time in the electrical characteristics is satisfied. For details of the electrical characteristics, see "Datasheet".

Table 23-9 and Table 23-10 show sampling time setting examples.

Table 23-9. Sampling time setting examples (for STX01, STX00/STX11, STX10 bits = 00)

Register value (N)	Sampling Time [µs]			Maximum External Impedance [kΩ]		
STx5 to STx0	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz
0	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
1	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
2	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
3	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
4	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
5	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
6	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
7	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
8	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
9	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
10	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
11	0.400	Setting prohibited	Setting prohibited	1.400	-	-
12	0.433	0.406	Setting prohibited	1.400	1.400	-
13	0.467	0.438	0.424	1.563	1.400	1.400
14	0.500	0.469	0.455	2.053	1.593	1.400
15	0.533	0.500	0.485	2.543	2.053	1.830
16	0.567	0.531	0.515	3.033	2.513	2.276
17	0.600	0.563	0.545	3.524	2.972	2.721
18	0.633	0.594	0.576	4.014	3.432	3.167
19	0.667	0.625	0.606	4.504	3.891	3.613
20	0.700	0.656	0.636	4.994	4.351	4.058
...	...	...	...	...	...	...
36	1.233	1.156	1.121	12.837	11.704	11.188
37	1.267	1.188	1.152	13.327	12.163	11.634
38	1.300	1.219	1.182	13.818	12.623	12.080
...	...	...	...	...	...	...
42	1.433	1.344	1.303	15.778	14.461	13.862
43	1.467	1.375	1.333	16.269	14.921	14.308
...	...	...	...	...	...	...
52	1.767	1.656	1.606	20.680	19.057	18.319
53	1.800	1.688	1.636	21.171	19.516	18.764
...	...	...	...	...	...	...
62	2.100	1.969	1.909	25.582	23.652	22.775
63	2.133	2.000	1.939	26.073	24.112	23.220

PCLK: Peripheral clock (PCLK) frequency

Table 23-10. Sampling time setting examples (for STX01, STX00/STX11, STX10 bits = 10)

Register Value (N)	Sampling Time [µs]			Maximum External Impedance [kΩ]		
STx5 to STx0	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz
0	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
1	0.533	0.500	0.485	2.543	2.053	1.830
2	0.800	0.750	0.727	6.465	5.729	5.395
3	1.067	1.000	0.970	10.386	9.406	8.960
4	1.333	1.250	1.212	14.308	13.082	12.525
5	1.600	1.500	1.455	18.229	16.759	16.090
6	1.867	1.750	1.697	22.151	20.435	19.655
7	2.133	2.000	1.939	26.073	24.112	23.220
8	2.400	2.250	2.182	29.994	27.788	26.786
9	2.667	2.500	2.424	33.916	31.465	30.351
10	2.933	2.750	2.667	37.837	35.141	33.916
11	3.200	3.000	2.909	41.759	38.818	37.481
12	3.467	3.250	3.152	45.680	42.494	41.046
13	3.733	3.500	3.394	49.602	46.171	44.611
14	4.000	3.750	3.636	53.524	49.847	48.176
15	4.267	4.000	3.879	57.445	53.524	51.741
16	4.533	4.250	4.121	61.367	57.200	55.306
17	4.800	4.500	4.364	65.288	60.876	58.871
18	5.067	4.750	4.606	69.210	64.553	62.436
19	5.333	5.000	4.848	73.131	68.229	66.001
20	5.600	5.250	5.091	77.053	71.906	69.566
...	...	...	...	...	...	...
36	9.867	9.250	8.970	139.798	130.729	126.607
37	10.133	9.500	9.212	143.720	134.406	130.172
38	10.400	9.750	9.455	147.641	138.082	133.737
...	...	...	...	...	...	...
42	11.467	10.750	10.424	163.327	152.788	147.998
43	11.733	11.000	10.667	167.249	156.465	151.563
...	...	...	...	...	...	...
52	14.133	13.250	12.848	202.543	189.553	183.648
53	14.400	13.500	13.091	206.465	193.229	187.213
...	...	...	...	...	...	...
62	16.800	15.750	15.273	241.759	226.318	219.299
63	17.067	16.000	15.515	245.680	229.994	222.864

PCLK: Peripheral clock (PCLK) frequency

- Compare time

  The compare time setting registers (ADCT0) specify the compare time.

  The formula for calculating the compare time is as follows:

  Compare time = {(CT + 1) × 10 + 4} × peripheral clock (PCLK) period

  CT: Value that is set in the CT2 to CT0 bits in a compare time setting register (ADCT0)

  shows a compare time setting example.

Table 23-11. Compare time setting example

Register Value (N) CT2 to CT0	Compare Time		
	PCLK = 30 MHz	PCLK = 32 MHz	PCLK = 33 MHz
0	Setting prohibited	Setting prohibited	Setting prohibited
1	0.80 µs	0.75 µs	0.73 µs
2	1.13 µs	1.06 µs	1.03 µs
3	1.47 µs	1.38 µs	1.33 µs
4	1.80 µs	1.69 µs	1.64 µs
5	2.13 µs	2.00 µs	1.94 µs
6	2.47 µs	2.31 µs	2.24 µs
7 (initial value)	2.80 µs	2.63 µs	2.55 µs

PCLK: Peripheral clock (PCLK) frequency

Note: This table covers only compare time data.

23.6.1 Operation of A/D Scan Conversion

Channels are selected by the scan conversion input select registers (SCIS00) sequentially.

Overview

A/D scan conversion is performed in one of the following two conversion modes:

- Single conversion mode

  The channel specified by a scan conversion input select register (SCIS00) is converted only once.
- Repeat conversion mode

  The channel specified by a scan conversion input select register (SCIS00) is converted repeatedly.

Also, the operation that is performed varies depending on whether only one channel is selected or multiple channels are selected with a scan conversion input select register (SCIS00).

Table 23-12 shows the order of conversion in each conversion mode.

Table 23-12. Conversion mode and order of conversion

Conversion Mode	Selected Channel	Conversion Order
Single conversion mode (RPT bit = 0 in an SCCR)	ch.0	ch.0 → conversion stopped
	ch.0, ch.4, ch.5, ch.7	ch.0 → ch.4 → ch.5 → ch.7 → conversion stopped
Repeat conversion mode (RPT bit = 1 in an SCCR)	ch.0	ch.0 → ch.0 → ch.0 → ch.0 ↑↓ ch.0 ← ch.0 ← ch.0 ← ch.0
	ch.0, ch.4, ch.5, ch.7	ch.0 → ch.4 → ch.5 → ch.7 ↑↓ ch.7 ← ch.5 ← ch.4 ← ch.0

SCCR: Scan Conversion Control Register (SCCR0)

Note: The 10-bit A/D converter enables A/D conversion by allowing analog signal input with the A/D channel enable register (ADCHE).
For details of the A/D channel enable register (ADCHE), see "13.4.6 A/D Channel Enable Register (ADCHE)".

Operation in single conversion mode

Writing "0" to the RPT bit in a scan conversion control register (SCCR0) sets single conversion mode.

In this mode, the channel specified by a scan conversion input select register (SCIS00) is converted only once.

- Activation

  The channel to be converted is selected with the scan conversion input select register (SCIS00) in one of the following ways, and the 10-bit A/D converter is then activated:

  - Write "1" to the SSTR bit in a scan conversion control register (SCCR0).
  - Set the SHEN bit in a scan conversion control register (SCCR0) to enable timer activation (SHEN = 1), and input the rising edge of a TOUT signal of base timer ch.0.

  If either of the above activation operations is performed during A/D scan conversion, the A/D scan conversion is immediately stopped/initialized, and the A/D scan conversion resumes later (reactivated).

- Single-channel conversion

  Only one channel to be converted is selected with a scan conversion input select register (SCIS00).

  When started, the 10-bit A/D converter activates the conversion operation for the selected channel, and the SCS bit in an A/DC status register (ADSR0) changes to "1".

  After the conversion of the selected channel is completed, the conversion result and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and the conversion operation is then stopped. Then, the SCS bit in the A/DC status register (ADSR0) is cleared to "0".

  The conversion results stored in the FIFO can be read from a scan conversion FIFO data register (SCFD0).

- Multichannel conversion

  Two or more channels to be converted are selected with a scan conversion input select register (SCIS00).

  When started, the 10-bit A/D converter begins converting the selected channels in ascending order of channel number. The SCS bit in an A/DC status register (ADSR0) then changes to "1".

  After the conversion of one channel is completed, the conversion results and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and the converter then begins converting the next channel.

  The channels not selected by the scan conversion input select register (SCIS00) remain unconverted.

  In the A/D scan conversion FIFO, the number of stages storing conversion results and information on each converted channel is changed every time the channel subject to conversion changes.

  The 10-bit A/D converter stops operating after converting on all the channels selected by the scan conversion input select register (SCIS00). Then, the SCS bit in the A/DC status register (ADSR0) is cleared to "0".

  The conversion results stored in the FIFO can be read sequentially from a scan conversion FIFO data register (SCFD0). For details of reading, see "Operation of A/D scan conversion" in "23.6.3 FIFO Operations".

Operation in repeat conversion mode

Writing "1" to the RPT bit in a scan conversion control register (SCCR0) enables single conversion mode.

In this mode, the channel specified by a scan conversion input select register (SCIS00) is converted repeatedly.

As in single conversion mode, to activate the 10-bit A/D converter, select a channel.

- Single-channel conversion

  Only one channel to be converted is selected with a scan conversion input select register (SCIS00).

  When started, the 10-bit A/D converter activates the conversion operation for the selected channel, and the SCS bit in an A/DC status register (ADSR0) changes to "1".

  After the conversion of the selected channel is completed, the conversion results and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and conversion of the same channel is then repeated.

  To stop conversion, write "0" to a scan conversion control register (SCCR0).

  The conversion results stored in the FIFO can be read sequentially from a scan conversion FIFO data register (SCFD0). For details of reading conversion results, see "Operation of A/D scan conversion" in "23.6.3 FIFO Operations".

- Multichannel conversion

  Two or more channels to be converted are selected with a scan conversion input select register (SCIS00).

  When activated, the 10-bit A/D converter begins converting on the selected channels sequentially in ascending order of channel number. The SCS bit in an A/DC status register (ADSR0) then changes to "1".

  After the conversion of one channel is completed, the conversion results and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and the converter then begins converting the next channel.

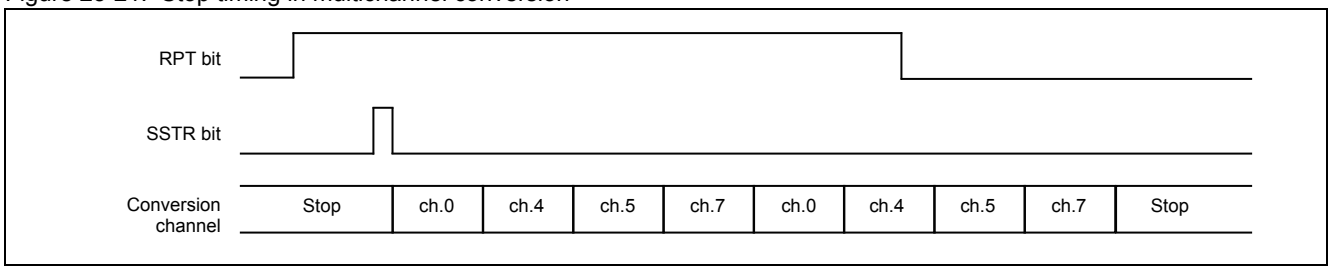
  The channels not selected by the scan conversion input select register (SCIS00) remain unconverted.

  After conversion of all the selected channels is completed, the second round of conversion begins for the channels sequentially in ascending order of channel number.

  To stop conversion, write "0" in the RPT bit of the scan conversion control register (SCCR0). Conversion stops when all channels selected by the scan conversion input select register (SCIS00) have been converted.

  Figure 23-21 shows the stop timing in multichannel conversion.

Figure 23-21. Stop timing in multichannel conversion



The conversion results stored in the FIFO can be read sequentially from a scan conversion FIFO data register (SCFD0). For details of reading, see "Operation of A/D scan conversion" in "23.6.3 FIFO Operations".

23.6.2 Operation of A/D Priority Conversion

High-priority A/D conversion is performed soon after an activation trigger for the conversion is generated, because the trigger stops A/D scan conversion. There are two priority levels.

Overview

Two priority levels can be selected depending on the activation trigger. Priority 1 has a higher priority than priority 2.

The channels that can be set vary depending on the priority.

Table 23-13 shows the relationship among priorities, channels, and activation triggers.

Table 23-13. Relationship among, priorities, channels, and activation triggers

	Priority 1	Priority 2
Priority	1	2
Supported channel	1 channel is selected from the 8 channels	1 channel is selected from the 8 channels.
Activation trigger	Detection of a falling edge at the ADTRG pin	Software Detection of a rising edge of the TOUT signal of base timer ch.2

Notes:

- The 10-bit A/D converter enables A/D conversion by allowing analog signal input with the A/D channel enable register (ADCHE).

  For details of the A/D channel enable register (ADCHE), see "13.4.6 A/D Channel Enable Register (ADCHE)".
- A/D conversion can be reactivated regardless of priority after A/D priority conversion.
- Only one channel can be converted in A/D priority conversion.

Conversion with priority 1

This conversion has the highest priority. When an activation trigger with priority 1 is generated, any A/D scan conversion or A/D priority conversion that is in progress is immediately stopped, and the conversion with priority 1 begins.

- Selecting a channel

  Only one channel is selected from the 8 channels and set in the P1A2 to P1A0 bits in a priority conversion input select register (PCIS0).
- Conversion

  An activation trigger for A/D priority conversion of priority 1 is generated when a falling edge is detected at the ADTRG pin after the PEEN bit in a priority conversion control register (PCCR0) is set to "1" to enable external activation.

  If A/D scan conversion or A/D priority conversion of priority 2 is in progress, it is immediately interrupted, and conversion of the specified channel with priority 1 begins. The PCS bit in an A/DC status register (ADSR0) then changes to "1".

  After the conversion is completed, the conversion results and information on the channel subject to conversion are stored in the FIFO for A/D priority conversion, and the PCS bit in the A/DC status register (ADSR0) is cleared to "0". The interrupted conversion is then restarted.

  The A/D priority conversion results stored in the FIFO can be read from a priority conversion FIFO data register (PCFD0). For details of reading, see "Operation of A/D priority conversion" in "23.6.3 FIFO Operations".

  For details of the operation performed when an activation trigger with a different priority is generated while A/D priority conversion of priority 1 is in progress, see "Priority and state transition" in "23.6 Explanation of Operations and Setting Procedure Examples".

Note: If an activation trigger for A/D conversion of the same level (priority 1) is generated while A/D priority conversion of priority 1 is in progress, the conversion in progress is continued and the new activation trigger is ignored.

Conversion with priority 2

This conversion has the second highest priority. When an activation trigger of priority 2 is generated, any A/D scan conversion in progress is immediately stopped, and the conversion with priority 2 begins.

- Selecting a channel

  Only one channel to be converted is selected from the 8 channels and set in the P2A4 to P2A0 bits in a priority conversion input select register (PCIS0).

- Conversion

  An activation trigger with priority 2 is generated in one of the following ways:

  - Write "1" to the PSTR bit in a priority conversion control register (PCCR0).
  - A rising edge of the TOUT signal of base timer ch.2 is detected after the PHEN bit in the priority conversion control register (PCCR0) is set to "1" to enable timer activation.

  When an activation trigger is generated, A/D priority conversion of priority 2 is activated and the PCS bit in an A/DC status register (ADSR0) changes to "1" as follows:

  - If the 10-bit A/D converter is not activated: The 10-bit A/D converter is activated to start conversion of the specified channel with priority 2.
  - If A/D scan conversion is in progress: The A/D scan conversion in progress is immediately interrupted, and conversion of the specified channel with priority 2 begins.
  - If A/D priority conversion of priority 1 is in progress: The activation trigger with priority 2 is held, and A/D priority conversion of priority 2 is started after A/D priority conversion of priority 1 is completed.

  After A/D priority conversion of priority 2 is completed, the conversion results and information on the channel subject to conversion are stored in the FIFO for A/D priority conversion, and the PCS bit in an A/DC status register (ADSR0) is cleared to "0". The interrupted conversion is then restarted.

  The A/D priority conversion results stored in the FIFO can be read from a priority conversion FIFO data register (PCFD0). For details of reading, see "Operation of A/D priority conversion" in "23.6.3 FIFO Operations".

  For details of the operation performed when an activation trigger with a different priority is generated while A/D priority conversion of priority 2 is in progress, see "Priority and state transition" in "23.6 Explanation of Operations and Setting Procedure Examples".

Note: No conversion operation can be reactivated during A/D priority conversion. If an activation trigger for A/D conversion of the same level (priority 2) is generated while A/D priority conversion of priority 2 is in progress, the conversion in progress is continued and the new activation trigger is ignored.
Example: The rising edge of a TOUT signal of base timer ch.2 may be detected after A/D priority conversion of priority 2 is activated by software. Even in this event, the conversion operation in progress is continued.

23.6.3 FIFO Operations

The 10-bit A/D converter provides 16 FIFO stages for A/D scan conversion and 4 FIFO levels for A/D priority conversion. A scan conversion interrupt request/priority conversion interrupt request can be generated when the number of stages storing the respective data reaches the predetermined number of FIFO stages.

This section explains FIFO operations and generation of interrupt requests.

Operation of A/D scan conversion

- Operation during A/D conversion

  The SEMP bit in a scan conversion control register (SCCR0) is "1", because the FIFO for A/D scan conversion contains no data (empty) after a reset is released.

  The SEMP bit changes to "0" when A/D scan conversion begins, and conversion results for 1 channel are stored in the first FIFO stage.

  After conversion of the next data is completed, the conversion results are stored in the second FIFO stage. Every time that conversion of 1 channel is completed after that, the conversion results are stored in the subsequent FIFO stage.

  After conversion results are written to all 16 FIFO levels, the A/D scan conversion FIFO becomes full and the SFUL bit in the scan conversion control register (SCCR0) changes to "1".

  If A/D scan conversion is performed again in this state, an overrun occurs and the SOVR bit in the scan conversion control register (SCCR0) changes to "1". In this event, the conversion results are not stored in the FIFO but abandoned.

- Read operation

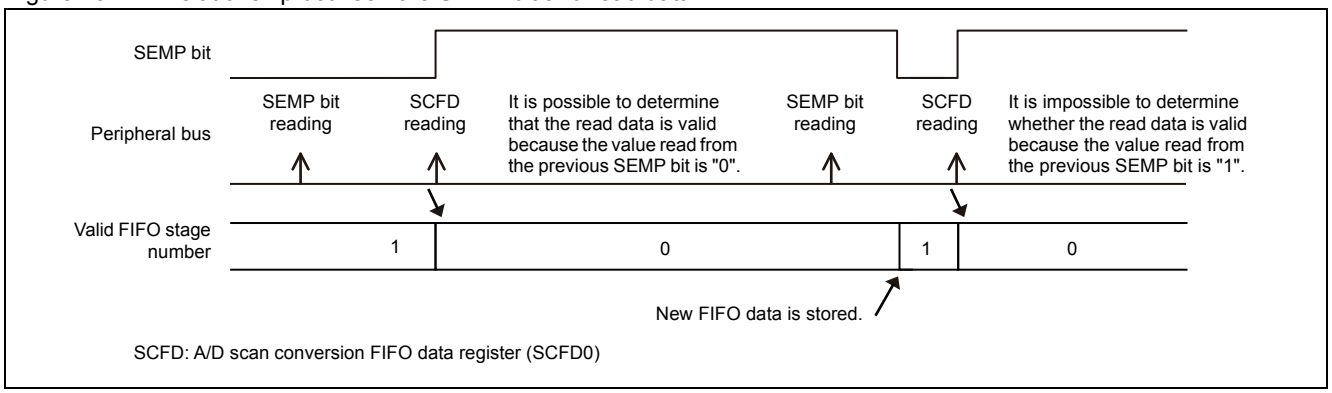
  Data stored in the A/D scan conversion FIFO can be read sequentially through reading with a scan conversion FIFO data register (SCFD0).

  A scan conversion FIFO data register (SCFD0) must always be read after the SEMP bit in a scan conversion control register (SCCR0) is checked to determine whether data remains in the A/D scan conversion FIFO (SEMP = 0).

  If an empty A/D scan conversion FIFO is read (SEMP = 1), it is difficult to determine whether the read data is valid, and valid data may be abandoned. (This is because conversion results may be stored in a scan conversion FIFO data register (SCFD0) immediately before the FIFO is read.)

  Figure 23-22 shows the relationship between the SEMP bit and read data.

Figure 23-22. Relationship between the SEMP bit and read data



Notes:

- The registers listed below are located at adjacent addresses. If these registers are accessed at one time by word access, the registers are read regardless of the setting of the SEMP bit in a scan conversion control register (SCCR0). Do not execute word access to these registers.
  - Scan conversion control register (SCCR0)
  - Scan conversion FIFO number setting register (SFNS0)
  - Scan conversion FIFO data register (SCFD0)
- The scan conversion FIFO data registers (SCFD0) can be byte-accessed. FIFO data is shifted after the high-order byte (bit15 to bit8) is read. FIFO data will not be shifted by reading of the lower-order byte (bit7 to bit0).

- Clear operation

  Writing "1" to the SFCLR bit in a scan conversion control register (SCCR0) clears the FIFO for A/D scanning conversion and changes the SEMP bit in the scan conversion control register (SCCR0) to "1".

- Scan conversion interrupt request

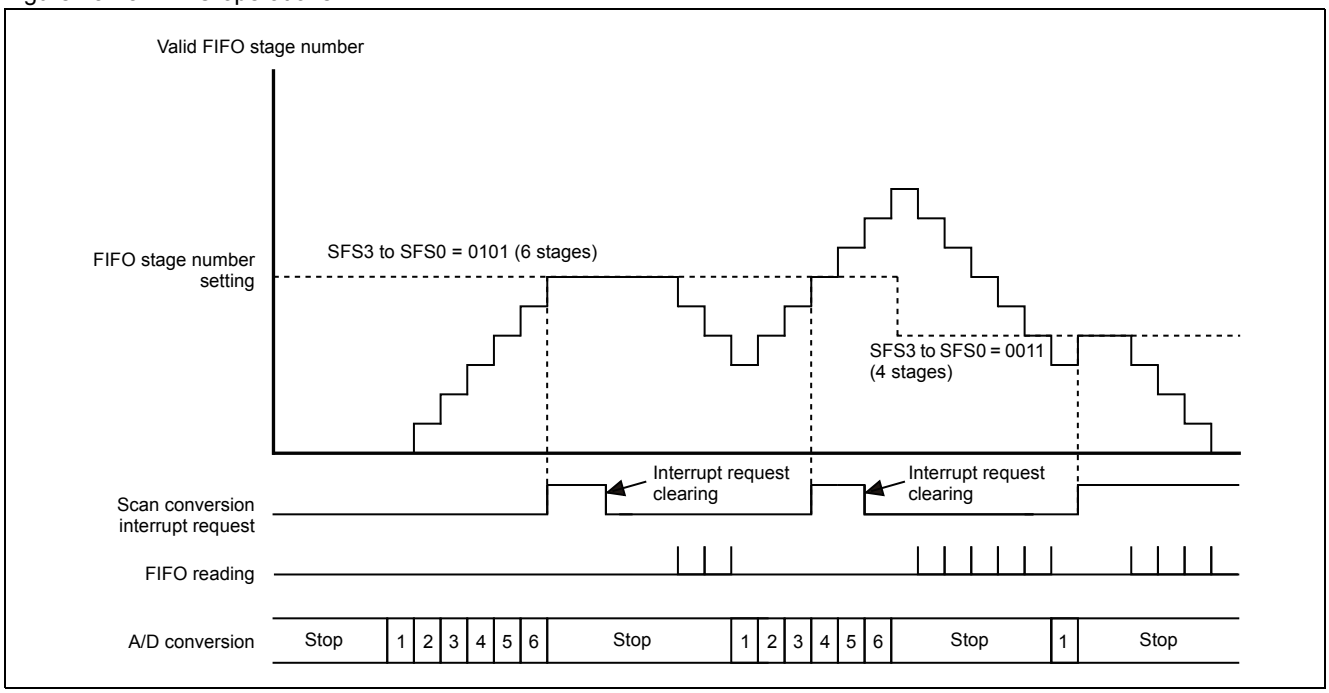
  A scan conversion interrupt request can be generated when the number of stages storing conversion results reaches the specified number of FIFO stages (SCIF bit = 1 in an A/DC control register (ADCR0)).

  To generate an A/D scan conversion interrupt request, perform the following processing:

  - Decide the number of FIFO stages at which an interrupt request is generated, and set the number in the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).
  - Set the SCIE bit in an A/DC control register (ADCR0) to "1" to enable generation of scan conversion interrupt requests.

  Figure 23-23 shows FIFO operations.

Figure 23-23. FIFO operations



The interrupt request generation examples shown below indicate the number of FIFO stages that is set for each conversion mode. The number of FIFO stages must be set in the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).

- **Single channel conversion in single conversion mode**

  If the number of FIFO stages at which a scan conversion interrupt request is generated is set at 1 (SFS3 to SFS0 = 0000), a scan conversion interrupt request is generated when conversion is completed. If the number of FIFO stages is set at 2 or more (SFS3 to SFS0 = 0001 or higher), an interrupt request is not generated even after conversion of the specified channel is completed.

- **Multichannel conversion in single conversion mode**

  If the set number of FIFO stages is the same as the number of channels to be subject to conversion, a scan conversion interrupt request is generated at the end of conversion.

  Example: Generation of a scan conversion interrupt request after conversion of 3 channels

  Set 3 (SFS3 to SFS0 = 0010) for the number of FIFO stages at which a scan conversion interrupt request is generated.

  Settings can be made such that a scan conversion interrupt request is generated at a number of FIFO stages that is less than the number of the channels to be subject to conversion. In such cases, a scan conversion interrupt request can be generated at any time before the end of A/D scan conversion.

- **Single channel conversion in repeat conversion mode**

  If the number of FIFO stages at which a scan conversion interrupt request is generated is set at 1 (SFS3 to SFS0 = 0000), a scan conversion interrupt request is generated when the first round of conversion is completed.

  To generate a scan conversion interrupt request after converting on the specified channel a certain number of times, match the conversion count to the number of FIFO stages.

  Example: Generation of a scan conversion interrupt request after conversion on a single channel is performed 4 times

  Set 4 (SFS3 to SFS0 = 0011) for the number of FIFO stages at which a scan conversion interrupt request is generated.

- **Multichannel conversion in repeat conversion mode**

  The desired generation time for scan conversion interrupt requests can be selected as shown below.

  Example: Conversion of 8 channels in repeat conversion mode

  1. Generate a scan conversion interrupt request after the end of the first round of conversion.
     Set 8 (SFS3 to SFS0 = 0111) for the number of FIFO stages at which a scan conversion interrupt request is generated.
  2. Generate an interrupt request after the end of the second round of conversion.
     Set 16 (twice the number of channels to be subject to conversion) (SFS3 to SFS0 = 1111) for the number of FIFO stages at which a scan conversion interrupt request is generated.

Note: DMA transfer of the data in the FIFO can be performed when a scan conversion interrupt request is generated. For details of DMA transfer, see "23.6.4 Activating the DMA Controller (DMAC)".

- FIFO overrun interrupt request

When data has been stored in all 16 FIFO levels and the FIFO becomes full, the SFUL bit in a scan conversion control register (SCCR0) changes to "1".

Also, the OVRIE bit in an A/DC control register (ADCR0) can be set to enable generation of FIFO overrun interrupt requests (OVRIE = 1). In this state, if an attempt is made to store the next conversion result in the FIFO when the SFUL bit is "1", an overrun interrupt is generated.

Notes:

- The data in the FIFO cannot be rewritten, even by the attempt to store the next conversion result in the full FIFO. The conversion result to be stored in this attempt is abandoned.
- The FIFO is emptied and the SEMP bit in a scan conversion control register (SCCR0) changes to "1" when the SFCLR bit in the scan conversion control register (SCCR0) is set to "1" to clear the FIFO.

Operation of A/D priority conversion

- Operation during A/D conversion

  The PEMP bit in an A/D priority conversion control register (PCCR0) is "1", because the FIFO for A/D scan conversion contains no data (empty) after a reset is released.

  The PEMP bit changes to "0" when A/D priority conversion begins, and conversion results for 1 channel are stored in the first FIFO stage.

  After the next A/D priority conversion is completed, the conversion results are stored in the second FIFO level. Every time that A/D priority conversion is completed after that, the conversion results are stored in the subsequent FIFO stage.

  After conversion results are written to all 4 FIFO levels, the FIFO for A/D priority conversion becomes full and the PFUL bit in the priority conversion control register (PCCR0) changes to "1".

  If A/D priority conversion is performed again in this state, an overrun occurs and the POVR bit in the priority conversion control register (PCCR0) changes to "1". In this event, the conversion results are not stored in the FIFO but abandoned.

- Read operation

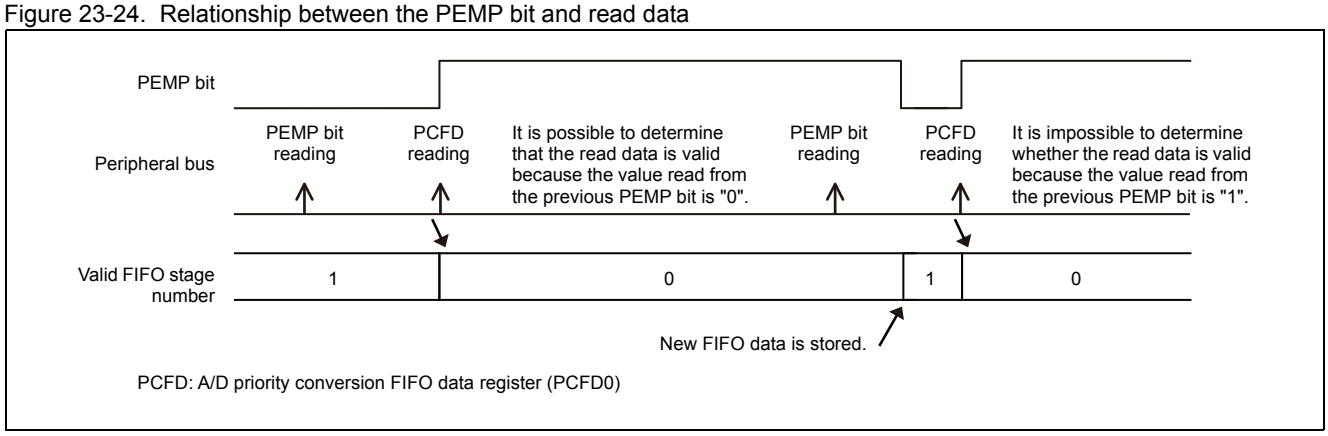
  Data stored in the A/D priority conversion FIFO can be read sequentially through reading with a priority conversion FIFO data register (PCFD0).

  A priority conversion FIFO data register (PCFD0) must always be read after the PEMP bit in a priority conversion control register (PCCR0) is checked to determine whether data remains in the A/D priority conversion FIFO (PEMP = 0).

  If an empty A/D priority conversion FIFO is read (PEMP = 1), it is difficult to determine whether the read data is valid, and valid data may be abandoned. (This is because conversion results may be stored in a priority conversion FIFO data register (PCFD0) immediately before the FIFO is read.)

  Figure 23-24 shows the relationship between the PEMP bit and read data.

Figure 23-24. Relationship between the PEMP bit and read data



Notes:

- The registers listed below are located at adjacent addresses. If these registers are accessed at one time by word access, the registers are read regardless of the setting of the PEMP bit in a priority conversion control register (PCCR0). Do not execute word access to these registers.
  - Priority conversion control register (PCCR0)
  - Priority conversion FIFO number setting register (PFNS0)
  - Priority conversion FIFO data register (PCFD0)
- The priority conversion FIFO data registers (PCFD0) can be byte accessed. FIFO data is shifted after the high-order byte (bit15 to bit8) is read. FIFO data will not be shifted by reading of the low-order byte (bit7 to bit0).

- Clear operation

  Writing "1" to the PFCLR bit in a priority conversion control register (PCCR0) clears the A/D priority conversion FIFO and changes the PEMP bit in an A/D priority conversion control register (PCCR0).

- Priority conversion interrupt request

  A priority conversion interrupt request can be generated when the number of stages storing conversion results reaches the specified number of FIFO stages (PCIF bit = 1 in an A/DC control register (ADCR0)).

  To generate an A/D priority conversion interrupt request, perform the following processing:

  - Decide the number of FIFO stages at which an interrupt request is generated, and set the number in the PFS1 and PFS0 bits in the priority conversion FIFO number setting register (PFNS0).
  - Set the PCIE bit in the A/DC control register (ADCR0) to "1" to enable generation of priority conversion interrupt requests.

  If the number of FIFO stages at which a priority conversion interrupt request is generated is set to "1" (PFS1, PFS0 = 00), a priority conversion interrupt request is generated when the conversion is completed.

  Notes:

  - If the number of FIFO stages at which a priority interrupt request is generated is set at 2 or more (PFS1, PFS0 = 01 or higher), no priority conversion interrupt request is generated even after A/D priority conversion is completed.
  - DMA transfer of the data in the FIFO can be performed when a priority conversion interrupt request is generated. For details of DMA transfer, see "23.6.4 Activating the DMA Controller (DMAC)".

- FIFO overrun interrupt request

  The PFUL bit in a priority conversion control register (PCCR0) changes to "1" when data has been stored in all of 4 FIFO levels and the FIFO becomes full.

  Also, the OVRIE bit in an A/DC control register (ADCR0) can be set to enable generation of FIFO overrun interrupt requests (OVRIE = 1). In this state, if an attempt is made to store the next conversion result in FIFO when the PFUL bit is "1", an overrun interrupt is generated.

  Notes:

  - The data in the FIFO cannot be rewritten, even by the attempt to store the next conversion result in the full FIFO. The conversion result to be stored in this attempt is abandoned.
  - The FIFO is emptied and the PEMP bit in a priority conversion control register (PCCR0) changes to "1" when the PFCLR bit in the priority conversion control register (PCCR0) is set to "1" to clear the FIFO.

23.6.4 Activating the DMA Controller (DMAC)

DMA transfer of FIFO data is possible through scan conversion interrupt requests and priority conversion interrupt requests generated by the 10-bit A/D converter.

If the same value is set for the number of FIFO stages at which a scan conversion interrupt request/priority conversion interrupt request is generated and the byte number for DMA transfer, DMA transfer of FIFO data can be performed in synchronization with A/D scan conversion. For details of setting the byte number for DMA transfer, see "31. DMA Controller (DMAC)".

- In single conversion mode

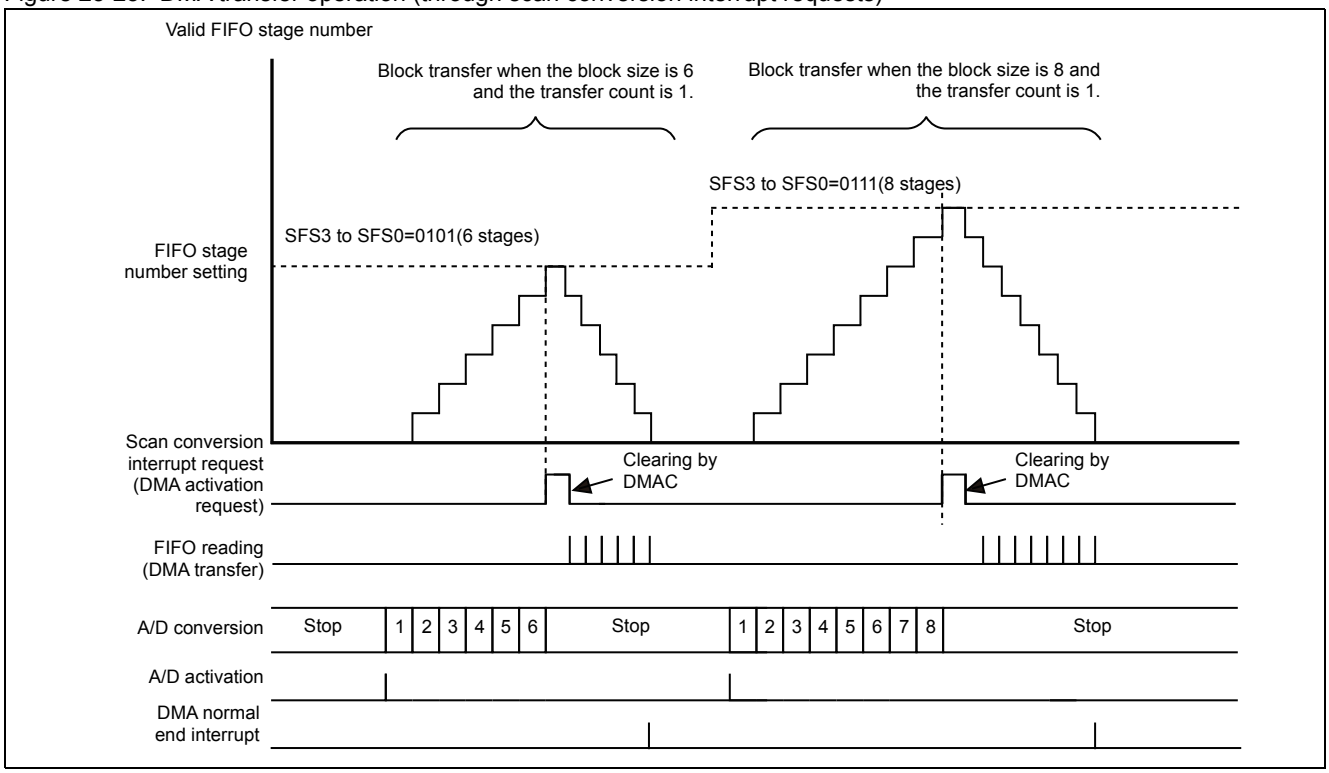
  To perform DMA transfer, set the same value to the DMA block size and the interrupt generation FIFO stage number, and perform the next A/D activation after DMA is completed.

- In repeat conversion mode

  To perform DMA transfer, set 1 to the DMA block size, and 1 for the interrupt generation FIFO stage number.

Figure 23-25 shows the DMA transfer operation.

Figure 23-25. DMA transfer operation (through scan conversion interrupt requests)



Notes:

- Set the same value to the DMA block size and the interrupt generation FIFO stage number.
- Perform the next A/D activation after performing DMA transfer of all FIFO data.

However, note that the event described below may occur while A/D conversion is repeated, such as in repeat conversion mode. In such cases, even after DMA transfer of data by the specified number of the bytes, the FIFO may still store more data corresponding to stages exceeding the number of stages at which a scan conversion interrupt request/priority conversion interrupt request is generated.

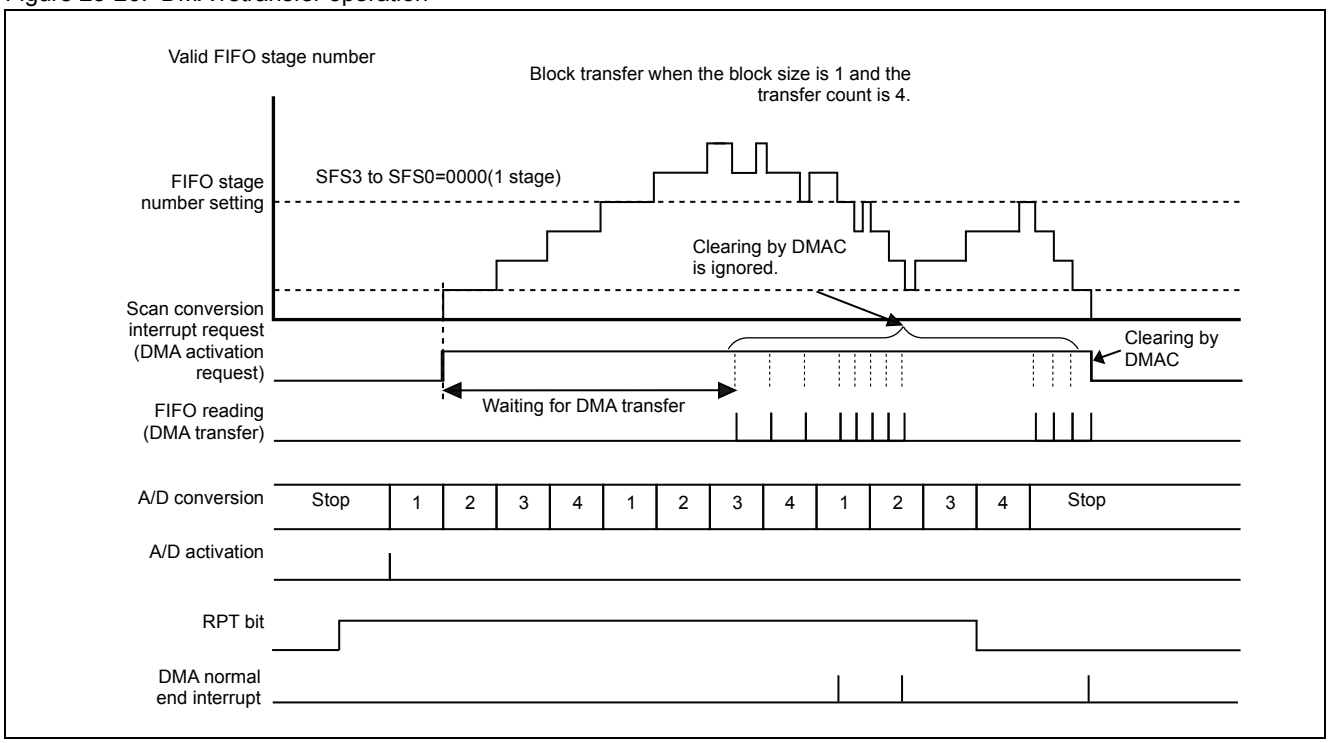
- A/D conversion of the signals from the next channel begins before DMA transfer of conversion results is completed.

  (Examples are when another DMA transfer is activated and DMA transfer of conversion results in progress is made to wait)

For this reason, if the FIFO is storing more data corresponding to stages exceeding the number of stages at which an interrupt request is generated, a clear operation by the DMA controller (DMAC) is ignored and DMA transfer is performed again.

Figure 23-26 shows the DMA retransfer operation.

Figure 23-26. DMA retransfer operation



Note: Set 1 to block size of DMA, and 1 for the interrupt generation FIFO stage number.

24. Multi-function Serial Interface

This chapter describes the functions and operations of the multi-function serial interface.

24.1 Characteristics of Multi-function Serial Interface

24.2 UART (Asynchronous Serial Interface)

24.3 Overview of UART (Asynchronous Serial Interface)

24.4 Registers of UART (Asynchronous Serial Interface)

24.5 Interrupts of UART

24.6 Operation of UART

24.7 Dedicated Baud Rate Generator

24.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)

24.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

24.10 Notes on UART Mode

24.11 CSIO (Clock Synchronous Serial Interface)

24.12 Overview of CSIO (Clock Synchronous Serial Interface)

24.13 Registers of CSIO (Clock Synchronous Serial Interface)

24.14 Interrupts of CSIO (Clock Synchronous Serial Interface)

24.15 Operation of CSIO (Clock Synchronous Serial Interface)

24.16 Dedicated Baud Rate Generator

24.17 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

24.18 Notes on CSIO Mode

24.19 I^2C Interface

24.20 Overview of I^2C Interface

24.21 Registers of I^2C Interface

24.22 Interrupts of I^2C Interface

24.23 Dedicated Baud Rate Generator

24.24 Notes on I^2C Mode

24.1 Characteristics of Multi-function Serial Interface

This multi-function serial interface has the following characteristics.

Interface mode

The following interface modes are selectable for the multi-function serial interface depending on the operation mode settings.

- UART0 (Asynchronous normal serial interface)
- UART1 (Asynchronous multi-processor serial interface)
- CSIO (Clock synchronous serial interface) (SPI can be supported)
- I^2C (I^2C bus interface)

Switching the interface mode

To communicate through each serial interface, the serial mode registers (SMR) shown in Table 24-1 should be used to set the operation mode before starting the communication.

Table 24-1. Switching Interface Mode

MD2	MD1	MD0	Interface mode
0	0	0	UART0 (Asynchronous normal serial interface)
0	0	1	UART1 (Asynchronous multi-processor serial interface)
0	1	0	CSIO (Clock synchronization serial interface) (SPI can be supported)
1	0	0	I^2C (I^2C bus interface)

Note: Settings other than above are prohibited.

Notes:

- Transmission and reception cannot be guaranteed when the operation mode is switched while one of the serial interfaces is still in use for transmission or reception operation.
- The operation mode must be set first. Otherwise, the part of registers of the same channel will be initialized when the operation mode is changed. For the registers to be initialized, see the notes for serial mode register (SMR) of each operation mode.

Number of channels

This product has 8 built-in channels for multi-function serial interface. There is no I^2C function for ch.0.

Transmission/reception FIFO

This UART has a 16-byte transmission FIFO and 16-byte reception FIFO. The FIFO steps should be converted to 16 bytes when reading through this text.

There is no FIFO between ch.0 and ch.3.

24.2 UART (Asynchronous Serial Interface)

Among all the functions of the multi-function serial interface, this section describes those supported in operation modes 0 and 1.

- UART (Asynchronous Serial Interface)
- Overview of UART (Asynchronous Serial Interface)
- Registers of UART (Asynchronous Serial Interface)
  - Serial Control Register (SCR)
  - Serial Mode Register (SMR)
  - Serial Status Register (SSR)
  - Extended Serial Control Register (ESCR)
  - Reception Data Register/Transmission Data Register (RDR/TDR)
  - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
  - FIFO Control Register 1 (FCR1)
  - FIFO Control Register 0 (FCR0)
  - FIFO Byte Register (FBYTE1/FBYTE2)
- Interrupts of UART
  - Occurrence of Reception Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing
  - Occurrence of Transmission Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing
- Operation of UART
- Dedicated Baud Rate Generator
  - Setting Baud Rate
- Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)
- Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

24.3 Overview of UART (Asynchronous Serial Interface)

UART (asynchronous serial interface) is a general-purpose serial data communication interface to perform asynchronous communication (start-stop synchronization) with an external unit. The UART supports a two-way communication function (normal mode) and a master/slave communication function (multi-processor mode: the master and slaves both supported). The UART also has transmission/reception FIFO.

Functions of UART (Asynchronous Serial Interface)

		Function
1	Data	■ Full-duplex double buffer (when FIFO is not used) ■ Transmission/reception FIFO (maximum size: 16 bytes each) (when FIFO is used)[1]
2	Serial input	Oversampling is performed for three times to determine the reception value by the majority of the sampling values achieved.
3	Transfer system	Asynchronous
4	Baud rate	■ Dedicated baud rate generator (15-bit reload counter configuration) ■ The reload counter can be used to adjust the external clock input.
5	Data length	5 to 9 bits (in normal mode), 7 or 8 bits (in multi-processor mode)
6	Signaling system	NRZ (Non Return to Zero), inverted NRZ
7	Start bit detection	■ Synchronized with the falling edge of a start bit (NRZ) ■ Synchronized with the rising edge of a start bit (inverted NRZ)
8	Reception error detection	■ Framing error ■ Overrun error ■ Parity error[2]
9	Interrupt request	■ Reception interrupt (completion of reception, framing error, overrun error, parity error)[2] ■ Transmission interrupt (transmission data empty, transmission bus idle) ■ Transmission FIFO interrupt (when transmission FIFO is empty) ■ DMA transfer support function for transmission and reception
10	Master/slave communication function (multi-processor mode)	Communication between 1 (master) and n (slaves) is enabled. (The master and slave systems are both supported.)
11	FIFO options	■ Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes)[1] ■ Transmission FIFO or reception FIFO selectable ■ Transmission data can be resent. ■ The interrupt timing for reception FIFO can be modified by software. ■ FIFO reset is supported separately.

[1]: There is no FIFO between ch.0 and ch.3.
[2]: The detection of a parity error is enabled only in normal mode.

24.4 Registers of UART (Asynchronous Serial Interface)

This section lists the registers of UART (asynchronous serial interface).

List of registers of UART (Asynchronous Serial Interface)

Table 24-2. List of Registers of UART (Asynchronous Serial Interface) (Sheet 1 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
0	SCR0	Serial control register 0	24.4.1
	SMR0	Serial mode register 0	24.4.2
	ESCR0	Extended serial control register 0	24.4.4
	BGR0	Baud rate generator register 0	24.4.6
	SSR0	Serial status register 0	24.4.3
	RDR0	Received data register 0	24.4.5
	TDR0	Transmitted data register 0	24.4.5
1	SCR1	Serial control register 1	24.4.1
	SMR1	Serial mode register 1	24.4.2
	ESCR1	Extended serial control register 1	24.4.4
	BGR1	Baud rate generator register 1	24.4.6
	SSR1	Serial status register 1	24.4.3
	RDR1	Received data register 1	24.4.5
	TDR1	Transmitted data register 1	24.4.5
2	SCR2	Serial control register 2	24.4.1
	SMR2	Serial mode register 2	24.4.2
	ESCR2	Extended serial control register 2	24.4.4
	BGR2	Baud rate generator register 2	24.4.6
	SSR2	Serial status register 2	24.4.3
	RDR2	Received data register 2	24.4.5
	TDR2	Transmitted data register 2	24.4.5
3	SCR3	Serial control register 3	24.4.1
	SMR3	Serial mode register 3	24.4.2
	ESCR3	Extended serial control register 3	24.4.4
	BGR3	Baud rate generator register 3	24.4.6
	SSR3	Serial status register 3	24.4.3
	RDR3	Received data register 3	24.4.5
	TDR3	Transmitted data register 3	24.4.5
8	SCR8	Serial control register 8	24.4.1
	SMR8	Serial mode register 8	24.4.2
	ESCR8	Extended serial control register 8	24.4.4
	BGR8	Baud rate generator register 8	24.4.6
	SSR8	Serial status register 8	24.4.3
	RDR8	Received data register 8	24.4.5
	TDR8	Transmitted data register 8	24.4.5
	FCR18	FIFO control register 18	24.4.7
	FCR08	FIFO control register 08	24.4.8
	FBYTE18	FIFO1 byte register 8	24.4.9
	FBYTE28	FIFO2 byte register 8	24.4.9

Table 24-2. List of Registers of UART (Asynchronous Serial Interface) (Sheet 2 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
9	SCR9	Serial control register 9	24.4.1
	SMR9	Serial mode register 9	24.4.2
	ESCR9	Extended serial control register 9	24.4.4
	BGR9	Baud rate generator register 9	24.4.6
	SSR9	Serial status register 9	24.4.3
	RDR9	Received data register 9	24.4.5
	TDR9	Transmitted data register 9	24.4.5
	FCR19	FIFO control register 19	24.4.7
	FCR09	FIFO control register 09	24.4.8
	FBYTE19	FIFO1 byte register 9	24.4.9
	FBYTE29	FIFO2 byte register 9	24.4.9
10	SCR10	Serial control register 10	24.4.1
	SMR10	Serial mode register 10	24.4.2
	ESCR10	Extended serial control register 10	24.4.4
	BGR10	Baud rate generator register 10	24.4.6
	SSR10	Serial status register 10	24.4.3
	RDR10	Received data register 10	24.4.5
	TDR10	Transmitted data register 10	24.4.5
	FCR110	FIFO control register 110	24.4.7
	FCR010	FIFO control register 010	24.4.8
	FBYTE110	FIFO1 byte register 10	24.4.9
	FBYTE210	FIFO2 byte register 10	24.4.9
11	SCR11	Serial control register 11	24.4.1
	SMR11	Serial mode register 11	24.4.2
	ESCR11	Extended serial control register 11	24.4.4
	BGR11	Baud rate generator register 11	24.4.6
	SSR11	Serial status register 11	24.4.3
	RDR11	Received data register 11	24.4.5
	TDR11	Transmitted data register 11	24.4.5
	FCR111	FIFO control register 111	24.4.7
	FCR011	FIFO control register 011	24.4.8
	FBYTE111	FIFO1 byte register 11	24.4.9
	FBYTE211	FIFO2 byte register 11	24.4.9

Table 24-3. Bit Assignment of UART (Asynchronous Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SBL	BDS	SCKE	SOE
SSR/ ESCR	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	-	ESBL	INV	PEN	P	L2	L1	L0
RDR/TDR	-							D8 (AD)	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

Operation mode

The UART (asynchronous serial interface) operates in two different modes. The mode selection is determined by MD2, MD1 and MD0 in the serial mode register (SMR).

Table 24-4. Operation Modes of UART (Asynchronous Serial Interface)

Operation mode	MD2	MD1	MD0	Type
0	0	0	0	UART0 (asynchronous normal mode)
1	0	0	1	UART1 (asynchronous multi-processor mode)

24.4.1 Serial Control Register (SCR)

The serial control register (SCR) enables or disables transmission/reception, transmission/reception interrupts, and transmission bus idle interrupts. SCR can also reset the UART.

Serial Control Register (SCR)

Figure 24-1 shows the bit structure of the serial control register (SCR), and Table 24-5. describes the function of each bit.

Figure 24-1. Bit Structure of Serial Control Register (SCR)

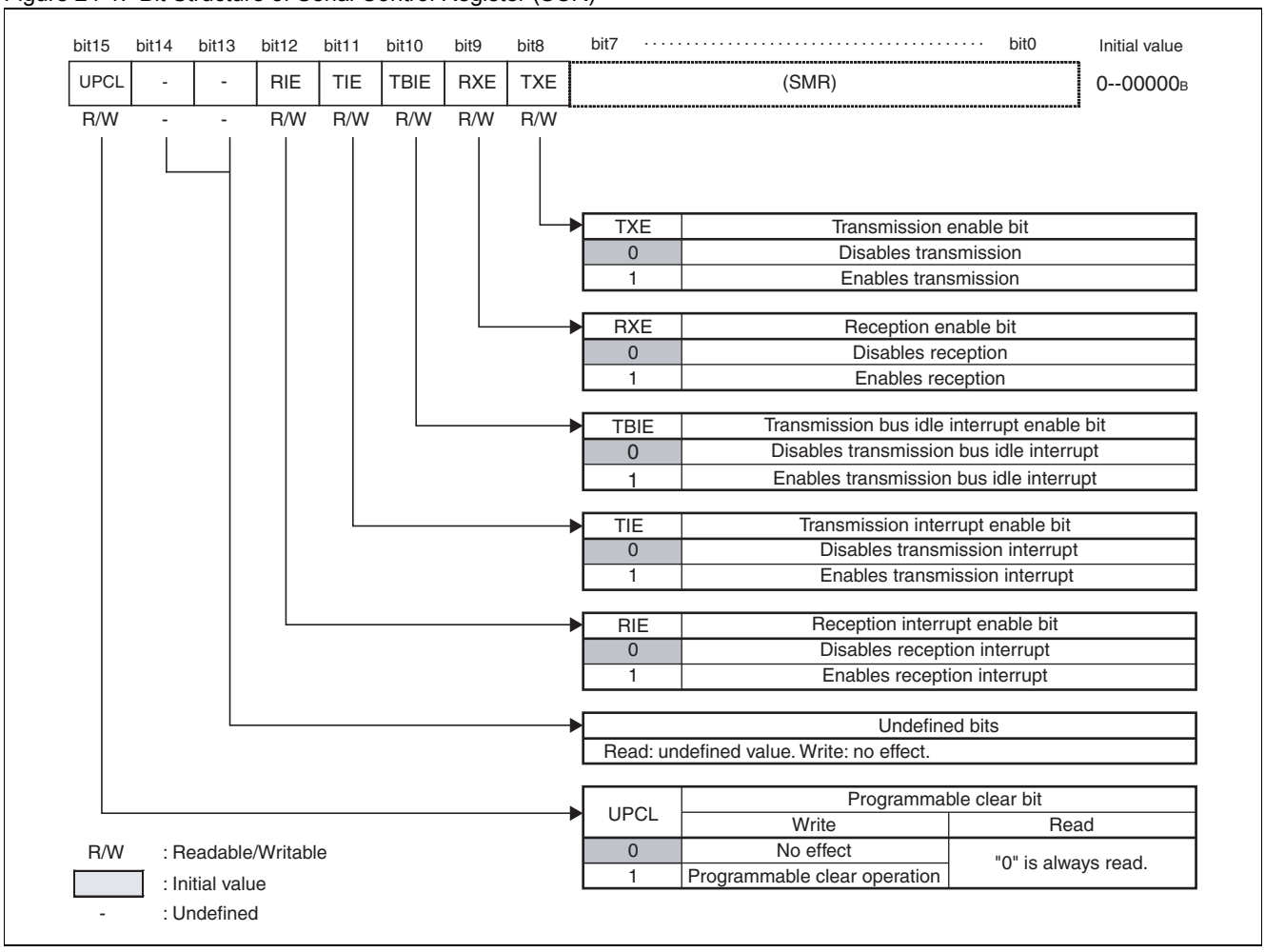


Table 24-5. Functional Description of Each Bit of Serial Control Register (SCR)

Bit name		Function
bit15	UPCL: Programmable clear bit	This bit is used to initialize the internal state of the UART. Setting the bit to "1": ■ The UART will be reset directly (software reset). The register setting, however, will be retained. In this case, communication of the data which is being transmitted or received will be cut off immediately. ■ The baud rate generator will reload the value set in BGR1/BGR0 registers, and then restart the operation. ■ All the transmission/reception interrupt sources (PE, FRE, ORE, RDRF, TDRE and TBI) will be initialized (000011_B). Setting the bit to "0": No effect on the operation Reading this bit always returns "0". **Note:** Execute the programmable clear operation after disabling interrupts. Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.
bit14, bit13	Undefined bits	Read:undefined value Write:no effect
bit12	RIE: Reception interrupt enable bit	■ This bit is used to enable/disable the output of reception interrupt requests to the CPU. ■ A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or when any of the error flag bits (PE, ORE or FRE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	■ This bit is used to enable/disable the output of transmission interrupt requests to the CPU. ■ A transmission interrupt request is output when the TIE and TDRE bits are set to "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	■ This bit is used to enable/disable the output of transmission bus idle interrupt requests to the CPU. ■ A transmission bus idle interrupt request is output when the TBIE and TBI bits are set to "1".
bit9	RXE: Reception enable bit	This bit is used to enable/disable UART reception operation. ■ Setting the bit to "0" disables the reception operation. ■ Setting the bit to "1" enables the reception operation. **Note:** Even when the reception operation is enabled (RXE = 1), such operation does not start until the falling edge of a start bit (in NRZ format: INV = 0) is input. (When the inverted NRZ format is selected (INV = 1), the reception operation does not start until the rising edge is input.) If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.
bit8	TXE: Transmission enable bit	This bit is used to enable/disable UART transmission operation. ■ Setting the bit to "0" disables the transmission operation. ■ Setting the bit to "1" enables the transmission operation. **Note:** If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.

24.4.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, selects the transfer direction, data length and stop bit length, and enables or disables the output to the serial data and serial clock pins.

Serial Mode Register (SMR)

Figure 24-2 shows the bit structure of the serial mode register (SMR), and Table 24-6 describes the function of each bit.

Figure 24-2. Bit Structure of Serial Mode Register (SMR)

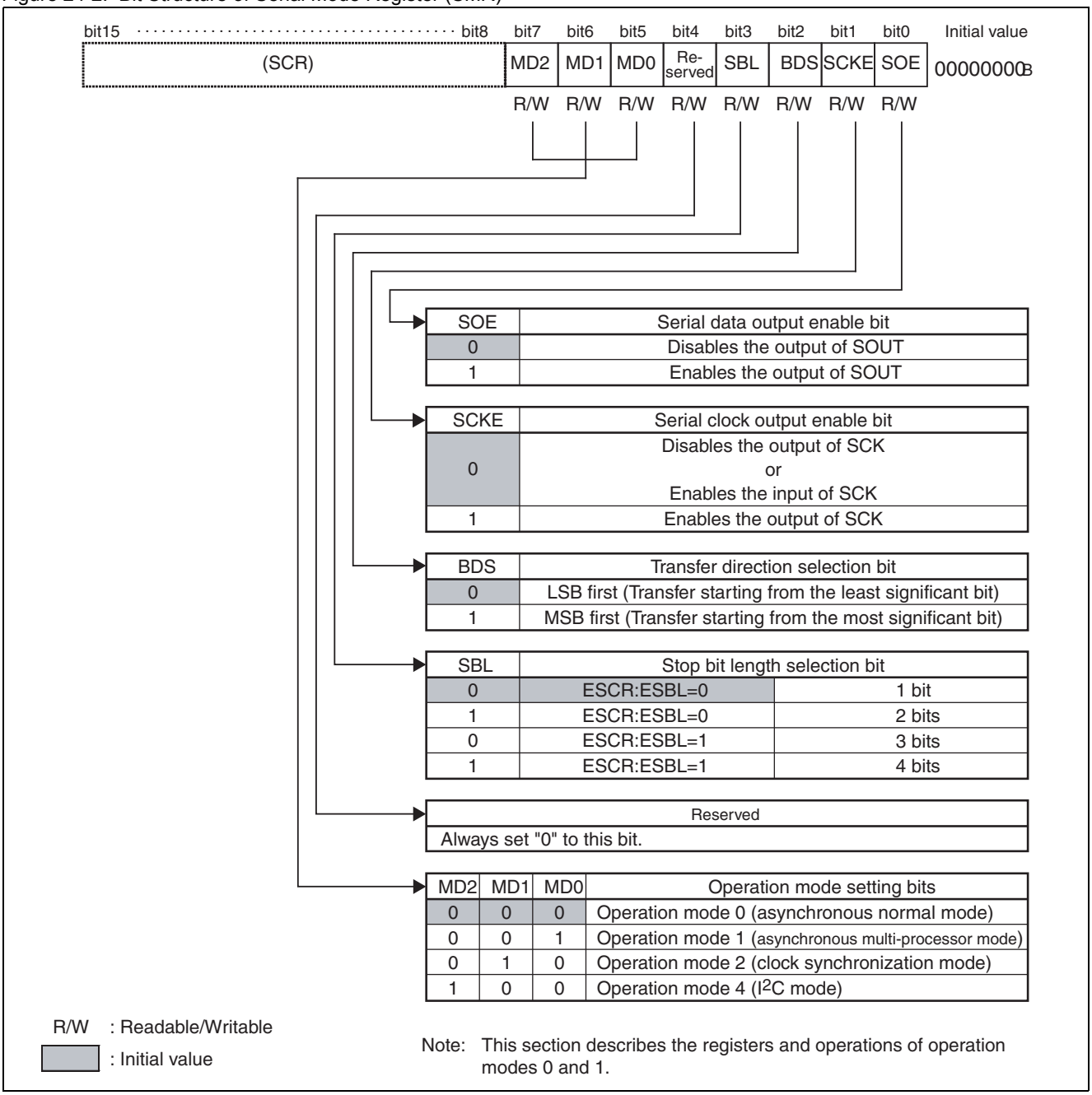


Table 24-6. Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2, MD1, MD0: Operation mode setting bits	These bits set the operation mode for the asynchronous serial interface. "000_B":Selects operation mode 0 (asynchronous normal mode) "001_B":Selects operation mode 1 (asynchronous multi-processor mode) "010_B":Selects operation mode 2 (clock synchronization mode) "100_B":Selects operation mode 4 (I^2C mode) This section describes the registers and operations of operation mode 0 (asynchronous normal mode) and operation mode 1 (asynchronous multi-processor mode). **Note:** Settings other than above are prohibited. To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1). And then, after setting the operation mode, set each register.
bit4	Reserved bit	Always set "0" to this bit.
bit3	SBL: Stop bit length selection bit	This bit is used to select a bit length for a stop bit (frame end mark of transmission data). Setting the bit to SBL=0, ESCR:ESBL=0 sets the stop bit to 1 bit in length. Setting the bit to SBL=1, ESCR:ESBL=0 sets the stop bit to 2 bits in length. Setting the bit to SBL=0, ESCR:ESBL=1 sets the stop bit to 3 bits in length. Setting the bit to SBL=1, ESCR:ESBL=1 sets the stop bit to 4 bits in length. **Note:** ■ In reception, only the first bit of each stop bit is always detected. ■ Set this bit when transmission is disabled (TXE=0).
bit2	BDS: Transfer direction selection bit	This bit is used to determine the transfer priority for transfer serial data: whether the least significant bit should be transferred first (LSB first, BDS = 0) or the most significant bit should be transferred first (MSB first, BDS = 1). **Note:** Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit1	SCKE: Serial clock output enable bit	This bit is used to control the I/O port of the serial clock. Setting the bit to "0": The output of SCK "H" or the input of SCK will be enabled. To use it as a SCK input, set a general-purpose I/O port as the input port. Also select the external clock (BGR:EXT = 1) using the external clock selection bit. Setting the bit to "1" enables the output of SCK.
bit0	SOE: Serial data output enable bit	This bit is used to enable/disable the output of serial data. Setting the bit to "0" disables the output. Setting the bit to "1" enables the output of SOUT.

Note:

The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed.

- Serial Control Register (SCR)
- Extended Serial Control Register (ESCR)

Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

24.4.3 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status, and also checks and clears the reception error flag.

Serial Status Register (SSR)

Figure 24-3 shows the bit structure of the serial status register (SSR) and Table 24-7. describes the function of each bit.

Figure 24-3. Bit Structure of Serial Status Register (SSR)

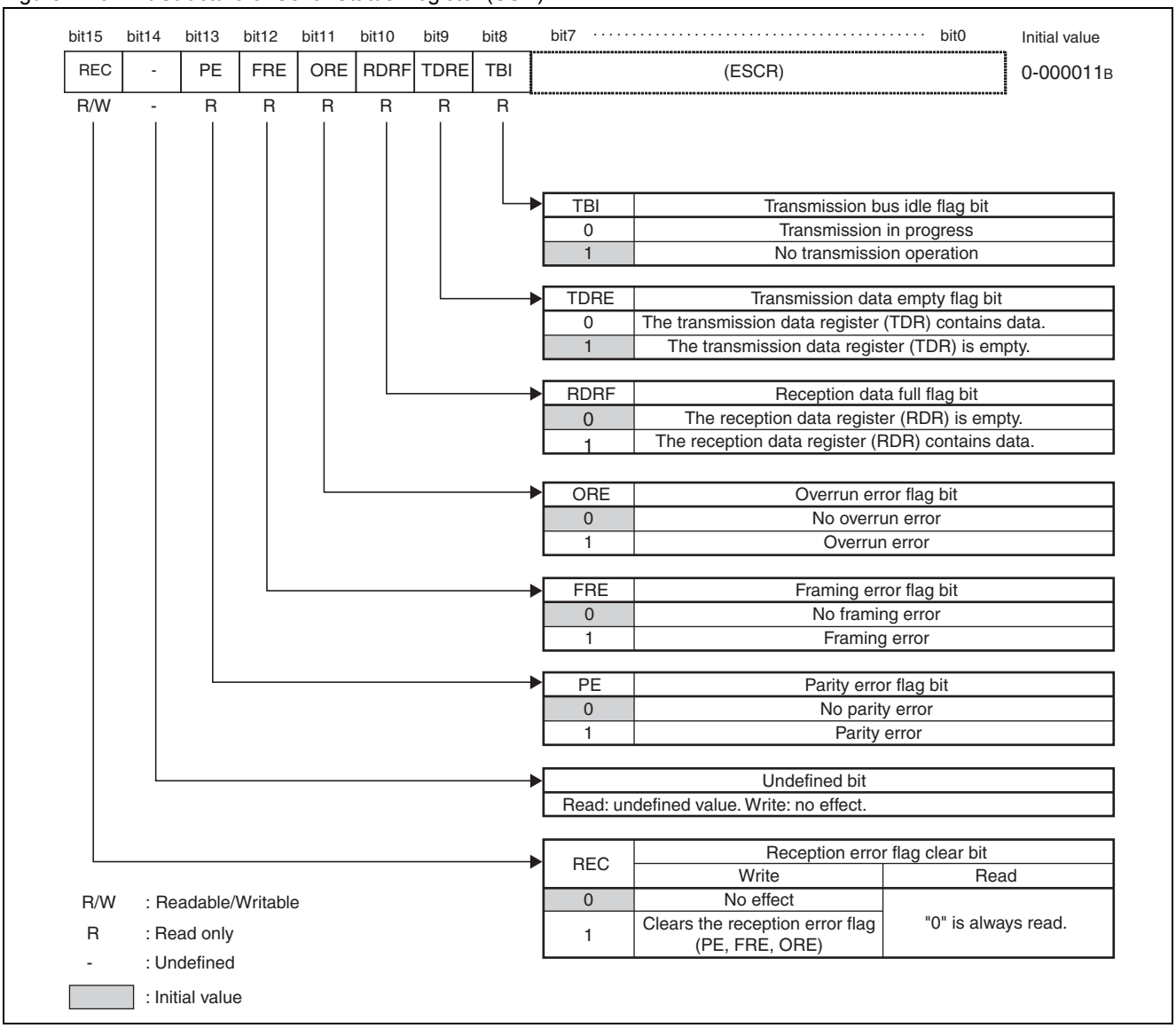


Table 24-7. Functional Description of Each Bit of Serial Status Register (SSR)

Bit name		Function
bit15	REC: Reception error flag clear bit	This bit is used to clear the PE, FRE and ORE flag in the serial status register (SSR). ■ Writing "1" clears the error flag. ■ Writing "0" has no effect. Reading this bit always returns "0".
bit14	Undefined bit	Read:undefined value Write:no effect
bit13	PE: Parity error flag bit (only available in operation mode 0)	■ The bit is set to "1" when a parity error occurs during reception (ESCR:PEN = 1). The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). ■ A reception interrupt request is output when the PE bit and the SCR:RIE bit are set to "1". ■ When this flag is set, the data in the reception data register (RDR) is invalid. ■ If this flag is set during the use of reception FIFO, the reception FIFO enable bit will be cleared and no reception data will be stored to the reception FIFO.
bit12	FRE: Framing error flag bit	■ This bit is set to "1" when a framing error occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). ■ A reception interrupt request is output when the FRE and RIE bits are set to "1". ■ When this flag is set, the data in the reception data register (RDR) is invalid. ■ If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
bit11	ORE: Overrun error flag bit	■ This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). ■ A reception interrupt request is output when the ORE and RIE bits are set to "1". ■ When this flag is set, the data in the reception data register (RDR) is invalid. ■ If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
bit10	RDRF: Reception data full flag bit	■ This flag indicates the status of the reception data register (RDR). ■ The bit is set to "1" when reception data is loaded to RDR. The bit is cleared to "0" when the reception data register (RDR) is read. ■ A reception interrupt request is output when the RDRF and RIE bits are set to "1". ■ RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. ■ When the reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1" during the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO. If RDR is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. ■ This bit is cleared to "0" when the reception FIFO, if used, becomes empty.
bit9	TDRE: Transmission data empty flag bit	■ This flag indicates the status of the transmission data register (TDR). ■ When transmission data is written to TDR, the bit becomes "0", indicating that TDR contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR no longer contains any valid data. ■ A transmission interrupt request is output when the TDRE and TIE bits are set to "1". ■ The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". ■ For information about the set/reset timings of the TDRE bit for when the transmission FIFO is used, refer to Section "24.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".
bit8	TBI: Transmission bus idle flag bit	■ This bit indicates that the UART is not performing transmission operation. ■ The bit is set to "0" when transmission data is written to the transmission data register (TDR). ■ The bit is set to "1" when the transmission data register is empty (TDRE =1) and no transmission operation is in progress. ■ The TBI bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". ■ A transmission interrupt request is output when this bit is "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).

24.4.4 Extended Serial Control Register (ESCR)

The extended serial control register (ESCR) can be used to set the transmission/reception data length, select the stop bit length, enable/disable the parity bit, select the parity bit, and invert the serial data format.

Bit structure of the Extended Serial Control Register (ESCR)

Figure 24-4 shows the bit structure of the extended serial control register (ESCR) and Table 24-8. describes the function of each bit.

Figure 24-4. Bit Structure of Extended Serial Control Register (ESCR)

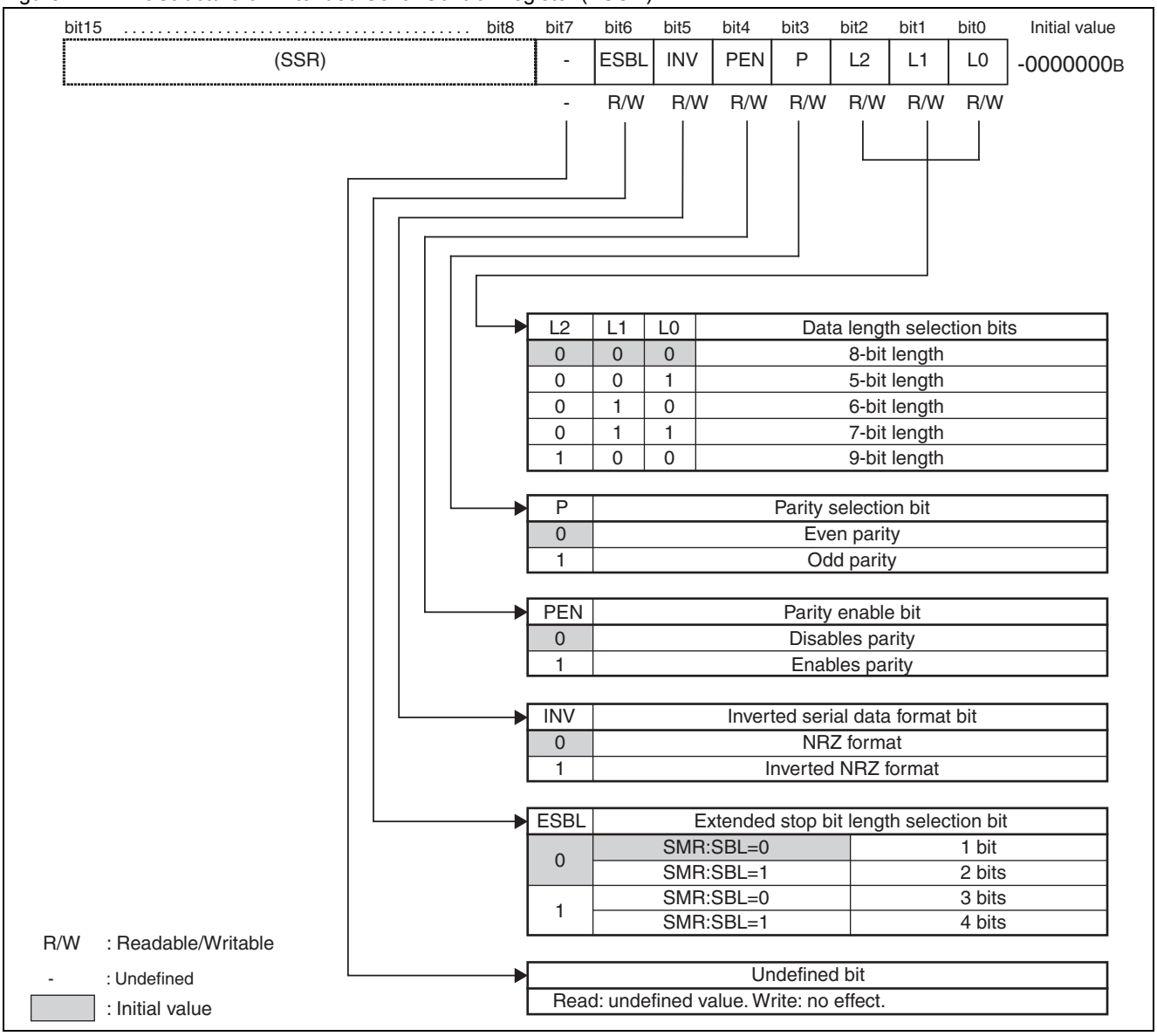


Table 24-8. Functional Description of Each Bit of Extended Serial Control Register (ESCR)

Bit name		Function
bit7	Undefined bit	Read:undefined value Write:no effect
bit6	ESBL: Extended stop bit length selection bit	Selects the bit length of stop bit (frame end mark of the transmitted data). Setting SMR: SBL=0, ESBL=0, sets the stop bit to 1 bit. Setting SMR: SBL=1, ESBL=0, sets the stop bit to 2 bits. Setting SMR: SBL=0, ESBL=1, sets the stop bit to 3 bits. Setting SMR: SBL=1, ESBL=1, sets the stop bit to 4 bits. **Notes:** ■ Always detects the first bit only of the stop bit during reception. ■ Set this bit when transmission is prohibited (TXE=0).
bit5	INV: Inverted serial data format bit	This bit is used to select the NRZ format or the inverted NRZ format as the serial data format.
bit4	PEN: Parity enable bit (only available in operation mode 0)	This bit is used to determine whether the parity bit should be added (in transmission) or detected (in reception). ■ When this bit is set to "0", the parity bit is not added. ■ When this bit is set to "1", the parity bit is added. **Note:** This bit is fixed to "0" internally in operation mode 1.
bit3	P: Parity selection bit (only available in operation mode 0)	This bit is used to select odd parity "1" or even parity "0" when parity is enabled (ESCR:PEN = 1). ■ Setting the bit to "0" selects even parity. ■ Setting the bit to "1" selects odd parity.
bit2 to bit0	L2, L1, L0: Data length selection bits	These bits are used to specify a data length for transmission/reception data. ■ Selecting "000_B" sets the data length to 8 bits. ■ Selecting "001_B" sets the data length to 5 bits. ■ Selecting "010_B" sets the data length to 6 bits. ■ Selecting "011_B" sets the data length to 7 bits. ■ Selecting "100_B" sets the data length to 9 bits. **Note:** Settings other than above are prohibited. For operation mode 1, set the data length to 7 or 8 bits. Any other setting is prohibited.

24.4.5 Reception Data Register / Transmission Data Register (RDR/TDR)

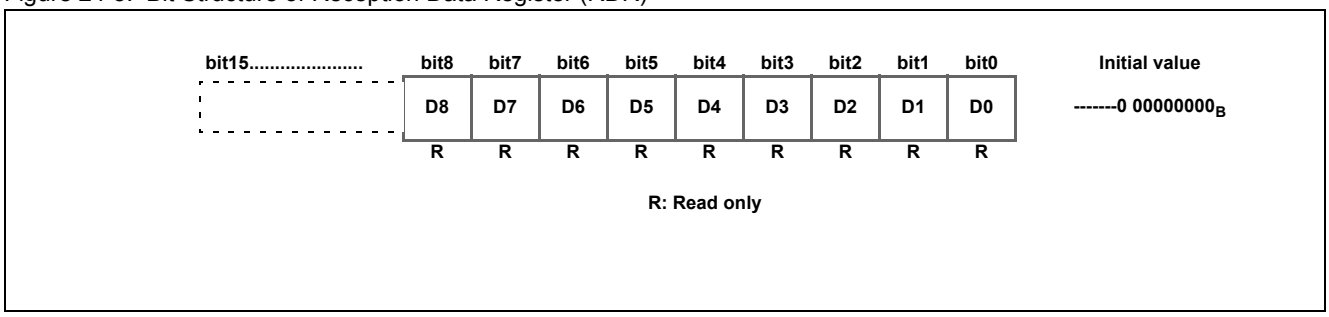
The reception data register and transmission data register are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

When FIFO operation is enabled, the RDR/TDR address becomes the read/write address for the FIFO.

Reception Data Register (RDR)

Figure 24-5 illustrates the bit structure of the serial reception register (RDR).

Figure 24-5. Bit Structure of Reception Data Register (RDR)



The reception data register (RDR) is a 9-bit data buffer register for serial data reception.

- A serial data signal sent to a serial input pin (SIN pin) is converted through the shift register and then stored in the reception data register (RDR).
- "0" is placed in one of the upper bits, depending on the data length, as shown below.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

X: Reception data bit

The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR). A reception interrupt request will be generated if reception interrupts have been enabled (SSR: RIE = 1).

- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is cleared to "0" automatically, when the reception data register (RDR) is read.
- If a reception error occurs (one of SSR:PE, ORE, or FRE is "1"), the data in the reception data register (RDR) becomes invalid.
- In operation mode 1 (multi-processor mode), 7-bit or 8-bit operation is performed and the received AD bit is stored in bit D8.
- 16-bit access is used to read RDR for a 9-bit transfer in operation mode 1.

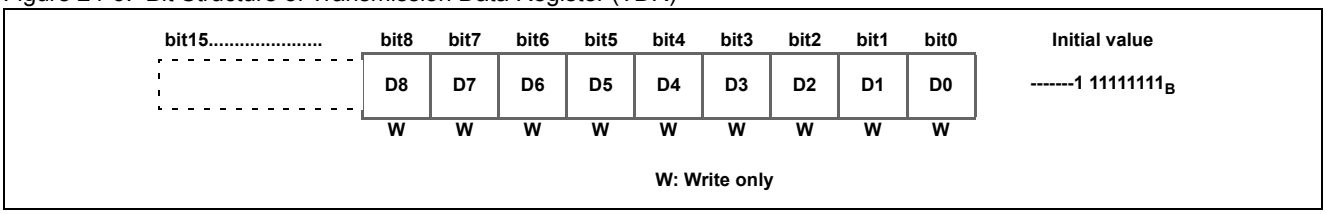
Notes:

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.
- If a reception error occurs (one of SSR:PE, ORE, or FRE is "1") when the reception FIFO is used, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.

Transmission Data Register (TDR)

Figure 24-6 illustrates the bit structure of the transmission data register.

Figure 24-6. Bit Structure of Transmission Data Register (TDR)



The transmission data register (TDR) is a 9-bit data buffer register for serial data transmission.

- If transmission data is written to the transmission data register (TDR) when transmission operation is enabled (SCR:TXE = 1), the transmission data will be transferred to the transmission shift register, converted into serial data and then sent from a serial data output pin (SOUT pin).
- As shown below, data becomes invalid from the upper bit in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

X: Reception data bit

- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) will be set to "1" when transmission data is transferred to the transmission shift register and the transmission starts.
- Transmission data can be written when the transmission data empty flag (SSR:TDRE) is set to "1". A transmission interrupt will occur if transmission interrupts have been enabled. Write transmission data by generating a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is set to "1".
- Transmission data cannot be written when the transmission data empty flag (SSR:TDRE) is set to "0" and the transmission FIFO is either disabled or full.
- In operation mode 1 (multi-processor mode), 7-bit or 8-bit operation is performed and the AD bit is sent by writing to bit D8.
- 16-bit access is used to write to TDR for a 9-bit transfer in operation mode 1.

Notes:

- The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. These registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.
- For information about the timing for setting the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, refer to Section "24.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".

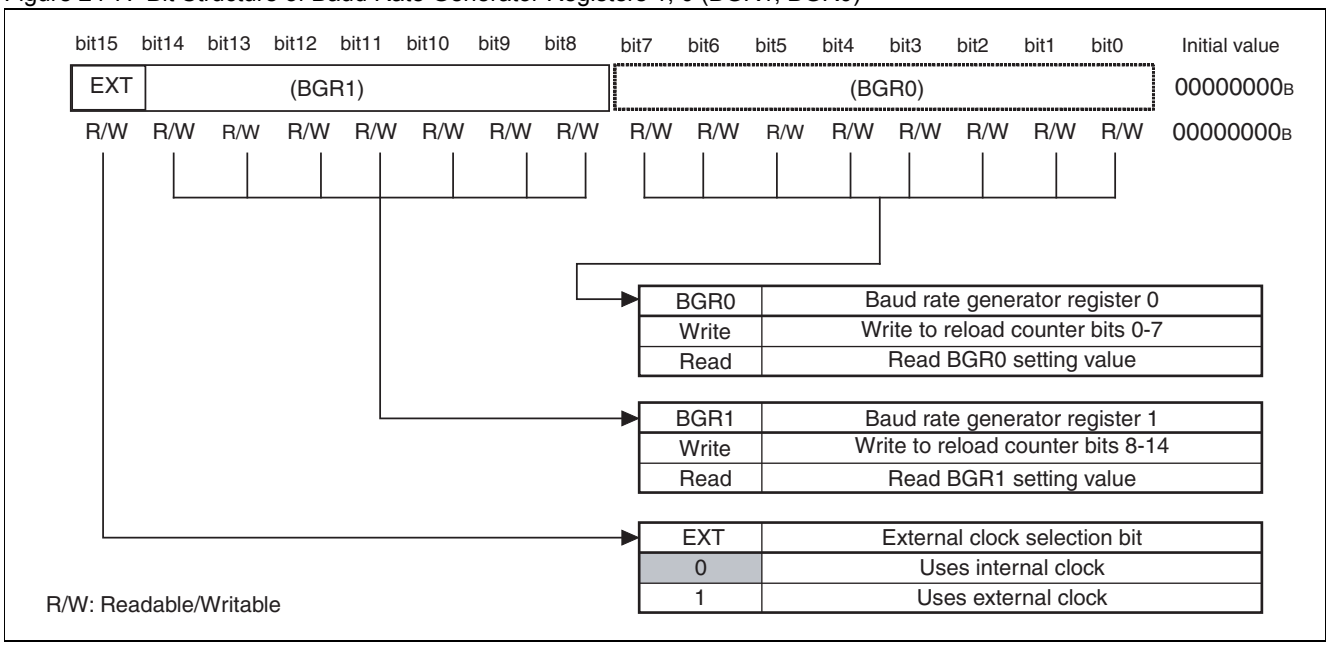
24.4.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock. They also allow an external clock to be selected as the clock source for the reload counter.

Bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0)

Figure 24-7 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 24-7. Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)



- The baud rate generator registers are used to set a division ratio for the serial clock.
- BGR1 and BGR0 correspond to the upper bits and lower bits respectively and they can write a reload value to be counted as well as read BGR1/BGR0 setting values.
- The reload counter starts counting when a reload value is written to the baud rate generator registers (BGR1/BGR0).
- The EXT bit (bit15) is used to determine whether the internal clock or external clock should be used as the clock source for the reload counter. Setting EXT to "0" selects the internal clock, while setting EXT to "1" selects the external clock.

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- When a setting value of the baud rate generator registers 1, 0 (BGR1, BGR0) is changed, the new setting value is not reloaded until the counter value becomes "0000_H". To make the new setting value valid immediately, therefore, execute a programmable clear (UPCL) operation after changing the BGR1/BGR0 setting value.
- When the reload value is even-numbered, the "L" width of the reception serial clock is one peripheral clock (PCLK) cycle longer than the "H" width of the same serial clock. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
- Select 4 or a larger value for BGR1/BGR0. However, data may not be able to be received properly, due to a baud rate error or reload settings.
- To change the setting to the external clock (EXT = 1) during the operation of the baud rate generator, write "0" to baud rate generator registers 1, 0 (BGR1, BGR0), execute a programmable clear (UPCL) operation, and then set to the external clock (EXT = 1).

24.4.7 FIFO Control Register 1 (FCR1)

The FIFO control register 1 (FCR1) selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

Bit structure of FIFO control register 1 (FCR1)

Figure 24-8 shows the bit structure of the FIFO control register 1 (FCR1) and Table 24-9 describes the function of each bit.

Figure 24-8. Bit Structure of FIFO Control Register 1 (FCR1)

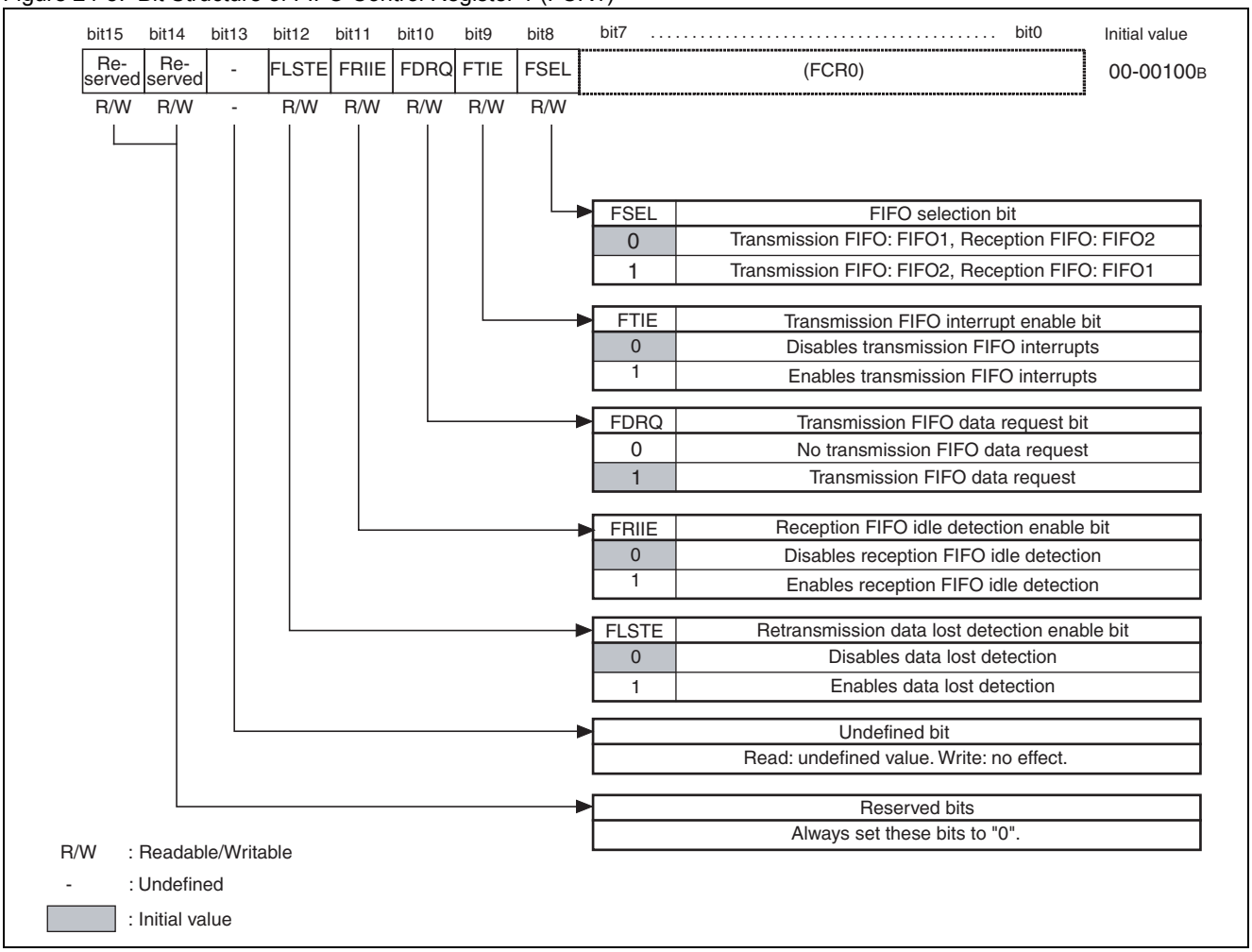


Table 24-9. Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	Reserved bits	Always set these bits to "0".
bit13	Undefined bit	Read:undefined value Write:no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. **Note:** To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8 clocks with the baud rate clock or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission FIFO interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. **FDRQ setting condition** FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) **FDRQ reset condition** ■ Writing "0" to this bit ■ When the transmission FIFO is full. **Note:** It is valid to write "0" when transmission FIFO has been enabled. It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. **Note:** This bit cannot be cleared by resetting FIFO (FCL2, FCL1 = 1). To modify this bit, disable FIFO operation beforehand (FCR:FE2, FE1 = 0).

24.4.8 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

Bit structure of FIFO control register 0 (FCR0)

Figure 24-9 shows the bit structure of the FIFO control register 0 (FCR0) and Table 24-10 describes the function of each bit.

Figure 24-9. Bit Structure of FIFO Control Register 0 (FCR0)

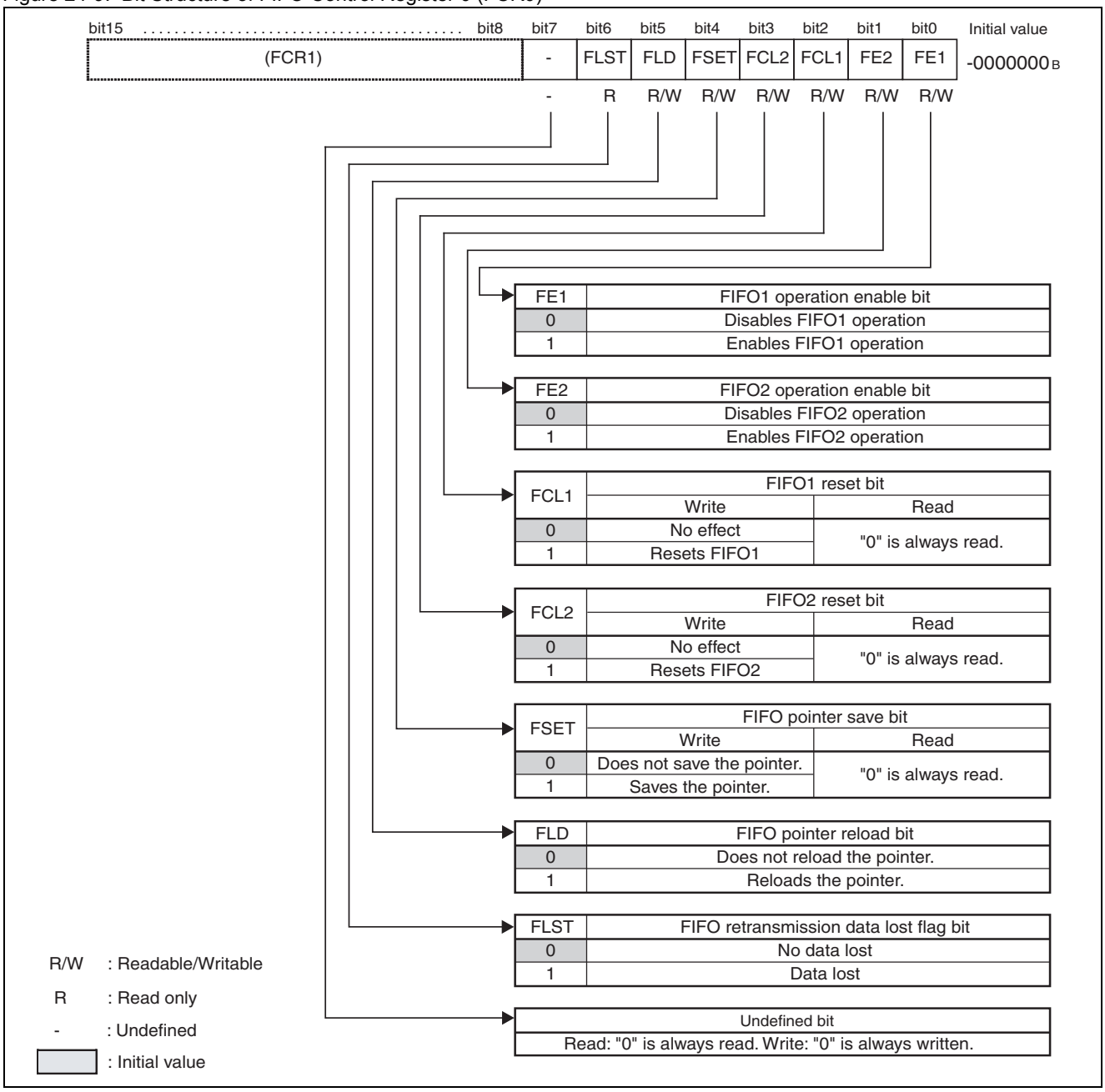


Table 24-10. Functional Description of Each Bit of FIFO Control Register 0 (FCR0)

Bit name		Function
bit7	Undefined bit	Read:"0" is always read. Write:Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition Writing (overwriting) to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. **FLST reset conditions** ■ FIFO reset (writing "1" to FCL) ■ Writing "1" to the FSET bit Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.
bit5	FLD: FIFO pointer reload bit	This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs. The bit becomes "0" when retransmission has been set. **Note:** Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset. It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress. Write "1" to this bit after setting the TIE and TBIE bits to "0". And then, set the TIE and TBIE bits to "1" when the transmission FIFO has been enabled.
bit4	FSET: FIFO pointer save bit	This bit is used to save the read pointer of the transmission FIFO. If the FLST bit is set to "0", saving the read pointer prior to communication will enable retransmission in case that an error such as a communication error occurs. Setting the bit to "1" saves the current read pointer value. Setting the bit to "0" has no effect. **Note:** Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".
bit3	FCL2: FIFO2 reset bit	This bit is used to reset FIFO2. Setting this bit to "1" initializes the internal state of FIFO2. Only the FCR1:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. **Note:** Disable transmission/reception before resetting FIFO2. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE2 register will become "0".
bit2	FCL1: FIFO1 reset bit	This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. **Note:** Disable transmission/reception before resetting FIFO1. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".
bit1	FE2: FIFO2 operation enable bit	This bit is used to enable/disable FIFO2 operation. ■ To use FIFO2, set this bit to "1". ■ When FIFO2 is set for the transmission FIFO (FCR1:FSEL = 1) and "1" is written to this bit, transmission will start immediately, if FIFO2 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". ■ This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. ■ Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. ■ Even when FIFO2 is disabled, its status is retained.
bit0	FE1: FIFO1 operation enable bit	This bit is used to enable/disable FIFO1 operation. ■ To use FIFO1, set this bit to "1". ■ When FIFO1 is set for the transmission FIFO (FCR1:FSEL = 0) and "1" is written to this bit, transmission will start immediately, if FIFO1 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". ■ This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. ■ Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. ■ Even when FIFO1 is disabled, its status is retained.

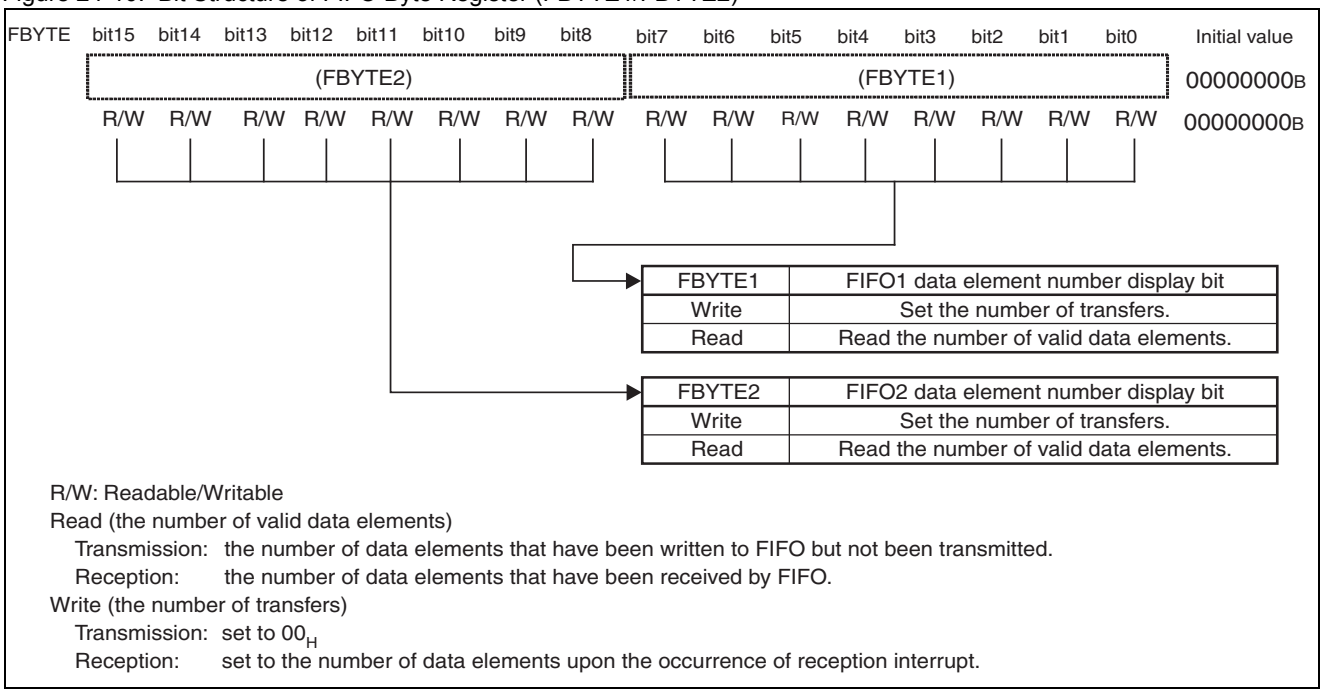
24.4.9 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO. It can also determine whether a reception interrupt should occur when the reception FIFO has received a specified number of data elements.

Bit Structure of FIFO byte register (FBYTE1/FBYTE2)

Figure 24-10 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 24-10. Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)



The FBYTE1/FBYTE2 register indicates the number of valid data elements written to or received by FIFO. The details are as follows, depending on the setting of the FCR1:FSEL bit.

Table 24-11. Displaying the Number of Data Elements

FSEL	FIFO selection	Number of data elements displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is 08_H.
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (SSR:RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.

Notes:

- Set 00_H to the FBYTE1/FBYTE2 register of the transmission FIFO.
- Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
- Change this register after prohibiting receiving.
- Read modify write (RMW) instructions cannot be used for this register.
- Settings that will exceed the capacity of FIFO are prohibited.

24.5 Interrupts of UART

The UART has the transmission/reception interrupt functionality. The following sources can be used to generate interrupt requests.

- When reception data is set in the reception data register (RDR) or a reception error occurs
- When transmission data is transferred from the transmission data register (TDR) to the transmission shift register and then transmission starts
- Transmission bus idle state (no transmission operation)
- Transmission FIFO data request

Interrupts of UART

Table 24-12 shows the interrupt control bits and interrupt sources of the UART.

Table 24-12. Interrupt Control Bits and Interrupt Sources of UART

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
			0	1			
Reception	RDRF	SSR	❍	❍	Reception of 1 byte	SCR:RIE	Reading reception data (RDR)
					Reception of the amount of data specified in FBYTE1/FBYTE2 setting value		Reading reception data (RDR) until reception FIFO becomes empty
					Detection of the idle state of reception for 8 clocks with the baud rate or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	ORE	SSR	❍	❍	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	❍	❍	Framing error		
	PE	SSR	❍	×	Parity error		
Transmission	TDRE	SSR	❍	❍	Transmission register being empty	SCR:TIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)[1]
	TBI	SSR	❍	❍	No transmission operation	SCR:TBIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)[1]
	FDRQ	FCR1	❍	❍	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

[1]: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

24.5.1 Occurrence of Reception Interrupts and Flag Set Timing

Reception interrupts are generated by the completion of reception (SSR:RDRF) and the occurrence of a reception error (SSR:PE, ORE, FRE).

Occurrence of reception interrupts and flag set timing

Reception data is stored to the reception data register (RDR) when the first stop bit is detected. Each flag is set when the reception has been completed (SSR:RDRF = 1) or a reception error has occurred (SSR:PE, ORE, FRE = 1). If reception interrupts have been enabled (SSR:RIE = 1), a reception interrupt will occur.

Note: If a reception error occurs, the data in the reception data register (RDR) will become invalid.

Figure 24-11. Timing for Setting RDRF (Reception Data Full) Flag Bit

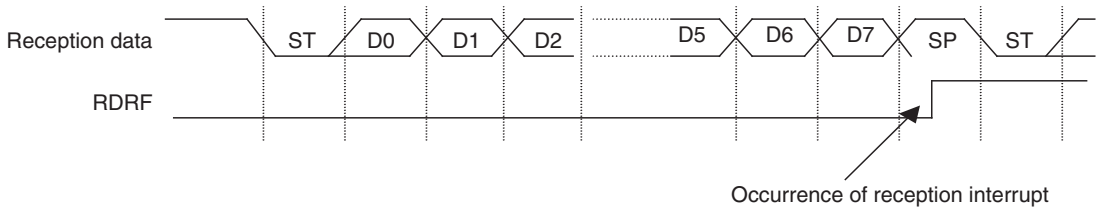
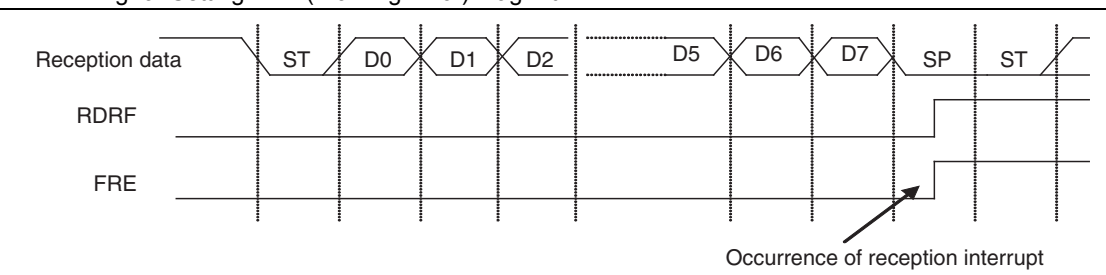
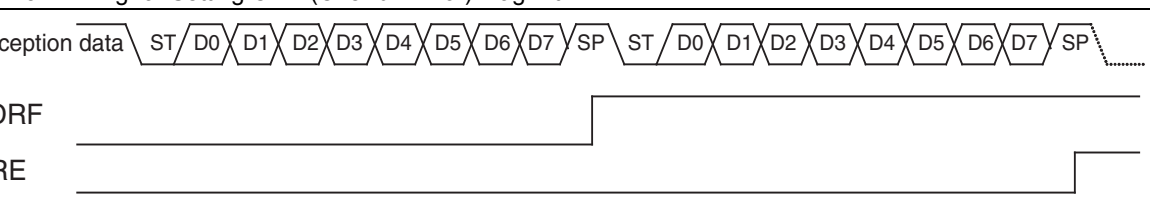


Figure 24-12. Timing for Setting FRE (Framing Error) Flag Bit



Notes: • A framing error occurs when the level of the first stop bit is "L".
• Although RDRF is set to "1" and data is received when a framing error occurs, the received data will be invalid.

Figure 24-13. Timing for Setting ORE (Overrun Error) Flag Bit



Note: An overrun error occurs when the next data is transferred before reception data is read (RDRF = 1).

24.5.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing

When the reception FIFO is used, an interrupt will occur, if the amount of data specified in the FBYTE1 register (FBYTE1/FBYTE2) is received.

Occurrence of reception interrupts when reception FIFO is used and flag set timing

Occurrence of interrupts when reception FIFO is used is determined by the setting value of the FBYTE1/FBYTE2 register.

- The reception data full flag of the serial status register (SSR:RDRF) is set to "1", when the received data is equivalent of the number of transfers specified in the FBYTE1/FBYTE2 register. At this point, a reception interrupt will occur, if reception interrupts have been enabled (SCR:RIE).
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- The reception data full flag (SSR:RDRF) is cleared when reception data (RDR) is read until the reception FIFO becomes empty.
- An overrun error occurs (SSR:ORE = 1) when the next data is received while the number of valid reception data elements is indicating the capacity of FIFO.

Figure 24-14. Timing for Generating Reception Interrupt when Reception FIFO is Used

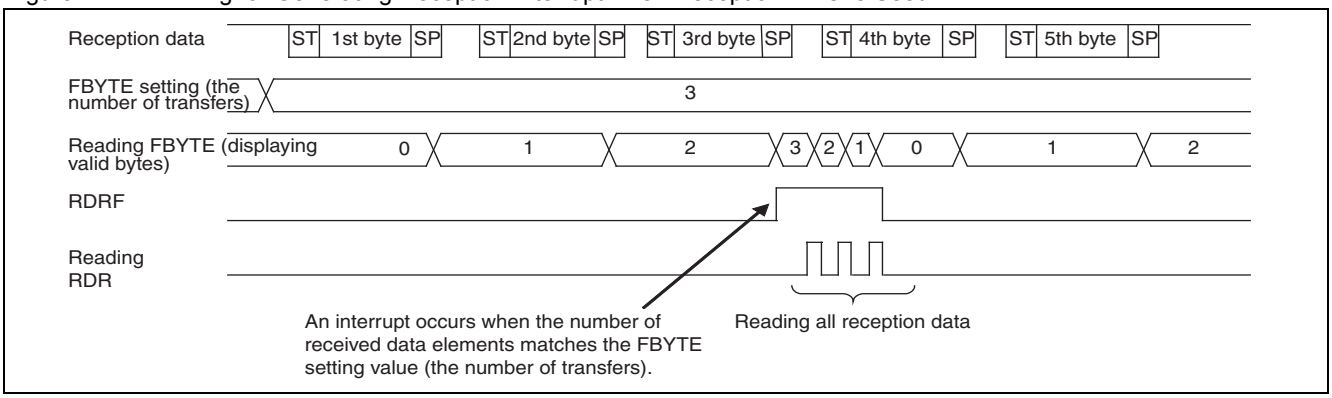
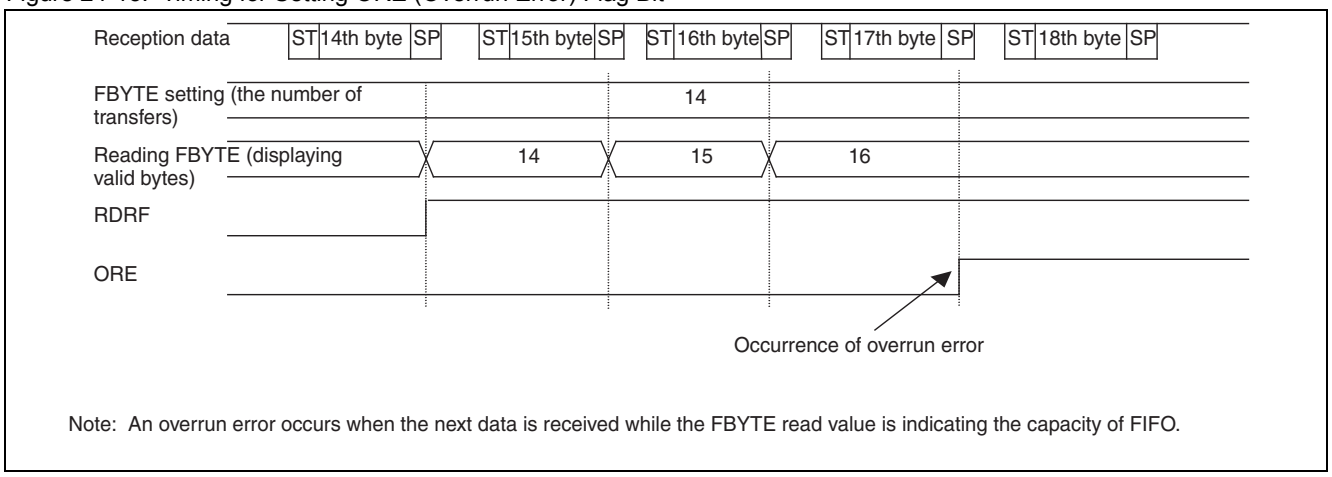


Figure 24-15. Timing for Setting ORE (Overrun Error) Flag Bit



Note: An overrun error occurs when the next data is received while the FBYTE read value is indicating the capacity of FIFO.

24.5.3 Occurrence of Transmission Interrupts and Flag Set Timing

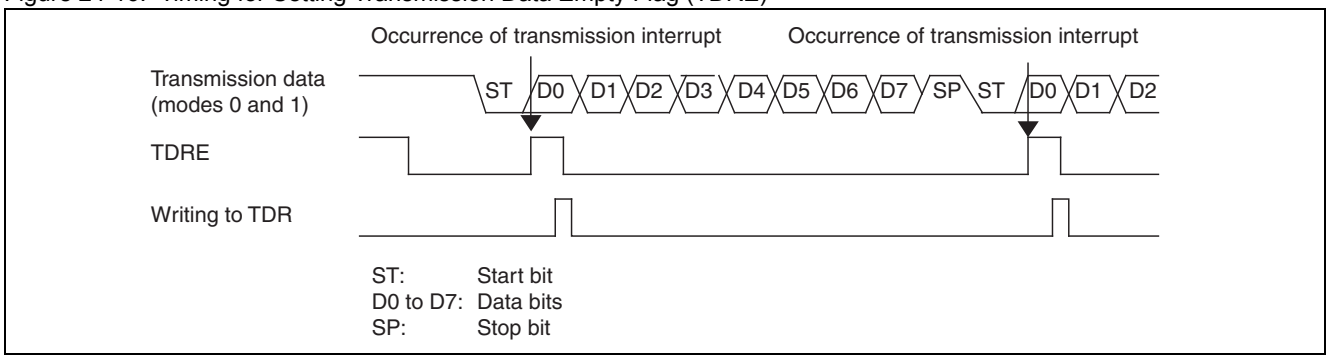
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

Occurrence of transmission interrupts and flag set timing

- Timing for setting the transmission data empty flag (TDRE)

  It is enabled to write the next data (SSR:TDRE = 1), when the data written to the transmission data register (TDR) is transferred to the transmission shift register. At this point, a transmission interrupt will occur, if transmission interrupts have been enabled (SCR:TIE = 1). As the TDRE bit is a read only bit, it is cleared by writing "0" to the transmission data register (TDR).

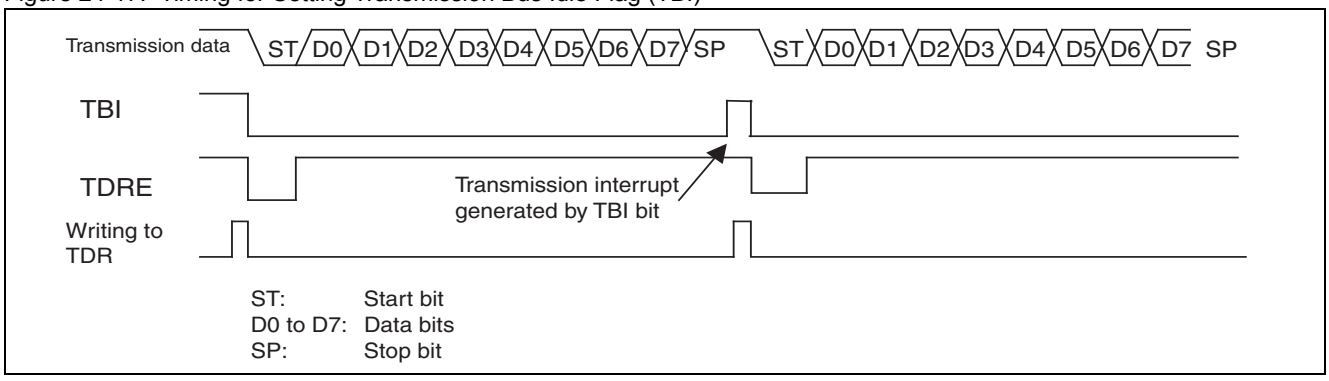
Figure 24-16. Timing for Setting Transmission Data Empty Flag (TDRE)



- Timing for setting the transmission bus idle flag (TBI)

  The SSR:TBI bit is set to "1", when the transmission data register is empty (TDRE = 1) and no transmission operation is in progress. At this point, a transmission interrupt occurs if transmission bus idle interrupts have been enabled (SCR:TBIE = 1). The TBI bit and transmission interrupt request are cleared when transmission data is set to the transmission data register (TDR).

Figure 24-17. Timing for Setting Transmission Bus Idle Flag (TBI)



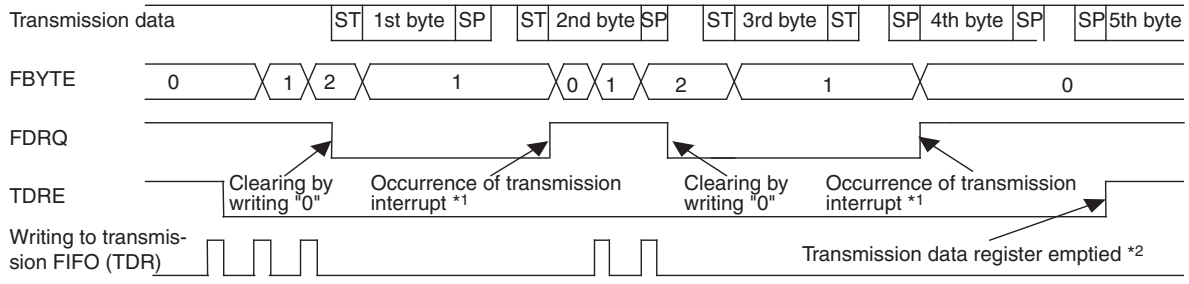
24.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

Occurrence of transmission interrupts when transmission FIFO is used and flag set timing

- The FIFO transmission data request bit (FCR1:FDRQ) is set to "1", when the transmission FIFO contains no data.
  At this point, a transmission interrupt will occur if FIFO transmission interrupts have been enabled (FCR1:FTIE = 1).
- Write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request when required data has been written to the transmission FIFO upon the occurrence of a transmission interrupt.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" when the transmission FIFO becomes full.
- The FIFO byte register (FBYTE1/FBYTE2) can be read to check if the transmission FIFO contains any data.
- FBYTE1/FBYTE2 = 00_H indicates that the transmission FIFO contains no data.

Figure 24-18. Occurrence Timing for Transmission Interrupts when Transmission FIFO is Used



*1: FDRQ is set to "1", as the transmission FIFO is empty.
*2: TDRE is set to "1", as the transmission shift register and the transmission buffer register contain no data.

24.6 Operation of UART

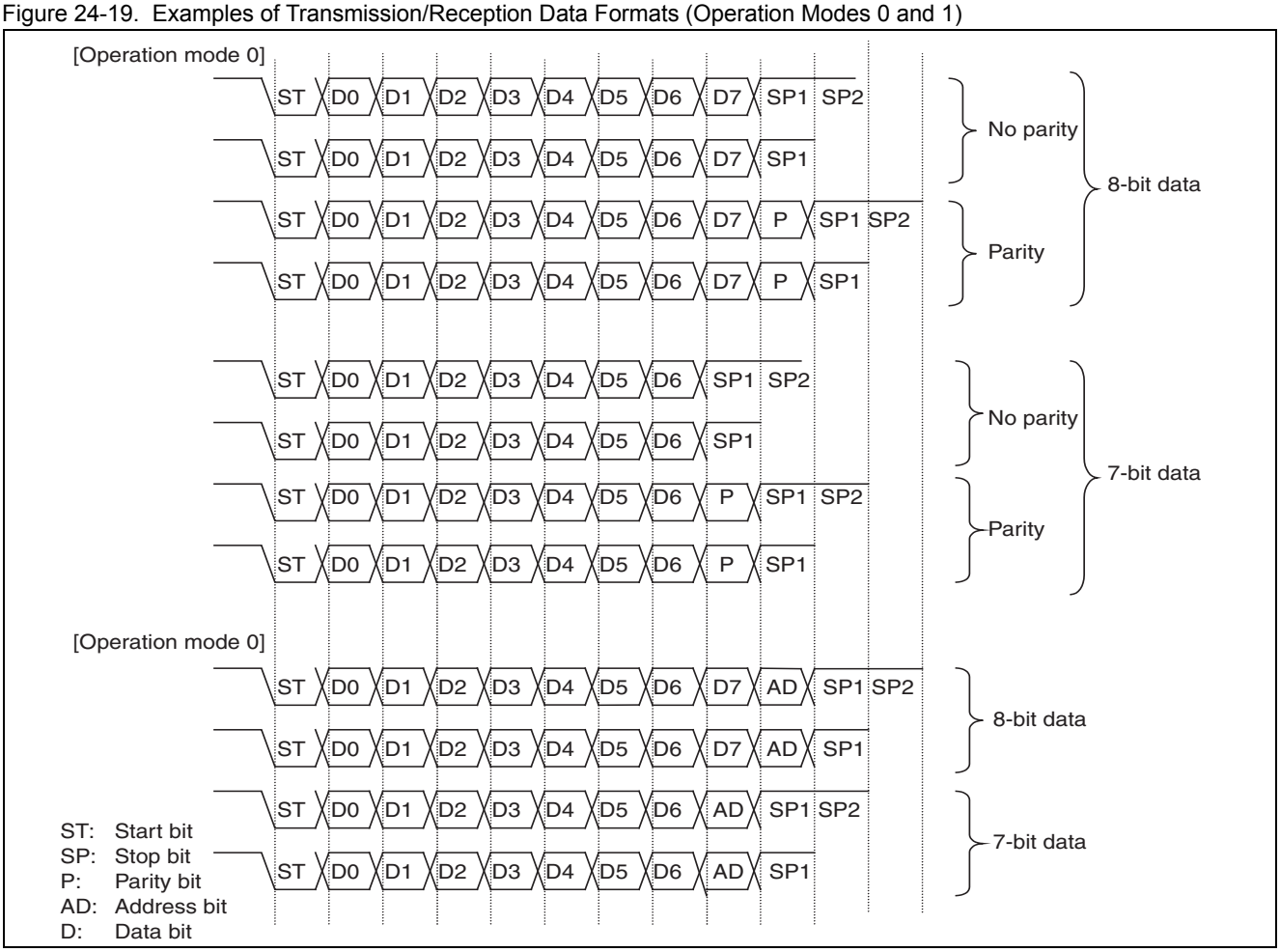
The UART operates in two-way serial asynchronous communications for mode 0 and in master/slave multi-processor communications for mode 1.

Operation of UART

- Transmission/reception data format
  - Transmission and reception data is transmitted or received for a specified data bit length, always starting from the start bit and finishing with the stop bit (at least 1 bit).
  - The data transfer direction (LSB or MSB first) is determined by the BDS bit in the serial mode register (SMR). When the addition of parity is selected, the parity bit is always placed between the last data bit and the first stop bit.
  - The addition or omission of parity can be selected in operation mode 0 (normal mode).
  - In operation mode 1 (multi-processor mode), the AD bit is added rather than parity.

  Figure 24-19 shows transmission/reception data formats for operation modes 0 and 1.

Figure 24-19. Examples of Transmission/Reception Data Formats (Operation Modes 0 and 1)



Notes:

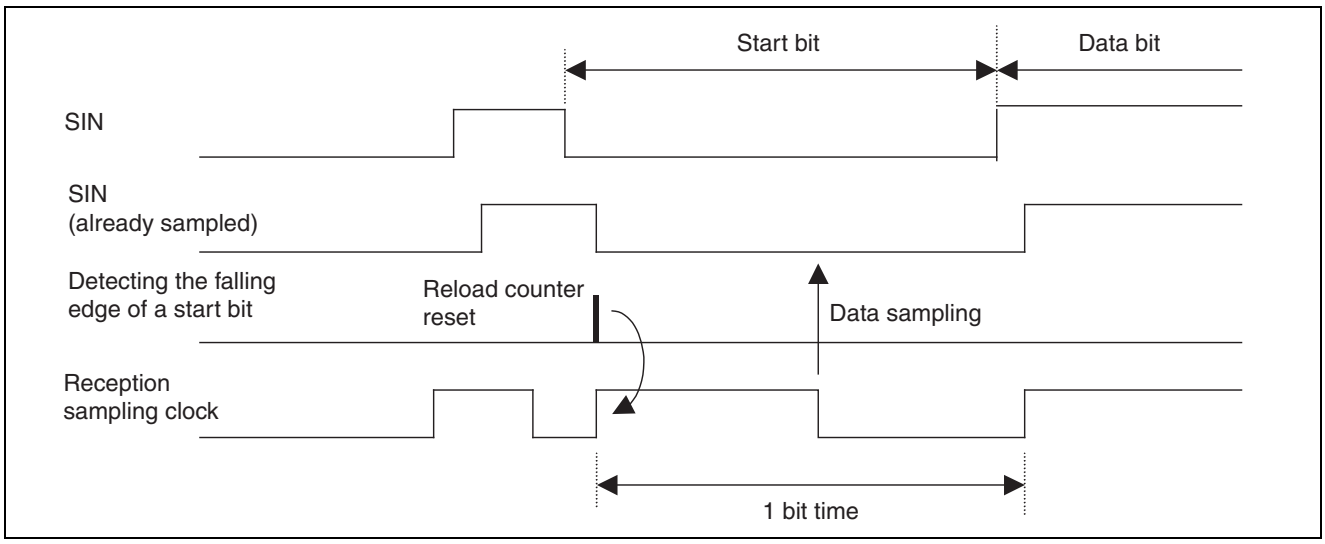
- Figure 24-19. shows cases where the data length is set to 7 or 8 bits. (The data length can be set to 5-9 bits for operation mode 0.)
- When the BDS bit in the serial mode register (SMR) is set to "1" (MSB first), the bits are processed in the following order: D7, D6, D5...D1, D0 (P).
- When the data length is set to X bits, the lower X bits of the transmission/reception data register (RDR/TDR) become valid.

- Transmission operation
  - Transmission data can be written to the transmission data register (TDR) when the transmission data empty flag bit (TDRE) in the serial status register (SSR) is set to "1". (If the transmission FIFO is enabled, transmission data can be written even when TDRE is set to "0".)
  - Writing transmission data to the transmission data register (TDR) sets the transmission data empty flag bit (TDRE) to "0".
  - When the transmission operation enable bit in the serial control register (SCR:TXE) is set to "1", transmission data is loaded to the transmission shift register and the transmission starts from the start bit.
  - Once transmission starts, the transmission data empty flag bit (TDRE) is set back to "1". At this point, a transmission interrupt will occur if transmission interrupts have been enabled (SCR:TIE = 1). The next transmission data can be written to the transmission data register through interrupt processing.

  Notes:
  - The initial value of the transmission data empty flag bit (SSR:TDRE) is "1". Therefore, a transmission interrupt occurs immediately after transmission interrupts are enabled (SCR:TIE=1).
  - The initial value of the FIFO transmission data request bit (FCR1:FDRQ) is "1". Therefore, a transmission interrupt occurs immediately after FIFO transmission interrupts are enabled (FCR1:FTIE = 1).
- Reception operation
  - Reception operation starts when such operation is enabled (SCR:RXE = 1).
  - When a start bit is detected, one frame of data is received according to the data format set in the extended serial control register (ESCR:PEN, P, L2, L1, L0) and the serial mode register (SMR:BDS).
  - Once one frame of data has been received, the reception data full flag bit (SSR:RDRF) is set to "1". At this point, a reception interrupt will occur if reception interrupts have been enabled (SCR:RIE = 1).
  - Read reception data after one frame of data has been received, and then check the error flag status of the serial status register (SSR). If a reception error is occurring, the error must be treated.
  - Reading reception data clears the reception data full flag bit (SSR:RDRF) to "0".
  - When the reception FIFO has been enabled, the reception data full flag bit (SSR:RDRF) will be set to "1", if the received data is equivalent of the number of frames specified in the reception FBYTE1/FBYTE2.
  - When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
  - If the error flag in the serial status register (SSR) is set to "1" when the reception FIFO has been enabled, the data in which the error has occurred will not be stored to the reception FIFO. At the same time, the reception data full flag bit (SSR:RDRF) will not be set to "1". (In case of an overrun error, however, the RDRF flag will be set to "1".) The reception FBYTE1/FBYTE2 indicates the number of data elements that was successfully received before the error occurs. The reception FIFO will not be enabled unless the error flag in the serial status register (SSR) is cleared to "0".
  - If the reception FIFO has been enabled, the reception data full flag bit (SSR:RDRF) will be cleared to "0" when the reception FIFO no longer has data.

  Note: The data in the reception data register (RDR) will become valid, if no reception error occurs (SSR:PE, ORE, FRE = 0) when the reception data register full flag bit (SSR:RDRF) is set to "1".
- Clock selection
  - The internal clock or external clock can be used.
  - To use the external clock, set BGR:EXT to "1". In this case, the external clock is divided by the baud rate generator.

- Detection of the start bit
  - In asynchronous mode, a start bit is identified by the falling edge of a SIN signal. Therefore, even when reception operation has been enabled (SCR:RXE = 1), such operation will not start unless the falling edge of a SIN signal is input.
  - When the falling edge of a start bit is detected, the reception reload counter of the baud rate generator is reset and reloaded to start counting down. This allows sampling to be always performed using the central part of data.

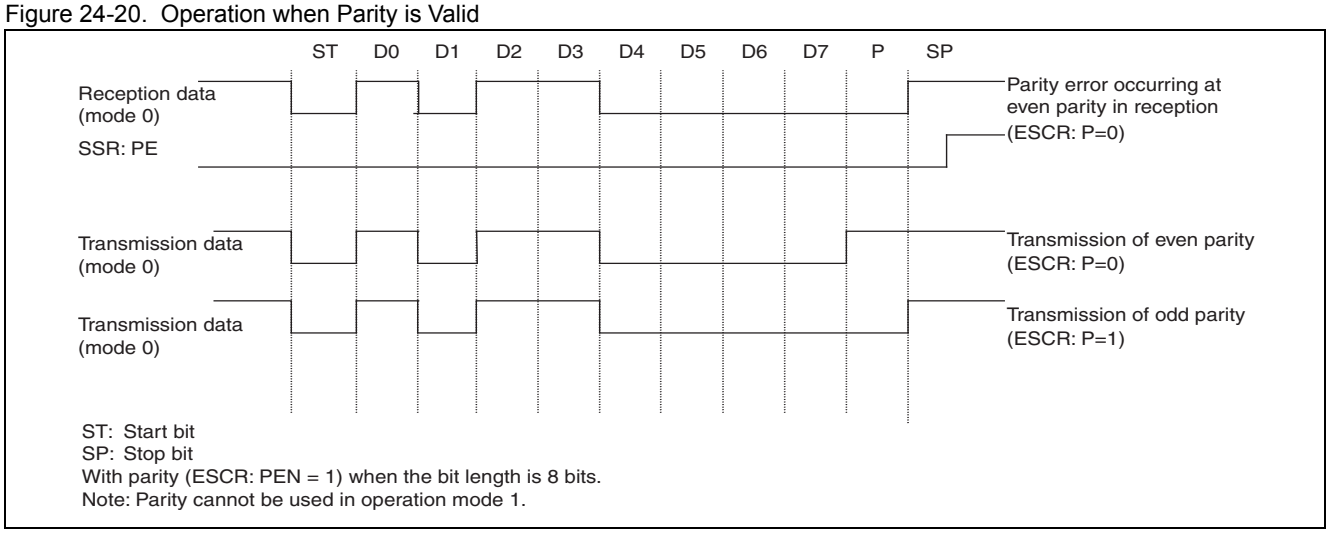


- Stop bit
  - 1 bit to 4 bits can be selected for the bit length.
  - The reception data full flag bit (SSR:RDRF) is set to "1" when the first stop bit is detected.
- Detection of errors
  - In operation mode 0, parity errors, overrun errors and frame errors can be detected.
  - In operation mode 1, overrun errors and frame errors can be detected. Parity errors, on the other hand, cannot be detected.

- Parity bit
  - Addition of the parity bit can be selected only in operation mode 0. The parity enable bit (ESCR:PEN) can be used to determine the addition or omission of parity, while the parity selection bit (ESCR:P) can be used to select even parity or odd parity.
  - Parity cannot be used in operation mode 1.

  Figure 24-20 shows transmission/reception data when parity is valid.

Figure 24-20. Operation when Parity is Valid

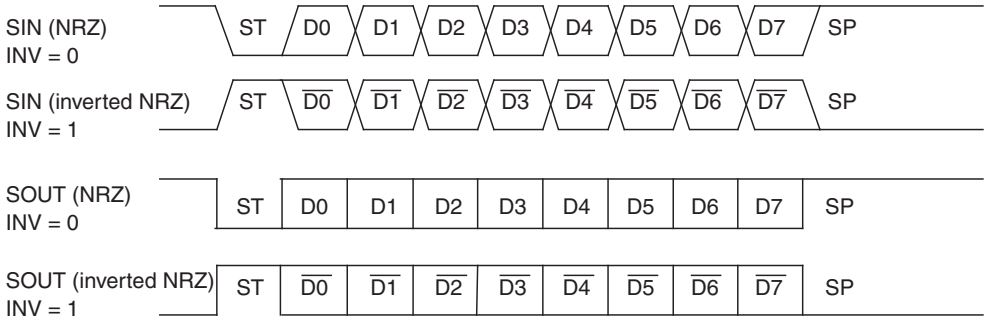


- Data signaling system

  The NRZ (Non Return to Zero) signaling system (ESCR:INV = 0) or the inverted NRZ signaling system (ESCR:INV = 1) can be selected by setting the INV bit in the extended communication control register.

  Figure 24-21 shows the NRZ and inverted NRZ signaling systems.

Figure 24-21. NRZ (Non Return to Zero) and Inverted NRZ Signaling Systems



- Data transfer system

  LSB-first or MSB-first data bit transfer system can be selected.

24.7 Dedicated Baud Rate Generator

One of the following options can be selected for the transmission/reception clock source of the UART.

- Dedicated baud rate generator (reload counter)
- External clock input to the baud rate generator (reload counter)

UART baud rate selection

One of the following two options can be selected for the baud rate.

- Baud rate achieved by dividing the internal clock using the dedicated baud rate generator (reload counter)

  There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

  The reload counter divides the internal clock, according to the set value.

  To set the clock source, select the internal clock (BGR:EXT = 0).

- Baud rate achieved by dividing the external clock using the dedicated baud rate generator (reload counter)

  The external clock is used as the clock source for the reload counter.

  The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

  The reload counter divides the external clock, according to the set value.

  To set the clock source, select the external clock and the baud rate generator clock (BGR:EXT = 1).

  This mode is made available on the assumption that an oscillator with a special frequency is divided for use.

  Notes:

  - Set the external clock (EXT = 1) while the reload counter is stopped (BGR1/BGR0 = 15’h00).
  - When the external clock has been selected (EXT = 1), the "H" and "L" widths of the external clock must be two or more peripheral clocks (PCLK).

24.7.1 Setting Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

Calculating the baud rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

- Reload value:
  $V = \phi / b - 1$

  V: Reload value
  b: Baud rate
  ϕ: Peripheral clock (PCLK), external clock frequency

- Example of calculation
  If the peripheral clock (PCLK) is 16MHz, the internal clock is used, and the baud rate is 19200bps, the reload value will be:
  Reload value:
  $V = (16 \times 1000000)/19200 - 1 = 832$
  As a result, the baud rate is:
  $b = (16 \times 1000000)/(832+1) = 19208$ bps

- Baud rate error
  The following formula is used to calculate a baud rate error.
  Error(%) = (calculated value - target value) / target value $\times$ 100

  Example: peripheral clock (PCLK) = 20MHz, target baud rate = 153600bps
  Reload value $=(20 \times 1000000)/153600 - 1 = 129$
  Baud rate (calculated value) $=(20 \times 1000000)/(129+1) = 153846$ (bps)
  Error (%) $=(153846 - 153600)/153600 \times 100 = 0.16$ (%)

Notes:

- The reload counter halts when the reload value is set to "0".
- When the reload value is even-numbered, the "L" width of the reception serial clock is one peripheral clock (PCLK) cycle longer than the "H" width of the same serial clock. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
- Select 4 or a larger value for the reload value. However, data may not be able to be received properly, due to a baud rate error or reload settings.

Reload values and baud rates for different peripheral clock (PCLK) frequencies

Table 24-13. Reload Values and Baud Rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4M	-	-	-	-	-	0	4	0	5	0	7	0
2.5M	-	-	-	0	-	-	-	-	-	-	-	-
2M	-	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

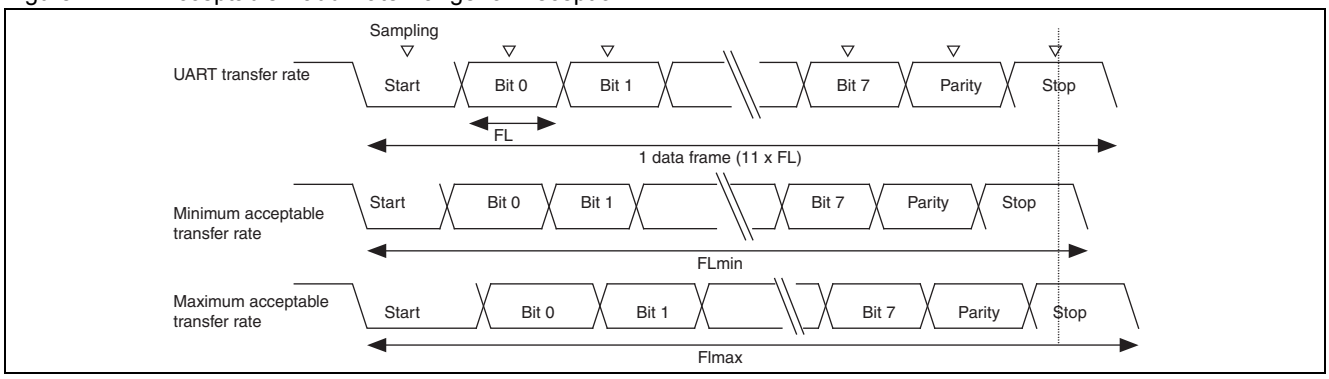
- Value:the value set in BGR1/BGR0 registers (decimal)
- ERR:baud rate error (%)

Acceptable baud rate range for reception

The following figure shows the range of acceptable baud rate differences at the transmission destination during reception.

The following calculation formula must be used to set a baud rate error for reception within the acceptable error range.

Figure 24-22. Acceptable Baud Rate Range for Reception



As shown in the figure, the sampling timing for reception data is determined by the counter selected by the BGR1/BGR0 registers after a start bit is detected. If all data including the last data (stop bit) can fit in this sampling timing, reception can be performed successfully.

In theory, the following is expected when this is applied to 11-bit reception.

When the sampling timing margin is equivalent of two clocks of the peripheral clock (PCLK) (ϕ), the minimum acceptable transfer rate (FLmin) is as follows:

FLmin = (11 bits $\times$ (V + 1) - (V + 1)/2 + 2)/ϕ = (21V + 25)/2ϕ (s)
V: reload valueϕ: peripheral clock (PCLK)

Consequently, the maximum receivable baud rate at the transmission destination (BGmax) is as follows:

BGmax = 11/FLmin = 22ϕ/(21V+25) (bps)
V: reload valueϕ: peripheral clock (PCLK)

Likewise, the maximum acceptable transfer rate (Flmax) can be calculated as shown below:

FLmax = (11 bits $\times$ (V + 1) + (V + 1)/2 - 2)/ϕ = (23V + 19)/2ϕ (s)
V: reload valueϕ: peripheral clock (PCLK)

Consequently, the minimum receivable baud rate at the transmission destination (BGmin) is as follows:

BGmin = 11/FLmax = 22ϕ/(23V+19) (bps)
V: reload valueϕ: peripheral clock (PCLK)

Based on the aforementioned calculation formulas for the minimum/maximum baud rates, the acceptable baud rate error between the UART and transmission destination can be calculated as shown below.

Table 24-14. Acceptable Baud Rate Error

Reload value (V)	Maximum acceptable baud rate error	Minimum acceptable baud rate error
3	0%	0
10	+2.98%	-2.81%
50	+4.37%	-4.02%
100	+4.56%	-4.18%
200	+4.66%	-4.26%
32767	+4.76%	-4.35%

Note: The accuracy of reception depends on the number of bits per frame, the peripheral clock (PCLK) and the reload value. The accuracy becomes higher as the peripheral clock (PCLK) and the division ratio become higher.

External clock

The baud rate generator divides the external clock, when "1" is written to the EXT bit in the baud rate generator register 1, 0 (BGR1, BGR0).

Note: The UART synchronizes external clock signals with the internal clock. Therefore, the operation becomes unstable when an external clock which cannot be synchronized is used.

Functions of reload counters

There are two reload counters, a transmission reload counter and a reception reload counter, which function as a dedicated baud rate generator. Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the external or internal clock.

Starting a count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

Restart

The reload counter restarts under the following conditions.

- For both transmission and reception reload counters
  Programmable reset (SCR:UPCL bit)
- For reception reload counter
  Detecting the falling edge of a start bit in asynchronous mode

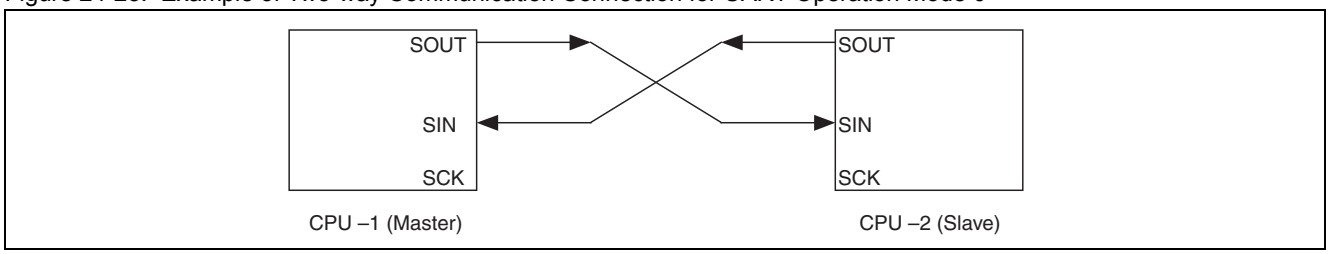
24.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)

Asynchronous serial two-way communication is enabled in operation mode 0.

Connection between CPUs

Two-way communication should be selected for operation mode 0 (normal mode). Two CPUs are connected to each other, as shown in Figure 24-23.

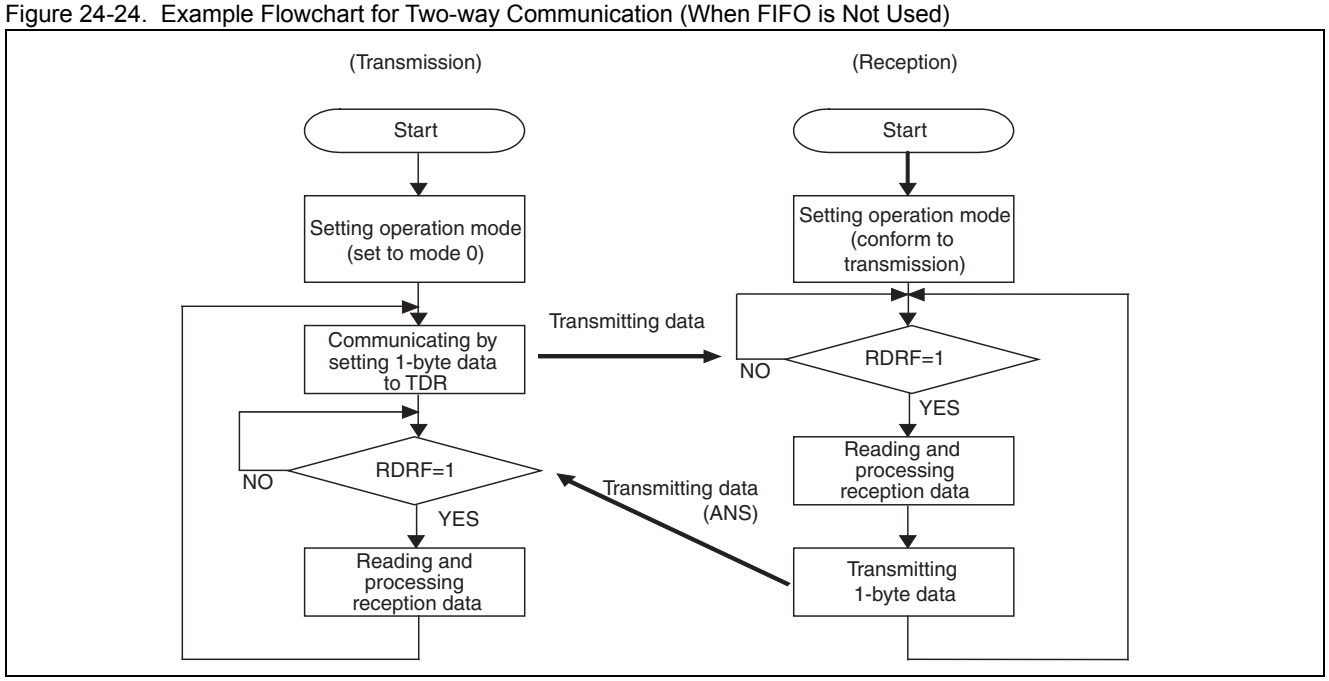
Figure 24-23. Example of Two-way Communication Connection for UART Operation Mode 0



Flowchart

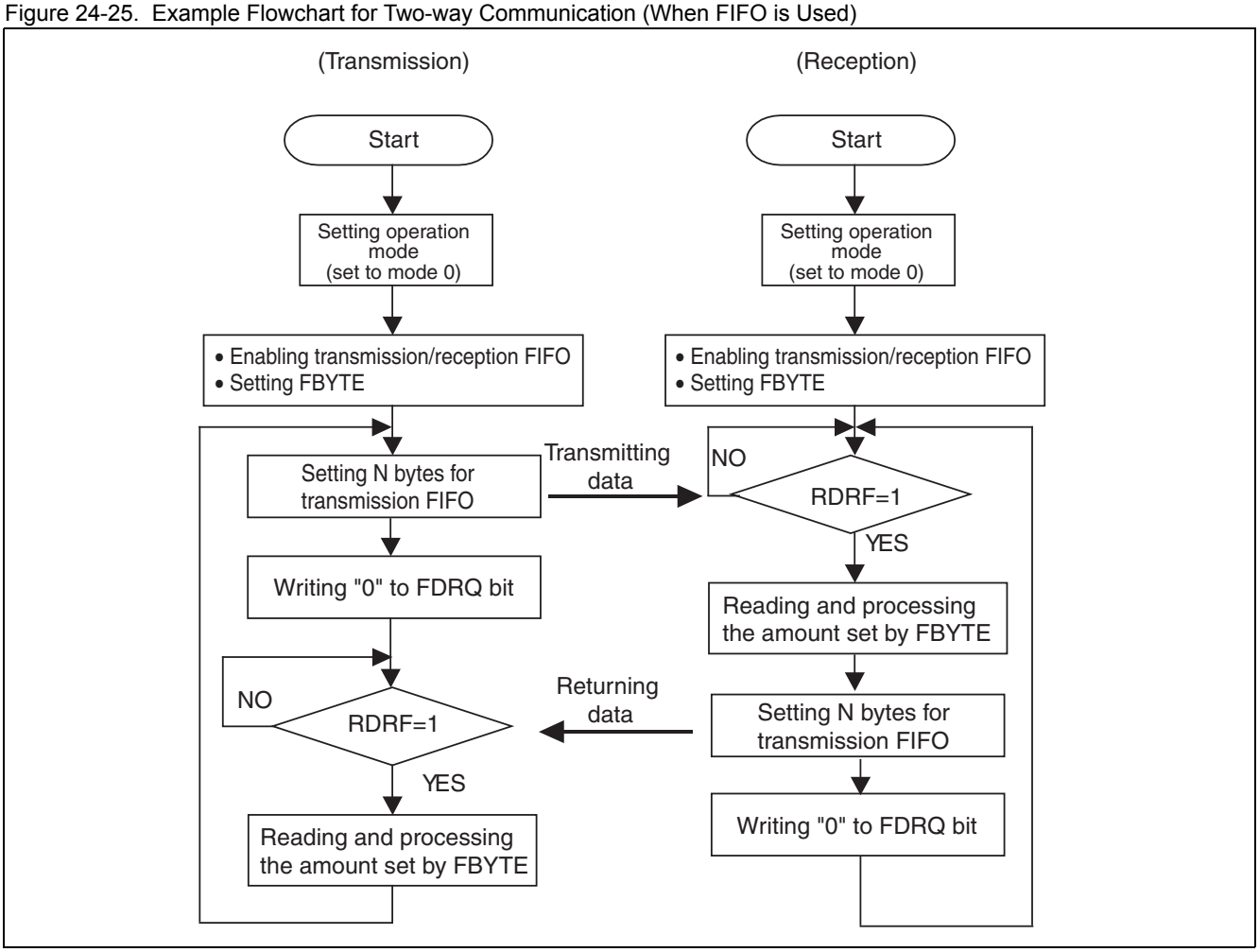
- When FIFO is not used

Figure 24-24. Example Flowchart for Two-way Communication (When FIFO is Not Used)



■ When FIFO is used

Figure 24-25. Example Flowchart for Two-way Communication (When FIFO is Used)



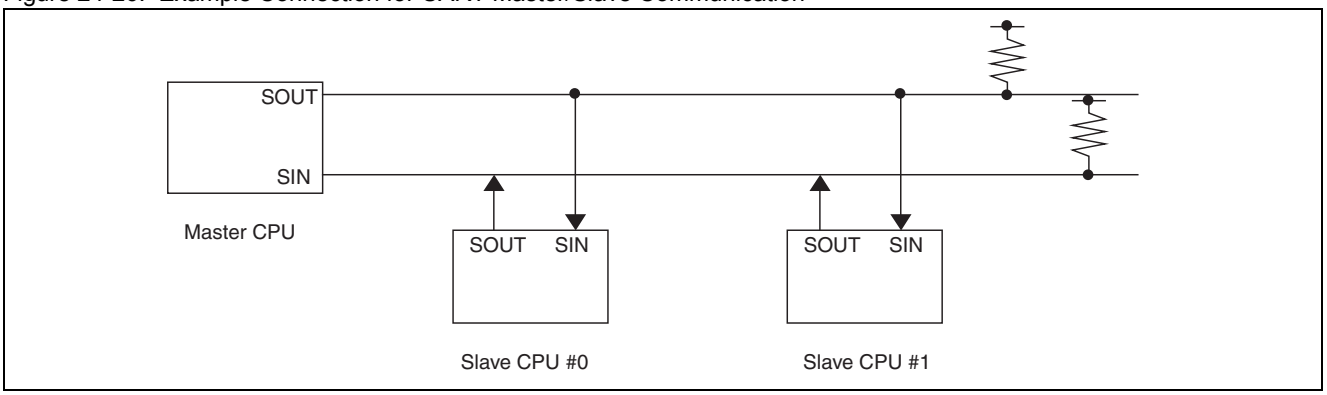
24.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

In operation mode 1 (multi-processor mode), communication among multiple CPUs is enabled through master/slave connection. The connected CPUs can be used as a master/slave.

Connection among CPUs

In this master/slave communication, one master CPU and more than one slave CPU are connected to two common communication lines, as shown in Figure 24-26, to configure a communication system. The UART can be used by both the master and slaves.

Figure 24-26. Example Connection for UART Master/Slave Communication



Function selection

For master/slave communication, select the operation mode and data transfer system shown in Table 24-15.

Table 24-15. Selection of Master/Slave Communication Function

	Operation mode		Data	Parity	Stop bit	Bit direction
	Master CPU	Slave CPU				
Address transmission/reception	Mode 1 (AD bit transmission)	Mode 1 (AD bit reception)	AD = 1 + 7-bit or 8-bit address	None	1 bit to 4 bits	LSB first or MSB first
Data transmission/reception			AD = 0 + 7-bit or 8-bit data			

Note: Use half word access for transmission/reception data (RDR/TDR) in operation mode 1.

- Communication procedure

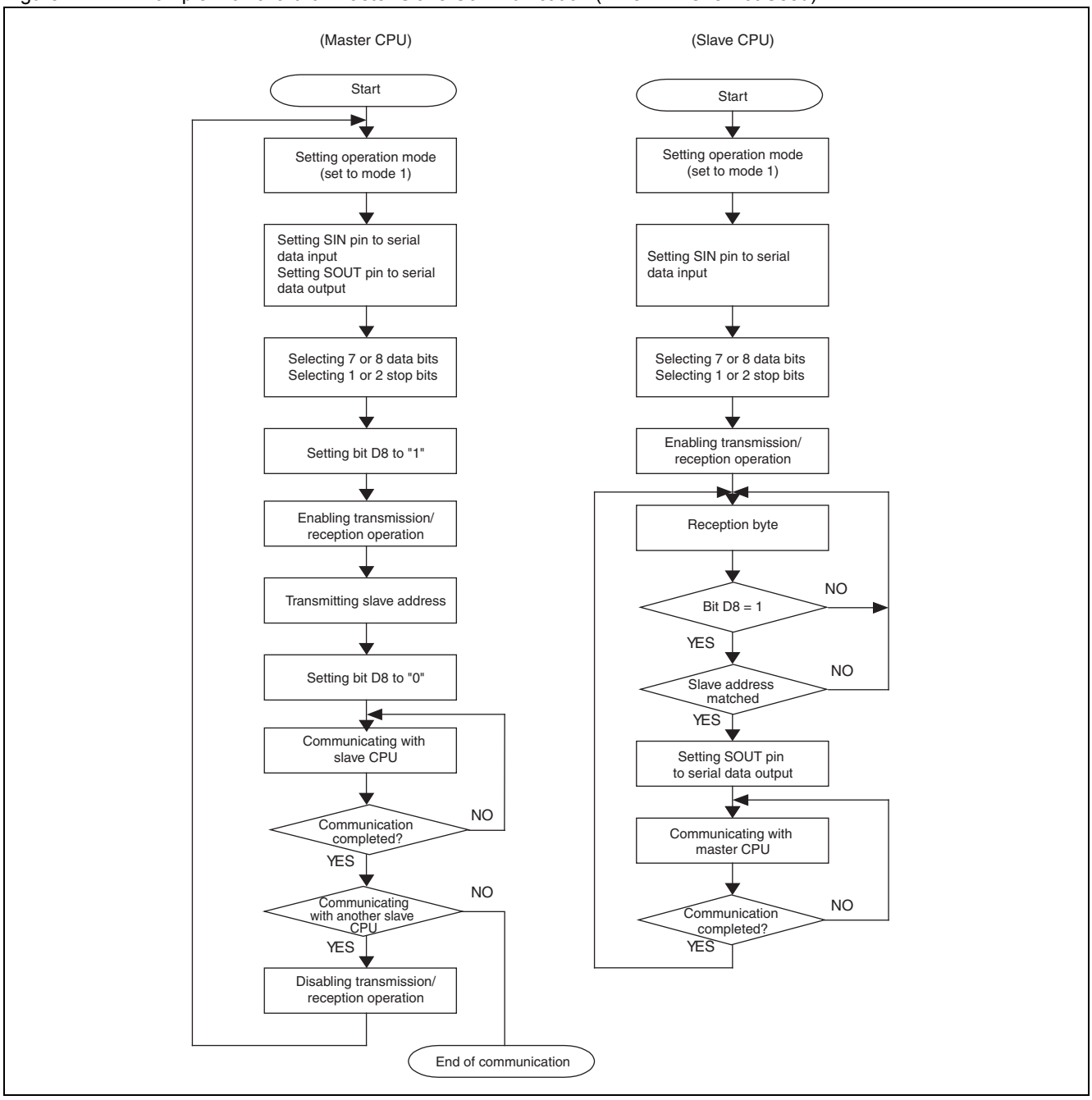
  Communication is started when the master CPU transmits address data, where bit D8 is treated as "1". This data is used to select a slave CPU which will be the communication destination. Each slave CPU judges the address data on a program, and communicates (normal data) with the master CPU when the data matches its assigned address.

  Figure 24-27 and Figure 24-28 show flowcharts for the master/slave communication (multi-processor mode).

Flowcharts

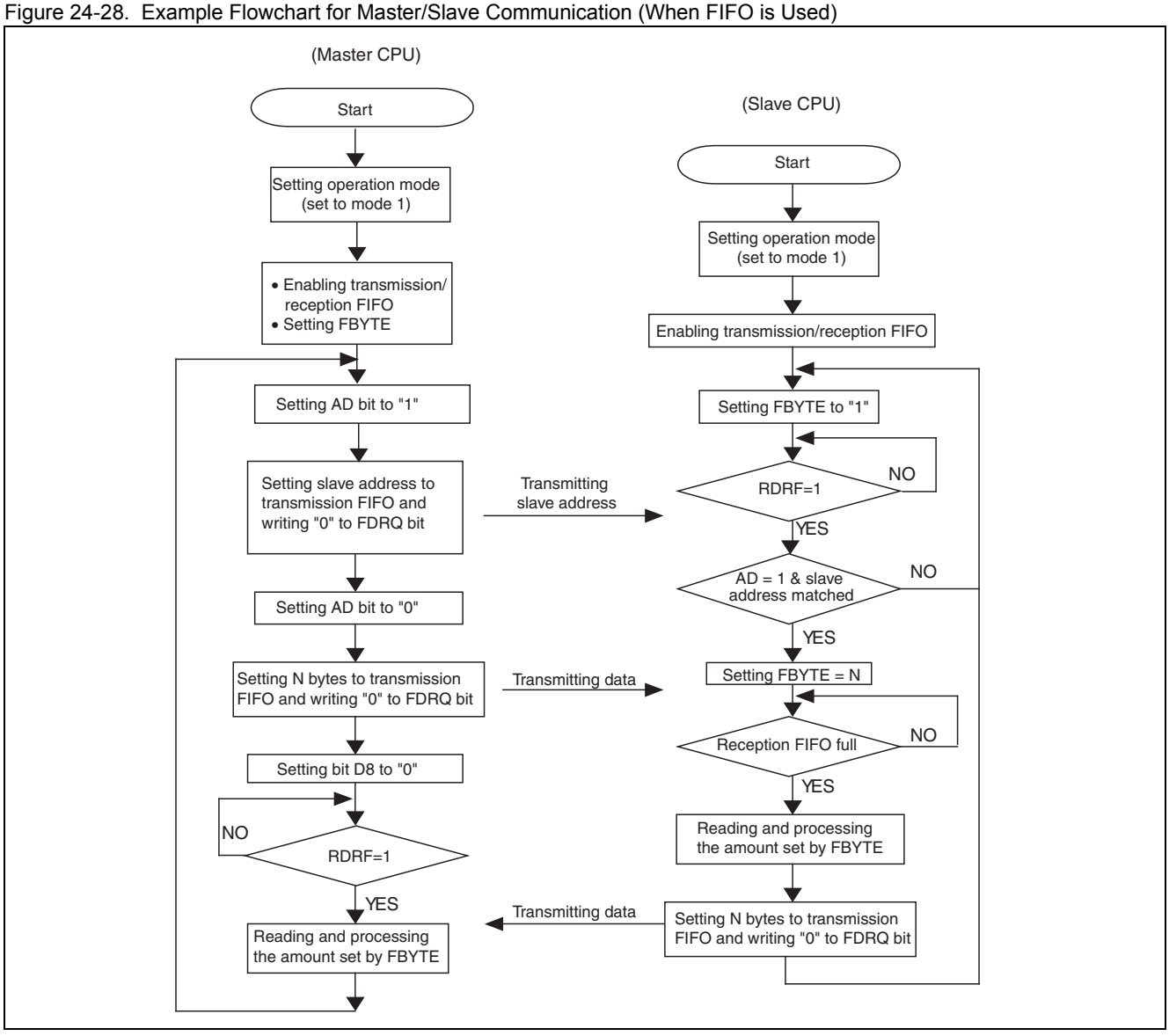
- When FIFO is not used:

Figure 24-27. Example Flowchart for Master/Slave Communication (When FIFO is Not Used)



■ When FIFO is used

Figure 24-28. Example Flowchart for Master/Slave Communication (When FIFO is Used)



24.10 Notes on UART Mode

The notes for when you use the UART mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.

24.11 CSIO (Clock Synchronous Serial Interface)

Among all the functions of the multi-function serial interface, this section describes the CSIO functions that are supported in operation mode 2.

- CSIO (Clock Synchronous Serial Interface)
- Overview of CSIO (Clock Synchronous Serial Interface)
- Registers of CSIO (Clock Synchronous Serial Interface)
  - Serial Control Register (SCR)
  - Serial Mode Register (SMR)
  - Serial Status Register (SSR)
  - Extended Serial Control Register (ESCR)
  - Reception Data Register / Transmission Data Register (RDR/TDR)
  - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
  - FIFO Control Register 1 (FCR1)
  - FIFO Control Register 0 (FCR0)
  - FIFO Byte Register (FBYTE1/FBYTE2)
  - Serial mode selection registers (SSEL0123)
  - Reception data mirror registers/ transmission data mirror registers (RDRM/TDRM)
- Interrupts of CSIO (Clock Synchronous Serial Interface)
  - Occurrence of Reception Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing
  - Occurrence of Transmission Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing
- Operation of CSIO (Clock Synchronous Serial Interface)
- Dedicated Baud Rate Generator
  Setting Baud Rate
- Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

24.12 Overview of CSIO (Clock Synchronous Serial Interface)

CSIO (Clock Synchronous Serial Interface) is a general-purpose interface for serial data communication, which allows synchronous communications with external units (SPI supported). In addition, this interface comes with transmission/reception FIFO (up to 16 bytes each).

Functions of CSIO (Clock Synchronous Serial Interface)

		Function
1	Data buffer	■ Full-duplex double buffer (when FIFO is not used) ■ Transmission/reception FIFO (up to 16 bytes each) (when FIFO is used)[1]
2	Transfer system	■ Clock synchronization (no start bit / no stop bit) ■ Master/slave function ■ SPI supported (both master & slaves supported)
3	Baud rate	■ Dedicated baud rate generator available (15-bit reload counter configuration, in master operation) ■ External clock can be input (in slave operation)
4	Data length	Variable from 5 bits to 9 bits
5	Reception error detection	Overrun error
6	Interrupt request	■ Reception interrupt (completion of reception, overrun error) ■ Transmission interrupt (transmission data empty, transmission bus idle) ■ Transmission FIFO interrupt (when transmission FIFO is empty) ■ DMA transfer support function for transmission and reception
7	Synchronous mode	Master or slave function
8	Pin access	Serial data output pin can be set to "H".
9	4-channel simultaneous communication	4-channel simultaneous communication is available for ch.0 to ch.3.
10	FIFO options	■ Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes)* ■ Transmission FIFO or reception FIFO selectable ■ Transmission data can be resent. ■ The interrupt timing for reception FIFO can be modified by software. ■ FIFO reset is supported separately.

[1]: There is no FIFO between ch.0 and ch.3

24.13 Registers of CSIO (Clock Synchronous Serial Interface)

This section lists the registers of CSIO (clock synchronous serial interface).

List of registers of CSIO (Clock Synchronous Serial Interface)

Table 24-16. List of Registers of CSIO (Clock Synchronous Serial Interface) (Sheet 1 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
Common to 0 to 3	SSEL0123	Serial mode select register 0123	24.13.10
0	SCR0	Serial control register 0	24.13.1
	SMR0	Serial mode register 0	24.13.2
	ESCR0	Extended serial control register 0	24.13.4
	BGR0	Baud rate generator register 0	24.13.6
	SSR0	Serial status register 0	24.13.3
	RDR0	Received data register 0	24.13.5
	TDR0	Transmitted data register 0	24.13.5
	RDRM0	Received data mirror register 0	24.13.11
	TDRM0	Transmitted data mirror register 0	24.13.11
1	SCR1	Serial control register 1	24.13.1
	SMR1	Serial mode register 1	24.13.2
	ESCR1	Extended serial control register 1	24.13.4
	BGR1	Baud rate generator register 1	24.13.6
	SSR1	Serial status register 1	24.13.3
	RDR1	Received data register 1	24.13.5
	TDR1	Transmitted data register 1	24.13.5
	RDRM1	Received data mirror register 1	24.13.11
	TDRM1	Transmitted data mirror register 1	24.13.11
2	SCR2	Serial control register 2	24.13.1
	SMR2	Serial mode register 2	24.13.2
	ESCR2	Extended serial control register 2	24.13.4
	BGR2	Baud rate generator register 2	24.13.6
	SSR2	Serial status register 2	24.13.3
	RDR2	Received data register 2	24.13.5
	TDR2	Transmitted data register 2	24.13.5
	RDRM2	Received data mirror register 2	24.13.11
	TDRM2	Transmitted data mirror register 2	24.13.11
3	SCR3	Serial control register 3	24.13.1
	SMR3	Serial mode register 3	24.13.2
	ESCR3	Extended serial control register 3	24.13.4
	BGR3	Baud rate generator register 3	24.13.6
	SSR3	Serial status register 3	24.13.3
	RDR3	Received data register 3	24.13.5
	TDR3	Transmitted data register 3	24.13.5
	RDRM3	Received data mirror register 3	24.13.11
	TDRM3	Transmitted data mirror register 3	24.13.11

Table 24-16. List of Registers of CSIO (Clock Synchronous Serial Interface) (Sheet 2 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
8	SCR8	Serial control register 8	24.13.1
	SMR8	Serial mode register 8	24.13.2
	ESCR8	Extended serial control register 8	24.13.4
	BGR8	Baud rate generator register 8	24.13.6
	SSR8	Serial status register 8	24.13.3
	RDR8	Received data register 8	24.13.5
	TDR8	Transmitted data register 8	24.13.5
	FCR18	FIFO control register 18	24.13.7
	FCR08	FIFO control register 08	24.13.8
	FBYTE18	FIFO1 byte register 8	24.13.9
	FBYTE28	FIFO2 byte register 8	24.13.9
9	SCR9	Serial control register 9	24.13.1
	SMR9	Serial mode register 9	24.13.2
	ESCR9	Extended serial control register 9	24.13.4
	BGR9	Baud rate generator register 9	24.13.6
	SSR9	Serial status register 9	24.13.3
	RDR9	Received data register 9	24.13.5
	TDR9	Transmitted data register 9	24.13.5
	FCR19	FIFO control register 19	24.13.7
	FCR09	FIFO control register 09	24.13.8
	FBYTE19	FIFO1 byte register 9	24.13.9
	FBYTE29	FIFO2 byte register 9	24.13.9
10	SCR10	Serial control register 10	24.13.1
	SMR10	Serial mode register 10	24.13.2
	ESCR10	Extended serial control register 10	24.13.4
	BGR10	Baud rate generator register 10	24.13.6
	SSR10	Serial status register 10	24.13.3
	RDR10	Received data register 10	24.13.5
	TDR10	Transmitted data register 10	24.13.5
	FCR110	FIFO control register 110	24.13.7
	FCR010	FIFO control register 010	24.13.8
	FBYTE110	FIFO1 byte register 10	24.13.9
	FBYTE210	FIFO2 byte register 10	24.13.9
11	SCR11	Serial control register 11	24.13.1
	SMR11	Serial mode register 11	24.13.2
	ESCR11	Extended serial control register 11	24.13.4
	BGR11	Baud rate generator register 11	24.13.6
	SSR11	Serial status register 11	24.13.3
	RDR11	Received data register 11	24.13.5
	TDR11	Transmitted data register 11	24.13.5
	FCR111	FIFO control register 111	24.13.7
	FCR011	FIFO control register 011	24.13.8
	FBYTE111	FIFO1 byte register 11	24.13.9
	FBYTE211	FIFO2 byte register 11	24.13.9

Table 24-17. Bit Assignment of CSIO (Clock Synchronous Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

24.13.1 Serial Control Register (SCR)

The serial control register (SCR) enables/disables transmission/reception interrupts, transmission idle interrupts and transmission/reception operations. This register can also set SPI connection and reset CSIO.

Serial Control Register (SCR)

Figure 24-29 shows the bit structure of the serial control register (SCR), and Table 24-18 describes the function of each bit.

Figure 24-29. Bit Structure of Serial Control Register (SCR)

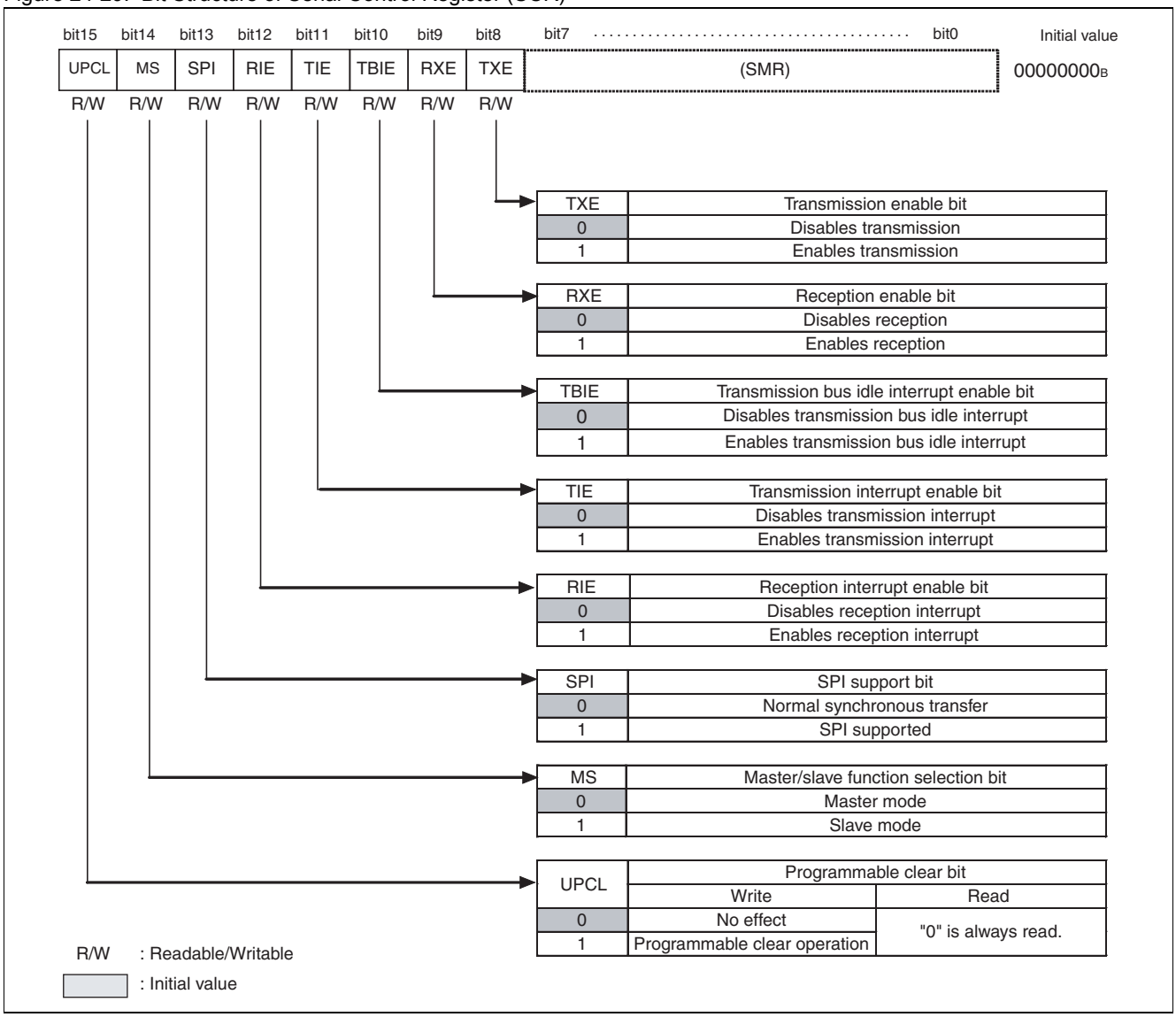


Table 24-18. Functional Description of Each Bit of Serial Control Register (SCR)

Bit name		Function
bit15	UPCL: Programmable clear bit	This bit is used to initialize the internal state of the CSIO. Setting the bit to "1": ■ The CSIO will be reset directly (software reset). The register setting, however, will be retained. In this case, communication of the data which is being transmitted or received will be cut off immediately. ■ The baud rate generator will reload the value set in BGR1/BGR0 registers, and then restart the operation. ■ All the transmission/reception interrupt sources (TDRE, TBI, RDRF and ORE) will be initialized ("1100_B"). ■ Setting the bit to "0": No effect on the operation ■ Reading this bit always returns "0". **Note:** Execute the programmable clear operation after disabling interrupts. Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.
bit14	MS: Master/slave function selection bit	This bit is used to select master or slave mode. Setting the bit to "0" selects master mode. Setting the bit to "1" selects slave mode. **Note:** The external clock will be input directly, if SMR:SCKE is set to "0" when slave mode is selected.
bit13	SPI: SPI support bit	This bit is used to enable communication supporting SPI. Setting the bit to "0" enables normal synchronous communication. Setting the bit to "1" enables SPI support.
bit12	RIE: Reception interrupt enable bit	■ This bit is used to enable/disable the output of reception interrupt requests to the CPU. ■ A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or when any of the error flag bits (ORE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	■ This bit is used to enable/disable the output of transmission interrupt requests to the CPU. ■ A transmission interrupt request is output when the TIE and TDRE bits are set to "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	■ This bit is used to enable/disable the output of transmission bus idle interrupt requests to the CPU. ■ A transmission bus idle interrupt request is output when the TBIE and TBI bits are set to "1".
bit9	RXE: Reception enable bit	This bit is used to enable/disable CSIO reception operation. Setting the bit to "0" disables data frame reception operation. Setting the bit to "1" enables data frame reception operation. **Note:** If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.
bit8	TXE: Transmission enable bit	This bit is used to enable/disable CSIO transmission operation. Setting the bit to "0" disables data frame transmission operation. Setting the bit to "1" enables data frame transmission operation. **Note:** If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.

24.13.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, selects the transfer direction and serial clock inversion, and enables or disables the output to the serial data and serial clock pins.

Serial Mode Register (SMR)

Figure 24-30 shows the bit structure of the serial mode register (SMR), and Table 24-19 describes the function of each bit.

Figure 24-30. Bit Structure of Serial Mode Register (SMR)

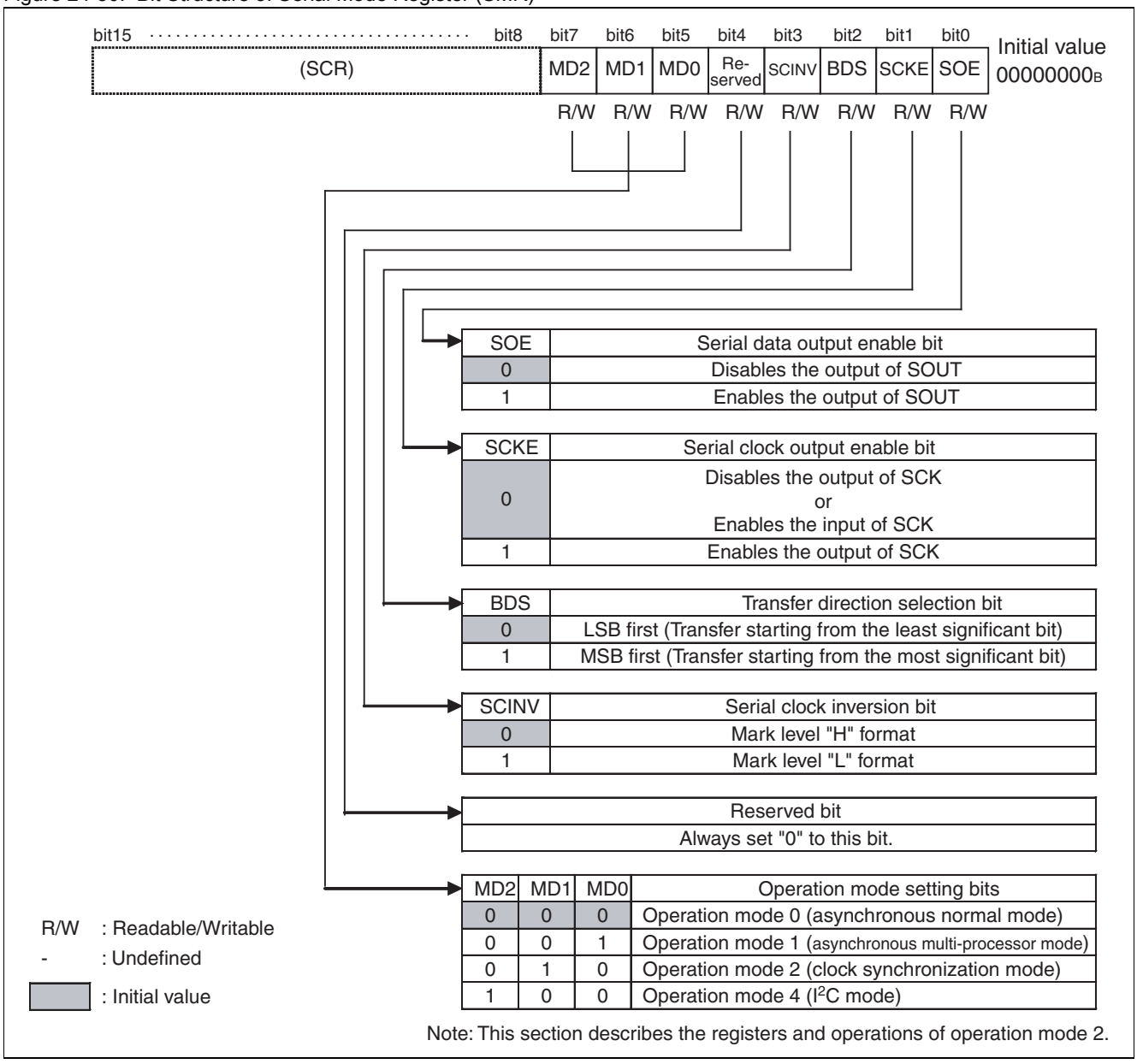


Table 24-19. Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	These bits are used to select the operation mode. "000_B":Selects operation mode 0 (asynchronous normal mode) "001_B":Selects operation mode 1 (asynchronous multi-processor mode) "010_B":Selects operation mode 2 (clock synchronization mode) "100_B":Selects operation mode 4 (I^2C mode) This section describes the registers and operations of operation mode 2 (clock synchronization mode). **Note:** Settings other than above are prohibited. To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1). And then, after setting the operation mode, set each register.
bit4	Reserved bit	Always set "0" to this bit.
bit3	SCINV: Serial clock inversion bit	This bit is used to invert the serial clock format. Setting the bit to "0": ■ Changes the mark level of the serial clock output to "H". ■ Transmission data is output, being synchronized with the falling edge (normal transfer) or the rising edge (SPI transfer) of the serial clock. ■ Reception data is sampled at the rising edge (normal transfer) or the falling edge (SPI transfer) of the serial clock. Setting the bit to "1": ■ Changes the mark level of the serial clock output to "L". ■ Transmission data is output, being synchronized with the rising edge (normal transfer) or the falling edge (SPI transfer) of the serial clock. ■ Reception data is sampled at the falling edge (normal transfer) or the rising edge (SPI transfer) of the serial clock. **Note:** Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit2	BDS: Transfer direction selection bit	This bit is used to determine the transfer priority for transfer serial data: whether the least significant bit should be transferred first (LSB first, BDS = 0) or the most significant bit should be transferred first (MSB first, BDS = 1). **Note:** Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit1	SCKE: Serial clock output enable bit	This bit is used to control the I/O port of the serial clock. Setting the bit to "0": ■ The output of SCK "H" or the input of SCK will be enabled. To use it as a SCK input, set a general-purpose I/O port as the input port. Setting the bit to "1" enables the output of SCK.
bit0	SOE: Serial data output enable bit	This bit is used to enable/disable the output of serial data. Setting the bit to "0" enables the output of the "H" level of SOUT. Setting the bit to "1" enables the output of SOUT.

Note:

The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed.

- Serial Control Register (SCR)
- Extended Serial Control Register (ESCR)

Note: Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

24.13.3 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status, and also checks and clears the reception error flag.

Serial Status Register (SSR)

Figure 24-31 shows the bit structure of the serial status register (SSR) and Table 24-20describes the function of each bit.

Figure 24-31. Bit Structure of Serial Status Register (SSR)

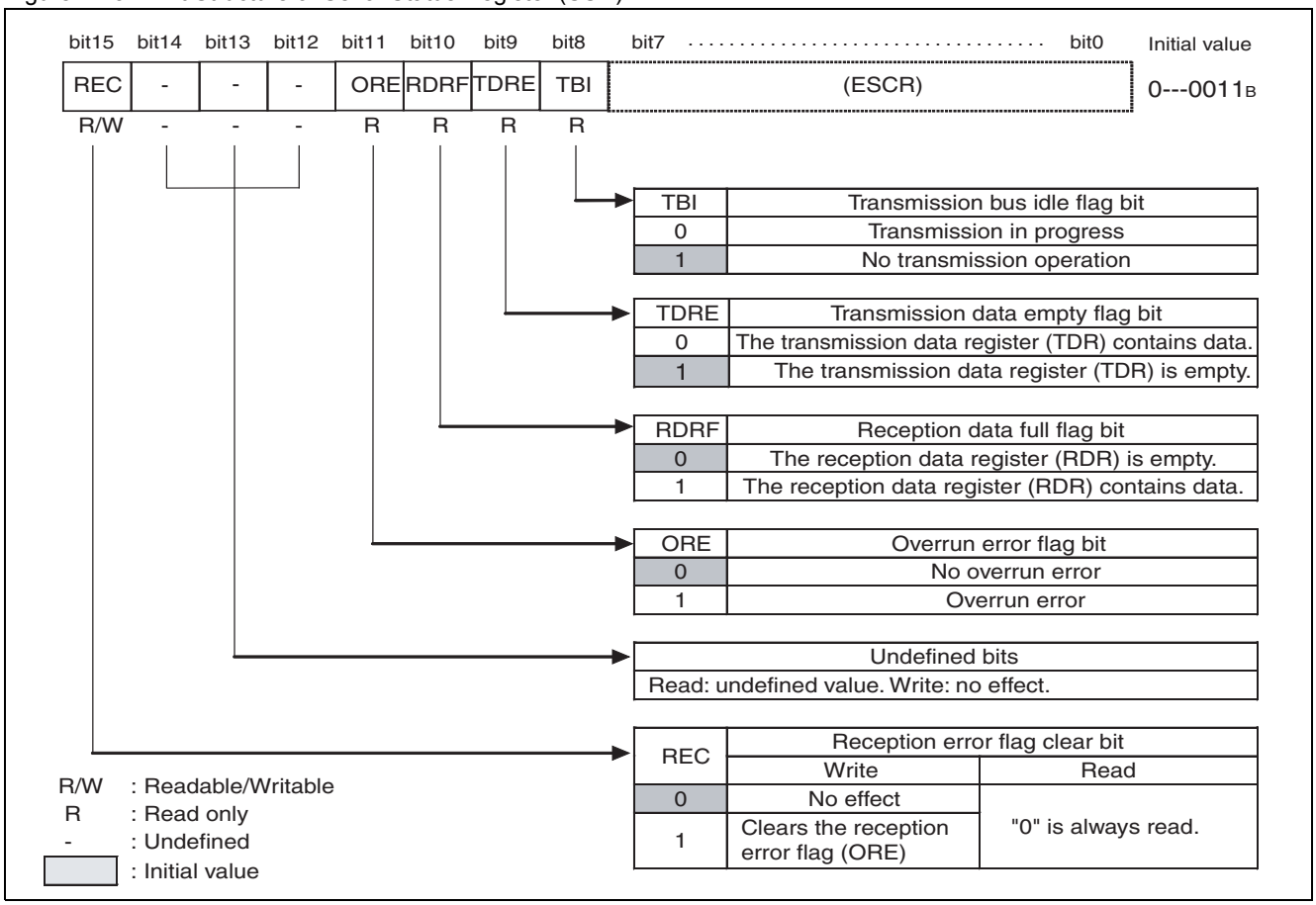


Table 24-20. Functional Description of Each Bit of Serial Status Register (SSR)

Bit name		Function
bit15	REC: Reception error flag clear bit	This bit is used to clear the ORE flag in the serial status register (SSR). ■ Writing "1" clears the error flag. ■ Writing "0" has no effect. Reading this bit always returns "0".
bit14 to bit12	Undefined bits	Read:undefined value Write:no effect
bit11	ORE: Overrun error flag bit	■ This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). ■ A reception interrupt request is output when the ORE and RIE bits are set to "1". ■ When this flag is set, the data in the reception data register (RDR) is invalid. ■ If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
bit10	RDRF: Reception data full flag bit	■ This flag indicates the status of the reception data register (RDR). ■ The bit is set to "1" when reception data is loaded to RDR. The bit is cleared to "0" when the reception data register (RDR) is read. ■ A reception interrupt request is output when the RDRF and RIE bits are set to "1". ■ RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. ■ During the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO. If RDR is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. ■ This bit is cleared to "0" when the reception FIFO, if used, becomes empty.
bit9	TDRE: Transmission data empty flag bit	■ This flag indicates the status of the transmission data register (TDR). ■ When transmission data is written to TDR, the bit becomes "0", indicating that TDR contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR no longer contains any valid data. ■ A transmission interrupt request is output when the TDRE and TIE bits are set to "1". ■ The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". ■ For information about the set/reset timings of the TDRE bit for when the transmission FIFO is used, refer to Section "24.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".
bit8	TBI: Transmission bus idle flag bit	■ This bit indicates that the CSIO is not performing transmission operation. ■ The bit is set to "0" when data is written to the transmission data register (TDR). ■ The bit is set to "1" when the transmission data register (TDR) is empty (TDRE = 1) and no transmission operation is in progress. ■ The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". ■ A transmission interrupt request is output when this bit is "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).

24.13.4 Extended Serial Control Register (ESCR)

The extended serial control register (ESCR) can be used to set the transmission/reception data length, select data transmission/reception wait, and fix the serial output to "H".

Bit structure of the extended serial control register (ESCR)

Figure 24-32shows the bit structure of the extended serial control register (ESCR) and Table 24-21 describes the function of each bit.

Figure 24-32. Bit Structure of Extended Serial Control Register (ESCR)

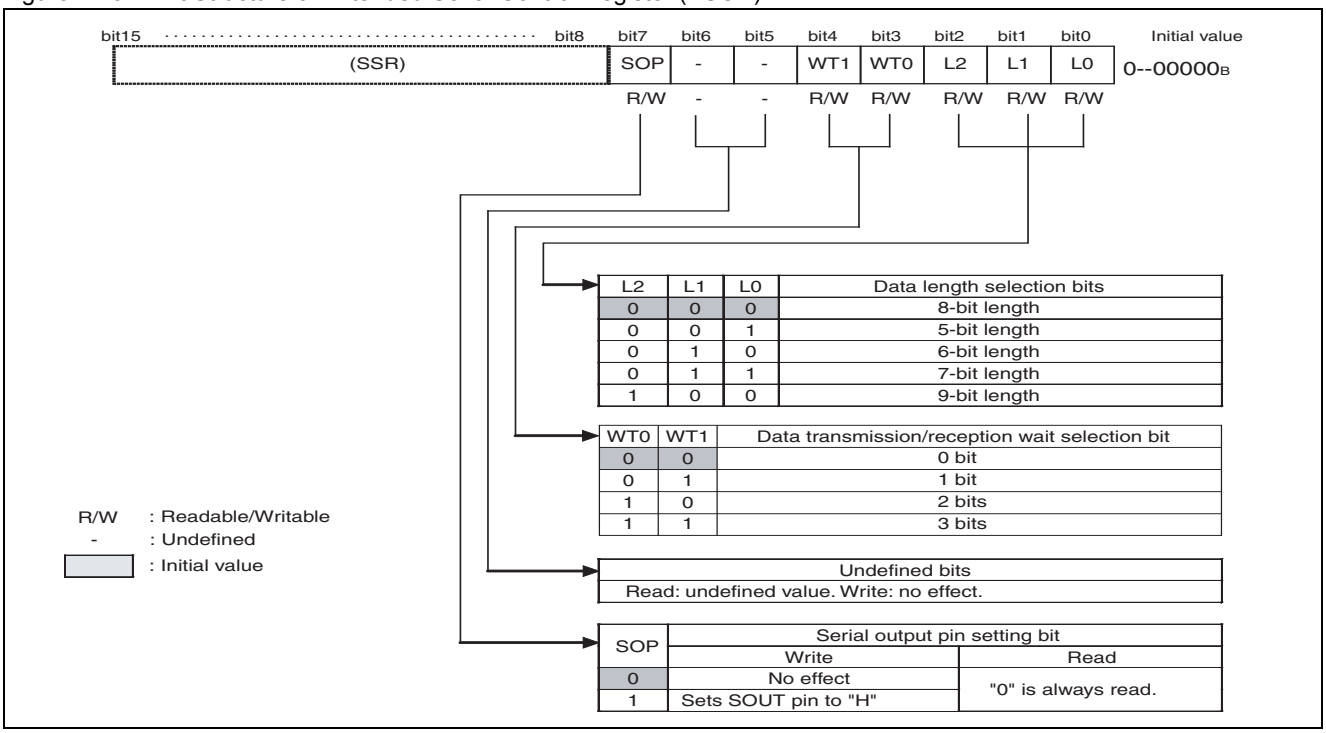


Table 24-21. Functional Description of Each Bit of Extended Serial Control Register (ESCR)

Bit name		Function
bit7	SOP: Serial output pin setting bit	■ This bit is used to set the serial output pin to "H". The SOUT pin is set to "H" when "1" is written to this bit. It is not necessary to write "0" to this bit after that. ■ Reading this bit always returns "0". **Note:** Do not set this bit during serial data transmission.
bit6, bit5	Undefined bits	Read:undefined value Write:no effect
bit4, bit3	WT1,WT0: Data transmission/ reception wait selection bit	In master mode, the wait number is specified to transmission/reception of sequential data. In slave mode, it will be "00" operation. ■ If "00_B" is set, SCK is output continuously. ■ If "01_B" is set, SCK is output after 1 bit time wait. ■ If "10_B" is set, SCK is output after 2 bits time wait. ■ If "11_B" is set, SCK is output after 3 bits time wait.
bit2 to bit0	L2 to L0: Data length selection bits	These bits are used to specify a data length for transmission/reception data. Selecting "000_B" sets the data length to 8 bits. Selecting "001_B" sets the data length to 5 bits. Selecting "010_B" sets the data length to 6 bits. Selecting "011_B" sets the data length to 7 bits. Selecting "100_B" sets the data length to 9 bits. **Note:** Settings other than above are prohibited.

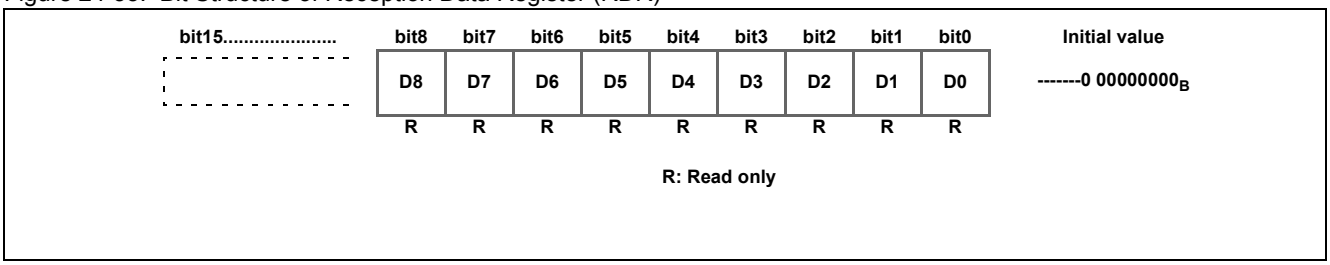
24.13.5 Reception Data Register / Transmission Data Register (RDR/TDR)

The reception data register and transmission data register are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

Reception Data Register (RDR)

Figure 24-33 illustrates the bit structure of the serial reception register (RDR).

Figure 24-33. Bit Structure of Reception Data Register (RDR)



The reception data register (RDR) is a 9-bit data buffer register for serial data reception.

- A serial data signal sent to a serial input pin (SIN pin) is converted through the shift register and then stored in the reception data register (RDR).
- "0" is placed in upper bits, as shown below, in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

X: Reception data bit

- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR). A reception interrupt request will be generated if reception interrupts have been enabled (SSR: RIE = 1).
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is cleared to "0" automatically, when the serial reception data register (RDR) is read.
- If a reception error occurs (SSR:ORE), the data in the reception data register (RDR) becomes invalid.
- 16-bit access is used to read RDR for a 9-bit transfer.

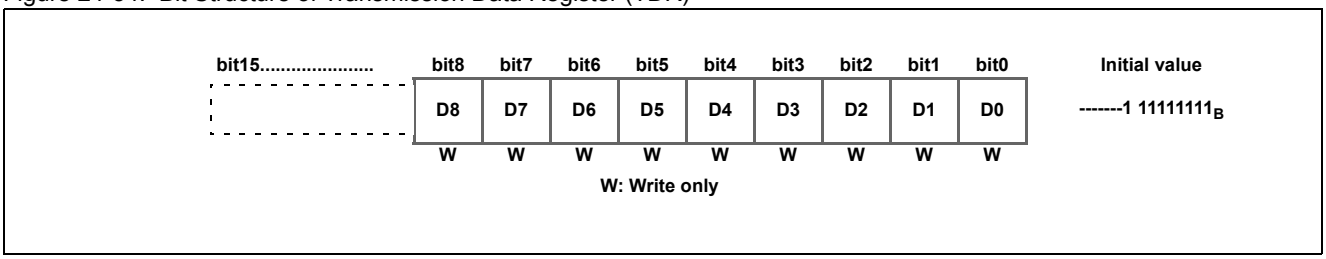
Notes:

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.
- If a reception error occurs (SSR:ORE = 1) when the reception FIFO is used, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.

Transmission Data Register (TDR)

Figure 24-34. illustrates the bit structure of the transmission data register.

Figure 24-34. Bit Structure of Transmission Data Register (TDR)



The transmission data register (TDR) is a 9-bit data buffer register for serial data transmission.

- If transmission data is written to the transmission data register (TDR) when transmission operation is enabled (SCR:TXE = 1), the transmission data will be transferred to the transmission shift register, converted into serial data and then sent from a serial data output pin (SOUT pin).
- As shown below, data becomes invalid from the upper bit in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

X: Transmission data bit

- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) will be set to "1" when transmission data is transferred to the transmission shift register and the transmission starts.
- The next transmission data can be written when the transmission data empty flag (SSR:TDRE) is set to "1". A transmission interrupt will occur if transmission interrupts have been enabled. Write the next transmission data by generating a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is set to "1".
- Transmission data cannot be written to the transmission data register (TDR) when the transmission data empty flag (SSR:TDRE) is set to "0" and the transmission FIFO is either disabled or full.
- 16-bit access is used to write to TDR for a 9-bit transfer.

Notes:

- The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. The two registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.
- For information about the timing for setting the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, refer to Section "24.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".

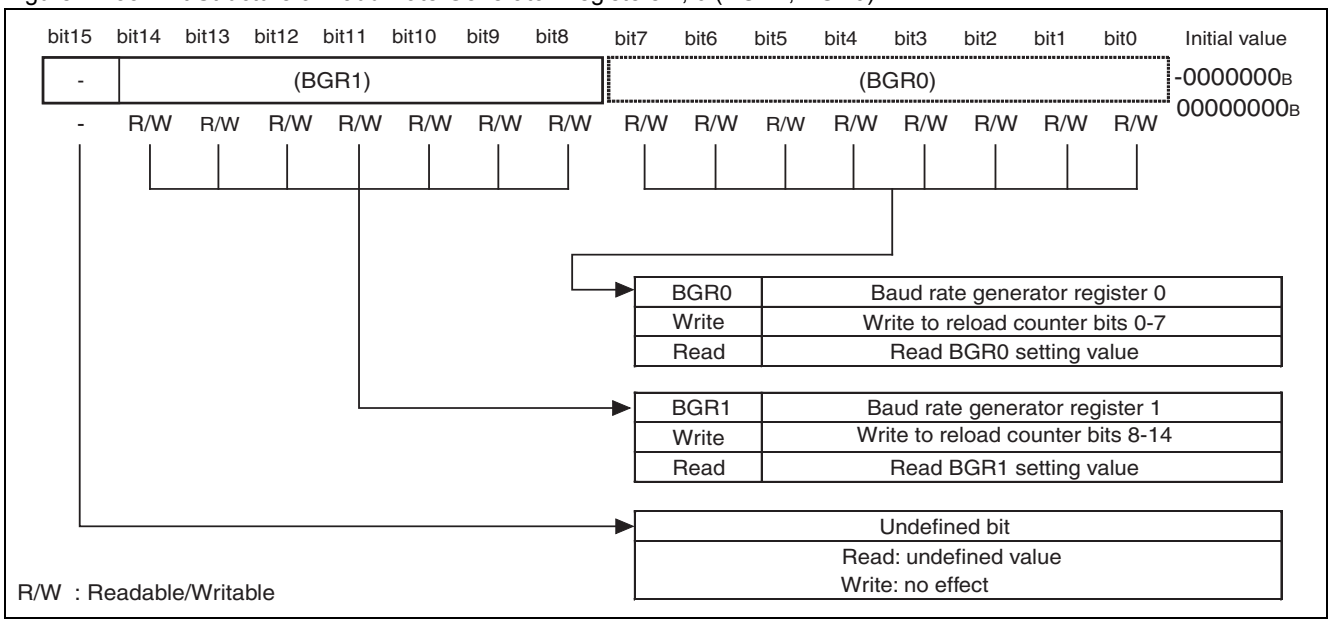
24.13.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock.

Bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0)

Figure 24-35 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 24-35. Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)



- A value is set to the baud rate generator registers 1, 0 (BGR1, BGR0).
- BGR0 and BGR1 correspond to the lower bits and upper bits respectively and they can write a reload value to be counted as well as read BGR0/BGR1 setting values.
- The reload counter starts counting when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the setting of the SCINV bit. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
  - SCINV = 0: The "H" width of the serial clock is 1 peripheral clock (PCLK) cycle longer.
  - SCINV = 1: The "L" width of the serial clock is 1 peripheral clock (PCLK) cycle longer.
- Select 1 or a larger number for the reload value. However, select 3 or a larger value for the reload value of the CSIO which will become the master, when using these CSIO's as the master and slave.
- When a setting value of the baud rate generator registers 1, 0 (BGR1, BGR0) is changed, the new setting value is not reloaded until the counter value becomes "0000_H". To make the new setting value valid immediately, therefore, execute a CSIO reset (UPCL) after changing the BGR0/BGR1 setting value.
- Set a baud rate to BGR0/BGR1 during the use of reception FIFO to set the reception FIFO idle detection enable bit (FCR1:FRIIE) to "1" and operate in slave mode.

24.13.7 FIFO Control Register 1 (FCR1)

The FIFO control register 1 (FCR1) selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

Bit structure of FIFO control register 1 (FCR1)

Figure 24-36 shows the bit structure of the FIFO control register 1 (FCR1) and Table 24-22 describes the function of each bit.

Figure 24-36. Bit Structure of FIFO Control Register 1 (FCR1)

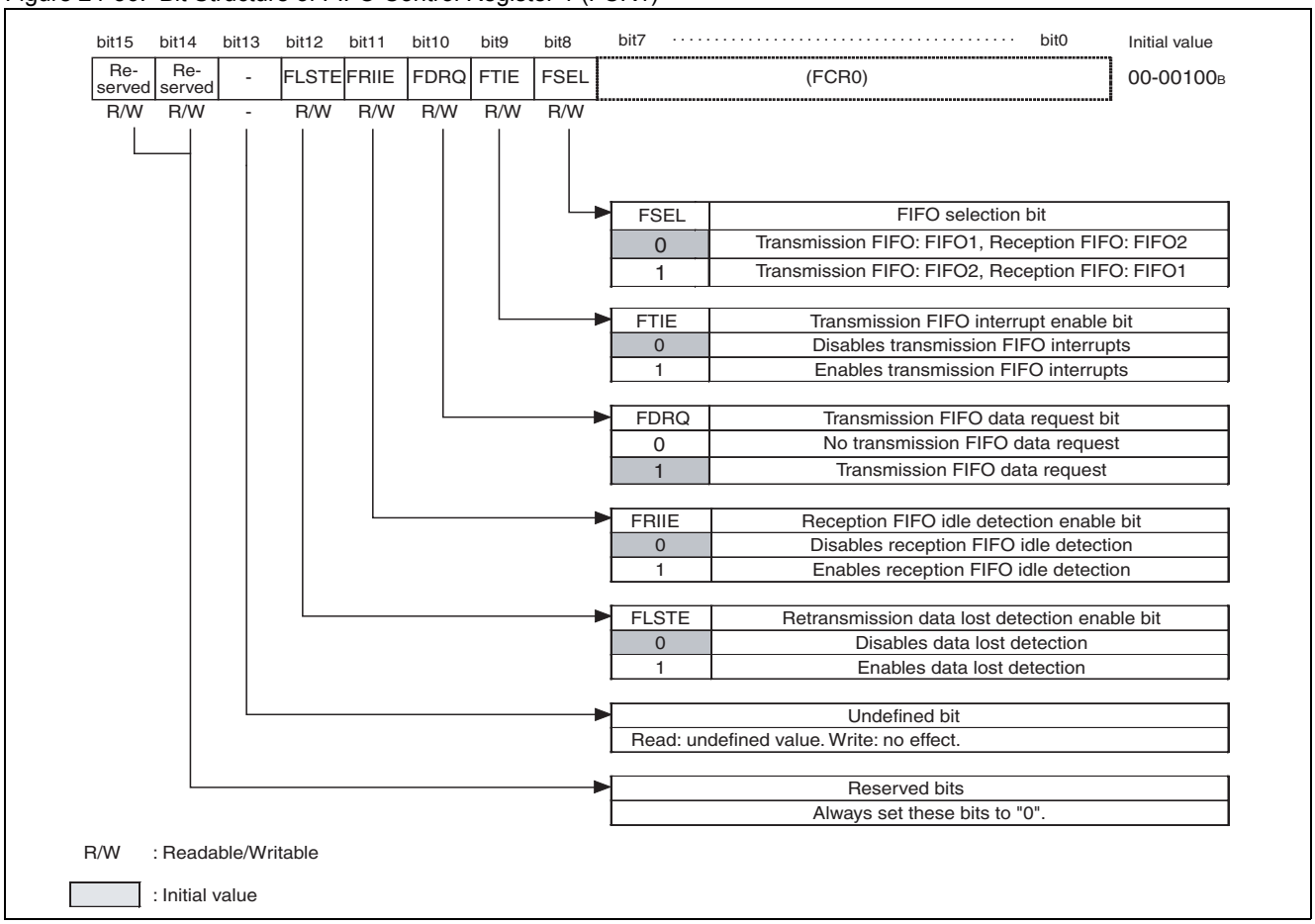


Table 24-22. Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	Reserved bits	Always set these bits to "00_B".
bit13	Undefined bit	Read: undefined value Write: no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. **Note:** To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8 clocks with the baud rate clock or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission FIFO interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition ■ FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) ■ Transmission FIFO reset FDRQ reset condition ■ Writing "0" to this bit ■ When the transmission FIFO is full. **Note:** It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". It is prohibited to modify the FSEL bit when this bit is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. **Note:** This bit cannot be cleared by resetting FIFO (FCR0:FCL2, FCL1 = 1). To modify this bit, disable FIFO operation (FCR0: FE2, FE1 = 0) in advance.

24.13.8 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

Bit structure of FIFO control register 0 (FCR0)

Figure 24-37 shows the bit structure of the FIFO control register 0 (FCR0) and Table 24-23 describes the function of each bit.

Figure 24-37. Bit Structure of FIFO Control Register 0 (FCR0)

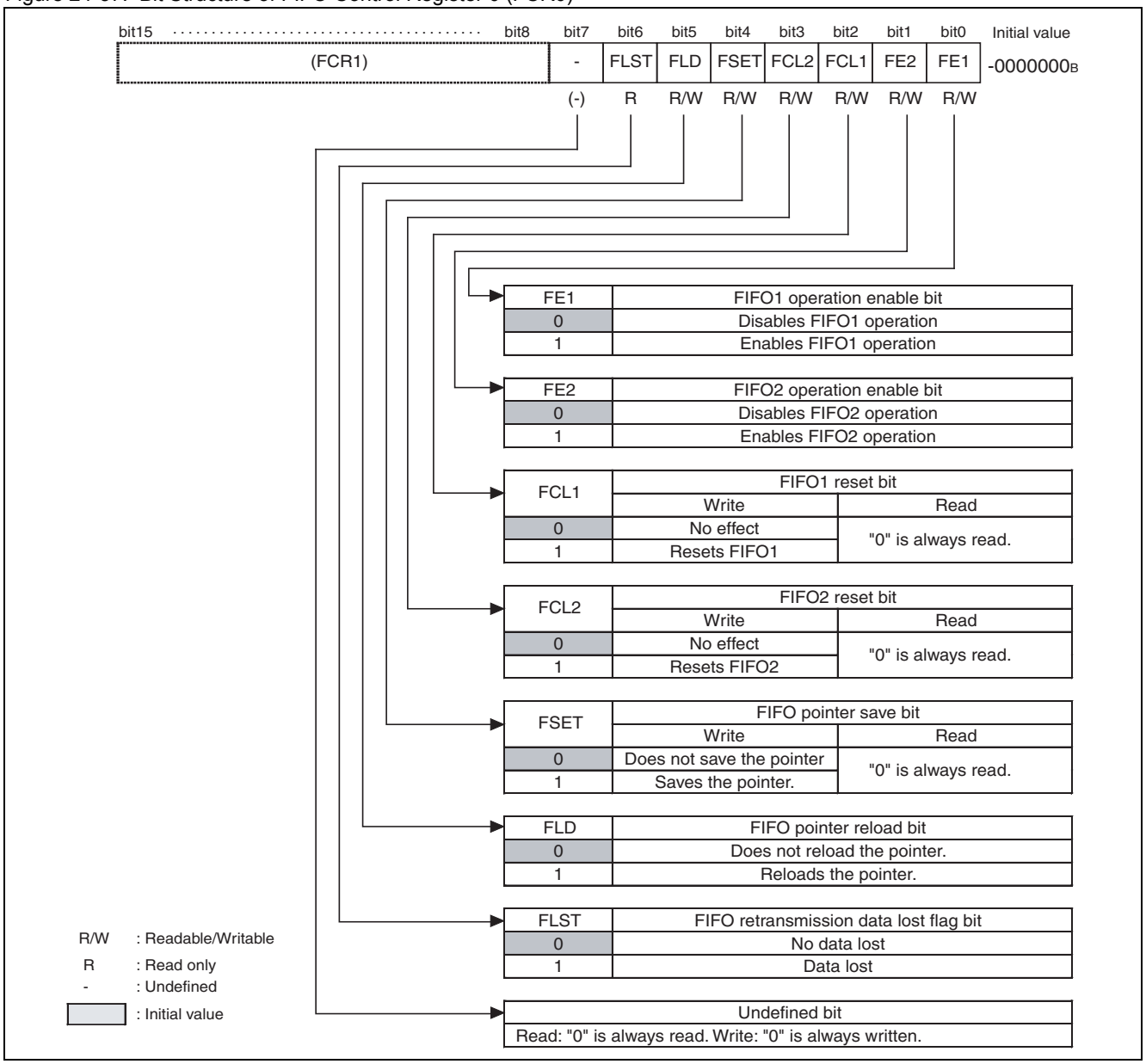


Table 24-23. Functional Description of Each Bit of FIFO Control Register 0 (FCR0)

Bit name		Function
bit7	Undefined bit	Read: "0" is always read. Write: Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition ■ Writing to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. FLST reset conditions ■ FIFO reset (writing "1" to FCL) ■ Writing "1" to the FLST bit Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.
bit5	FLD: FIFO pointer reload bit	This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs. The bit becomes "0" when retransmission has been set. **Note:** Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset. It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress. Write "1" to this bit after setting the TIE and TBIE bits to "0". And then, set the TIE and TBIE bits to "1" when the transmission FIFO has been enabled.
bit4	FSET: FIFO pointer save bit	This bit is used to save the read pointer of the transmission FIFO. If the FLST bit is set to "0", saving the read pointer prior to transmission will enable retransmission in case that an error such as a communication error occurs. Setting the bit to "1" saves the current read pointer value. Setting the bit to "0" has no effect. **Note:** Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".
bit3	FCL2: FIFO2 reset bit	This bit is used to reset FIFO2. Setting this bit to "1" initializes the internal state of FIFO2. Only the FCR0: FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. **Note:** Disable transmission/reception before resetting FIFO2. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE2 register will become "0".
bit2	FCL1: FIFO1 reset bit	This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. **Note:** Disable transmission/reception before resetting FIFO1. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".
bit1	FE2: FIFO2 operation enable bit	This bit is used to enable/disable FIFO2 operation. ■ To use FIFO2, set this bit to "1". ■ When FIFO2 is set for the transmission FIFO (FCR1:FSEL = 1) and "1" is written to this bit, transmission will start immediately, if FIFO2 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". ■ This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. ■ Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. ■ Even when FIFO2 is disabled, its status is retained.
bit0	FE1: FIFO1 operation enable bit	This bit is used to enable/disable FIFO1 operation. ■ To use FIFO1, set this bit to "1". ■ When FIFO1 is set for the transmission FIFO (FCR1:FSEL = 0) and "1" is written to this bit, transmission will start immediately, if FIFO1 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". ■ This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. ■ Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. ■ Even when FIFO1 is disabled, its status is retained.

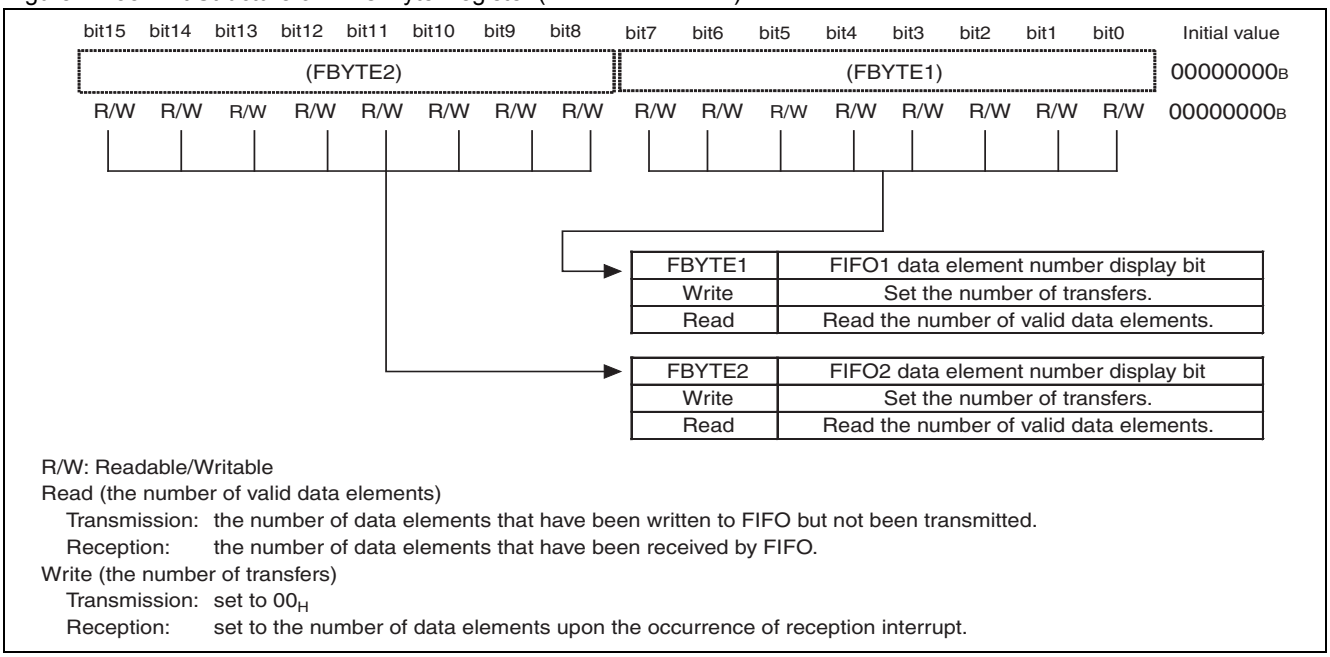
24.13.9 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO.

Bit structure of FIFO byte register (FBYTE1/FBYTE2)

Figure 24-38shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 24-38. Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)



The FBYTE1/FBYTE2 register indicates the number of valid data elements for FIFO. The details are as follows, depending on the setting of the FCR1:FSEL bit.

Table 24-24. Displaying the Number of Data Elements

FSEL	FIFO selection	Number of bytes displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is 08_H.
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- To receive data in master operation (master reception), set the TIE and TBIE bits to "0", set the number of data elements to be received to the FBYTE1/FBYTE2 register of the transmission FIFO, and write "0" to the FDRQ bit. When the TXE bit is set to "1", the serial clock will be output for a specified amount of data so that the specified amount of data can be received. To set the TIE and TBIE bits to "1", wait until the FDRQ becomes "1".

Notes:

- In master operation, set "00_H" to FBYTE1/FBYTE2 of the transmission FIFO except when receiving data.
- Set the number of transmission data elements when receiving data in master operation, when the transmission FIFO is empty and the TIE and TBIE bits are set to "0".
- To disable reception (RXE = 0) while receiving data in master operation, disable transmission/reception after disabling the transmission FIFO.
- Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
- To modify the FBYTE1/FBYTE2 of the reception FIFO, disable reception beforehand.
- Read modify write (RMW) instructions cannot be used for this register.
- Settings that will exceed the capacity of FIFO are prohibited.

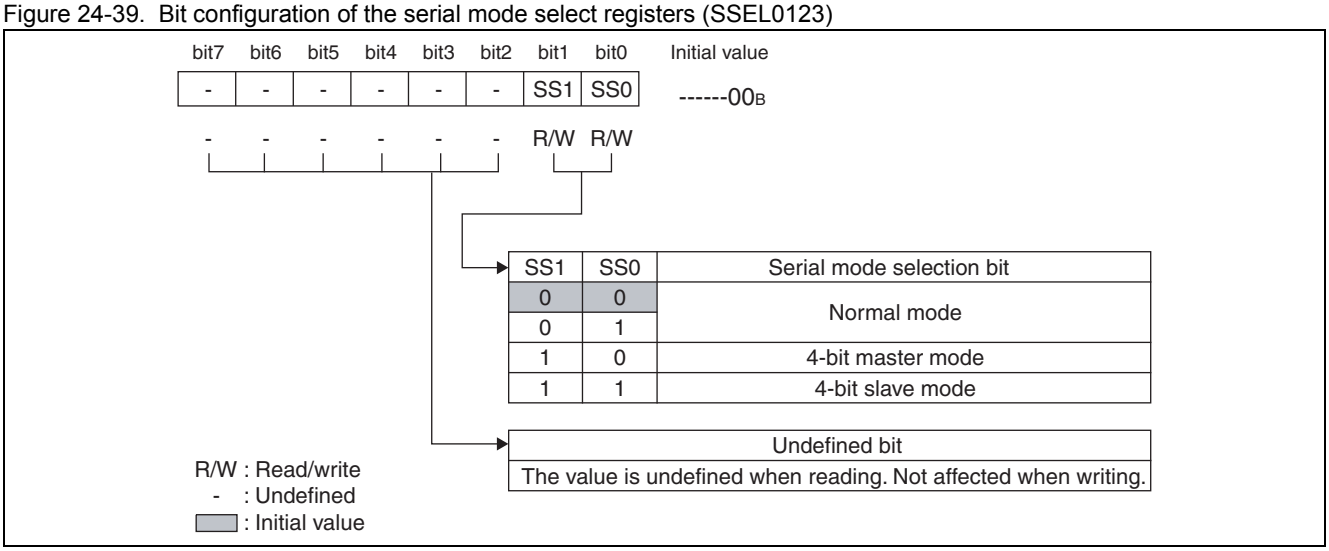
24.13.10 Serial Mode Select Registers (SSEL0123)

These registers operate the CSIO with 4-channels by using 1 clock simultaneously, enabling 4-bit serial communication.

The combination of ch.0 to ch.3 enable 4-channel simultaneous communication.

Figure 24-39 shows the bit configuration of the serial mode select registers (SSEL0123).

Figure 24-39. Bit configuration of the serial mode select registers (SSEL0123)



Note: Set these registers when the operation of the CSIO is stopped.

[bit7 to bit2]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit1, bit0]: SS1, SS0 (Serial mode select bit)

This bit determines whether to enable 4-channel simultaneous CSIO communication. In addition, it also selects an operation mode in the case of 4-channel simultaneous communication.

The operation modes are as follows:

- Normal mode: 4-channel simultaneous communication is not used.
- 4-bit master mode: 4-channel simultaneous communication using ch.0 to ch.3 is allowed in master mode.
- 4-bit slave mode: 4-channel simultaneous communication using ch.0 to ch.3 is allowed in slave mode.

SS1	SS0	Explanation
0	0	Normal mode is set.
0	1	
1	0	4-bit master mode is set.
1	1	4-bit slave mode is set.

Notes:

- To set as 4-bit master mode, make the following settings using the MS bit of the serial control register (SCR0 to SCR3).
  - ch.0 to ch.2: Slave mode
  - ch.3: Master mode
- To set as 4-bit slave mode, set slave mode for all channels used for simultaneous communication using the MS bit of the serial control register (SCR0 to SCR3).

24.13.11 Received Data Mirror Registers/Transmitted Data Mirror Registers (RDRM/TDRM)

The received data mirror register (RDRM) is a mirror register of the received data register (RDR) lower 8 bits.

The transmitted data mirror register (TDRM) is a mirror register of the transmitted data register (TDR) lower 8 bits.

Access to these registers allows access to the received data register (RDR) lower 8 bits the transmitted data register (TDR) lower 8 bits.

Use these registers when 4-channel simultaneous communication is used.

Received data mirror registers (RDRM)

The received data mirror register 0 (RDRM0) corresponds to lower 8 bits of the received data register 0 (RDR0) while the received data mirror register 3 (RDRM3) corresponds to lower 8 bits of the received data register 3 (RDR3).

The received data mirror registers (RDRM0 to RDRM3) of ch.0 to ch.3 are arranged in line. So, word access allows the registers to be read at one time. Use this for DMA transfer and other purposes.

For details, see "Operation in 4-channel simultaneous communication mode" in "24.15 Operation of CSIO (Clock Synchronous Serial Interface)".

Note: When 4-channel simultaneous communication is used, data with 9-bit length cannot be used.

Transmitted data mirror registers (TDRM)

The transmitted data mirror register 0 (TDRM0) corresponds to lower 8 bits of the transmitted data register 0 (TDR0) while the transmitted data mirror register 3 (TDRM3) corresponds to lower 8 bits of the transmitted data register 3 (TDR3).

The transmitted data mirror registers (TDRM0 to TDRM3) of ch.0 to ch.3 are arranged in line. So, word access allows the registers to be written to at one time. Use this for DMA transfer and other purposes.

For details, see "Operation in 4-channel simultaneous communication mode" in "24.15 Operation of CSIO (Clock Synchronous Serial Interface)".

Note: When 4-channel simultaneous communication is used, data with 9-bit length cannot be used.

24.14 Interrupts of CSIO (Clock Synchronous Serial Interface)

CSIO (clock synchronous serial interface) has the transmission/reception interrupt functionality. The following sources can be used to generate interrupt requests.

- When reception data is set in the reception data register (RDR) or a reception error occurs
- When transmission data is transferred from the transmission data register (TDR) to the transmission shift register and then transmission starts
- Transmission bus idle state (no transmission operation)
- Transmission FIFO data request

Interrupts of CSIO

Table 24-25 lists the interrupt control bits and interrupt sources of the CSIO.

Table 24-25. Interrupt Control Bits and Interrupt Sources of CSIO

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Reception	RDRF	SSR	Reception of 1 byte	SCR:RIE	Reading reception data (RDR)
			Reception of the amount of data specified in FBYTE1/FBYTE2 setting value		Reading reception data (RDR) until reception FIFO becomes empty
			Detection of the idle state of reception for 8 clocks with the baud rate clock or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	ORE	SSR	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)
Transmission	TDRE	SSR	Transmission register being empty	SCR:TIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)[1]
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)[1]
	FDRQ	FCR1	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

[1]: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

24.14.1 Occurrence of Reception Interrupts and Flag Set Timing

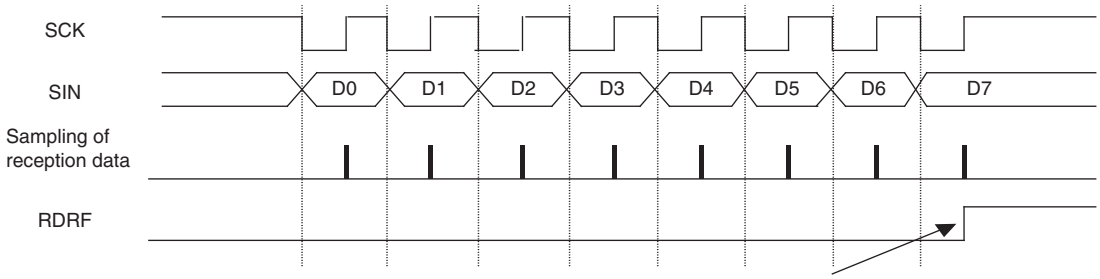
Reception interrupts are generated by the completion of reception (SSR:RDRF) and the occurrence of a reception error (SSR:ORE).

Occurrence of reception interrupts and flag set timing

Reception data is stored to the reception data register (RDR) when the last data bit is detected. Each flag is set when the reception has been completed (SSR:RDRF = 1) or a reception error has occurred (SSR:ORE = 1). If reception interrupts have been enabled (SSR:RIE = 1), a reception interrupt will occur.

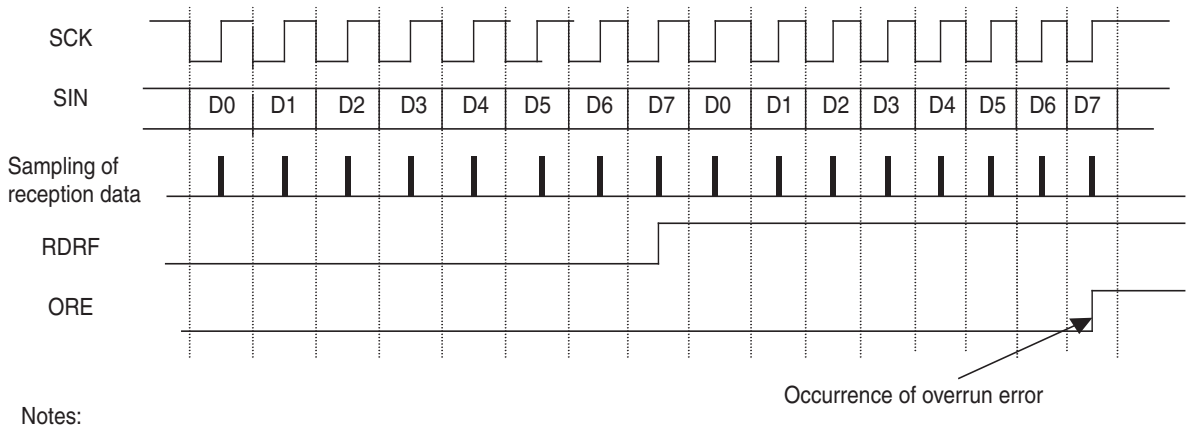
Note: If a reception error occurs, the data in the reception data register (RDR) will become invalid.

Figure 24-40. Reception Operation and Flag Set Timing



Note:
The figure shows the timing based on the following conditions.
SCR: MS=1, SPI=0
ESCR: L2-L0=000_B
SMR: SCINV=0, BDS=0, SCKE=0, SOE=0

Figure 24-41. Timing for Setting ORE (Overrun Error) Flag



Notes:

- The figure shows the timing based on the following conditions.
  SCR: MS=1, SPI=0
  ESCR: L2-L0=000_B
  SMR: SCINV=0, BDS=0, SCKE=0, SOE=0
- An overrun error occurs when the next data is transferred before reception data is read (RDRF = 1).

24.14.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing

When the reception FIFO is used, an interrupt will occur, if the amount of data specified in the FBYTE1/FBYTE2 register (FBYTE1/FBYTE2) is received.

Occurrence of reception interrupts when reception FIFO is used and flag set timing

Occurrence of interrupts when reception FIFO is used is determined by the setting value of the FBYTE1/FBYTE2 register.

- The reception data full flag of the serial status register (SSR:RDRF) is set to "1", when the received data is equivalent of the number of transfers specified in the FBYTE1/FBYTE2 register. At this point, a reception interrupt will occur, if reception interrupts have been enabled (SCR:RIE).
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- The reception data full flag (SSR:RDRF) is cleared when reception data (RDR) is read until the reception FIFO becomes empty.
- An overrun error occurs (SSR:ORE = 1) when the next data is received while the number of valid reception data elements is indicating the capacity of FIFO.

Figure 24-42. Timing for Generating Reception Interrupt when Reception FIFO is Used

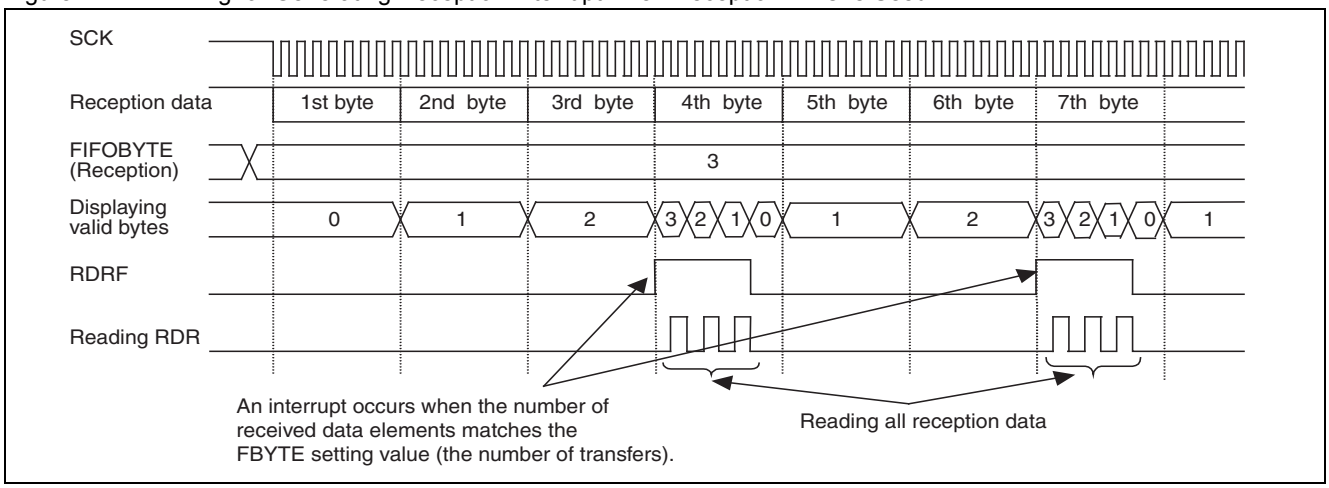
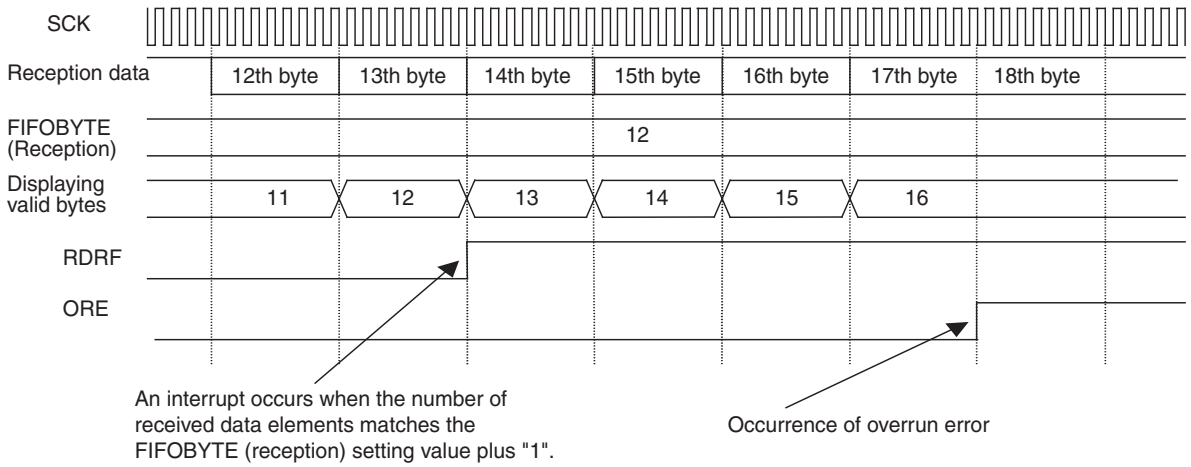


Figure 24-43. Timing for Setting ORE (Overrun Error) Flag Bit



Note: An overrun error occurs when the next data is received while the FIFO display is showing its capacity.

24.14.3 Occurrence of Transmission Interrupts and Flag Set Timing

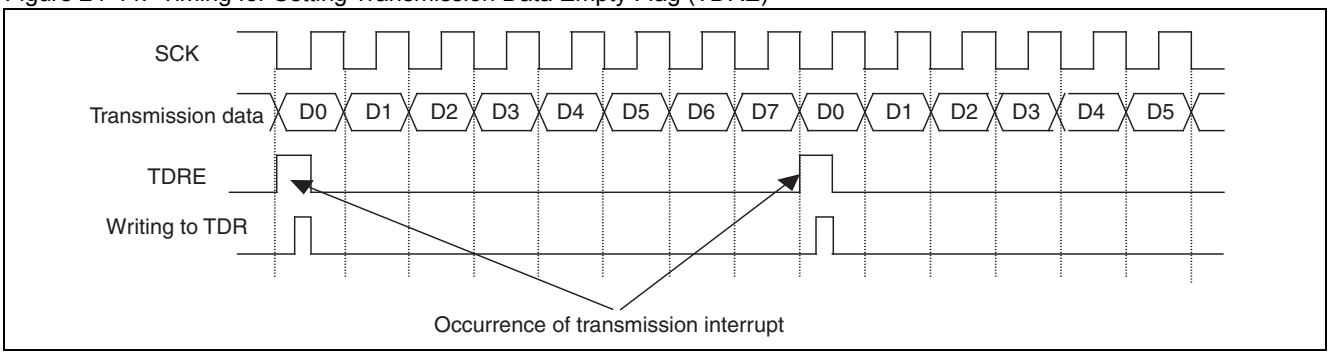
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR) to the shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

Occurrence of transmission interrupts and flag set timing

- Timing for setting the transmission data empty flag (TDRE)

  It is enabled to write the next data (SSR:TDRE = 1), when the data written to the transmission data register (TDR) is transferred to the transmission shift register. At this point, a transmission interrupt will occur, if transmission interrupts have been enabled (SCR:TIE = 1). As the TDRE bit is a read only bit, it is cleared by writing "0" to the transmission data register (TDR).

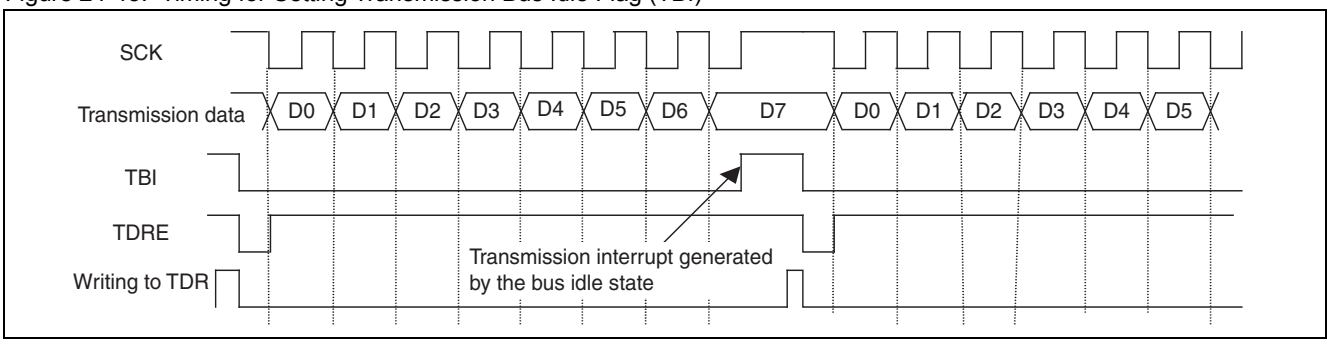
Figure 24-44. Timing for Setting Transmission Data Empty Flag (TDRE)



- Timing for setting the transmission bus idle flag (TBI)

  The SSR:TBI bit is set to "1", when the transmission data register is empty (TDRE = 1) and no transmission operation is in progress. At this point, a transmission interrupt occurs if transmission bus idle interrupts have been enabled (SCR:TBIE = 1). The TBI bit and transmission interrupt request are cleared when transmission data is set to the transmission data register (TDR).

Figure 24-45. Timing for Setting Transmission Bus Idle Flag (TBI)



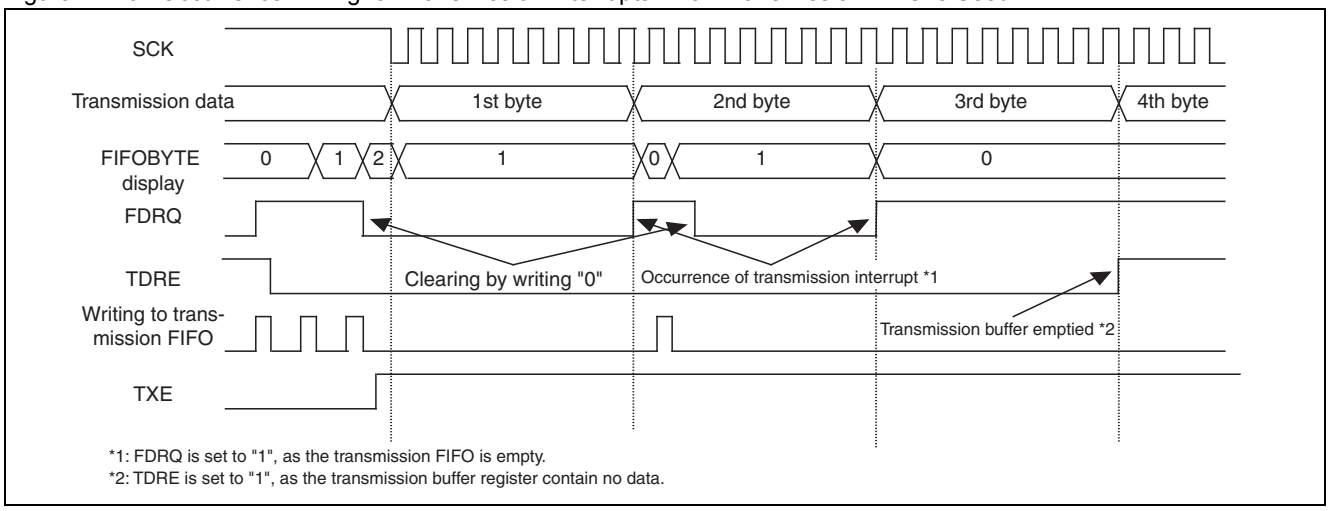
24.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

Occurrence of transmission interrupts when transmission FIFO is used and flag set timing

- The FIFO transmission data request bit (FCR1:FDRQ) is set to "1", when the transmission FIFO contains no data. At this point, a transmission interrupt will occur if FIFO transmission interrupts have been enabled (FCR1:FTIE = 1).
- Write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request when required data has been written to the transmission FIFO upon the occurrence of a transmission interrupt.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" when the transmission FIFO becomes full.
- The FIFO byte register (FBYTE1/FBYTE2) can be read to check if the transmission FIFO contains any data. FBYTE1/FBYTE2 = 00_H indicates that the transmission FIFO contains no data.

Figure 24-46. Occurrence Timing for Transmission Interrupts when Transmission FIFO is Used



24.15 Operation of CSIO (Clock Synchronous Serial Interface)

CSIO uses clock synchronization for its transfer system.

Operation of CSIO (Clock Synchronous Serial Interface)

Normal Transfer (I)

- Features

Table 24-26. Features of Normal Transfer (I)

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Timing for transmission data output	Falling edge of SCK
3	Sampling of reception data	Rising edge of SCK
4	Data length	5 bits to 9 bits

- Register settings

  The setting values of registers required for the normal transfer (I) are shown below.

Table 24-27. Register Settings for Normal Transfer (I)

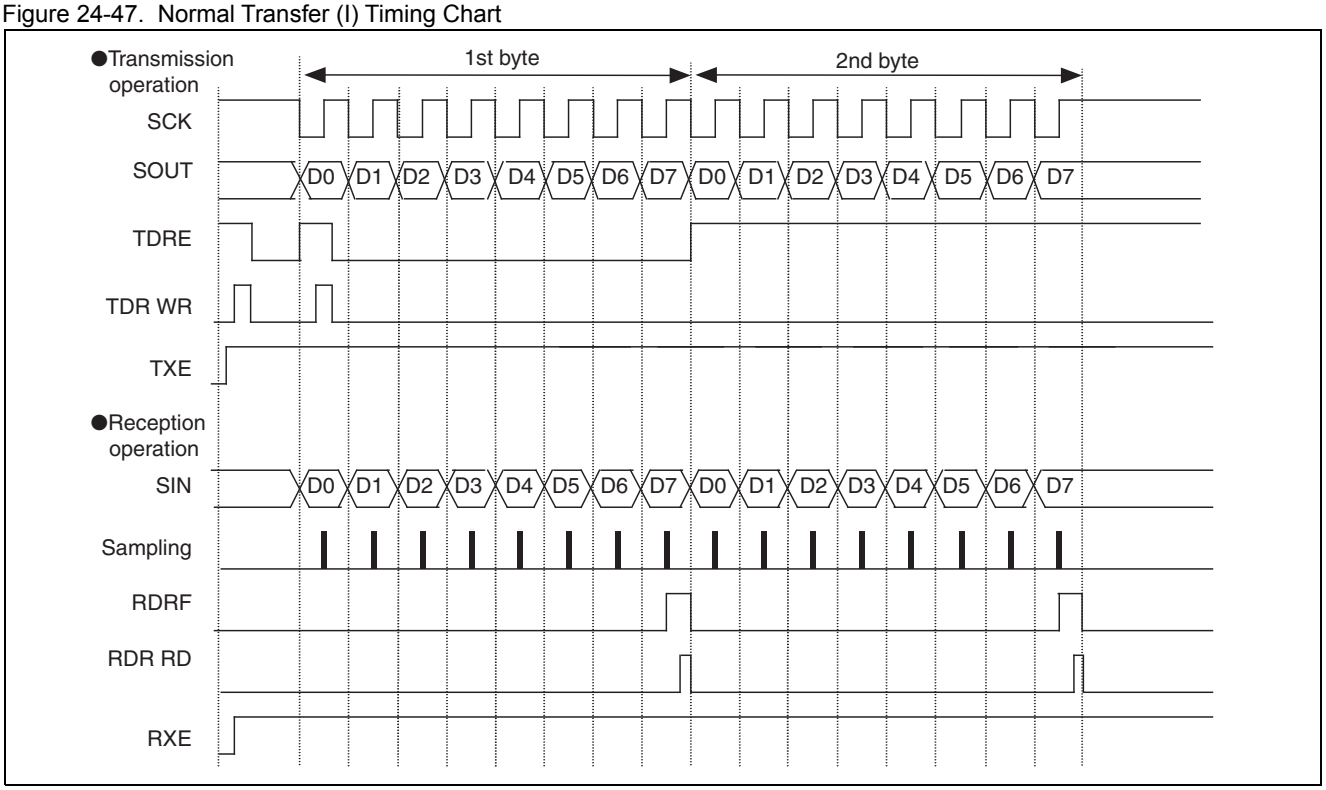
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	[1]	[1]	[1]	[1]	[1]	0	1	0	0	0	[1]	1/0	[1]
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	[1]	[1]	[1]	[1]	[1]
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

1: Set to "1"
0: Set to "0"
[1]: User-defined setting

Note: The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.
Master operation: SCR:MS = 0, SMR:SCKE = 1
Slave operation: SCR:MS = 1, SMR:SCKE = 0

- Normal transfer (I) timing chart

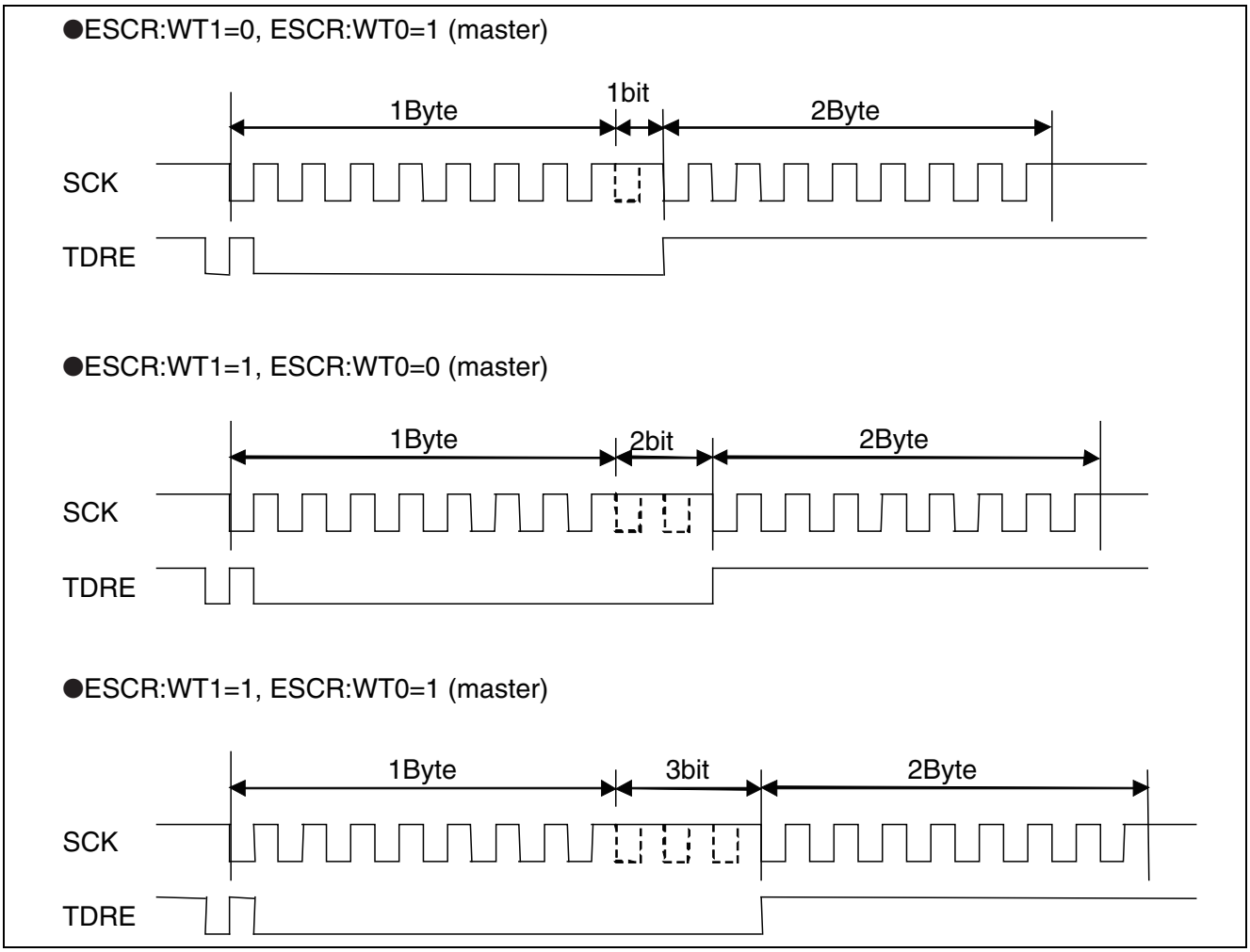
Figure 24-47. Normal Transfer (I) Timing Chart



- Operational description
  - Master operation (SCR:MS = 0, SMR:SCKE = 1)
    1. Transmission operation
       a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output.
       b. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
    2. Reception operation
       a. If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock output (SCK).
       b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.
       c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

       Notes:
       - To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).
       - A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

3. Transmission/reception operation
   a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
   b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
   c. Reception data will be sampled at the rising edge of the serial clock (SCK) output. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
4. Sequential data transmission/reception wait operation
   a. If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



- Slave operation (SCR:MS = 1, SMR:SCKE = 0)
  1. Transmission operation
     a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input.
     b. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
  2. Reception operation
     a. If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE=1), reception data will be sampled at the rising edge of the serial clock input (SCK).
     b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.
     c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
  3. Transmission/reception operation
     a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
     b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
     c. Reception data will be sampled at the rising edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

Normal transfer (II)

- Features

Table 24-28. Features of Normal Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Output timing for transmission data	Rising edge of SCK
3	Sampling of reception data	Falling edge of SCK
4	Data length	5 bits to 9 bits

- Register settings

  The table below shows the register setting values required for the normal transfer (II).

Table 24-29. Register Settings for Normal Transfer (II)

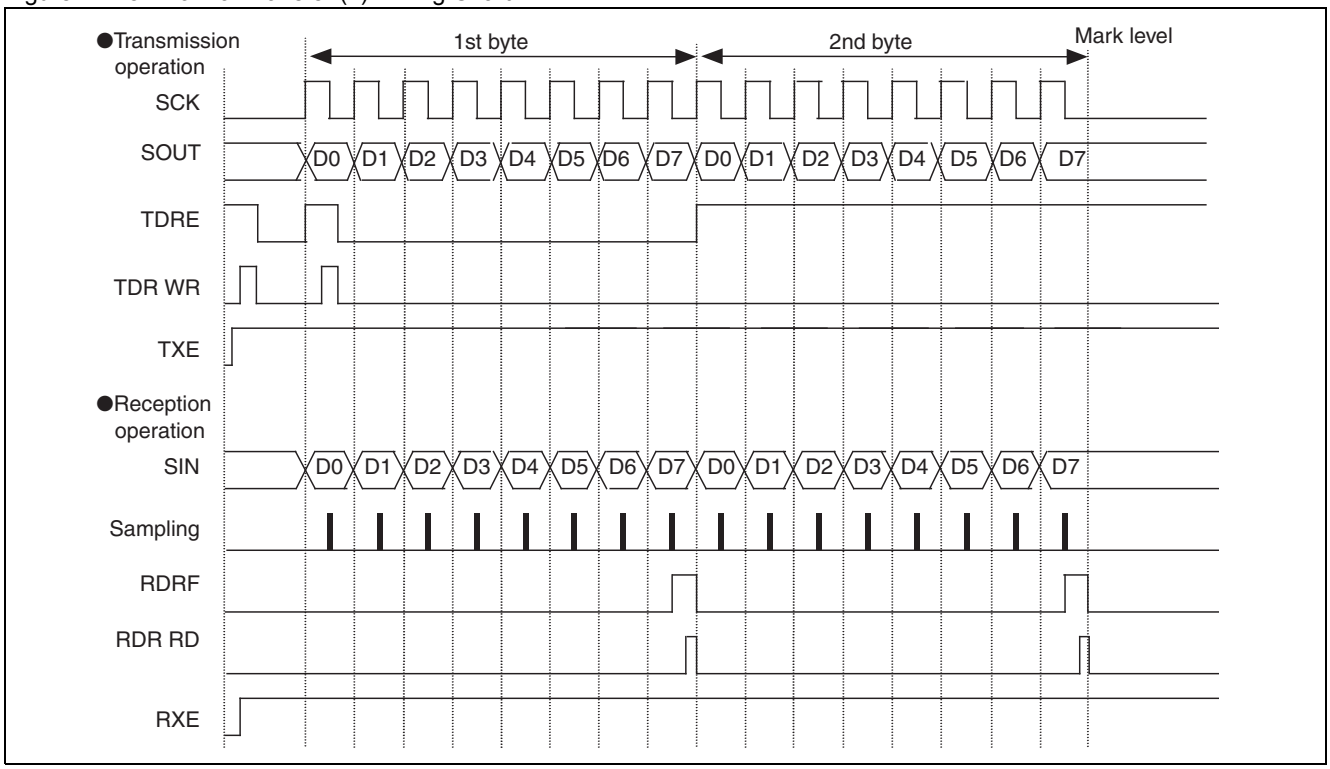
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	[1]	[1]	[1]	[1]	[1]	0	1	0	0	1	[1]	1/0	[1]
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	[1]	[1]	[1]	[1]	[1]
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

1: Set to "1"
0: Set to "0"
[1]: User-defined setting

Note: The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.
Master operation:SCR:MS = 0, SMR:SCKE = 1
Slave operation:SCR:MS = 1, SMR:SCKE = 0

Normal transfer (II) timing chart

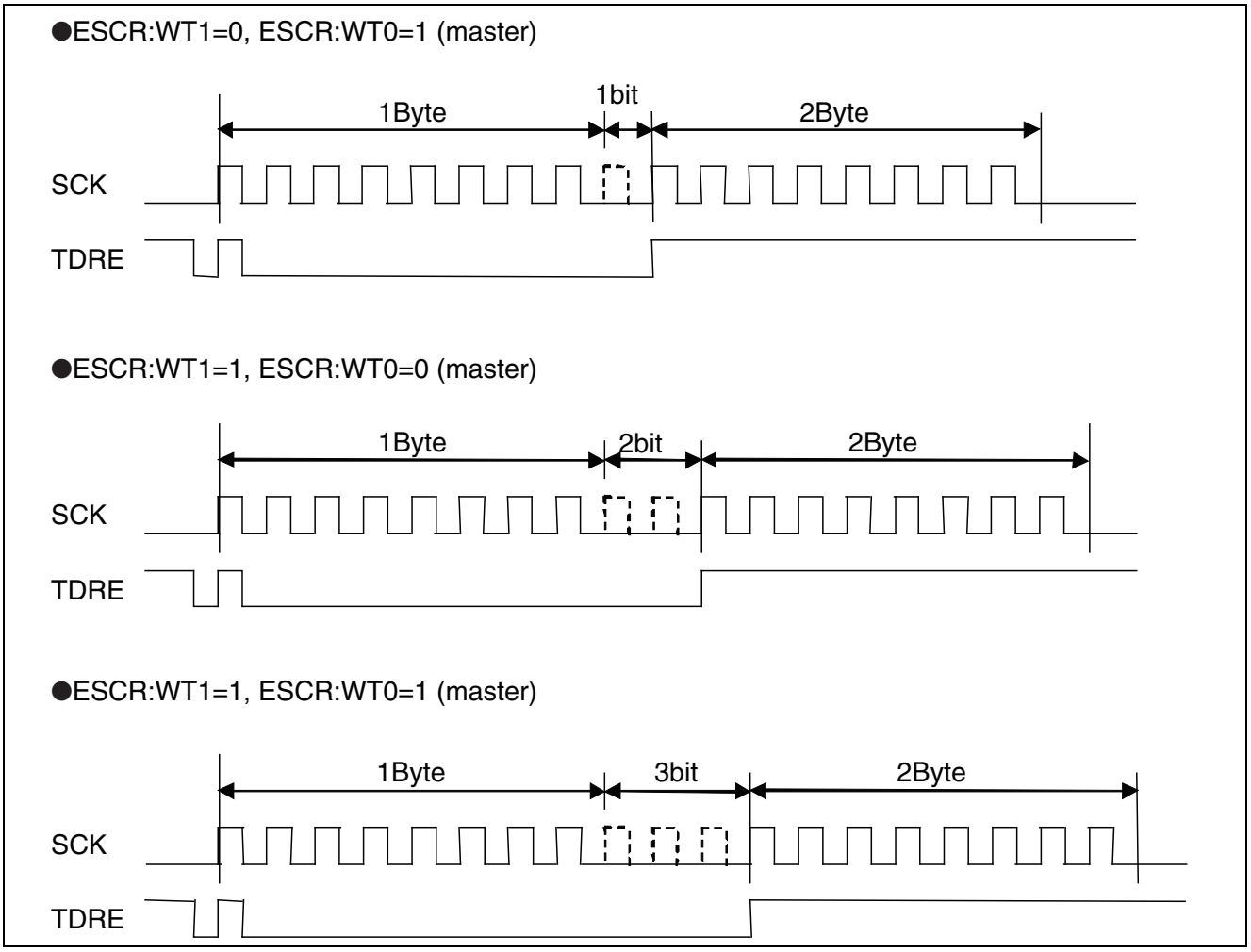
Figure 24-48. Normal Transfer (II) Timing Chart



- Operational description
  - Master operation (SCR:MS = 0, SMR:SCKE = 1)
    1. Transmission operation
       a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.
       b. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
    2. Reception operation
       a. If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).
       b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.
       c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

       Notes:
       - To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).
       - A specified number of frames of the serial clock (SCK) will be output if the number of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

3. Transmission/reception operation
   a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
   b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
   c. Reception data will be sampled at the falling edge of the serial clock (SCK) output. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
4. Sequential data transmission/reception wait operation
   a. If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



- Slave operation (SCR:MS = 1, SMR:SCKE = 0)
  1. Transmission operation
     a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) input.
     b. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
  2. Reception operation
     a. If the output of serial data is disabled (SMR:SOE = 0) and reception operations is enabled (SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock input (SCK).
     b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.
     c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
  3. Transmission/reception operation
     a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
     b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
     c. Reception data will be sampled at the falling edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

SPI transfer (I)

■ Features

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Output timing for transmission data	Rising edge of SCK
3	Sampling of reception data	Falling edge of SCK
4	Data length	5 bits to 9 bits

■ Register settings

The table below shows the register setting values required for the SPI transfer (I).

Table 24-30. SPI Transfer (I) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	[1]	[1]	[1]	[1]	[1]	0	1	0	0	0	[1]	1/0	[1]
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	[1]	[1]	[1]	[1]	[1]
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

1: Set to "1"

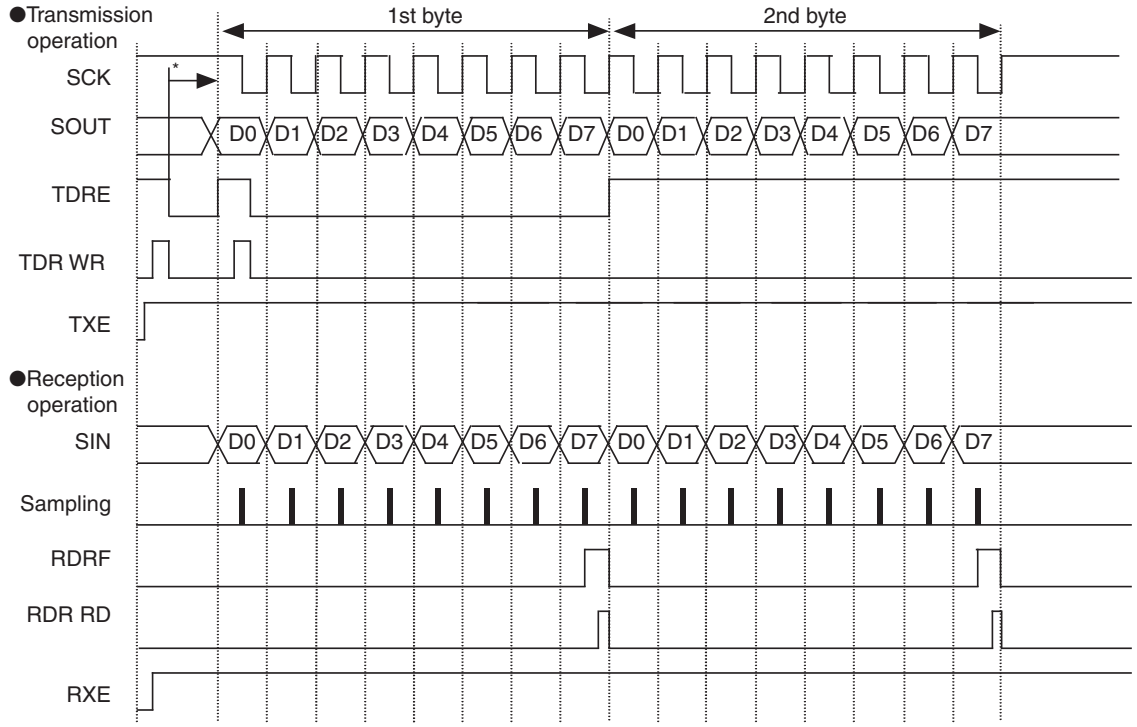
0: Set to "0"

*: User-defined setting

Note: The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.
Master operation: SCR:MS = 0, SMR:SCKE = 1
Slave operation: SCR:MS = 1, SMR:SCKE = 0

■ SPI transfer (I) timing chart

Figure 24-49. SPI Transfer (I) Timing Chart



■ Operational description

❒ Master operation (SCR:MS = 0, SMR:SCKE = 1)

1. Transmission operation

   a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.

   b. SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock (SCK) output.
   A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

2. Reception operation

   a. If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).

   b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1).
   At this point, reception data (RDR) can be read.

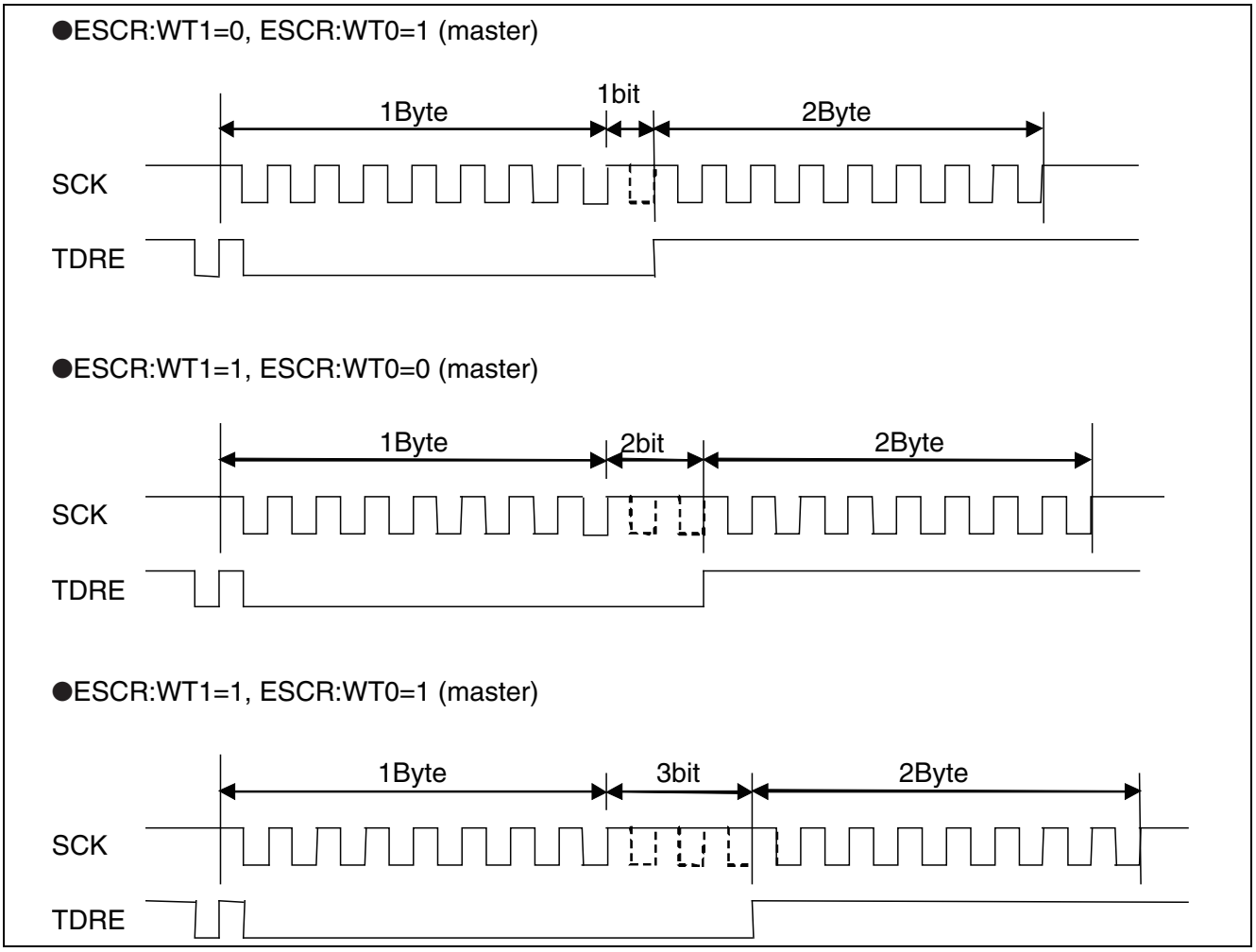
   c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

   Notes:

   ■ To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).

   ■ A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

3. Transmission/reception operation
   a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
   b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output. SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock (SCK) output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
   c. Reception data will be sampled at the falling edge of the serial clock (SCK) output. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
4. Sequential data transmission/reception wait operation
   a. If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



- Slave operation (SCR:MS = 1, SMR:SCKE = 0)
  1. Transmission operation
     a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.
     b. SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1).
     At this point, the transmission data for the second byte can be written.
  2. Reception operation
     a. If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock input (SCK).
     b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.
     c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
  3. Transmission/reception operation
     a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
     b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
     c. Reception data will be sampled at the falling edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

SPI transfer (II)

■ Features

Table 24-31. Features of SPI Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Output timing for transmission data	Falling edge of SCK
3	Sampling of reception data	Rising edge of SCK
4	Data length	5 to 9 bits

■ Register settings

The table below shows the register setting values required for the SPI transfer (II).

Table 24-32. SPI Transfer (II) Register Settings

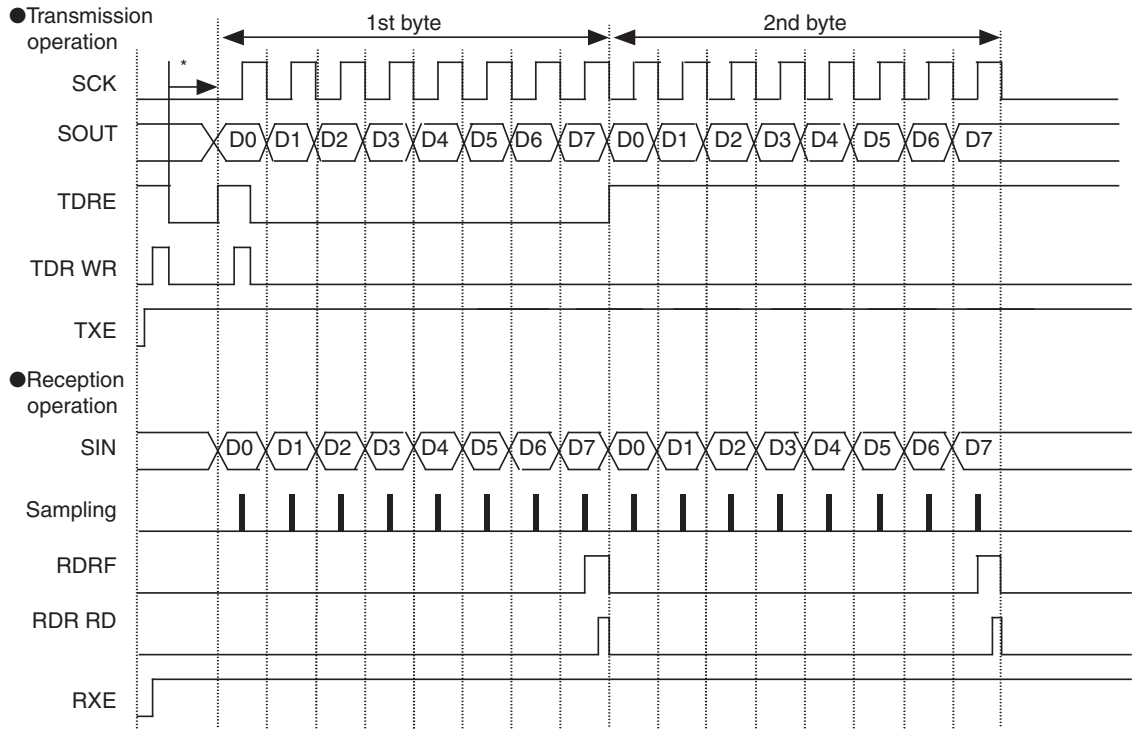
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	[1	[1]	[1]	[1]	[1]	0	1	0	0	1	[1]	1/0	[1]
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	[1]	[1]	[1]	[1]	[1]
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

1: Set to "1"
0: Set to "0"
[1]: User-defined setting

Note: The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.
Master operation:SCR:MS = 0, SMR:SCKE = 1
Slave operation:SCR:MS = 1, SMR:SCKE = 0

■ SPI transfer (II) timing chart

Figure 24-50. SPI Transfer (II) Timing Chart



■ Operational description

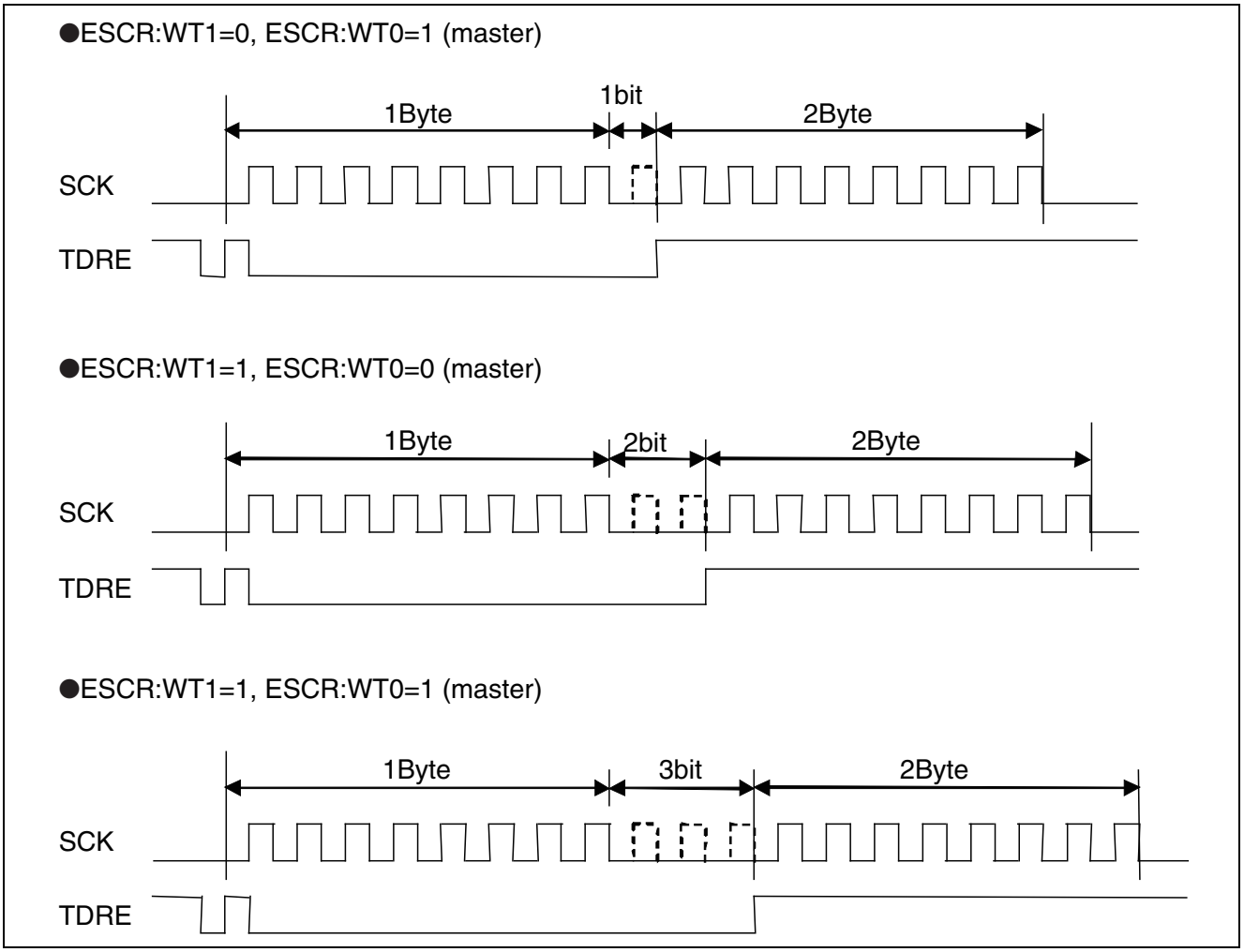
❒ Master operation (SCR:MS = 0, SMR:SCKE = 1)

1. Transmission operation
   a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output.
   b. SSR:TDRE is set to "1", half a cycle before the rising edge of the first serial clock (SCK) output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
2. Reception operation
   a. If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock output (SCK).
   b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.
   c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

Notes:

- To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).
- A specified number of frames of the serial clock (SCK) will be output if the byte number of frames to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

3. Transmission/reception operation
   a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
   b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output. SSR:TDRE is set to "1", half a cycle before the rising edge of the first serial clock (SCK) output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
   c. Reception data will be sampled at the rising edge of the transmission clock. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
4. Sequential data transmission/reception wait operation
   a. If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



- Slave operation (SCR:MS = 1, SMR:SCKE = 0)
  1. Transmission operation
     a. If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input.
     b. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
  2. Reception operation
     a. If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock input (SCK).
     b. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.
     c. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
  3. Transmission/reception operation
     a. To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
     b. If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.
     c. Reception data will be sampled at the rising edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

Operation in 4-channel simultaneous communication mode

It is possible to make the CSIO of 4-channels of ch.0 to ch.3 communicate simultaneously to transmit and receive 4-bit data at one time.

The 4-channels can be used both in master mode and in slave mode. This section explains the operation in 4-channel simultaneous communication mode.

- Overview

  To allow 4-channel simultaneous communication, setting is made by the SS1 and SS0 bit of the serial mode select register (SSEL0123).

  In addition, the required settings vary according to whether communication is executed in master mode or in slave mode.

  Table 24-33 shows the settings required for the 4-channel simultaneous communication mode.

Table 24-33. Settings in 4-channel simultaneous communication mode

Mode	Setting		ch.0	ch.1	ch.2	ch.3
4-bit master	SSEL	SS1/SS0 bit	10	10	10	10
	SCR	MS bit	1	1	1	0
4-bit slave	SSEL	SS1/SS0 bit	11	11	11	11
	SCR	MS bit	1	1	1	1

SSEL: Serial mode select register (SSEL0123)
SCR: Serial control register (SCR0 to SCR3)

The serial clock input methods vary according to whether the mode is 4-bit master mode or 4-bit slave mode.

Table 24-34 lists the input sources of the serial clock.

Table 24-34. Input Sources of the Serial Clock

Mode	ch.0	ch.1	ch.2	ch.3
4-bit master (SS1, SS0 = 10)	Output from ch.3	Output from ch.3	Output from ch.3	SCK3 pin
4-bit slave (SS1, SS0 = 11)	SCK3 pin	SCK3 pin	SCK3 pin	SCK3 pin

Figure 24-51 shows the input sources of the serial clock in 4-bit master mode and in 4-bit slave mode.

Figure 24-51. The Input Sources of the Serial Clock

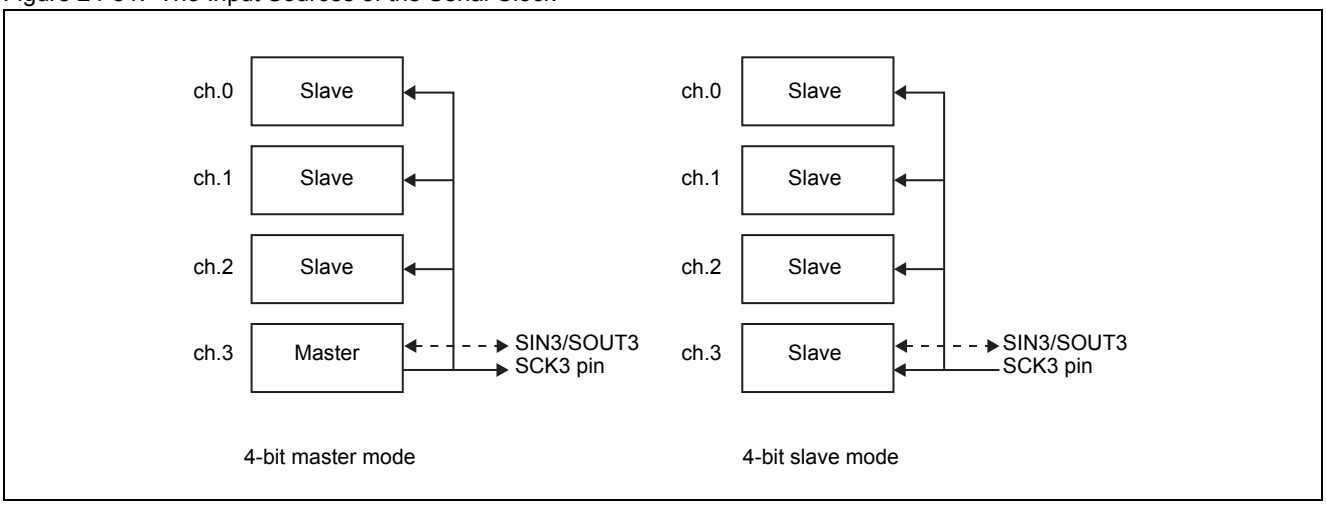


Table 24-35 shows the available pin combination for the four-channel simultaneous communication mode.

Table 24-35. Available Pin Combination

	ch.0	ch.1	ch.2	ch.3
Combination 1	SCK0_1 SIN0_1 SOUT0_1	SCK1 SIN1 SOUT1	SCK2 SIN2 SOUT2	SCK3 SIN3 SOUT3

- Operation

  When 4-channel simultaneous communication mode is used, the receive operation/transmit operation is the same as the operation of 1 channel.

  However, to allow 4-bit simultaneous transmission and reception, the following registers are provided.

  - Received data mirror register (RDRM0 to RDRM3)
  - Transmitted data mirror registers (TDRM0 to TDRM3)

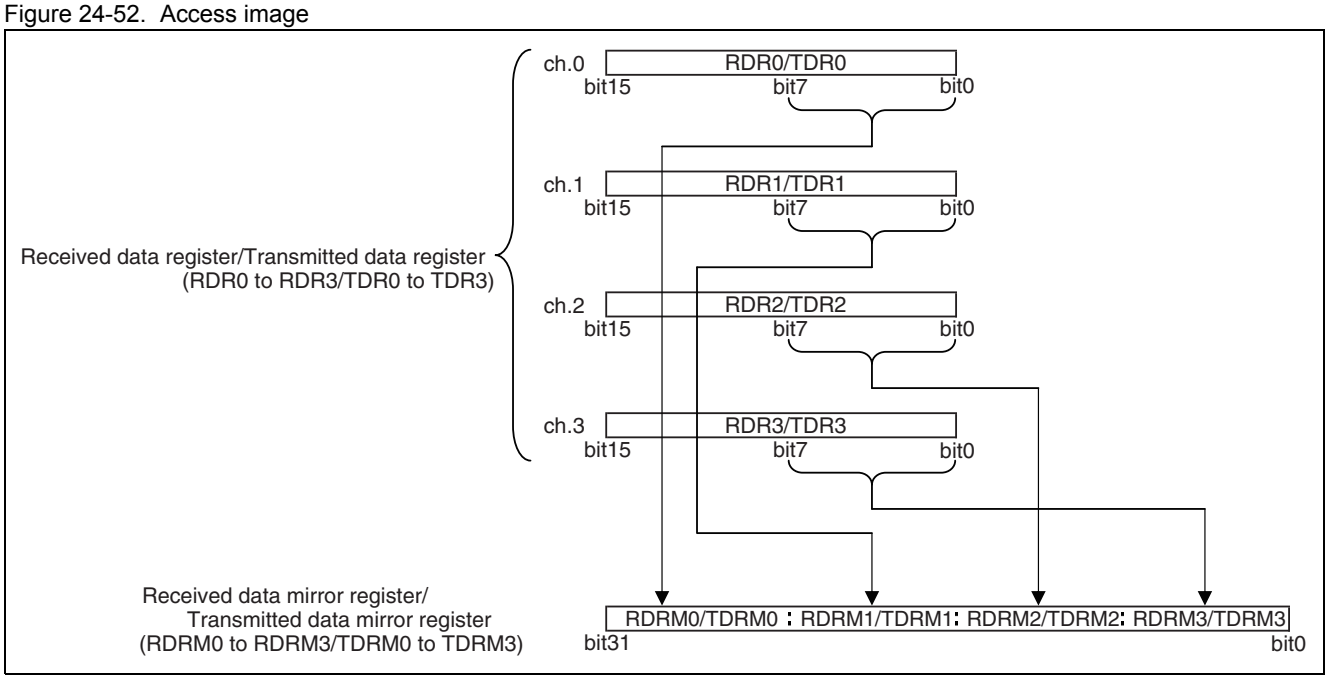
  Access to these registers allows access to the received data register (RDR) lower 8 bits or the transmitted data register (TDR) lower 8 bits.

  The received data mirror registers (RDRM0 to RDRM3)/transmitted data mirror registers (TDRM0 to TDRM3) of ch.0 to ch.3 are arranged in line. So, word access allows the registers to be written to at one time. Use this for DMA transfer and other purposes.

  Note: The interrupt during 4 channels are simultaneously activating, it is recommended to allow using 1 channel only out of 4 channels.

  Figure 24-52 shows the images of the received data mirror register (RDRM0 to RDRM3)/transmitted data mirror register (TDRM0 to TDRM3).

Figure 24-52. Access image



Note: When 4-channel simultaneous communication is used, data with 9-bit length cannot be used.

24.16 Dedicated Baud Rate Generator

The dedicated baud rate generator only functions in master operation. However, set the dedicated baud rate generator even for slave operation, when using the reception FIFO.

Baud rate selection for CSIO (Clock Synchronous Serial Interface)

The dedicated baud rage generator settings are different between master and slave operations.

- Master operation

  The baud rate is selected by dividing the internal clock using the dedicated baud rate generator.

  - There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).
  - The reload counter divides the internal clock, according to the set value.

- Slave operation

  In slave operation (SCR:MS = 1), the dedicated baud rate generator does not function.
  (The slave operation directly uses the external clock which is input from the clock input pin SCK.)

  Note: Set the dedicated baud rate generator even for slave operation, when using the reception FIFO.

24.16.1 Setting Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

Calculating the baud rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

- Reload value:
  $V = \phi / b - 1$

  V:Reload value

  b:Baud rate

  ϕ:Peripheral clock (PCLK) frequency

- Example of calculation
  If the peripheral clock (PCLK) is 16MHz, the internal clock is used, and the baud rate is 19200bps, the reload value will be:

  Reload value:

  $V = (16 \times 1000000)/19200 - 1 = 832$

  So baud rate is:

  $b = (16 \times 1000000)/(832+1) = 19208$ bps

- Baud rate error

  The following formula is used to calculate a baud rate error.
  Error (%) = (calculated value - target value) / target value $\times$ 100

  Example: peripheral clock (PCLK) = 20MHz, target baud rate = 153600bps

  Reload value $=(20 \times 1000000)/153600 - 1 = 129$

  Baud rate (calculated value) $=(20 \times 1000000)/(129+1) = 153846$ (bps)

  Error (%) $=(153846 - 153600)/153600 \times 100 = 0.16$ (%)

Notes:

- The reload counter halts when the reload value is set to "0".

- When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the SCINV bit settings. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
  - When SPI is set to "0" and when SCINV is set to "0", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer.
  - When SPI is set to "0" and when SCINV is set to "1", the "L" width of the serial clock is one peripheral clock (PCLK) cycle longer.
  - When SPI is set to "1" and SCINV is set to "0", the "L" width of the serial clock is one peripheral clock (PCLK) cycle longer.
  - When SPI is set to "1" and SCINV is set to "1", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer.
- Select 3 or a larger value for the reload value.

Reload values and baud rates for different peripheral clock (PCLK) frequencies

Table 24-36. Reload Values and Baud Rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	-	-	-	-	-	-	-	-	-	-	3	0
6M	-	-	-	-	-	-	-	-	3	0	-	-
5M	-	-	-	-	-	-	3	0	-	-	-	-
4M	-	-	-	-	3	0	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	-	-	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: The value set in BGR1/BGR0 registers
- ERR: Baud rate error (%)

Functions of reload counters

There are two reload counters, a transmission reload counter and a reception reload counter, which function as a dedicated baud rate generator. Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the internal clock.

Starting a count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

Restart

The reload counter restarts under the following conditions.

- For both transmission and reception reload counters
  Programmable reset (SCR:UPCL bit)

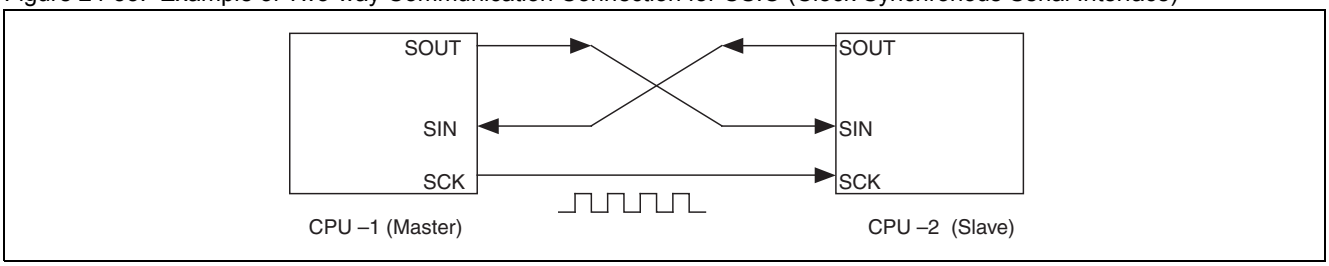
24.17 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

Two-way serial synchronous communication is enabled in CSIO (clock synchronous serial interface).

Connection between CPUs

Two-way communication should be selected for CSIO (clock synchronous serial interface). Two CPUs are connected to each other, as shown in Figure 24-53.

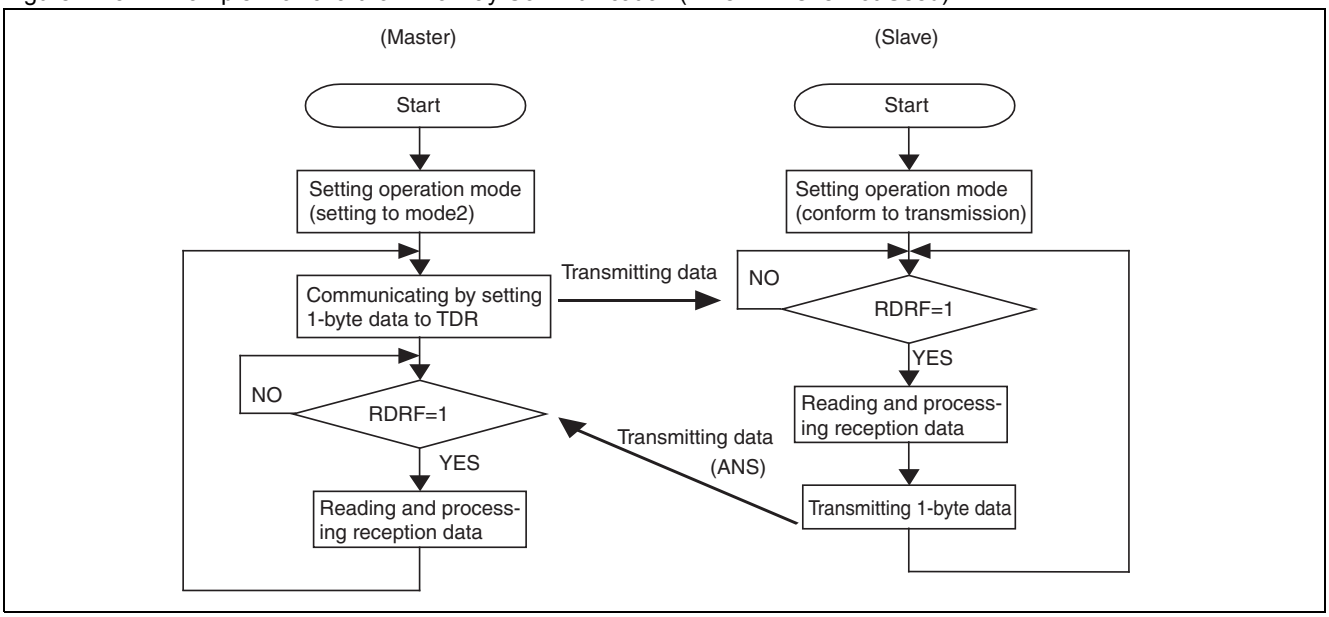
Figure 24-53. Example of Two-way Communication Connection for CSIO (Clock Synchronous Serial Interface)



Flowchart

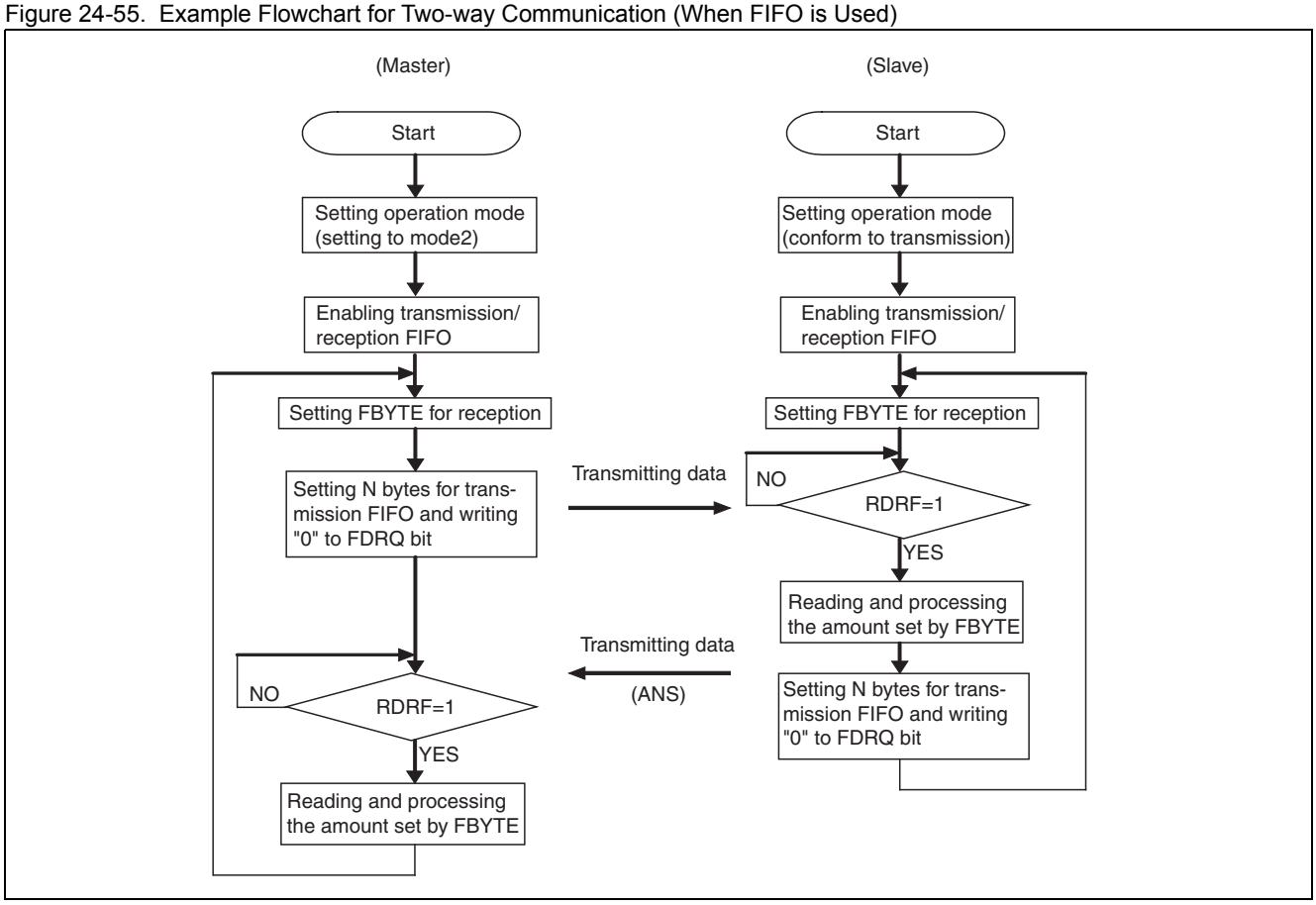
- When FIFO is not used

Figure 24-54. Example Flowchart for Two-way Communication (When FIFO is Not Used)



■ When FIFO is used

Figure 24-55. Example Flowchart for Two-way Communication (When FIFO is Used)



24.18 Notes on CSIO Mode

The notes for when you use the CSIO mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.

24.19 I²C Interface

Of all the functions of the multi-function serial interface, this section describes the I²C interface that is supported in operation mode 4.

- I²C Interface
- Overview of I²C Interface
- Registers of I²C Interface
  - I²C Bus Control Register (IBCR)
  - Serial Mode Register (SMR)
  - I²C Bus Status Register (IBSR)
  - Serial Status Register (SSR)
  - Reception Data Register / Transmission Data Register (RDR/TDR)
  - 7-bit Slave Address Mask Register (ISMK)
  - 7-bit Slave Address Register (ISBA)
  - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
  - FIFO Control Register 1 (FCR1)
  - FIFO Control Register 0 (FCR0)
  - FIFO Byte Register (FBYTE1/FBYTE2)
- Interrupts of I²C Interface
  - Operation of I²C Interface Communication
  - Master Mode
  - Slave Mode
  - Bus Error
- Dedicated Baud Rate Generator
  - Example Flowchart for I²C Interface

24.20 Overview of I²C Interface

The I²C interface supports a bus between ICs and operates as a master/slave device on the I²C bus. This interface also comes with transmission/reception FIFO (up to 16 bytes each). There is no I²C function for ch.0.

Functions of I²C interface

The I²C interface has the following functions.

- Master/slave transmission & reception functionality
- Arbitration function
- Clock synchronization
- Transmission direction detection
- Generation and detection of repeated start condition
- Bus error detection
- General call addressing
- 7-bit addressing as master/slave
- Interrupts can be generated during transmission and bus errors.
- 10-bit addressing can be supported by a program.

Functions of FIFO

The FIFO has the following functions.

- Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes)[1]
- Transmission FIFO or reception FIFO selectable
- Transmission data can be resent.
- The interrupt timing for reception FIFO can be modified by software.
- FIFO reset is supported separately.

[1]: There is no FIFO between ch.0 and ch.3.

24.21 Registers of I²C Interface

This section lists the registers of the I²C interface.

List of registers of I²C interface

Table 24-37. Registers of the I²C (Sheet 1 of 3)

Channel	Abbreviated Register Name	Register Name	Reference
1	IBCR1	I²C bus control register 1	24.21.1
	SMR1	Serial mode register 1	24.21.2
	IBSR1	I²C bus status register 1	24.21.3
	BGR1	Baud rate generator register 1	24.21.8
	SSR1	Serial status register 1	24.21.4
	RDR1	Received data register 1	24.21.5
	TDR1	Transmitted data register 1	24.21.5
	ISMK1	7-bit slave address mask register 1	24.21.6
	ISBA1	7-bit slave address register 1	24.21.7
2	IBCR2	I²C bus control register 2	24.21.1
	SMR2	Serial mode register 2	24.21.2
	IBSR2	I²C bus status register 2	24.21.3
	BGR2	Baud rate generator register 2	24.21.8
	SSR2	Serial status register 2	24.21.4
	RDR2	Received data register 2	24.21.5
	TDR2	Transmitted data register 2	24.21.5
	ISMK2	7-bit slave address mask register 2	24.21.6
	ISBA2	7-bit slave address register 2	24.21.7
3	IBCR3	I²C bus control register 3	24.21.1
	SMR3	Serial mode register 3	24.21.2
	IBSR3	I²C bus status register 3	24.21.3
	BGR3	Baud rate generator register 3	24.21.8
	SSR3	Serial status register 3	24.21.4
	RDR3	Received data register 3	24.21.5
	TDR3	Transmitted data register 3	24.21.5
	ISMK3	7-bit slave address mask register 3	24.21.6
	ISBA3	7-bit slave address register 3	24.21.7

Table 24-37. Registers of the I²C (Sheet 2 of 3)

Channel	Abbreviated Register Name	Register Name	Reference
8	IBCR8	I²C bus control register 8	24.21.1
	SMR8	Serial mode register 8	24.21.2
	IBSR8	I²C bus status register 8	24.21.3
	BGR8	Baud rate generator register 8	24.21.8
	SSR8	Serial status register 8	24.21.4
	RDR8	Received data register 8	24.21.5
	TDR8	Transmitted data register 8	24.21.5
	FCR18	FIFO control register 18	24.21.9
	FCR08	FIFO control register 08	24.21.10
	FBYTE18	FIFO1 byte register 8	24.21.10
	FBYTE28	FIFO2 byte register 8	24.21.11
	ISMK8	7-bit slave address mask register 8	24.21.6
	ISBA8	7-bit slave address register 8	24.21.7
9	IBCR9	I²C bus control register 9	24.21.1
	SMR9	Serial mode register 9	24.21.2
	IBSR9	I²C bus status register 9	24.21.3
	BGR9	Baud rate generator register 9	24.21.8
	SSR9	Serial status register 9	24.21.4
	RDR9	Received data register 9	24.21.5
	TDR9	Transmitted data register 9	24.21.5
	FCR19	FIFO control register 19	24.21.9
	FCR09	FIFO control register 09	24.21.10
	FBYTE19	FIFO1 byte register 9	24.21.10
	FBYTE29	FIFO2 byte register 9	24.21.11
	ISMK9	7-bit slave address mask register 9	24.21.6
	ISBA9	7-bit slave address register 9	24.21.7
10	IBCR10	I²C bus control register 10	24.21.1
	SMR10	Serial mode register 10	24.21.2
	IBSR10	I²C bus status register 10	24.21.3
	BGR10	Baud rate generator register 10	24.21.8
	SSR10	Serial status register 10	24.21.4
	RDR10	Received data register 10	24.21.5
	TDR10	Transmitted data register 10	24.21.5
	FCR110	FIFO control register 110	24.21.9
	FCR010	FIFO control register 010	24.21.10
	FBYTE110	FIFO1 byte register 10	24.21.10
	FBYTE210	FIFO2 byte register 10	24.21.11
	ISMK10	7-bit slave address mask register 10	24.21.6
	ISBA10	7-bit slave address register 10	24.21.7

Table 24-37. Registers of the I²C (Sheet 3 of 3)

Channel	Abbreviated Register Name	Register Name	Reference
11	IBCR11	I²C bus control register 11	24.21.1
	SMR11	Serial mode register 11	24.21.2
	IBSR11	I²C bus status register 11	24.21.3
	BGR11	Baud rate generator register 11	24.21.8
	SSR11	Serial status register 11	24.21.4
	RDR11	Received data register 11	24.21.5
	TDR11	Transmitted data register 11	24.21.5
	FCR111	FIFO control register 111	24.21.9
	FCR011	FIFO control register 011	24.21.10
	FBYTE111	FIFO1 byte register 11	24.21.10
	FBYTE211	FIFO2 byte register 11	24.21.11
	ISMK11	7-bit slave address mask register 11	24.21.6
	ISBA11	7-bit slave address register 11	24.21.7

Table 24-38. Bit Assignment of I²C Interface

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IBCR/ SMR	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	MD2	MD1	MD0	-	RIE	TIE	-	-
SSR/ IBSR	REC	TSET	-	-	ORE	RDRF	TDRE	-	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
RDR/ TDR	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
ISMK/ ISBA	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

24.21.1 I²C Bus Control Register (IBCR)

The I²C bus control register (IBCR) selects master/slave mode, generates a repeated start condition, enables the acknowledge function, enables interrupts, bus error detection and displays an interrupt flag.

I²C bus control register (IBCR)

Figure 24-56 shows the bit structure of the I²C bus control register (IBCR), and Table 24-39 describes the function of each bit.

Figure 24-56. Bit Structure of I²C Bus Control Register (IBCR)

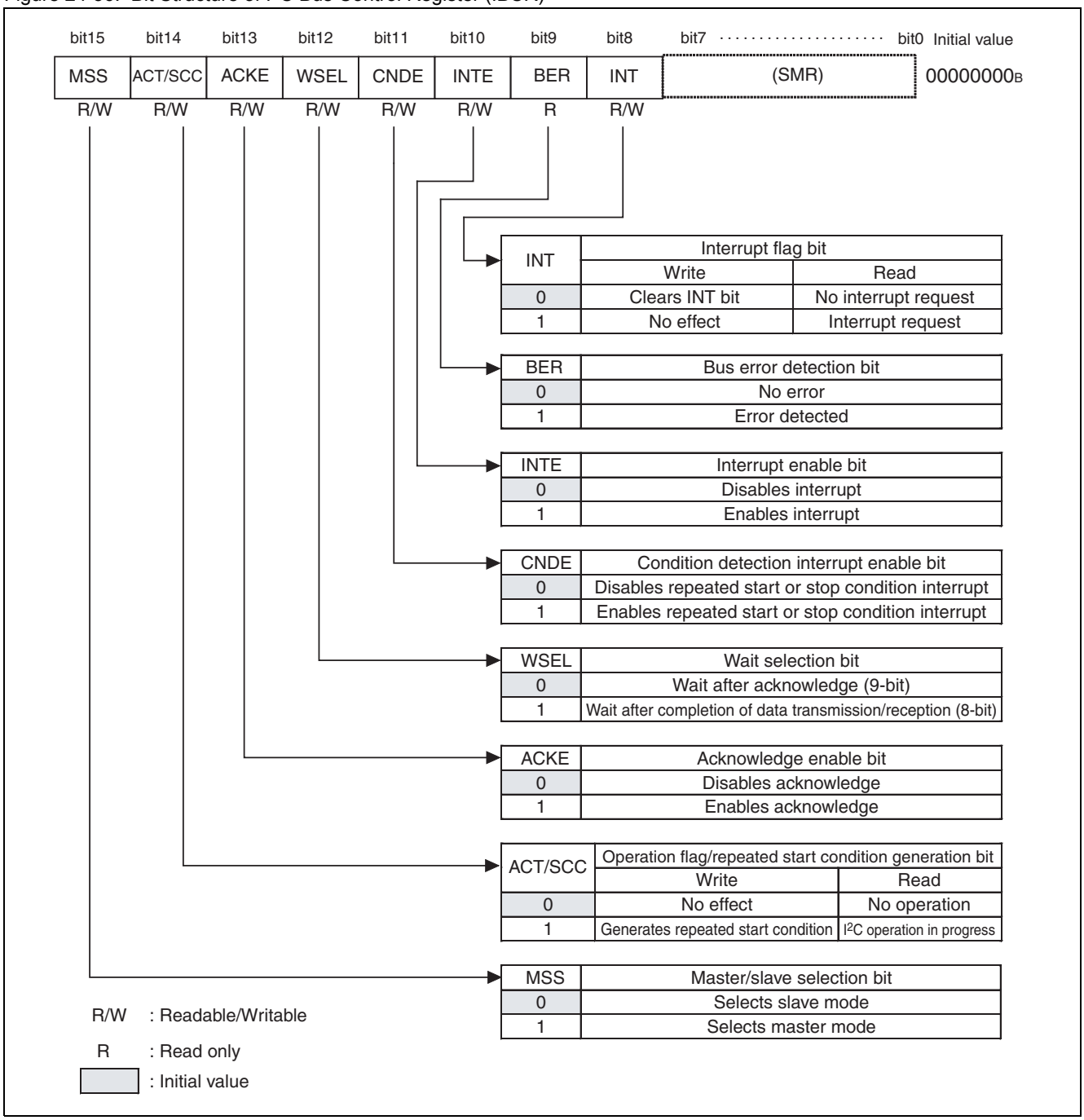


Table 24-39. Functional Description of Each Bit of I²C Bus Control Register (IBCR) (Sheet 1 of 3)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit15</td><td>MSS:
Master/slave selection bit</td><td>
■ Master mode will be selected if this bit is set to "1" when the I²C bus is in the idle state (EN = 1, BB = 0).
■ If this bit is set to "1" when the BB bit in the IBSR register is set to "1", the register will wait to generate a start condition until the BB bit becomes "0". If a slave address match occurs during that wait and the device operates as a slave, this bit will be set to "0" and the AL bit in the IBSR register will be set to "1".
■ A stop condition will be generated if "0" is written to this bit when the device is operating as the master (MSS = 1, ACT = 1) and the interrupt flag (INT) is set to "1".
The MSS bit is cleared under the following conditions.
■ The I²C interface is disabled (EN bit = 0).
■ An arbitration lost condition occurs.
■ A bus error is detected (BER bit = 1).
■ "0" is written to the MSS bit when INT is "1".
The relationship between the MSS and ACT bits is shown below.
<table>
<tr><th>MSS bit</th><th>ACT bit</th><th>Status</th></tr>
<tr><td>0</td><td>0</td><td>Idle</td></tr>
<tr><td>0</td><td>1</td><td>Slave operation in progress (slave mode) due to slave address match or ACK response[1] to reserved address</td></tr>
<tr><td>1</td><td>0</td><td>Master operation on standby</td></tr>
<tr><td>1</td><td>1</td><td>Master operation in progress (master mode)</td></tr>
</table>
[1]: ACK response: indicates that SDA of the I²C bus is at "L" during acknowledge.
Note: Change the MSS bit from "1" to "0" when the INT bit and MSS bit are set to "1". If "0" is written to the MSS bit when the ACT bit is set to "1", the INT bit will also be cleared to "0".
Writing "0" to the MSS bit returns "1" during master operation, as long as the ACT bit is set to "1".
</td></tr>
<tr><td>bit14</td><td>ACT/SCC:
Operation flag / repeated start condition generation bit</td><td>
This bit has different meanings between read and write.
<table>
<tr><th>Read</th><th>Write</th></tr>
<tr><td>ACT bit</td><td>SCC bit</td></tr>
</table>
The ACT bit indicates that the device is operating in master or slave mode.
Setting conditions for the ACT bit:
■ A start condition is output to the I²C bus (master mode).
■ A slave address matches the address transmitted from the master (slave mode).
■ A reserved address is detected and then an acknowledge is returned as a response (MSS = 0: slave mode).
Reset conditions for the ACT bit:
<Master mode>
■ A stop condition is detected.
■ An arbitration lost condition is detected.
■ A bus error is detected.
■ The I²C interface is disabled (EN bit = 0).
<Slave mode>
■ A (repeated) start condition is detected.
■ A stop condition is detected.
■ An acknowledge is not returned although a reserved address is detected (RSA bit = 1).
■ The I²C interface is disabled (EN bit = 0).
■ A bus error occurs (BER bit = 1).
A repeated start is performed when "1" is written to this bit during master mode.
Writing "0" is invalid.
Note: Write "1" to the SCC bit while an interrupt is occurring in master mode (MSS = 1, ACT = 1, INT = 1). The INT bit will be cleared to "0" if "1" is written to the SCC bit when the ACT bit is set to "1".
In slave mode (MSS = 0, ACT = 1), it is prohibited to write "1" to this bit.
The MSS bit has higher priority than the SCC bit, when "1" is written to the SCC bit and "0" is written to the MSS bit.
The SCC bit is read when a read modify write (RMW) instruction is used.
</td></tr>
</table>

Table 24-39. Functional Description of Each Bit of I²C Bus Control Register (IBCR) (Sheet 2 of 3)

Bit name		Function
bit13	ACKE: Acknowledge enable bit	■ If this bit is set to "1", "L" will be output when an acknowledge is returned. ■ When ACT is set to "1", this bit must be modified, if necessary, while the INT bit is set to "1". This bit is invalid under the following conditions. ■ An acknowledge is returned to an address field other than the reserved address (automatic generation). ■ Data transmission (RSA = 0, TRX = 1, FBT = 0) ■ An ACK is returned whenever the reception FIFO is enabled and slave reception is selected (FE = 1, MSS = 0, ACT = 1). ■ When the reception FIFO is enabled, WSEL is "0", and master reception is selected (FE = 1, MSS = 1, ACT = 1, WSEL = 0), setting the TDRE bit to "0" returns an ACK while setting it to "1" returns a NACK. An ACK is always returned when the reception FIFO is enabled, WSEL is "0", and slave transmission is performed through reserved address detection (RSA = 1, TRX = 1, FBT = 1). To allow a NACK to be returned, disable the reception FIFO and set ACKE to "0" during an interrupt after the reserved address detection. ■ The reception FIFO is enabled, WSEL is "1", and the transmission data register contains data in master reception (FE = 1, MSS = 1, ACT = 1, WSEL = 1, TDRE = 0).
bit12	WSEL: Wait selection bit	■ This bit is used to determine whether an interrupt should occur (INT = 1) before or after an acknowledgment to put the I²C bus in a wait state. ■ The WSEL bit is invalid under the following conditions. ❒ An interrupt occurs for the first byte[1] (INT = 1). ❒ A reserved address is detected (FBT = 1, RSA = 1). ❒ A NACK response[2] is detected during a data transfer when FIFO is used (FE = 1, RACK = 1, ACT = 1). ❒ The reception FIFO becomes full when it is used. [1]: First byte: indicates the data after a (repeated) start condition [2]: NACK response: indicates that SDA of the I²C bus is at "H" during acknowledgment.
bit11	CNDE: Condition detection interrupt enable bit	This bit is used to enable the occurrence of interrupts when a stop condition or a repeated start condition is detected in master or slave mode (ACT = 1). An interrupt occurs when the RSC or SPC bit in the IBSR register is set to "1" and this bit is set to "1".
bit10	INTE: Interrupt enable bit	This bit is used to enable an interrupt (INT = 1) for data transmission/reception and a bus error in master or slave mode.
bit9	BER: Bus error detection bit	This bit indicates that an error is detected on the I²C bus. Setting conditions for the BER bit: ■ A start condition or stop condition is detected during the transfer of the first byte[1]. ■ A (repeated) start condition or stop condition is detected at the 2nd bit - 9th (acknowledge) bit of data in the second or succeeding byte. Reset conditions for the BER bit: ■ "0" is written to the INT bit when BER is set to "1". ■ The I²C interface is disabled (EN = 0). [1]: First byte: indicates the data after (repeated) start condition **Note:** Data cannot be transmitted or received properly if this bit is set to "1" when the interrupt flag (INT bit) is set to "1". In this case, take an action such as retransmitting the data.

Table 24-39. Functional Description of Each Bit of I²C Bus Control Register (IBCR) (Sheet 3 of 3)

Bit name		Function
bit8	INT: Interrupt flag bit	This bit is set to "1" after the 8th or 9th bit (ACK) of data transmission/reception in master or slave mode, or upon the occurrence of a bus error. In cases other than the occurrence of a bus error, SCL is set to "L" when the INT bit is set to "1". When the INT bit is set to "0", SCL is released from the "L" state. Setting conditions for the INT bit: <8th bit> ■ A reserved address is detected in the first byte. ■ WSEL is "1", and an arbitration lost condition is detected in the second or succeeding byte. ■ WSEL is "1", and the TDRE bit is set to "1" in the second or succeeding byte during master operation. ■ WSEL is "1", the reception FIFO is disabled and the TDRE bit is set to "1" in the second or succeeding byte during slave operation. ■ WSEL is set to "1", and the TDRE bit is set to "1" in the second or succeeding byte during slave transmission. <9th bit> ■ An arbitration lost condition is detected in the first byte. ■ A NACK is received at times other than when a stop condition output is set ("0" written to the MSS bit during master operation). ■ The TDRE bit is set to "1" in the transmission direction (TRX = 1) of master or slave mode without the detection of a reserved address in the first byte. ■ The reception FIFO contains data when the reception FIFO is enabled in the reception direction (TRX = 0) of master or slave mode without the detection of a reserved address in the first byte. ■ The TDRE bit is set to "1" when the reception FIFO is disabled in the reception direction (TRX = 0) of master or slave mode without the detection of a reserved address in the first byte. ■ WSEL is set to "0", and an arbitration lost condition is detected in the second or succeeding byte. ■ WSEL is set to "0", and the TDRE bit is set to "1" in the second or succeeding byte during master mode operation. ■ WSEL is set to "0", and the TDRE bit is set to "1" in the second or succeeding byte during slave transmission. ■ WSEL is set to "0", the reception FIFO is disabled, and slave reception is selected. In slave reception, however, an interrupt does not occur in the 9th bit of the first byte in which a reserved address is detected. ■ The reception FIFO is enabled, and it becomes full in slave reception. <Other condition> A bus error is detected. Reset conditions for the INT bit: ■ "0" is written to the INT bit. ■ "0" is written to the MSS bit when the INT bit is "1" and the ACT bit is "1". ■ "1" is written to the SCC bit when the INT bit is "1" and the ACT bit is "1". Writing "1" to the INT bit is invalid. **Note:** Setting the EN bit to "0" may set the RDRF and INT bits to "1", depending on the reception timing. In this case, read the reception data to clear the INT bit. "1" is read when a read modify write (RMW) instruction is used. The INT bit cannot be set to "1" even if the reception FIFO is full in master reception operation when the reception FIFO is enabled.

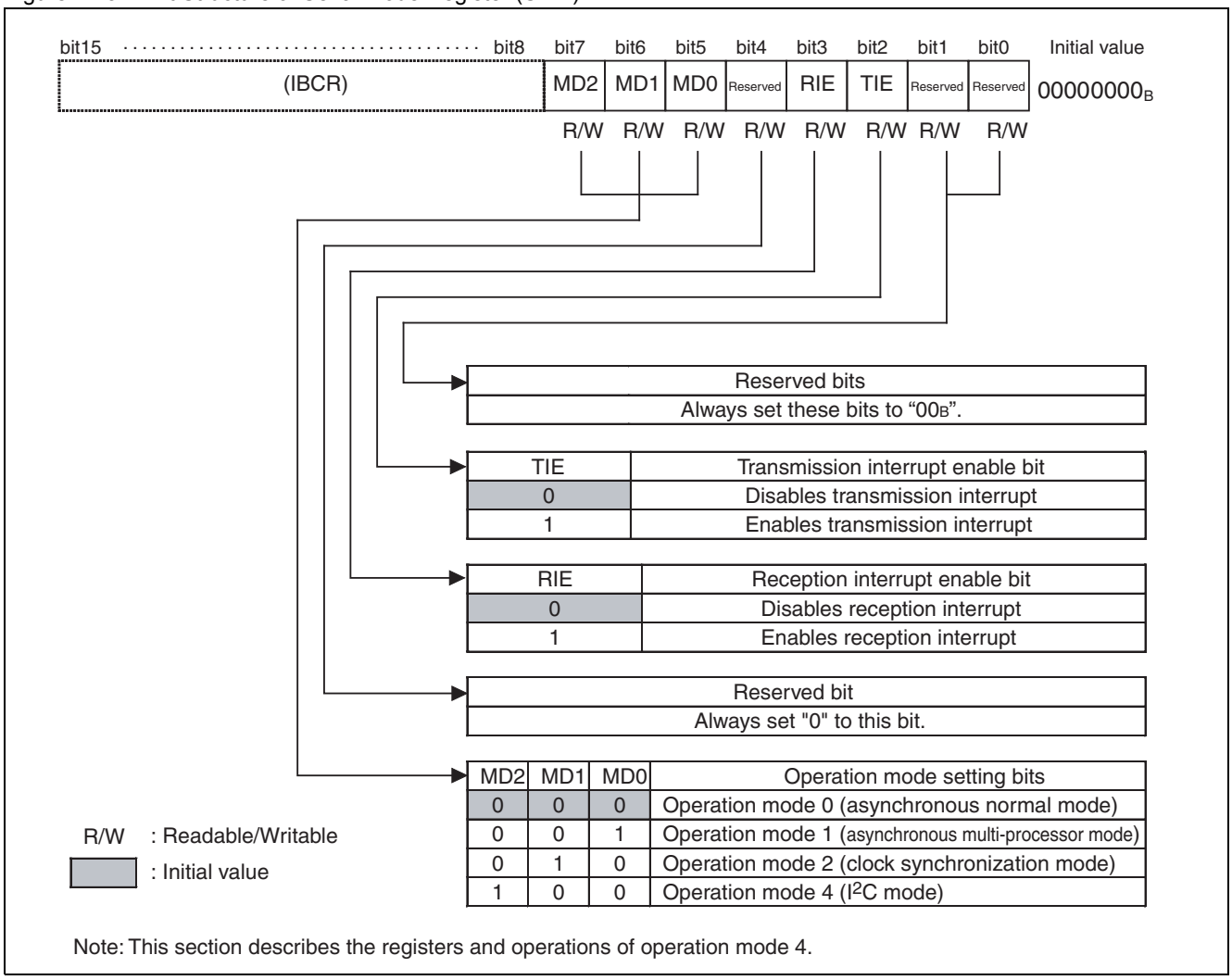
24.21.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, and enables or disables transmission/reception interrupts.

Serial Mode Register (SMR)

Figure 24-57 shows the bit structure of the serial mode register (SMR), and Table 24-40 describes the function of each bit.

Figure 24-57. Bit Structure of Serial Mode Register (SMR)



Note: This section describes the registers and operations of operation mode 4.

Table 24-40. Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	These bits are used to select the operation mode. "000_B": Selects operation mode 0 (asynchronous normal mode) "001_B": Selects operation mode 1 (asynchronous multi-processor mode) "010_B": Selects operation mode 2 (clock synchronization mode) "100_B": Selects operation mode 4 (I^2C mode) This section describes the registers and operations of operation mode 4 (I^2C mode). **Note:** Settings other than above are prohibited. To switch the operation mode, disable I^2C first (ISMK:EN = 0). Set each register after selecting the operation mode.
bit4	Reserved bit	Always set "0" to this bit.
bit3	RIE: Reception interrupt enable bit	■ This bit is used to enable or disable the output of reception interrupt requests to the CPU. ■ A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or the error flag bit (ORE) is set to "1". **Note:** Set this bit to "0" when receiving data using the INT bit in the I^2C bus control register (IBCR).
bit2	TIE: Transmission interrupt enable bit	■ This bit is used to enable or disable the output of transmission interrupt requests to the CPU. ■ A transmission interrupt request is output when the TIE and TDRE bits are set to "1". **Note:** Set this bit to "0" when transmitting data using the INT bit in the I^2C bus control register (IBCR).
bit1, bit0	Reserved bits	Always set these bits to "00_B".

Note:

The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed.

- I^2C Bus Control Register (IBCR)
- I^2C Bus Status Register (IBSR)

Note, however, that when IBCR and SMR are written simultaneously with 16-bit write access, IBCR reflects the written content.

24.21.3 I²C Bus Status Register (IBSR)

The I²C bus status register (IBSR) indicates the detection of a first byte, a reserved address, a repeated start condition, acknowledge, data direction, arbitration lost condition, stop condition, and I²C bus status.

I²C bus status register (IBSR)

Figure 24-58 shows the bit structure of the I²C bus status register (IBSR) and Table 24-41 describes the function of each bit.

Figure 24-58. Bit Structure of I²C Bus Status Register (IBSR)

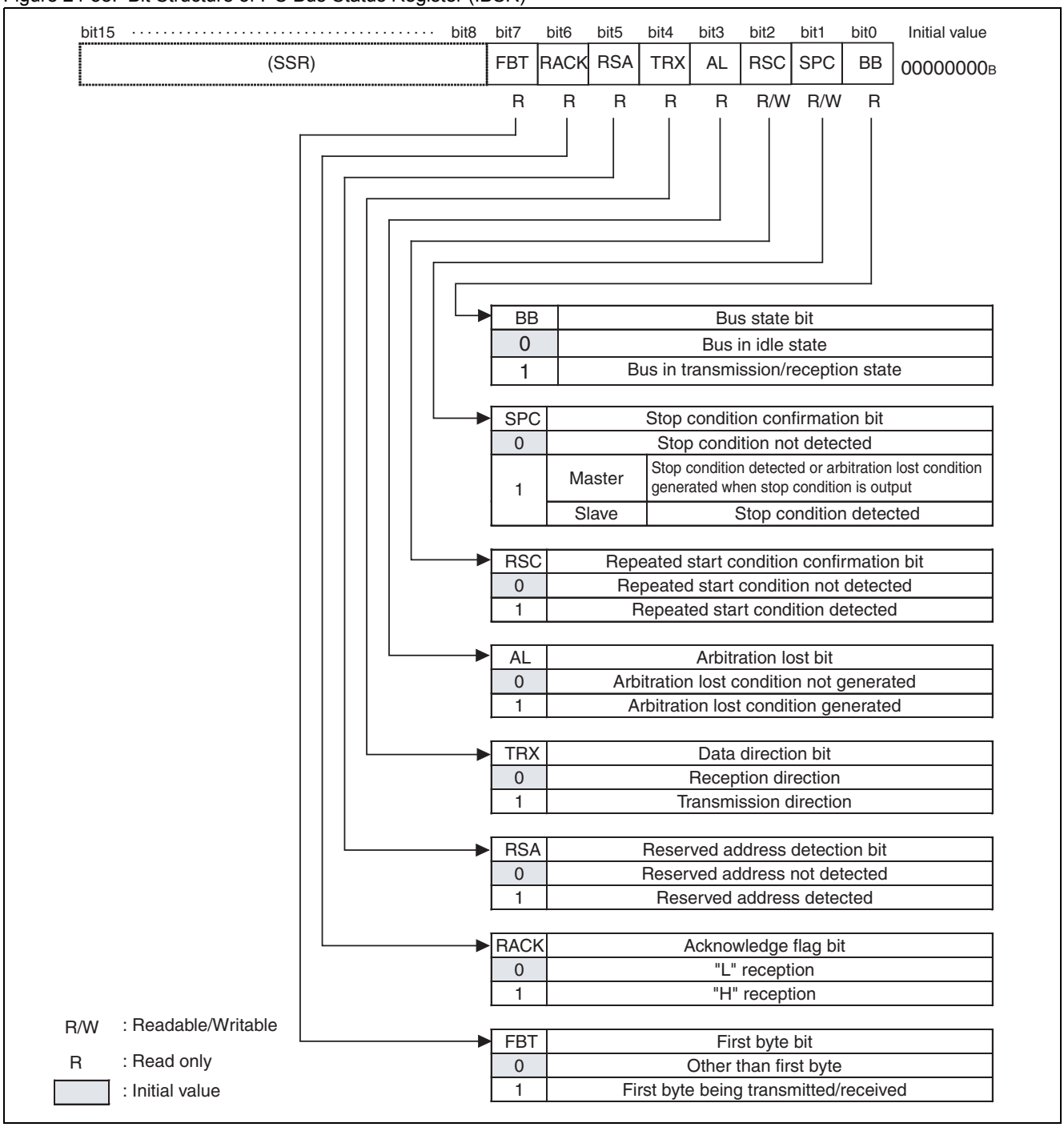


Table 24-41. Functional Description of Each Bit of I²C Bus Status Register (Sheet 1 of 2)

Bit name		Function
bit7	FBT: First byte bit	This bit indicates the first byte. Setting condition for the FBT bit: ■ A (repeated) start condition is detected. Clearing conditions for the FBT bit: 1. The second byte is transmitted or received. 2. A stop condition is detected. 3. The I²C interface is disabled (EN bit = 0). 4. A bus error is detected (BER bit = 1).
bit6	RACK: Acknowledge flag bit	This bit is used to indicate the acknowledge received for the first byte during master or slave mode. Update conditions for the RACK bit 1. Acknowledge for the first byte 2. Acknowledge for data in master or slave mode Clearing conditions for the RACK bit (RACK bit = 0) 1. A (repeated) start condition is detected. 2. The I²C interface is disabled (EN bit = 0). 3. A bus error is detected (BER bit = 1).
bit5	RSA: Reserved address detection bit	This bit indicates the detection of a reserved address. Setting condition for the RSA bit (RSA = 1) ■ The first byte is set to "0000XXXX" or "1111XXXX". "X" can be "0" or "1". Reset conditions for the RSA bit (RSA = 0) 1. A (repeated) start condition is detected. 2. A stop condition is detected. 3. The I²C interface is disabled (EN bit = 0). 4. A bus error is detected (BER bit = 1). When the RSA bit is set to "1" in the first byte, the interrupt flag (INT) is set to "1" at the falling edge of SCL in the 8th bit of the first byte to set the SCL to "L", whether the FIFO is enabled or disabled. In this case, ACKE should be set to "1" and the interrupt flag (INT) should be cleared to "0" in order to read reception data and allow the device to operate as a slave. If the TRX bit is set to "0", the device will receive data as a slave. To disable data reception in the middle of the operation, set the ACKE bit to "0". No more data will be received afterward. **Note:** When ACKE is set to "0" during a data transfer, it is prohibited to set ACKE to "1" until a stop condition or repeated start condition is detected. If slave transmission is confirmed during an interrupt by the detection of a reserved address, an ACK will be returned when the reception FIFO has been enabled. Therefore, disable the reception FIFO and set ACKE to "0".
bit4	TRX: Data direction bit	This bit indicates the data direction. Setting conditions for the TRX bit: 1. A (repeated) start condition is transmitted in master mode. 2. The 8th bit of the first byte is "1" in slave mode (transmission direction as a slave). Reset conditions for the TRX bit: 1. An arbitration lost condition is generated (AL = 1). 2. The 8th bit of the first byte is "0" in slave mode (reception direction as a slave). 3. The 8th bit of the first byte is "1" in master mode (reception direction as the master). 4. A stop condition is detected. 5. A (repeated) start condition is detected in modes other than master mode. 6. The I²C interface is disabled (EN bit = 0). 7. A bus error is detected (BER bit = 1).

Table 24-41. Functional Description of Each Bit of I²C Bus Status Register (Sheet 2 of 2)

Bit name		Function
bit3	AL: Arbitration lost bit	This bit indicates an arbitration lost condition. Setting conditions for the AL bit: 1. The output data is different from the received data in master mode. 2. The device is operating as a slave although the MSS bit has been set to "1". 3. A repeated start condition is detected in the first bit of the data contained in the second or succeeding byte in master mode. 4. A stop condition is detected in the first bit of the data contained in the second or succeeding byte in master mode. 5. A repeated start condition cannot be generated in master mode, despite attempts to do so. 6. A stop condition cannot be generated in master mode, despite attempts to do so. Reset conditions for the AL bit: 1. "1" is written to the MSS bit. 2. "0" is written to the INT bit. 3. "0" is written to the SPC bit when the AL and SPC bits are set to "1". 4. The I²C interface is disabled (EN bit = 0). 5. A bus error is detected (BER bit = 1).
bit2	RSC: Repeated start condition confirmation bit	This bit indicates that a repeated start condition has been detected in master or slave mode. Setting condition for the RSC bit: A repeated start condition is detected after acknowledgment during slave or master mode operation. Reset conditions for the RSC bit: 1. "0" is written to the RSC bit. 2. "1" is written to the MSS bit. 3. The I²C interface is disabled (EN bit = 0). Writing "1" to this bit is invalid. **Note:** Slave mode will end unless ACK is returned when the device is operating reception in slave mode by the detection of a reserved address. Consequently, this bit will not be set to "1" even if a repeated start condition is detected. "1" is read when a read modify write (RMW) instruction is used.
bit1	SPC: Stop condition confirmation bit	This bit indicates that a stop condition has been detected in master or slave mode. Setting conditions for the SPC bit: 1. A stop condition is detected during slave or master mode operation. 2. An arbitration lost condition is generated when a stop condition is generated in master mode. Reset conditions for the SPC bit: 1. "0" is written to this bit. 2. "1" is written to the MSS bit. 3. The I²C interface is disabled (EN bit = 0). Writing "1" to this bit is invalid. **Note:** Slave mode will end unless ACK is returned when the device is operating reception in slave mode by the detection of a reserved address. Consequently, this bit will not be set to "1" even if a stop condition is detected. "1" is read when a read modify write (RMW) instruction is used.
bit0	BB: Bus state bit	This bit indicates the bus state. Setting condition for the BB bit: ■ "L" is detected at SDA or SCL of the I²C bus. Reset conditions for the BB bit: 1. A stop condition is detected. 2. The I²C interface is disabled (EN bit = 0). 3. A bus error is detected (BER bit = 1).

24.21.4 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status.

Serial Status Register (SSR)

Figure 24-59 shows the bit structure of the serial status register (SSR) and Table 24-42 describes the function of each bit.

Figure 24-59. Bit Structure of Serial Status Register (SSR)

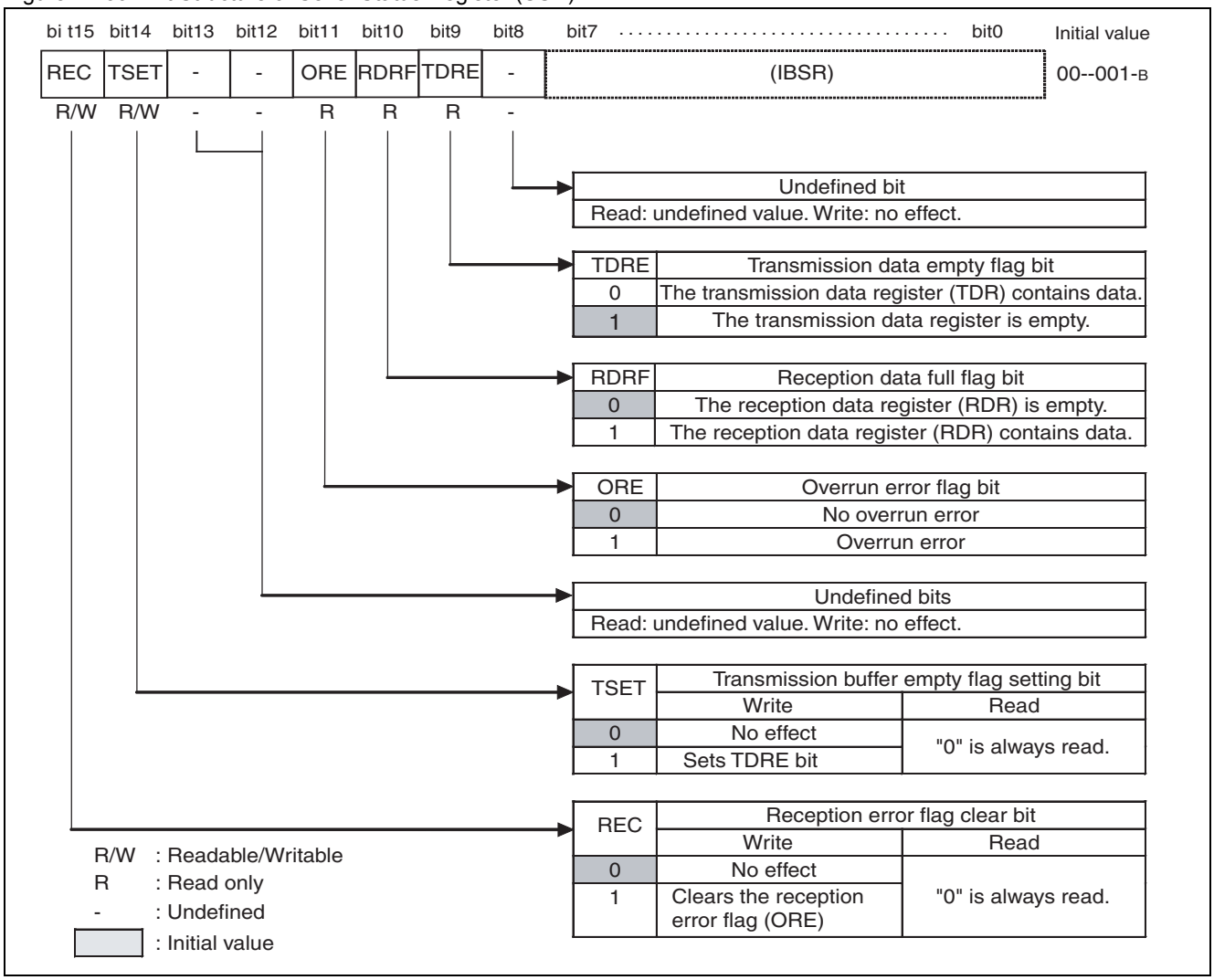


Table 24-42. Functional Description of Each Bit of Serial Status Register (SSR)

Bit name		Function
bit15	REC: Reception error flag clear bit	This bit is used to clear the ORE bit in the serial status register (SSR). ■ Writing "1" clears the ORE bit. ■ Writing "0" has no effect. Reading this bit always returns "0".
bit14	TSET: Transmission buffer empty flag setting bit	This bit is used to set the TDRE bit in the serial status register (SSR). ■ Writing "1" sets the TDRE bit. ■ Writing "0" has no effect. Reading this bit always returns "0".
bit13, bit12	Undefined bits	Read: undefined value Write: no effect
bit11	ORE: Overrun error flag bit	■ This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). ■ A reception interrupt request is output when the ORE and RIE bits are set to "1". ■ When this flag is set, the data in the reception data register (RDR) is invalid. ■ If this flag is set during the use of the reception FIFO, the reception data will not be stored to the reception FIFO.
bit10	RDRF: Reception data full flag bit	■ This flag indicates the status of the reception data register (RDR). ■ A reception interrupt request is output when the RIE bits and the reception data flag bit (RDRF) are set to "1". ■ The bit is set to "1" when reception data is loaded to RDR. The bit is cleared to "0" when the reception data register (RDR) is read. ■ This bit is set at the falling edge of SCL in the 8th bit of data. ■ It is also set by a NACK response. ■ RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. ■ This bit is cleared to "0" when the reception FIFO, if used, becomes empty. ■ During the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the reception baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO as well as the BER bit is set to "0". If RDR is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. **Note:** NACK response: indicates that SDA of the I^2C bus is at "H" during acknowledge.
bit9	TDRE: Transmission data empty flag bit	■ This flag indicates the status of the transmission data register (TDR). ■ A transmission interrupt request is output when the TIE and TDRE bits are set to "1" ■ When transmission data is written to TDR, the bit becomes "0", indicating that TDR contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR no longer contains any valid data. ■ This bit is set when "1" is written to the TSET bit in the serial status register (SSR). This bit is used to set the TDRE bit to "1" when an arbitration lost condition or bus error is detected.
bit8	Undefined bit	Read: undefined value Write: no effect

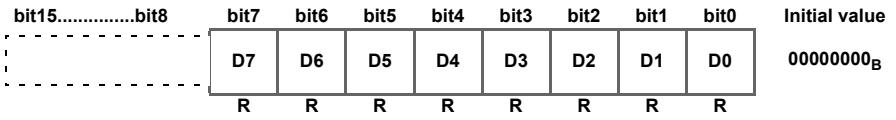
24.21.5 Reception Data Register / Transmission Data Register (RDR/TDR)

The reception data register and transmission data register are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

Reception Data Register (RDR)

Figure 24-60 illustrates the bit structure of the serial reception register (RDR).

Figure 24-60. Bit Structure of Reception Data Register (RDR)



R: Read only bit8 is unused in I^2C mode.

The reception data register (RDR) is a data buffer register for serial data reception.

- A serial data signal sent to a serial data line (SDA pin) is converted through the shift register and then stored in the reception data register (RDR).
- When the first byte[1] is received, the least significant bit (RDR: D0) becomes the data direction bit.
- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR).
- The reception data full flag bit (SSR: RDRF) is cleared to "0" automatically, when the reception data register (RDR) is read.

[1]: Indicates the data after a (repeated) start condition.

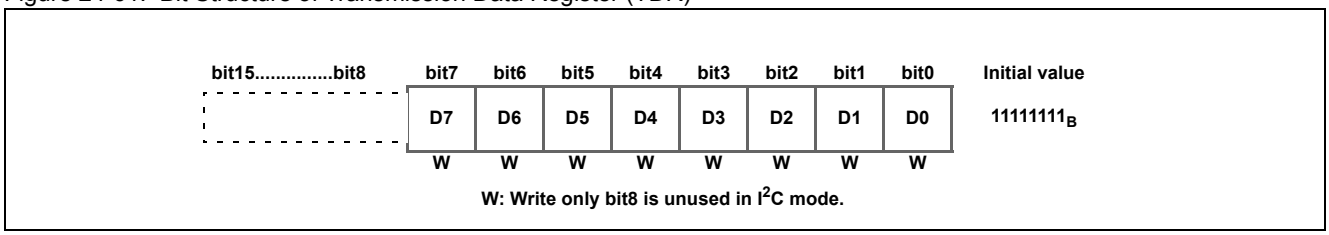
Notes:

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.

Transmission Data Register (TDR)

Figure 24-61. illustrates the bit structure of the transmission data register.

Figure 24-61. Bit Structure of Transmission Data Register (TDR)



The transmission data register (TDR) is a data buffer register for serial data transmission.

- Data is output to the serial data line (SDA pin), based on the transmission data register (TDR) value (MSB first).
- The least significant bit (TDR: D0) becomes the data direction bit when transmitting the first byte.
- The transmission data empty flag (SSR: TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- The transmission data empty flag (SSR: TDRE) is set to "1" when transmission data is transferred to the transmission shift register.
- Write the next transmission data under the following conditions.
  - The interrupt flag (INT bit) is set to "1".
  - No bus error is occurring (BER bit = 0).
  - Acknowledge is returned as ACK response ("0" is received as acknowledgment).
- Transmission data cannot be written to the transmission data register (TDR) if the data empty flag (SSR: TDRE) is set to "0" when the transmission FIFO is disabled.
- Transmission data can be written up to the capacity of the transmission FIFO, even if the data empty flag (SSR: TDRE) is set to "0" when the transmission FIFO is used.

Note: The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. The two registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.

24.21.6 7-bit Slave Address Mask Register (ISMK)

The 7-bit slave address mask register (ISMK) determines whether each bit of a slave address should be compared.

7-bit slave address mask register (ISMK)

Figure 24-62 shows the bit structure of the 7-bit slave address mask register (ISMK) and Table 24-43 describes the function of each bit.

Figure 24-62. Bit Structure of 7-bit Slave Address Mask Register (ISMK)

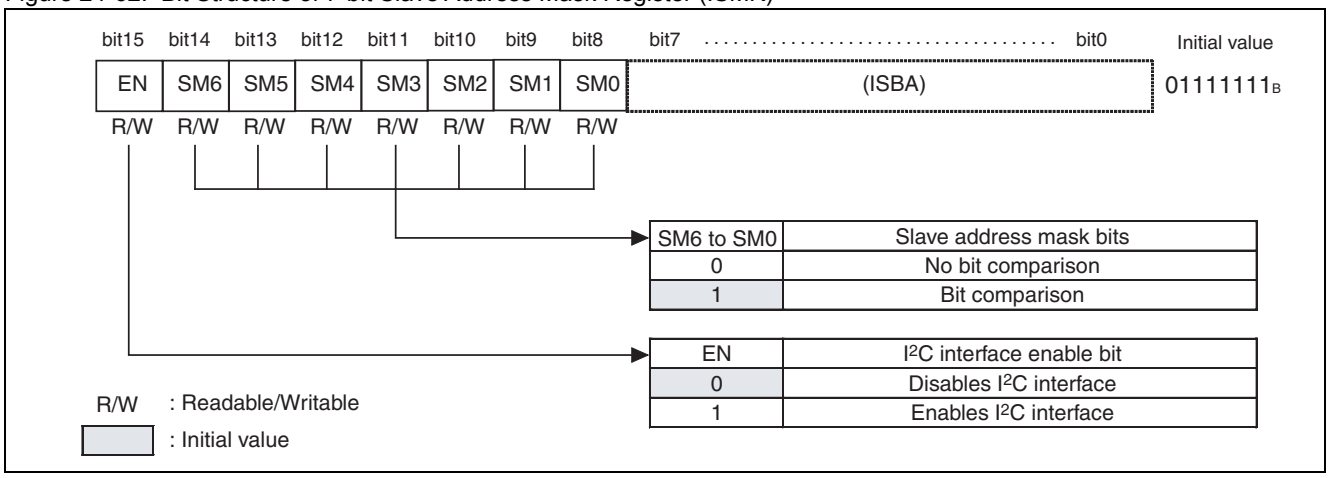


Table 24-43. Functional Description of Each Bit of 7-bit Slave Address Mask Register (ISMK)

Bit name		Function
bit15	EN: I^2C interface enable bit	This bit is used to enable or disable the operation of the I^2C interface. Setting the bit to "0" disables the operation of the I^2C interface. Setting the bit to "1" enables the operation of the I^2C interface. **Note:** This bit is not cleared to "0" even when the BER bit in the IBSR register is set to "1". Set the baud rate generator when this bit is set to "0". Set a 7-bit slave address and the 7-bit slave mask register when this bit is set to "0". Setting the EN bit to "0" during transmission may generate a pulse at SDA/SCL of the I^2C bus. If FIFO has been enabled, write "0" to the EN bit after disabling FIFO.
bit14 to bit8	SM6 to SM0: Slave address mask bits	These bits are used to determine whether to compare the 7-bit slave address with the received address. Bit set to "1": compared Bit set to "0": handled as matched **Note:** Set this register when the EN bit is "0".

24.21.7 7-bit Slave Address Register (ISBA)

The 7-bit slave address register (ISBA) sets a slave address.

7-bit slave address register (ISBA)

Figure 24-63 shows the bit structure of the 7-bit slave address register (ISBA) and Table 24-44 describes the function of each bit.

Figure 24-63. Bit Structure of 7-bit Slave Address Register (ISBA)

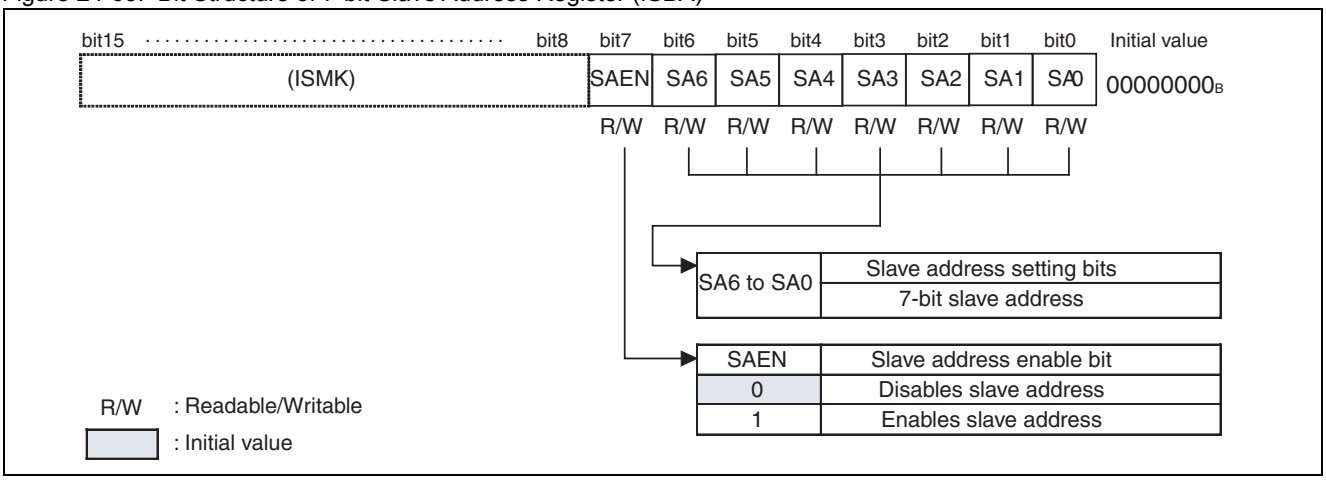


Table 24-44. Functional Description of Each Bit of 7-bit Slave Address Register (ISBA)

Bit name		Function
bit7	SAEN: Slave address enable bit	This bit is used to enable the detection of a slave address. Setting the bit to "0":Slave address not detected Setting the bit to "1":ISBA/ISMK setting compared with the first byte of received data
bit6 to bit0	SA6 to SA0: 7-bit slave address	If slave address detection has been enabled (SAEN = 1), the 7-bit data which is received after the detection of a (repeated) start condition will be compared with the value contained in the 7-bit slave address register (ISBA). If all the bits match, the device will operate in slave mode and output an ACK. At this point, the received slave address will be set to this register (An ACK will not be output if SAEN is set to "0"). The address bits which are set to "0" in the ISMK register are not subject to this comparison. **Note:** It is prohibited to set a reserved address. Set this register when the EN bit in the ISMK register is "0".

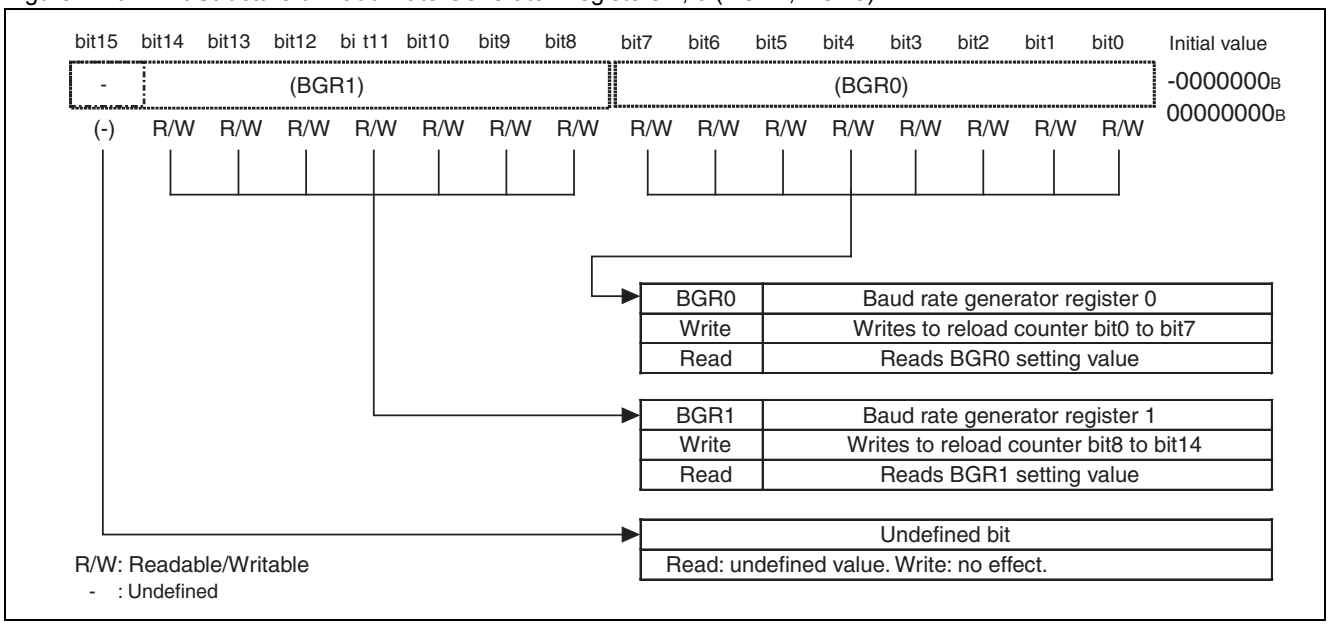
24.21.8 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock.

Bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0)

Figure 24-64 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 24-64. Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)



The baud rate generator registers are used to set a division ratio for the serial clock.

BGR0 and BGR1 correspond to the upper bits and lower bits respectively and they can write a reload value to be counted as well as read BGR1/BGR0 setting values.

The reload counter starts counting when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- Set the baud rate generator registers when the EN bit in the ISMK register is "0".
- Set a baud rate regardless of master or slave mode.
- Use the peripheral clock (PCLK) at 8 MHz or higher in operation mode 4 (I^2C mode). It is prohibited to set the baud rate generator to higher than 400kbps.

24.21.9 FIFO Control Register 1 (FCR1)

The FIFO control register 1 (FCR1) selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

Bit structure of FIFO control register 1 (FCR1)

Figure 24-65 shows the bit structure of the FIFO control register 1 (FCR1) and Table 24-45 describes the function of each bit.

Figure 24-65. Bit Structure of FIFO Control Register 1 (FCR1)

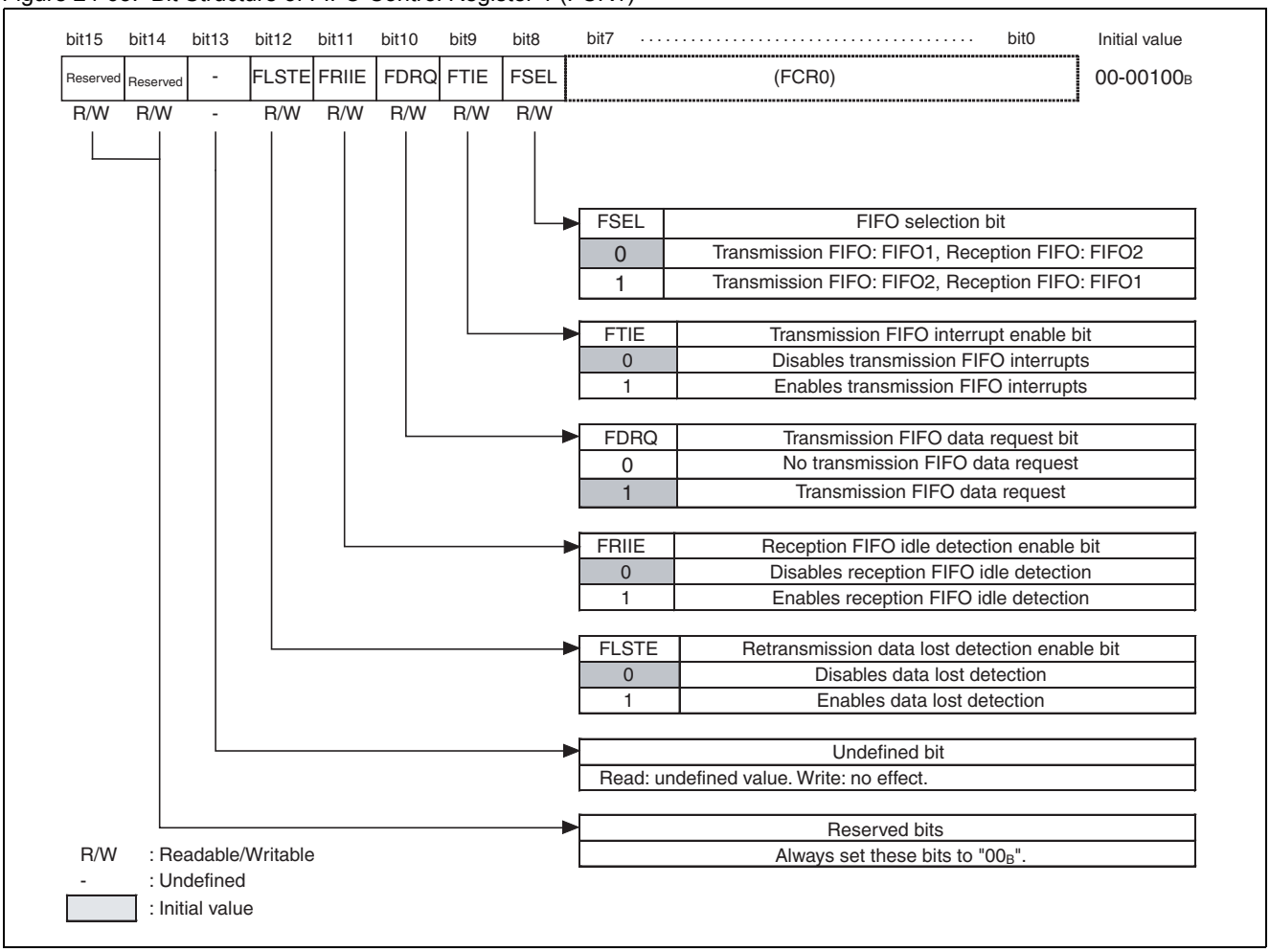


Table 24-45. Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	Reserved bits	Always set these bits to "00_B".
bit13	Undefined bit	Read: undefined value Write: no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. **Note:** To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8 clocks with the baud rate clock or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition ■ FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) ■ Transmission FIFO reset FDRQ reset condition ■ Writing "0" to this bit ■ When the transmission FIFO is full. **Note:** It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". It is prohibited to modify the FSEL bit when this bit is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. **Note:** This bit cannot be cleared by resetting FIFO (FCL2, FCL1 = 1). To modify this bit, disable FIFO operation (FCR0:FE2, FE1 = 0) in advance.

24.21.10 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

Bit structure of FIFO control register 0 (FCR0)

Figure 24-66 shows the bit structure of the FIFO control register 0 (FCR0) and Table 24-46 describes the function of each bit.

Figure 24-66. Bit Structure of FIFO Control Register 0 (FCR0)

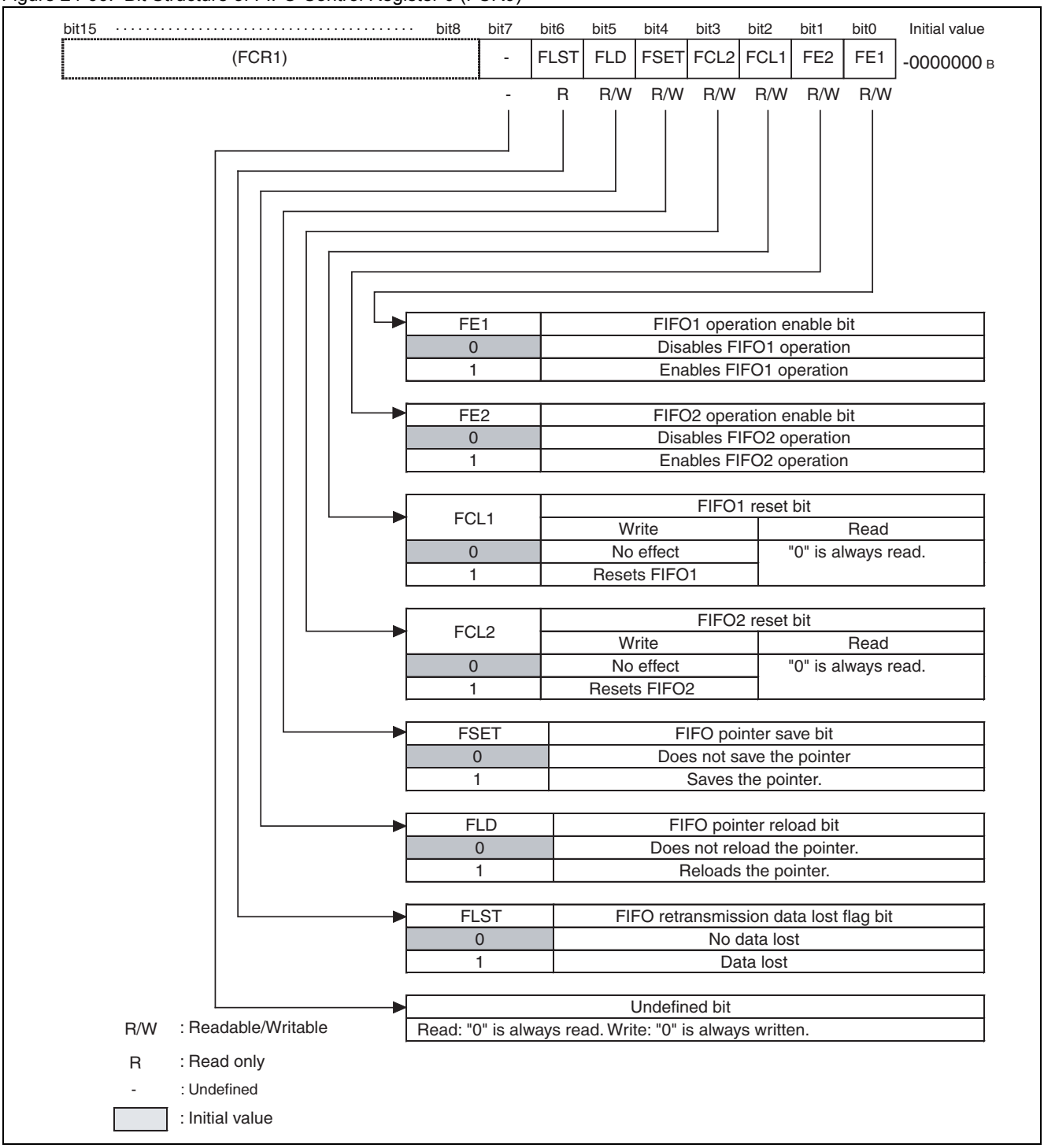


Table 24-46. Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (Sheet 1 of 2)

Bit name		Function
bit7	Undefined bit	Read:"0" is always read. Write:Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition ■ Writing to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. FLST reset conditions ■ FIFO reset (writing "1" to FCL) ■ Writing "1" to the FSET bit Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.
bit5	FLD: FIFO pointer reload bit	This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs. The bit becomes "0" when retransmission has been set. **Note:** Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset. It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress. Write "1" to this bit after setting the TIE bit to "0". And then, set the TIE bit to "1" when the transmission FIFO has been enabled.
bit4	FSET: FIFO pointer save bit	This bit is used to save the read pointer of the transmission FIFO. If the FLST bit is set to "0", saving the read pointer prior to transmission will enable retransmission in case that an error such as a communication error occurs. Setting the bit to "1" saves the current read pointer value. Setting the bit to "0" has no effect. **Note:** Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".
bit3	FCL2: FIFO2 reset bit	This bit is used to reset FIFO2. Setting this bit to "1" initializes the internal state of FIFO2. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. **Note:** Disable FIFO2 before resetting it. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE2 register will become "0".
bit2	FCL1: FIFO1 reset bit	This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. **Note:** Disable FIFO1 before resetting it. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".
bit1	FE2: FIFO2 operation enable bit	This bit is used to enable/disable FIFO2 operation. ■ To use FIFO2, set this bit to "1". ■ This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. ■ Set this bit to "1" or "0" when the transmission data is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception data is empty (RDRF = 0) to use it as the reception FIFO. ■ Even when FIFO2 is disabled, its status is retained. **Note:** Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1". To allow the device to operate in slave transmission when it has been selected as the reception FIFO to detect a reserved address, use an interrupt generated by the detection of a reserved address to set this bit to "0" and ACKE to "0". It the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0". To switch this bit from "0" to "1", set the TIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO2 contains data.

Table 24-46. Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (Sheet 2 of 2)

Bit name		Function
bit0	FE1: FIFO1 operation enable bit	This bit is used to enable/disable FIFO1 operation. ■ To use FIFO1, set this bit to "1". ■ This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. ■ Set this bit to "1" or "0" when the transmission data is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception data is empty (RDRF = 0) to use it as the reception FIFO. ■ Even when FIFO1 is disabled, its status is retained. **Note:** Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1". To allow the device to operate in slave transmission when it has been selected as the reception FIFO to detect a reserved address, use an interrupt generated by the detection of a reserved address to set this bit to "0" and ACKE to "0". It the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0". To switch this bit from "0" to "1", set the TIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO1 contains data.

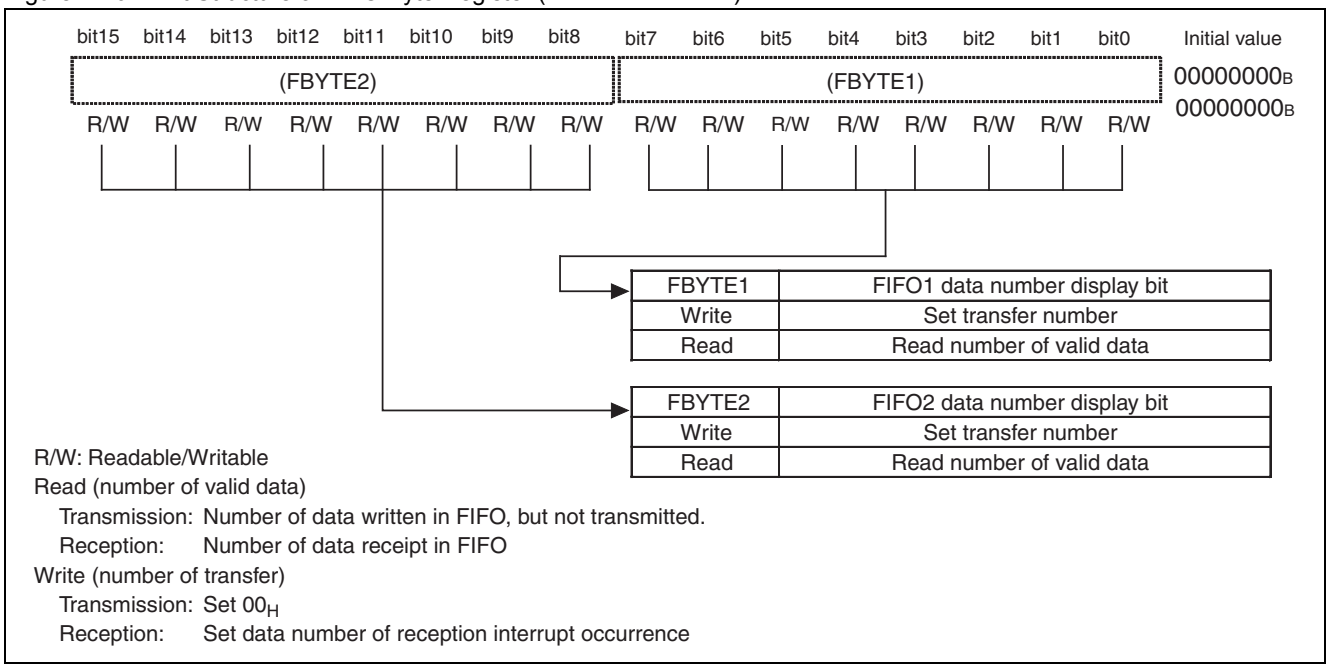
24.21.11 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO. This register can also be used to determine whether a reception interrupt should occur when the reception FIFO receives a specified number of data elements.

Bit structure of FIFO byte register (FBYTE1/FBYTE2)

Figure 24-67 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 24-67. Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)



The FBYTE1/FBYTE2 register indicates the number of valid data elements for FIFO. The details are as follows, depending on the FCR1:FSEL bit setting.

Table 24-47. Displaying the Number of Data Elements

FSEL	FIFO selection	Displaying the number of data elements
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is "08_H".
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- To receive data in master operation (master reception), set the TIE bit to "0", set the number of data elements to be received to the FBYTE1/FBYTE2 register of the transmission FIFO, and write "0" to the FDRQ bit. The SCL clock will be output for a specified amount of data, and then the INT bit will be set to "1". To set the TIE bit to "1", wait until the FDRQ becomes "1".

Notes:

- In master operation, set "00_H" to FBYTE1/FBYTE2 of the transmission FIFO except when receiving data.
- Set the number of transmission data elements when receiving data in master operation, when the transmission FIFO is empty and the TIE bit is set to "0".
- To disable the I^2C interface (EN = 0) during data reception in master operation, disable transmission/reception FIFO first.
- Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
- Modify the register after disabling transmission/reception.
- Read modify write (RMW) instructions cannot be used for this register.
- Settings that will exceed the capacity of FIFO are prohibited.

24.22 Interrupts of I²C Interface

The following sources can be used to generate interrupt requests for the I²C interface.

- After the transmission/reception of the first byte or data
- Stop condition
- Repeated start condition
- FIFO transmission data request
- Completion of FIFO reception data

Interrupts of I²C interface

Table 24-48 shows the interrupt control bits and interrupt sources of the I²C interface.

Table 24-48. Interrupt Control Bits and Interrupt Sources of I²C Interface

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Status	INT	IBCR	After transmission/reception of 1st byte[1]	IBCR:INTE	Writing "0" to interrupt flag bit (IBCR:INT)
			After transmission/reception of data[1]		
			Detection of bus error		
			Detection of arbitration lost condition		
			Detection of reserved address		Writing "0" to interrupt flag bit (IBCR:INT) after reading reception data (RDR) until reception FIFO becomes empty
	SPC	IBSR	Stop condition	IBCR:CNDE	Writing "0" to stop condition detection bit (IBSR:SPC)
	RSC	IBSR	Repeated start condition		Writing "0" to repeated start detection flag bit (IBSR:RSC)
Reception	RDRF	SSR	After reception of reserved address	SMR:RIE	Reading reception data (RDR)
			After reception of data		
			Reception of the amount set by FBYTE		Reading reception data (RDR) until reception FIFO becomes empty
			Detection of the idle state of reception for 8 clocks with the baud rate clock or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	ORE	SSR	Overrun error		Writing "1" to reception error flag bit (SSR:REC)
Transmission	TDRE	SSR	Transmission register being empty	SMR:TIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)[2]
			Writing "1" to transmission buffer empty flag setting bit (SSR:TSET)		
	FDRQ	FCR1	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit, or transmission FIFO being full

[1]: Normal data can be transmitted or received. An interrupt does not occur when TDRE is set to "0". This function is designed to support DMA transfer. To perform DMA transfer at reception, it is required to write to the transmission buffer for each 1 byte reception, and set TDRE to "0". Please perform a dummy write to TDR at the different ch of DMA. It is recommended to perform DMA transfer at ch.1 and ch.2 of I²C, which separately has interrupt vectors for reception/ transmission/ status interrupt.
The TDRE bit must be set to "1" before the INT flag is set, in order to generate the INT flag in data transmission/reception.
[2]: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

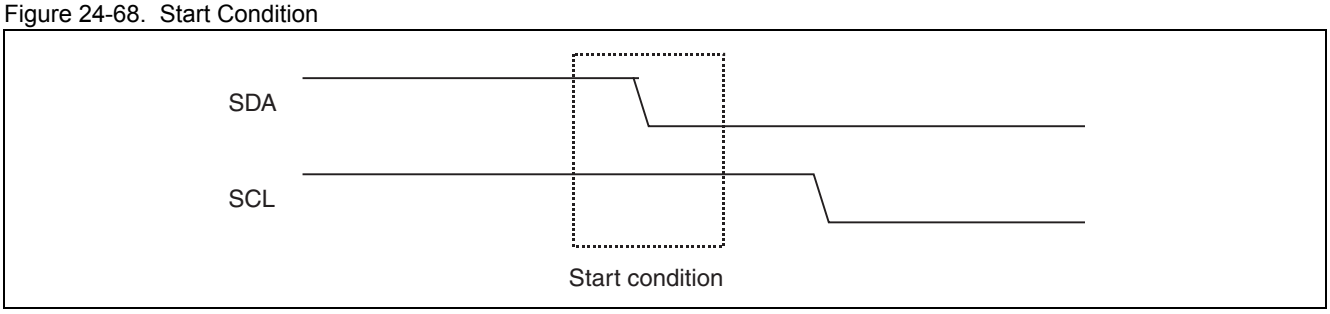
24.22.1 Operation of I²C Interface Communication

The I²C interface communication uses 2 two-way bus lines, a serial data line (SDA) and a serial clock line (SCL).

Start condition for I²C bus

The start condition for the I²C bus is shown below.

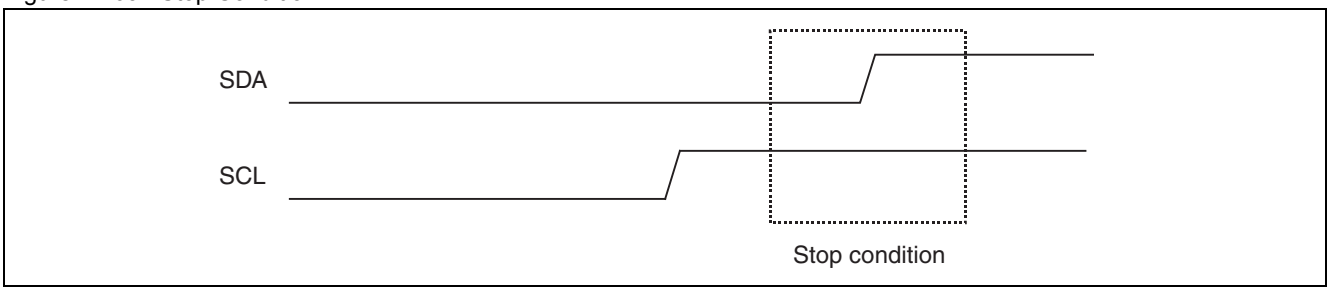
Figure 24-68. Start Condition



Stop condition for I²C bus

The stop condition for the I²C bus is shown below.

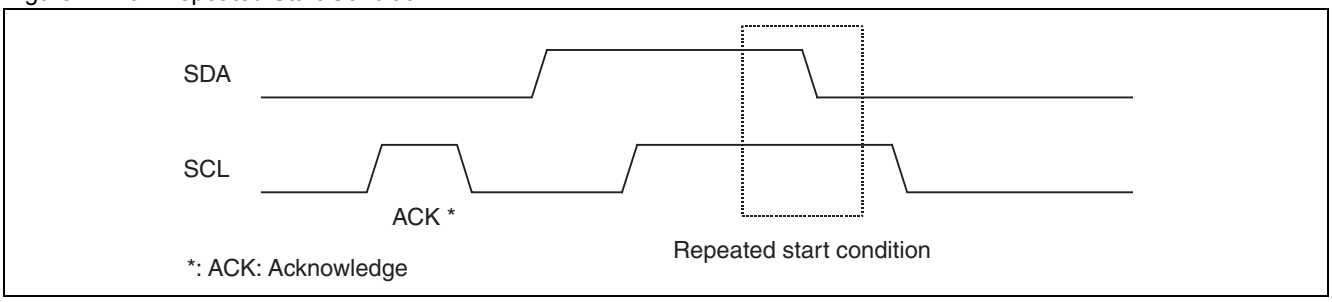
Figure 24-69. Stop Condition



Repeated start condition for I²C bus

The repeated start condition for the I²C bus is shown below.

Figure 24-70. Repeated Start Condition



24.22.2 Master Mode

Master mode generates a start condition for the I^2C bus and outputs a clock to the I^2C bus. Master mode will be selected and the ACT bit in the IBCR register will be set to "1", if the MSS bit in the IBCR register is set to "1" when the I^2C bus is in an idle state (SCL = "H", SDA = "H").

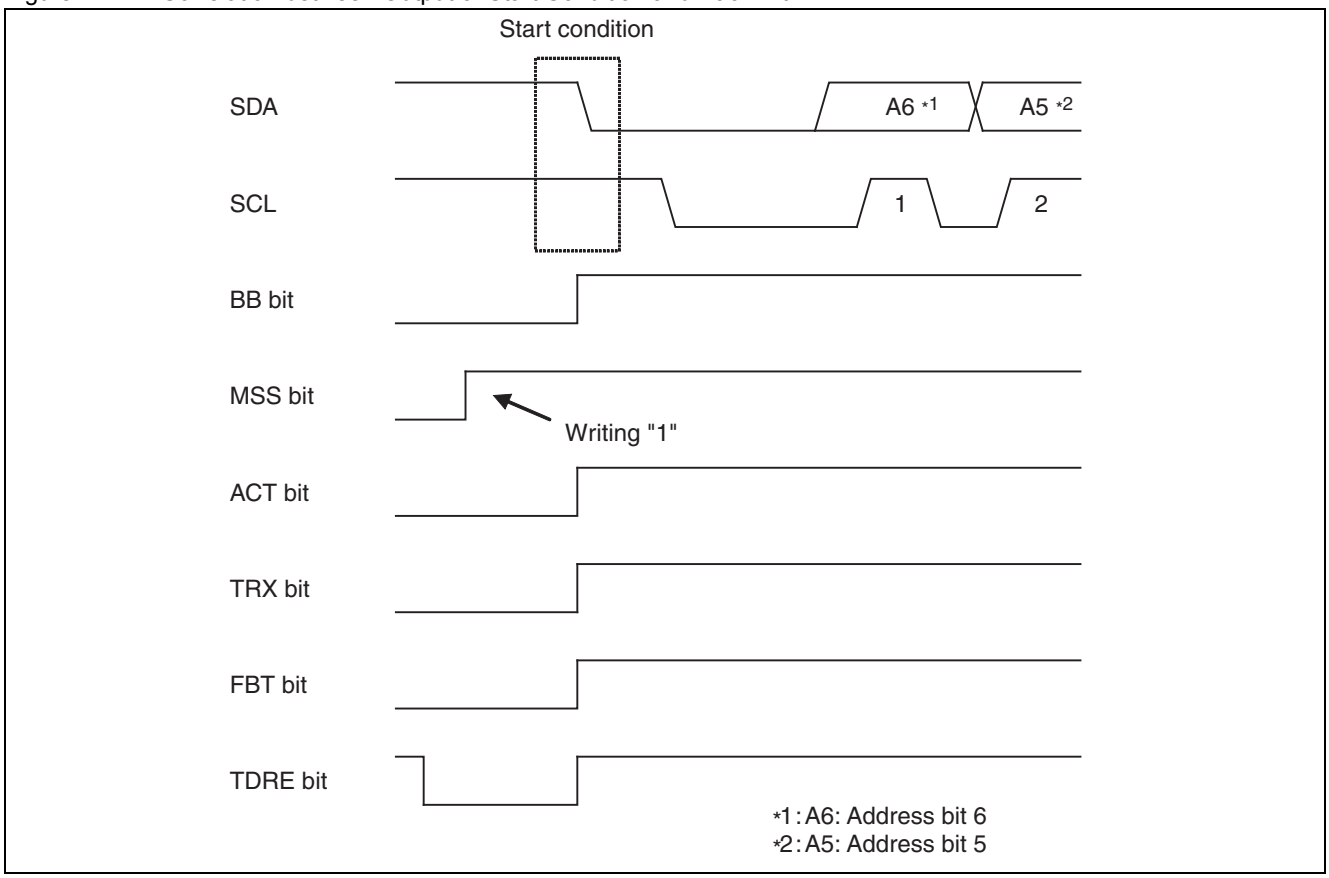
Generating a start condition

A start condition is output under the following conditions.

- "1" is written to the MSS bit when SDA = "H", SCL = "H", EN = 1, and BB = 0.

Outputting a start condition to the I^2C bus sets the ACT bit to "1". After that, the BB bit is set to "1", indicating that the I^2C bus is in the middle of communication, once the start condition is received (see Figure 24-71).

Figure 24-71. Correlation between Output of Start Condition and Each Bit



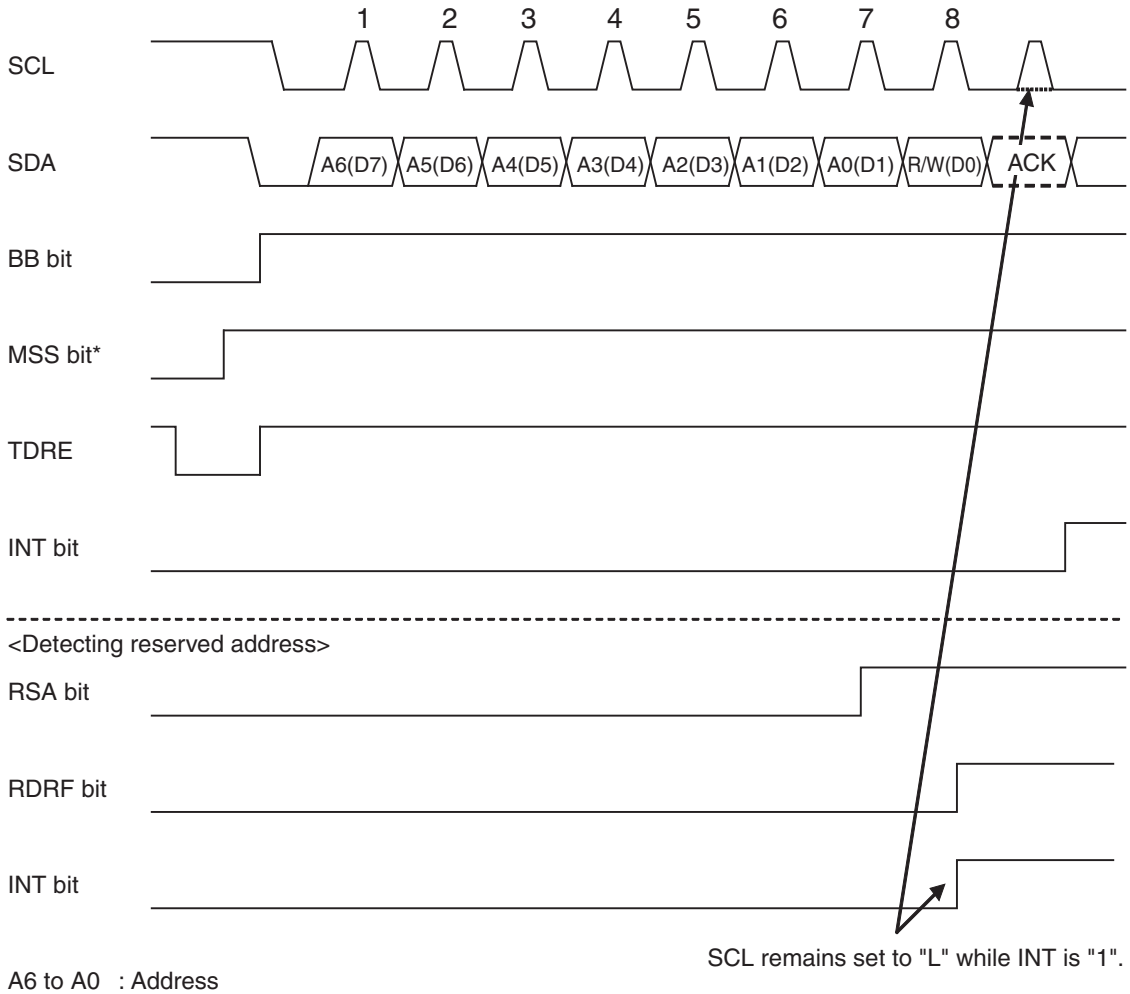
Note: Use the peripheral clock (PCLK) at 8 MHz or higher in operation mode 4 (I^2C mode). It is prohibited to set the baud rate generator to higher than 400kbps.

Outputting a slave address

When a start condition is output, the data set in the TDR register is output from bit7 as an address. When FIFO has been enabled, the first data written in the TDR register is output. Bit0 is used as the data direction bit (R/W), and the data indicates the write direction (master → slave) when the data direction bit (R/W) is set to "0". Set an address to the TDR register before writing "1" to MSS or SCC.

Figure 24-72 and Figure 24-73 show the address and data direction output timings.

Figure 24-72. Address and Data Direction (When FIFO is Disabled)



A6 to A0 : Address

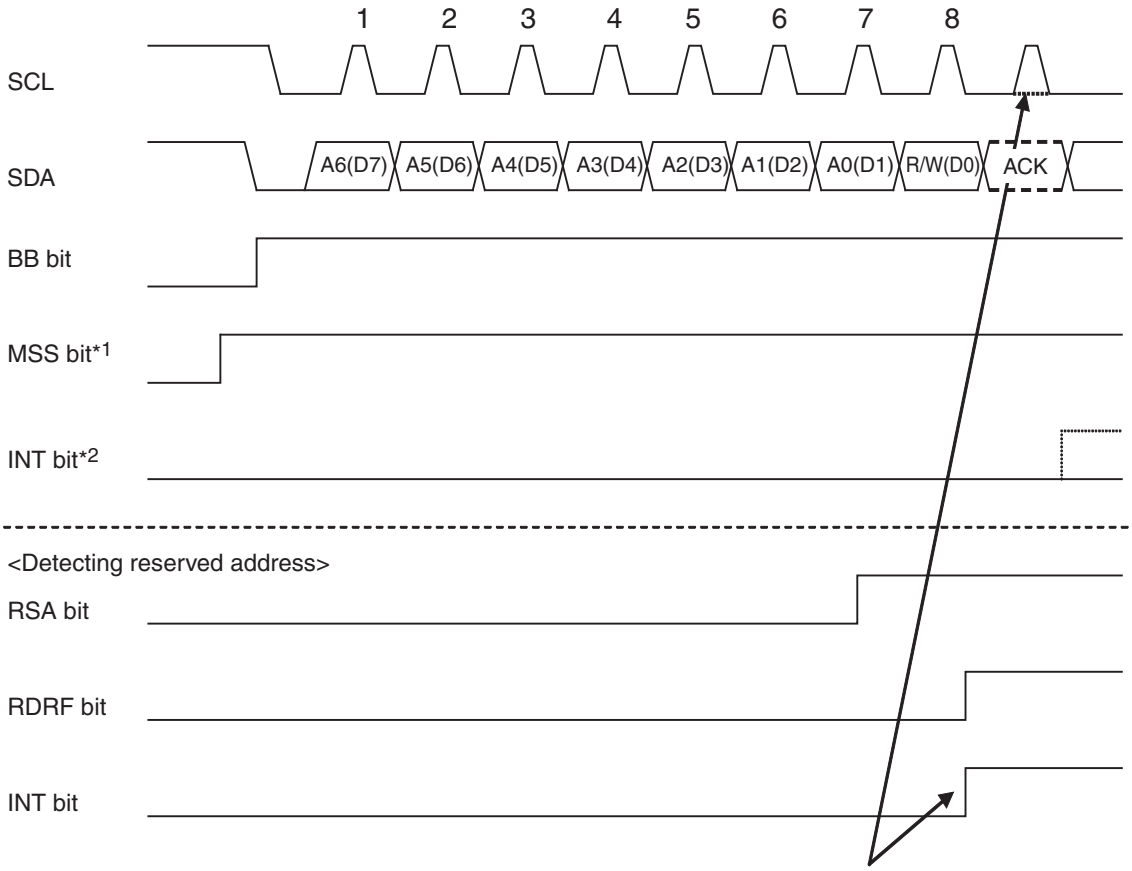
D7 to D0 : TDR register bit

R/W : Data direction ("L", write direction)

ACK : Acknowledge ("L", acknowledge, output from slave)

*: Set the address to the TDR register before writing "1" to the MSS bit.

Figure 24-73. Address and Data Direction (When Transmission/Reception FIFO is Enabled)



A6 to A0 : Address

D7 to D0 : TDR register bit

R/W : Data direction ("L", write direction)

ACK : Acknowledge ("L", acknowledge, output from slave)

*1: Set the address to the TDR register before writing "1" to the MSS bit.

*2: The INT bit will not be set to "1" if the transmission FIFO contains data when acknowledge is "L" and R/W is "L" or if the reception FIFO contains no data when acknowledge is "L" and R/W is "H".

Receiving acknowledge after transmitting 1st byte

The I^2C interface receives an acknowledge from the slave when the data direction bit (R/W) is output. The following operations are performed when FIFO is enabled and disabled.

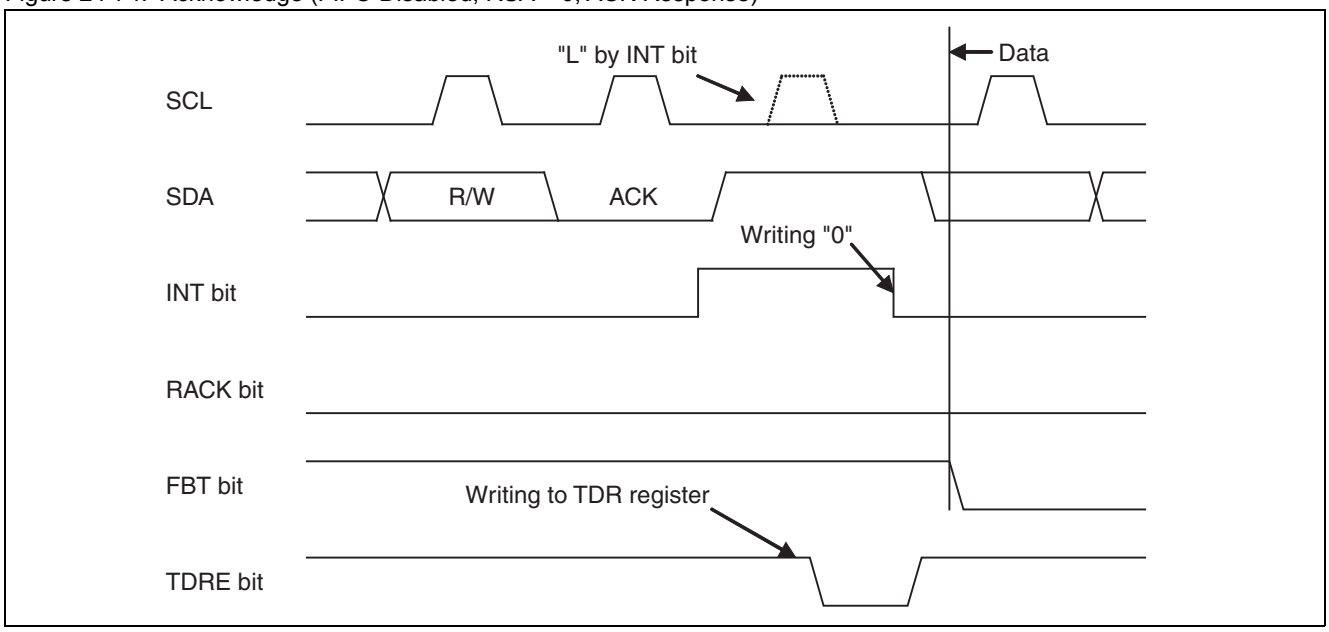
Table 24-49. Operations after Reception of Acknowledge (RSA Bit = 0)

Transmission FIFO	Reception FIFO	Transmission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after reception of acknowledge	
					Acknowledge = ACK	Acknowledge = NACK
Disabled	Disabled	-	-	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
				1		
Disabled	Enabled	-	No data contained	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
			Data contained		The INT bit is set to "1", causing a wait	
			-	1	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	
Enabled	Disabled	-	-	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
				1		
Enabled	Enabled	-	No data contained	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
			Data contained		The INT bit is set to "1", causing a wait	
			-	1	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	

- FIFO disabled (both transmission FIFO and reception FIFO disabled)
  - The interrupt flag (INT) will be set to "1", causing a wait while maintaining SCL at "L", if the TDRE bit is set to "1" after the reception of an acknowledge when the RSA bit is set to "0". The wait is cancelled when "0" is written to the interrupt flag to set it to "0". If the TDRE bit has been set to "0", a clock will be generated to SCL without setting the interrupt flag to "1" when an ACK is received.
  - When the RSA bit is set to "1", the interrupt flag (INT) is set to "1", causing a wait while maintaining SCL at "L", after a reserved address is received (before acknowledge). The interrupt flag will be set to "0" to cancel the wait, if the ACKE bit and transmission data are set and "0" is written to the interrupt flag after the RDR register has been read.
  - The received acknowledge is set to the RACK bit. If NACK is identified when the RACK bit is checked during the wait, a stop condition or repeated start condition will be generated by writing "0" to the MSS bit or writing "1" to the SCC bit. In this case, the INT bit will be cleared to "0" automatically.

- FIFO enabled
  - The following FIFO settings must be performed before the MSS bit is set to "1".
    1. For transmission to the slave (data direction bit = 0), data including a slave address should be set to the transmission FIFO.
    2. For reception of data from the slave (data direction bit = 1), the number of receptions should be set to the FIFO byte number register to write to the transmission data register using dummy data for the number of data elements to be received for the slave address and data direction bit.
  - When the RSA bit is set to "0", the interrupt flag (INT) will not be set to "1" and data will be transmitted/received according to the data direction bit, if the received acknowledge is an ACK (no wait). If the acknowledge is a NACK, the interrupt flag (INT) will be set to "1", causing a wait while maintaining SCL at "L".
  - The received acknowledge is stored in the RACK bit. If the acknowledge is a NACK when the RACK bit is checked during the wait, a stop condition or a repeated start condition will be generated by writing "0" to the MSS bit or writing "1" to the SCC bit. In this case, the INT bit will be cleared to "0" automatically.

Figure 24-74. Acknowledge (FIFO Disabled, RSA = 0, ACK Response)



Address wait timings:

- RSA = 0: after receiving acknowledge
- RSA = 1: before receiving acknowledge

The above timings are not dependent on the WSEL setting.

Figure 24-75. Acknowledge (FIFO Disabled, RSA = 0, NACK Response)

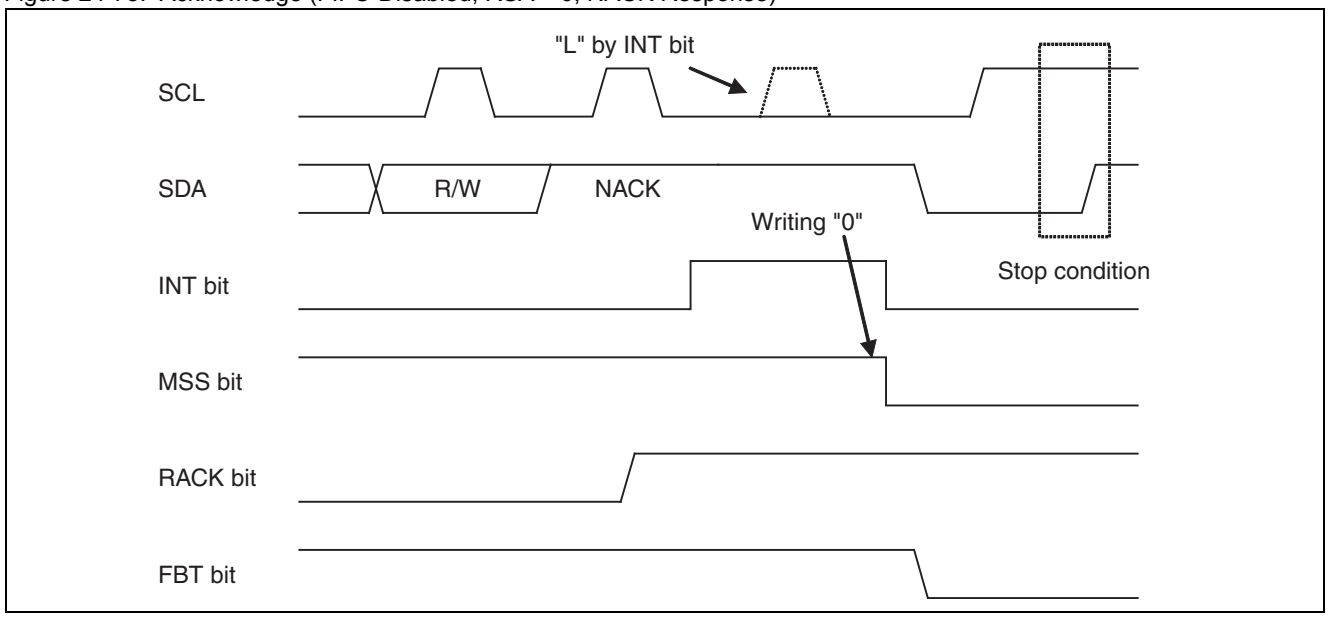


Figure 24-76. Acknowledge (FIFO Disabled, RSA = 1, ACK Response)

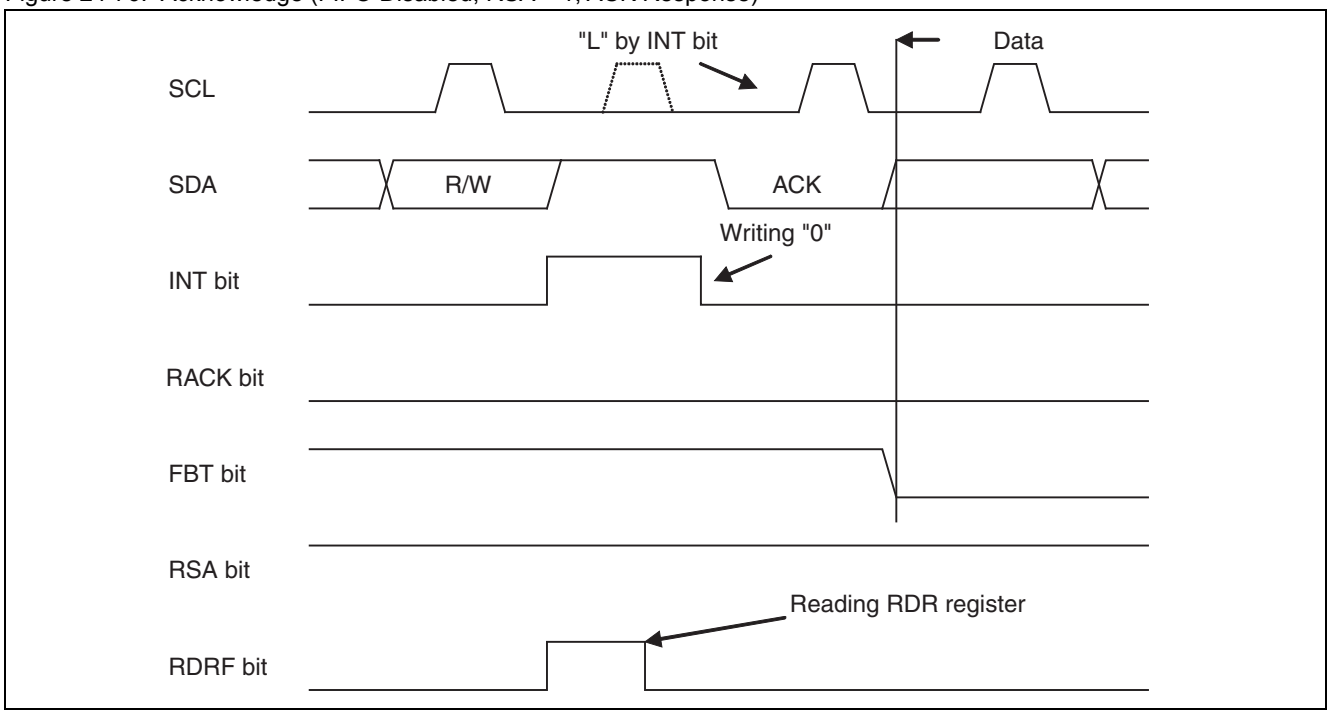


Figure 24-77. Acknowledge (FIFO Disabled, RSA = 1, NACK Response)

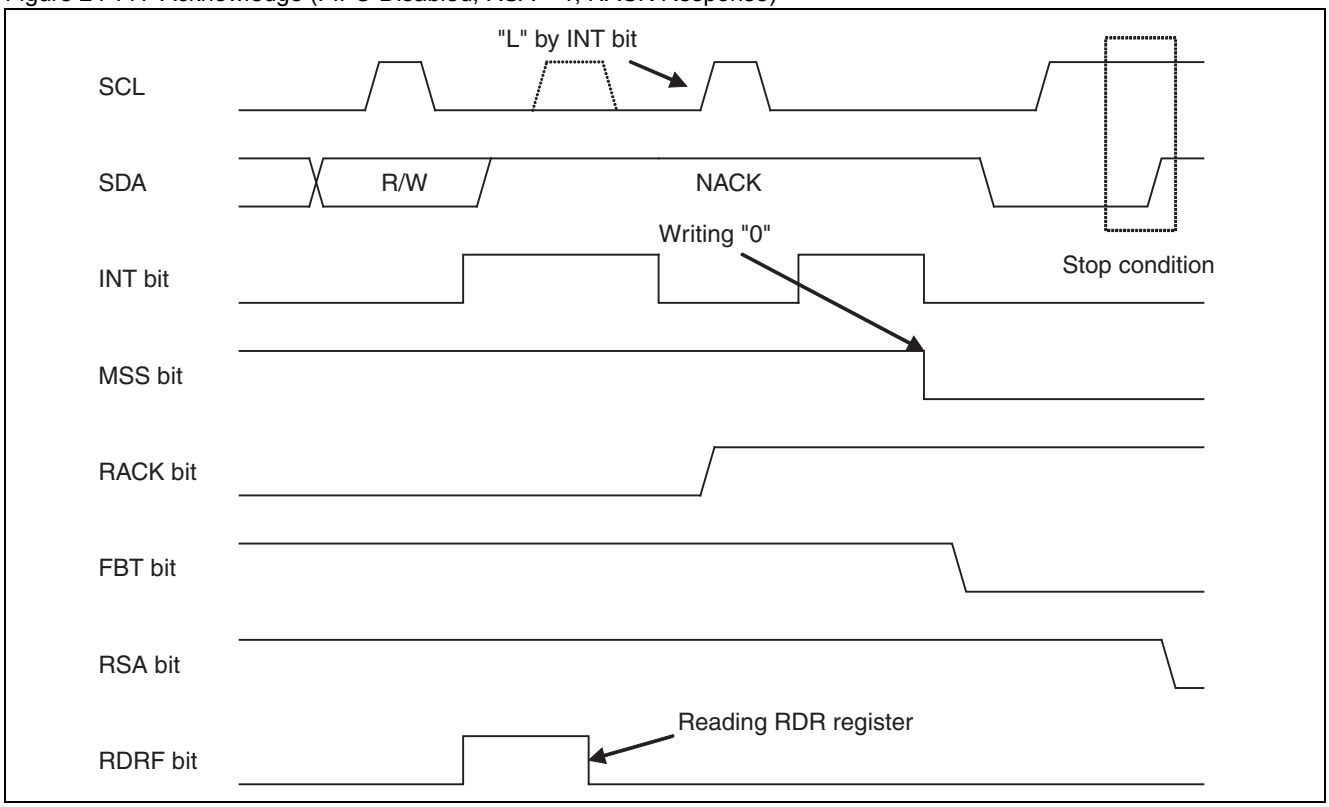
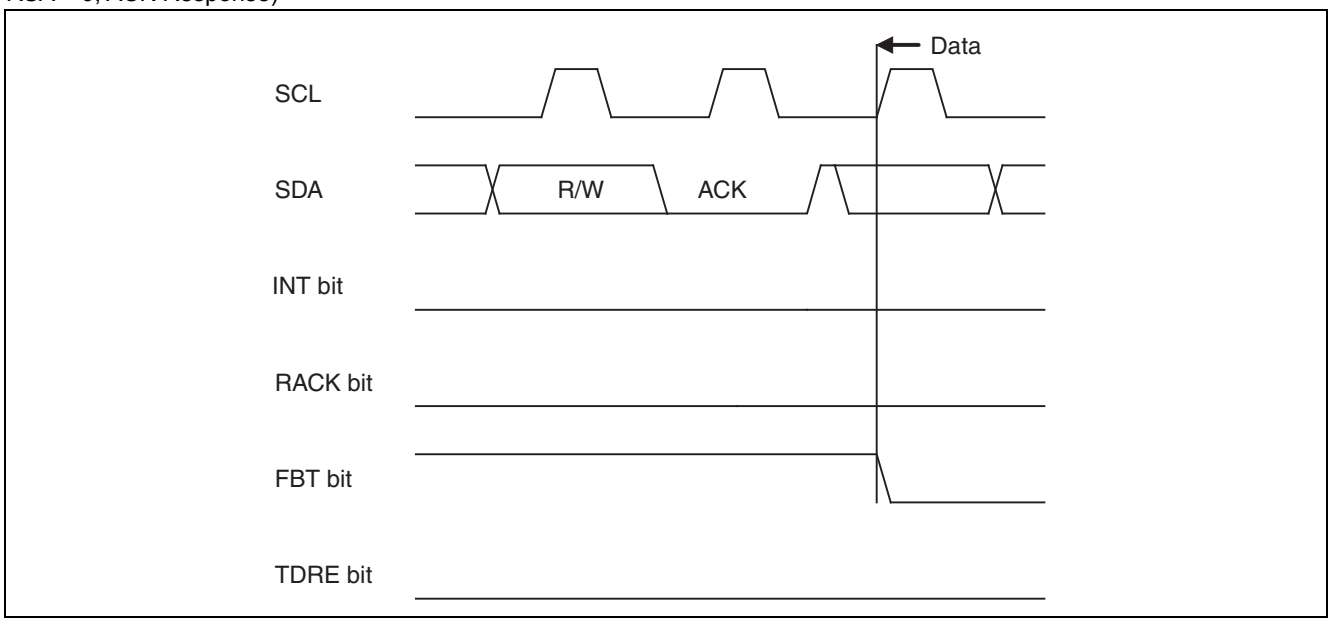


Figure 24-78. Acknowledge (FIFO Enabled, Transmission FIFO Containing Data, Reception FIFO Containing No Data, RSA = 0, ACK Response)



Master data transmission

Data is transmitted from the master when the data direction bit (R/W) is set to "0". The slave returns an ACK or NACK response for each byte transmitted.

The location in which a wait occurs is as follows, depending on the WSEL bit setting.

Table 24-50. WSEL Bit During Master Data Transmission

WSEL bit	Operation
0	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after an acknowledge by setting the TDRE bit to "1" or detecting an arbitration lost condition. If FIFO is enabled, a wait will be generated when the interrupt flag (INT) is set to "1" after an acknowledge by detecting an arbitration lost condition, or when the transmission data register no longer contains valid data (TDRE = 1).
1	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after the master transmits 1-byte data by setting the TDRE bit to "1" or detecting an arbitration lost condition. If FIFO is enabled, a wait will be generated when the interrupt flag (INT) is set to "1" after data transmission by detecting an arbitration lost condition, or when the transmission data register no longer contains valid data (TDRE = 1).

However, if a NACK is received at times other than when a stop condition is set (MSS = 0, ACT = 1), the interrupt flag (INT) is set after an acknowledge, regardless of the WSEL setting.

An example procedure for transmitting data to the slave is shown below.

- Transmitting data to any address other than reserved address
  - When transmission FIFO is disabled:
    1. Set the slave address (including the data direction bit) to the TDR register and write "1" to the MSS bit.
    2. An ACK will be received after the slave address is transmitted, and then the interrupt flag (INT) will be set to "1".
    3. Write the data to be transmitted to the TDR register.
    4. Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel a wait for the I^2C bus.
    5. Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (2) to (4) until a specified number of data elements are transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is canceled with WSEL set to "1".
    6. Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.
  - When transmission FIFO is enabled:
    1. Write the slave address (including the data direction bit) and transmission data to the TDR register.
    2. Set the WSEL bit and write "1" to the MSS bit.
    3. If a NACK is received during transmission, set the interrupt flag (INT) to "1" immediately after the reception to put the I^2C bus in a wait. If all receive an ACK response, set the interrupt flag to "1" after the last byte has been transmitted, according to the WSEL setting, to put the I^2C bus in a wait.
    4. Write "0" to the MSS bit to generate a stop condition.

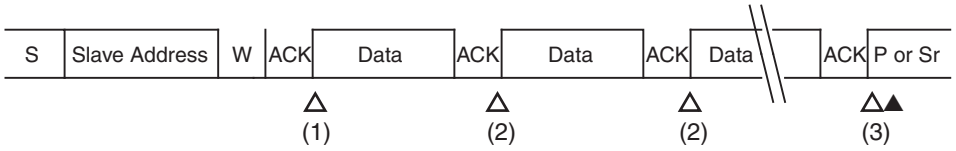
- Transmitting data to reserved address
  - When transmission FIFO is disabled:
    1. Set the reserved address to the TDR register as the slave address and write "1" to the MSS bit.
    2. The interrupt flag (INT) will be set to "1" once the slave address has been transmitted.
    3. Read from the RDR register to check the reserved address.*
    4. Write the data to be transmitted to the TDR register.
    5. Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel the wait for the I^2C bus.
    6. Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (4) to (6) until a specified number of data elements are transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is canceled with WSEL set to "1".
    7. Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.
  - When transmission FIFO is enabled:
    1. Set the reserved address to the TDR register as the slave address and write "1" to the MSS bit.
    2. The interrupt flag (INT) will be set to "1" once the slave address has been transmitted.
    3. Read from the RDR register to check the reserved address.[1]
    4. Write all the data to be transmitted (in case that the transmission FIFO becomes full, write as much until reaching that state) to the TDR register.
    5. If a NACK is received during transmission, set the interrupt flag (INT) to "1" immediately after the reception to put the I^2C bus in a wait. If all receive an ACK response, set the interrupt flag to "1" after the last byte has been transmitted, according to the WSEL setting, to put the I^2C bus in a wait.
    6. Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

    [1]: When the reserved address is a general call address in multi-master operation, it is necessary to confirm whether the device will operate as the master or slave for the next data by setting the ACKE and WSEL bits to "1", if the device may operate as the slave due to the generation of an arbitration lost condition.

Notes:

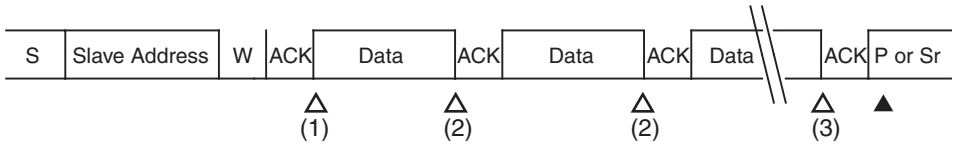
- To modify the IBCR register during transmission or reception, modify it when the interrupt flag (INT) is set to "1".
- When the WSEL bit has been modified, this will be used as a condition for generating the interrupt flag (INT) for the next data.
- If transmission data is written to the TDR register and an ACK response is detected when the TDRE is set to "1" during data transmission, the written data will be transmitted without setting the interrupt flag (INT) to "1".
- If transmission data is written to the TDR register and an ACK is returned when the TDRE is set to "1" during data reception, only RDRF will be set to "1" without setting the interrupt flag (INT) to "1" (when the reception FIFO is enabled, and the amount set in the FBYTE1/FBYTE2 register is received).

Figure 24-79. Master Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)



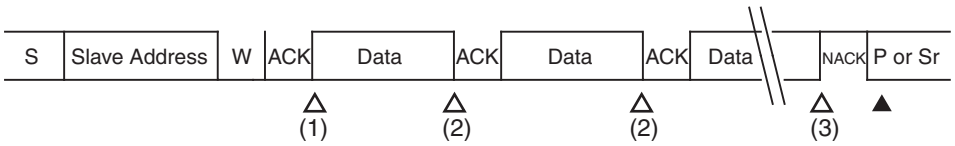
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte + reception of acknowledge
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-80. Master Transmission Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0, ACK Response)



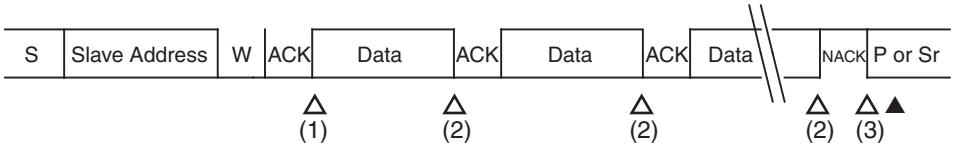
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-81. Master Transmission Interrupt (3) - when FIFO is Disabled (WSEL = 1, RSA = 0, NACK Response)



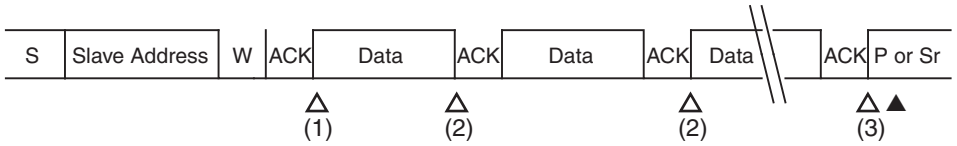
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-82. Master Transmission Interrupt (4) - when FIFO is Disabled (WSEL = 1, RSA = 0, NACK Response in the Middle of Operation)



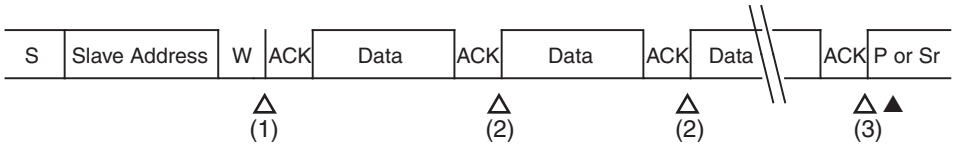
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by NACK response
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-83. Master Transmission Interrupt (5) - when FIFO is Disabled (WSEL = 1 -> 0, RSA = 0, ACK Response)



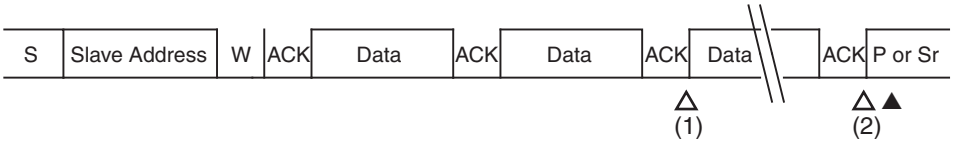
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to the transmission buffer
(2) Interrupt generated by transmission of 1 byte
Writing "0" to WSEL and INT after writing transmission data to the transmission buffer
(3) Interrupt generated by transmission of 1 byte
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-84. Master Interrupt (6) - when FIFO is Disabled (WSEL = 0, RSA = 1)



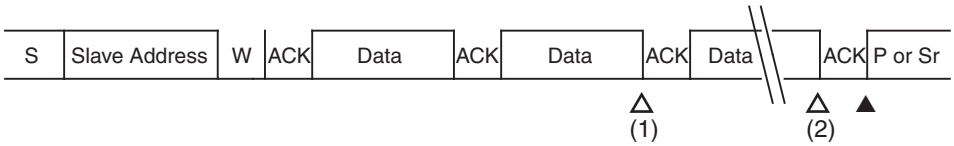
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address (reserved address) + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte + reception of acknowledge
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-85. Master Transmission Interrupt (7) - when FIFO is Enabled (WSEL = 0, RSA = 0, ACK Response)



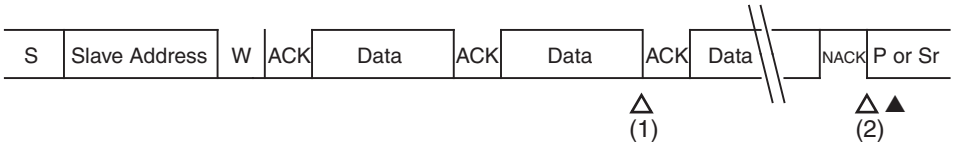
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
- Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by transmission of last byte (transmission FIFO being empty)
+ reception of acknowledge
Setting MSS=0 or MSS=1, and SCC=1

Figure 24-86. Master Transmission Interrupt (8) - when FIFO is Enabled (WSEL = 1, RSA = 0)



S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by transmission of last byte (transmission FIFO being empty)
Setting MSS=0 or MSS=1, and SCC=1

Figure 24-87. Master Transmission Interrupt (9) - when FIFO is Enabled (WSEL = 1, RSA = 0, NACK Response)



S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by NACK response
Setting MSS=0 or MSS=1, and SCC=1

Master data reception

The data transmitted from the slave is received when the data direction bit (R/W) is set to "1".

When FIFO is disabled, the master will generate a wait for reception of each byte if the TDRE bit is set to "1" (INT = 1, RDRF = 1), and an ACK or NACK will be returned by the setting of the ACKE bit in the IBCR register, according to the WSEL bit. When the TDRE bit is set to "0", a wait will not be generated (INT = 0) and the next data will be received if ACK has been selected by the ACKE bit in the IBCR register, or a wait will be generated (INT = 1) if NACK has been selected.

When FIFO is enabled, the RDRF bit will be set if the same number of bytes as a specified number of bytes to be received is received. The interrupt flag is set and puts the I^2C bus in a wait, when the TDRE bit is set to "1". When WSEL is set to "0", setting the TDRE bit to "1" returns a NACK and sets the interrupt flag to "1". When WSEL is set to "1", a wait is generated after the last byte has been received. Therefore, the ACKE bit should be set during that wait to clear the interrupt flag to "0", and then an ACK or NACK should be returned depending on the ACKE setting. Even when a NACK is output, it will be stored as reception data to the reception FIFO.

For interrupt-triggered waits, refer to the following section

Table 24-51. WSEL Bit During Master Data Reception

WSEL bit	Operation
0	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after an acknowledge by setting the TDRE bit to "1".
1	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after the master transmits 1-byte data by setting the TDRE bit to "1".

An example procedure for receiving data from the slave is shown below.

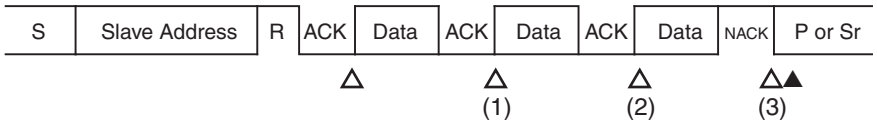
- When reception FIFO is disabled:
  1. Set the slave address (including the data direction bit) to the TDR register and write "1" to the MSS bit.
  2. An ACK will be received after the slave address is transmitted, and then the interrupt flag (INT) will be set to "1".
  3. Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel a wait for the I^2C bus.
  4. Put the I^2C bus in a wait by setting the interrupt flag to "1", after transmitting an acknowledge upon the reception of one byte when WSEL is set to "0", or immediately after one byte has been received when WSEL is set to "1". Repeat (2) to (4) until a specified number of data elements are received.
  5. Output a NACK after the reception of the last data, and set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

- When transmission/reception FIFO is enabled:
  1. Set the number of receptions to the FBYTE1/FBYTE2 register.
  2. Write the slave address (including the data direction bit) and dummy data for the number of receptions to the TDR register.
  3. Write "1" to the MSS bit.
  4. ACK will be returned and reception will continue as long as the TDRE bit is set to "0". RDRF is set to "1" once the amount set in FBYTE1/FBYTE2 is received during that reception. When RDRF is set to "1", the RDR register is read.
  5. When the TDRE bit is set to "1", the interrupt flag will be set to "1" to put the I^2C bus in a wait, after the output of a NACK if WSEL is set to "0", or immediately after the reception of one byte if WSEL is set to "1".
  6. The ACKE bit should be set to "0" when WSEL is set to "1", or the ACKE bit does not have to be set when WSEL is set to "0". Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

Notes:

- During enabling the detection of 7-bit slave address (ISBA:SAEN=1), it is prohibited to specify 7-bit slave address in master mode.
- An acknowledge will be output to handle the next data according to the setting of the ACKE bit, even if an overrun error occurs when TDRE is set to "0".
- Modify the IBCR register during transmission/reception, if necessary, when the interrupt flag (INT) is set to "1".
- In master reception, the next data will be received with the interrupt flag (INT) still set to "0", if the TDRE bit is set to "0" when dummy data is written to the TDR register and the interrupt flag (INT) is set to "1".
- If data is received when the reception FIFO has been enabled and WSEL is set to "0", the RDRF bit will be set to "1" upon the reception of the last bit and the interrupt flag (INT) will be set to "1" after an ACK is transmitted.

Figure 24-88. Master Reception Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)



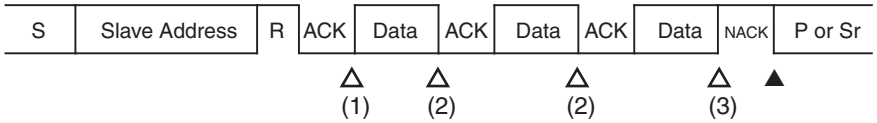
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
- Interrupt cleared to "0" by writing "0" to INT

(2) Interrupt generated by reception of 1 byte + transmission of acknowledge
- Setting ACKE to "0" and writing "0" to INT after reading reception data

(3) Interrupt generated by reception of 1 byte + transmission of acknowledge
- Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-89. Master Reception Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
- Interrupt cleared to "0" by writing "0" to INT
(2) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reading reception data
(3) Interrupt generated by reception of 1 byte
- Setting ACKE=0, and then setting MSS=0 or MSS=1, and SCC=1 after reading reception data
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 24-90. Master Reception Interrupt (3) - when FIFO is Enabled (WSEL = 0, ACKE = 0, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by TDRE=1
- Setting MSS=0 or MSS=1, and SCC=1 after reading all data from reception FIFO

Figure 24-91. Master Reception Interrupt (4) - when FIFO is Enabled (WSEL = 1, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by TDRE=1
- Setting ACKE=0, MSS=0 or MSS=1, and SCC=1 after reading all data from reception FIFO

Arbitration lost condition

When a master receives data which is different from the transmitted data due to a data collision with the data from another master, this is determined as an arbitration lost condition. Consequently, the MSS bit is set to "0" and the AL bit to "1" to allow the device to operate in slave mode.

The AL bit can be cleared to "0" under the following conditions.

- "1" is written to the MSS bit.
- "0" is written to the INT bit.
- "0" is written to the SPC bit when the AL and SPC bits are set to "1".
- The I^2C interface is disabled (EN bit = 0).

When an arbitration lost condition occurs, the interrupt flag (INT) is set to "1" and the SCL of the I^2C bus is set to "L", according to the setting of WSEL.

Wait in master mode

If the device is not operating in slave mode when the MSS bit is set to "1" with the BB bit set to "1", the master mode will be put in a wait as long as the BB bit remains set to "1". It will transmit a start condition once the BB bit becomes "0". The MSS and ACT bits can be used to determine whether the master mode is in a wait or not (MSS = 1, ACT = 0: wait state). To allow the device to operate in slave mode after the MSS bit is set to "1", set AL = 1, MSS = 0, and ACT = 1.

24.22.3 Slave Mode

In slave mode, the device detects a (repeated) start condition and returns an ACK when the combination of the ISBA and ISMK registers matches the received address, in order to operate in slave mode.

Slave address match detection

When a (repeated) start condition is detected, the next 7-bit data is received as an address. Each bit of the ISBA register is compared with the corresponding bit of the received address for the bits which are set to "1" in the ISMK register. An ACK will be output if there is a match.

Table 24-52. Operation Immediately after Output of Acknowledge for Slave Address

Transmission FIFO	Reception FIFO	Transmission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after acknowledge	
					Acknowledge = ACK	Acknowledge = NACK
Disabled	Disabled	-	-	0	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	The INT bit remains set to "0", causing no wait.
				1		
Disabled	Enabled	-	No data contained	0	The INT bit remains set to "0", causing no wait.	The INT bit remains set to "0", causing no wait.
			Data contained		The INT bit is set to "1", causing a wait.	
			-	1	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	
Enabled	Disabled	-	-	0	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	The INT bit remains set to "0", causing no wait.
				1		
Enabled	Enabled	-	No data contained	0	The INT bit remains set to "0", causing no wait.	The INT bit remains set to "0", causing no wait.
			Data contained		The INT bit is set to "1", causing a wait.	
			-	1	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	

- Reserved address detection

  When the first byte matches a reserved address ("$0000XXXX_B$" or "$1111XXXX_B$"), the INT bit is set to "1" to put the I^2C bus in a wait upon the reception of data from the 8th bit, whether or not the transmission/reception FIFO is enabled. At this point, ACKE is set to "1" and the INT bit is cleared when allowing the device to operate as a slave. The device will then start slave operation. When ACKE is set to "0", the device does not operate as a slave after the output of an acknowledge.

Data direction bit

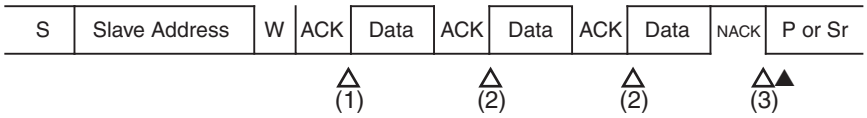
The data direction bit, which determines data transmission or reception, is received after an address is received. When this bit is set to"0", this indicates transmission from the master, therefore, as a slave, the device will receive data.

Slave reception

Reception is performed in slave mode when there is a slave address match and the data direction bit is set to "0". An example procedure for reception in slave mode is shown below.

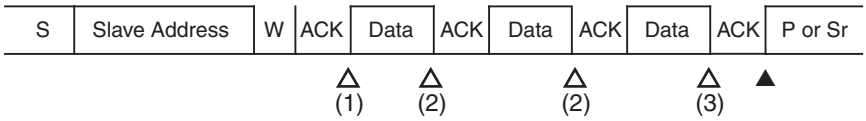
- When reception FIFO is disabled:
  1. Set the interrupt flag (INT) to "1" to put the I^2C bus in a wait after an ACK is transmitted. When the MSS, ACT and FBT bits determine that the interrupt is caused by a slave address match, set the ACKE bit to "1" and write "0" to the interrupt flag (INT) to cancel the I^2C bus wait. (Refer to Table 24-52.)
  2. After 1-byte data is received, set the interrupt flag (INT) to "1" according to the WSEL setting to put the I^2C bus in a wait.
  3. Read the data received from the RDR register, set the ACKE bit and then write "0" to the interrupt flag (INT) to cancel the I^2C bus wait.
  4. Repeat (2) and (3) until a stop condition or a repeated start condition is detected.
- When reception FIFO is enabled:
  1. The interrupt flag (INT) is set to "1" to put the I^2C bus in a wait when a NACK is detected or the reception FIFO becomes full. When a stop condition or a repeated start condition is detected, the SPC and RSC bits are set to "1" but not the interrupt flag (INT) (no I^2C bus wait). The reception FIFO sets the RDRF bit to "1" when the value set in the FBYTE1/FBYTE2 register matches the number of data elements received. At this point, a reception interrupt will occur if the RIE bit has been set to "1".
  2. When the interrupt flag (INT) is set to "1", read the data received from the RDR register. After reading all the data, write "0" to the interrupt flag to cancel the I^2C bus wait. Read all the data received from the RDR register and clear the SPC or RSC bit to "0", if a stop condition or a repeated start condition is detected.

Figure 24-92. Slave Reception Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)



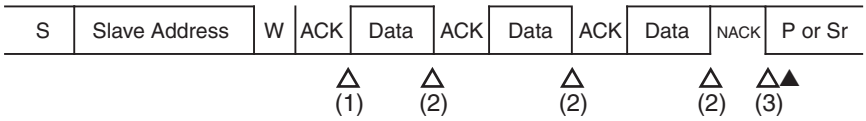
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by ACK output due to slave address match
- Writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte + ACK response
- Writing "0" to INT after reception data is read from reception buffer
(3) Interrupt generated by reception of 1 byte + NACK response
- Writing "0" to INT after reception data is read from reception buffer

Figure 24-93. Slave Reception Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by ACK output due to slave address match
- Writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reception data is read from reception buffer
(3) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reception data is read from reception buffer

Figure 24-94. Slave Reception Interrupt (3) - when FIFO is Disabled (WSEL = 1, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by ACK output due to slave address match
- Writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reception data is read from reception buffer
(3) Interrupt generated by NACK response
- Writing "0" to INT

Figure 24-95. Slave Reception Interrupt (4) - when Reception FIFO is Enabled (RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by detection of stop condition or repeated start condition
- Reading all data from reception FIFO

Figure 24-96. Slave Reception Interrupt (5) - when Reception FIFO is Enabled (RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by reception FIFO being full
- Reading all data from reception FIFO and writing "0" to INT

Figure 24-97. Slave Reception Interrupt (6) - when FIFO is Disabled (WSEL = 0, RSA = 1)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by reserved address ("0000XXXX$_B$" or "1111XXXX$_B$") match
- Reading reception data and writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte + output of acknowledge
- Writing "0" to INT
(3) Interrupt generated by reception of 1 byte + output of acknowledge
- Interrupt by writing "0" to INT

Slave transmission

Transmission is performed in slave mode when there is a slave address match and the data direction bit is set to "1". When FIFO is disabled, a wait is generated by setting the interrupt flag (INT) to "1" after transmitting one byte or after returning an acknowledge, depending on the WSEL setting (see Table 24-52.).

The RACK bit can be used to confirm the acknowledge output from the master. It indicates the end of the data reception, determining whether or not the master succeeded in the reception at a time of NACK response. An interrupt will occur to generate a wait if a NACK is detected when WSEL is set to "1".

24.22.4 Bus Error

A case where a stop condition or a (repeated) start condition is detected during data transmission/reception on the I^2C bus is handled as a bus error.

Conditions for the occurrence of bus errors

A bus error sets the BER bit to "1" under the following conditions.

- A (repeated) start condition or a stop condition is detected during the transfer of the first byte.
- A (repeated) start condition or a stop condition is detected in the 2nd bit - 9th (acknowledge) bit of data.

Bus error operation

Check the BER bit when transmission/reception sets the interrupt flag (INT) to "1". If the BER bit is set to "1", the error must be treated. The BER bit is cleared when "0" is written to the INT bit.

Although a bus error sets the INT bit to "1", the I^2C bus does not enter a wait state with SCL set to "L".

24.23 Dedicated Baud Rate Generator

The dedicated baud rate generator sets a serial clock frequency.

Baud rate selection

- Baud rate achieved by dividing the internal clock using the dedicated baud rate generator (reload counter)

  There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

  The reload counter divides the internal clock, according to the set value.

Calculating the baud rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

- Reload value:
  $V = \phi / b - 1$

  V: Reload value b: Baud rate ϕ: Peripheral clock (PCLK) frequency

  Note: The set baud rate may not be generated depending on the SCL rising time of the I^2C bus. In that case, the reload value must be adjusted.

- Example of calculation:
  If the peripheral clock (PCLK) is 16MHz and the baud rate is 400kbps, the reload value will be:

  Reload value:

  $V = (16 \times 1000000)/400000 - 1 = 39$

  As a result, the baud rate is:

  $b = (16 \times 1000000)/(38+2) = 400$ kbps

Notes:

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- Set the baud rate generator registers when the EN bit in the ISMK register is "0".
- Use the peripheral clock (PCLK) at 8 MHz or higher in operation mode 4 (I^2C mode). It is prohibited to set the baud rate generator to higher than 400kbps.
- The reload counter stops when the reload value is set to "0".

Reload values and baud rates for different peripheral clock (PCLK) frequencies

Table 24-53. Reload Values and Baud Rates

Baud rate [bps]	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32MHz
	Reload value	Reload value	Reload value	Reload value	Reload value	Reload value
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

These numerical values are based on the SCL rising time of I^2C bus set to "0". If the rising is slower, the actual baud rates should also be slower than the numerical values above.

Functions of reload counters

Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the internal clock. In addition, the count value of the transmission reload counter can be read from the baud rate generator registers 1, 0 (BGR1, BGR0).

Starting a count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

24.23.1 Example of I²C Flowcharts

Below are some example flowcharts for I²C communication.

I²C master reception/ slave transmission FIFO communication flow

Figure 24-98. Master Reception Main Settings

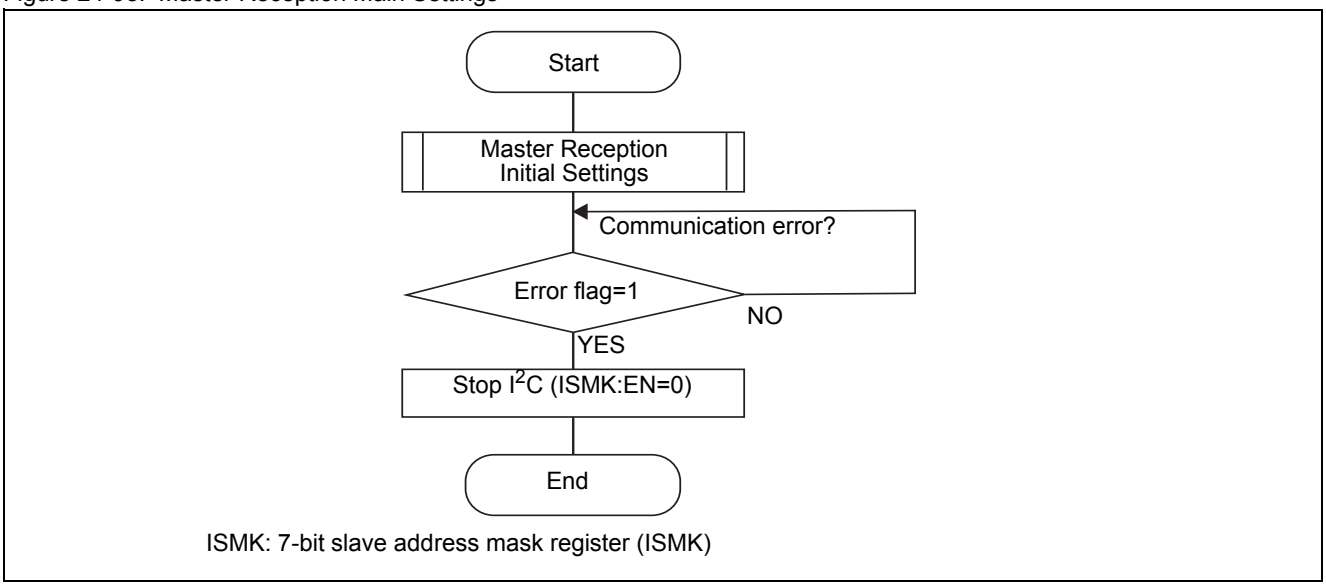
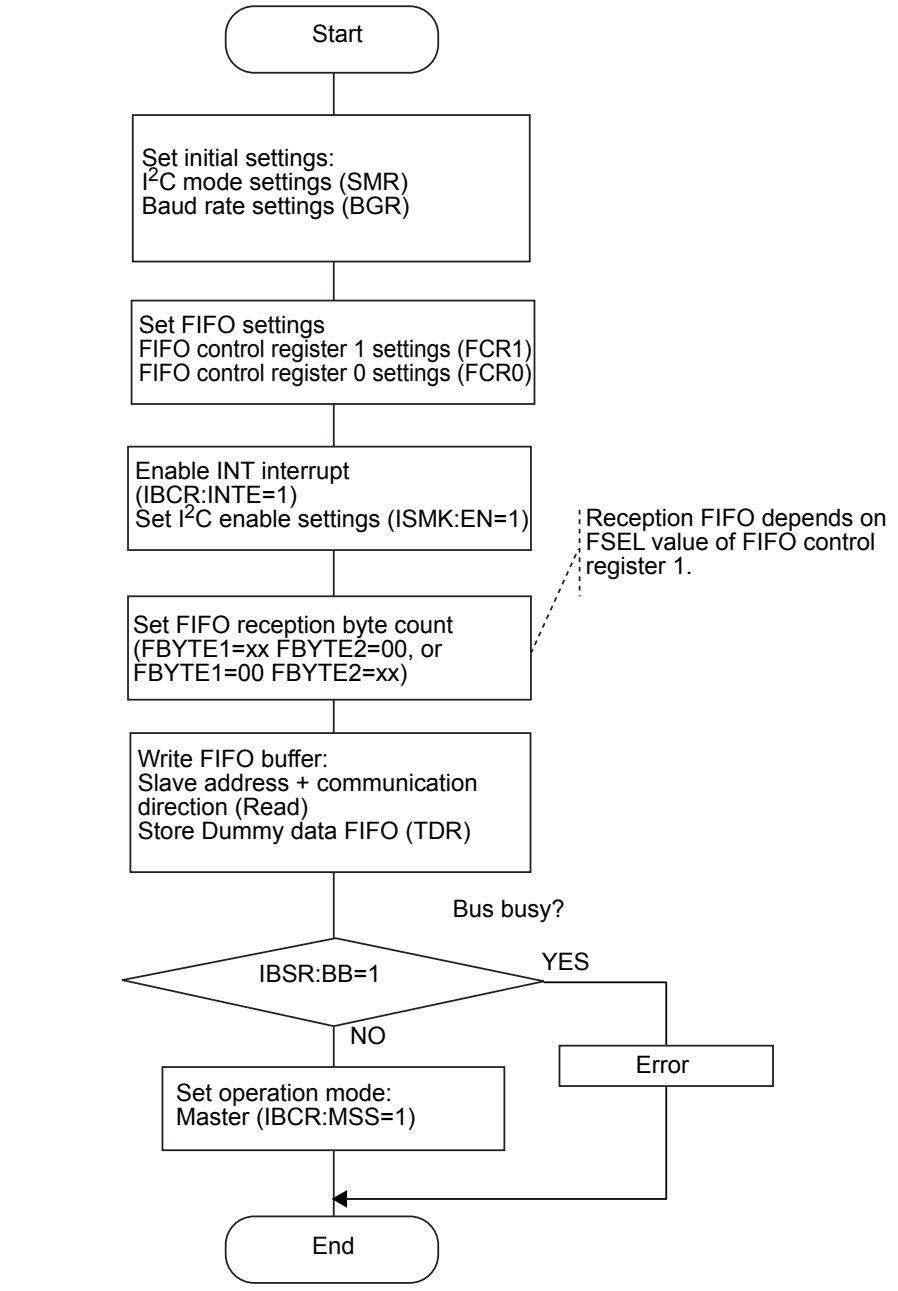
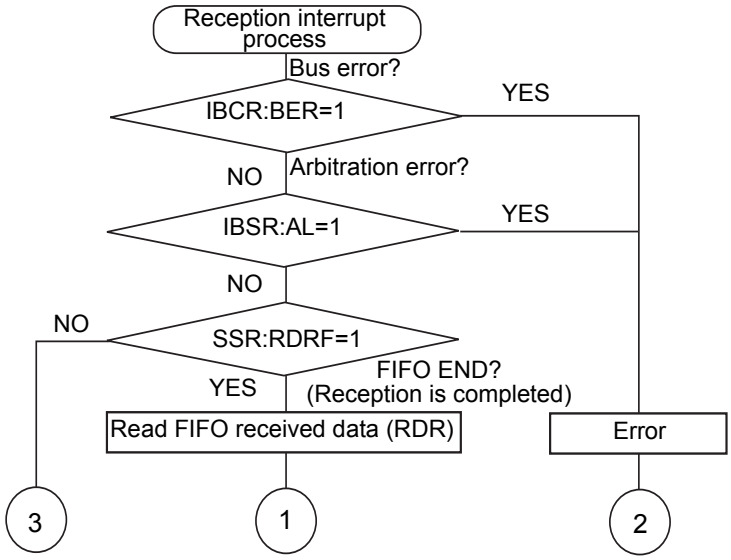


Figure 24-99. Master Reception Initial Settings



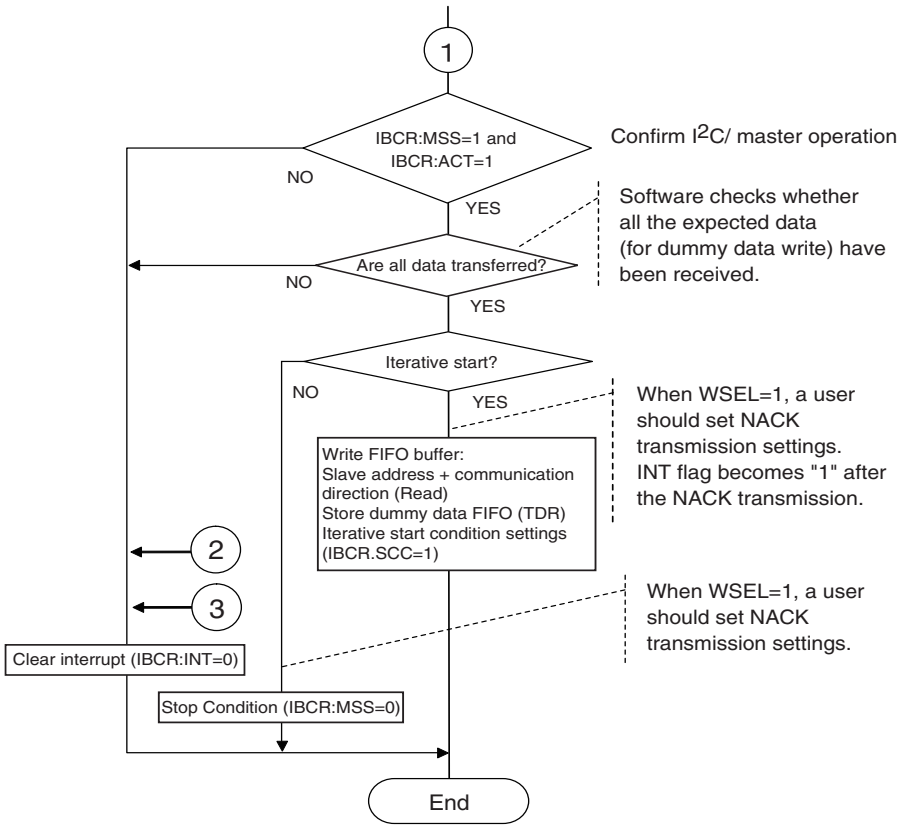
SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
IBCR: I^2C bus control register (IBCR)
IBSR: I^2C bus status register (IBSR)
FBYTE1: FIFO1 byte register (FBYTE1)
FBYTE2: FIFO2 byte register (FBYTE2)
TDR: Transmitted data register (TDR)

Figure 24-100. Master Reception Interrupt Process



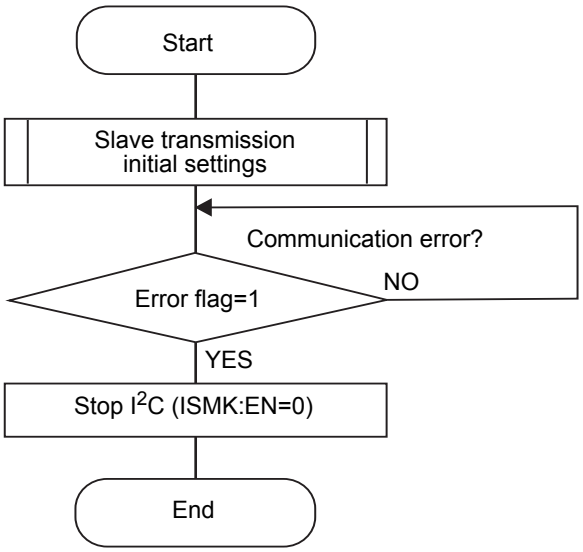
IBCR: I2C bus control register (IBCR)
IBSR: I2C bus status register (IBSR)
SSR: serial status register (SSR)
RDR: received data register (RDR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.
* For final ACK process when receiving FIFO continuously, a wait will be generated after receiving the final data, if setting WSEL=1. Please respond with NACK by setting IBSR.RACK=1 during this wait.



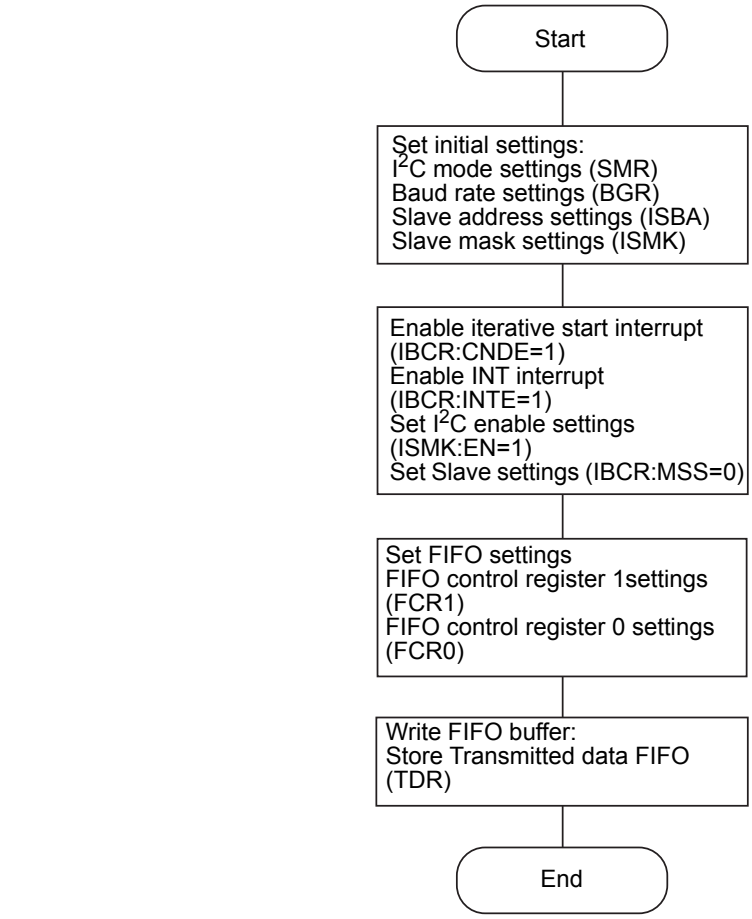
IBCR: I2C bus control register (IBCR)
TDR: Transmitted data register (TDR)

Figure 24-101. Slave Transmission Main Settings



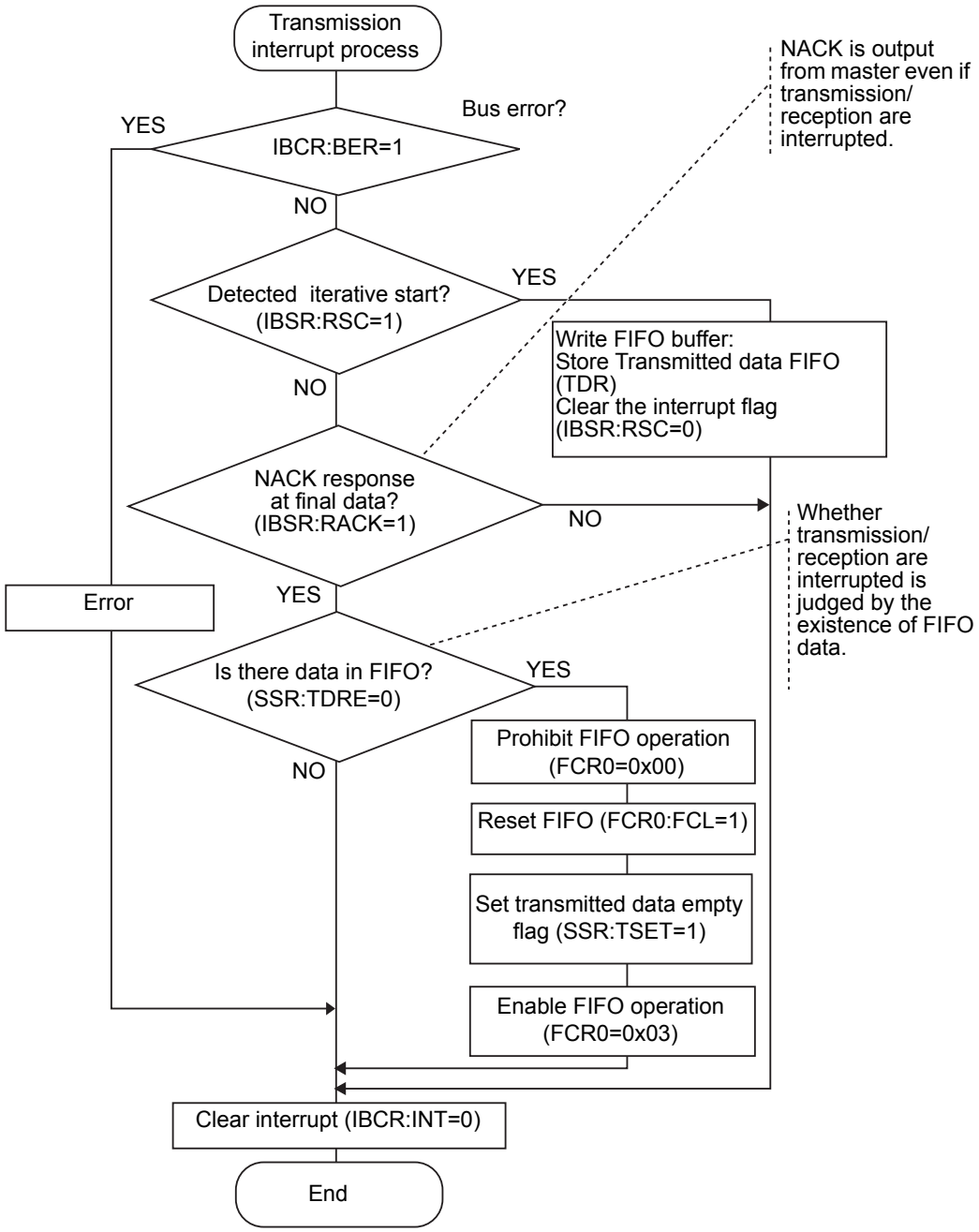
ISMK: 7-bit slave address mask register (ISMK)

Figure 24-102. Slave Transmission Initial Settings



SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
IBCR: I^2C bus control register (IBCR)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
TDR: Transmitted data register (TDR)

Figure 24-103. Slave Transmission Interrupt Process

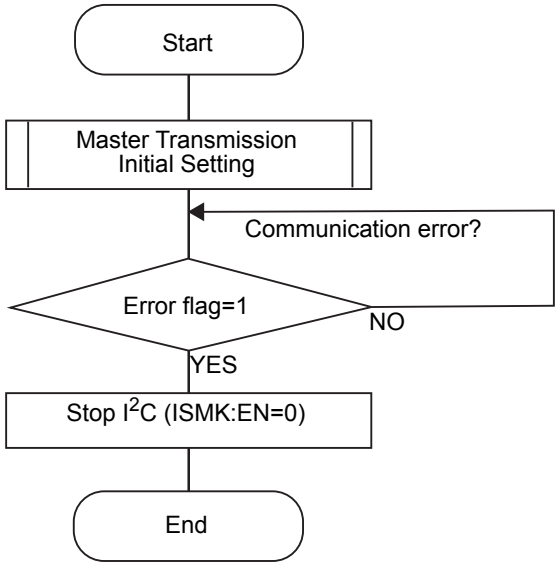


IBSR: I^2C bus status register (IBSR)
TDR: Transmitted data register (TDR)
SSR: Serial status register (SSR)
FCR0: FIFO control register 0 (FCR0)
IBCR: I^2C bus control register (IBCR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

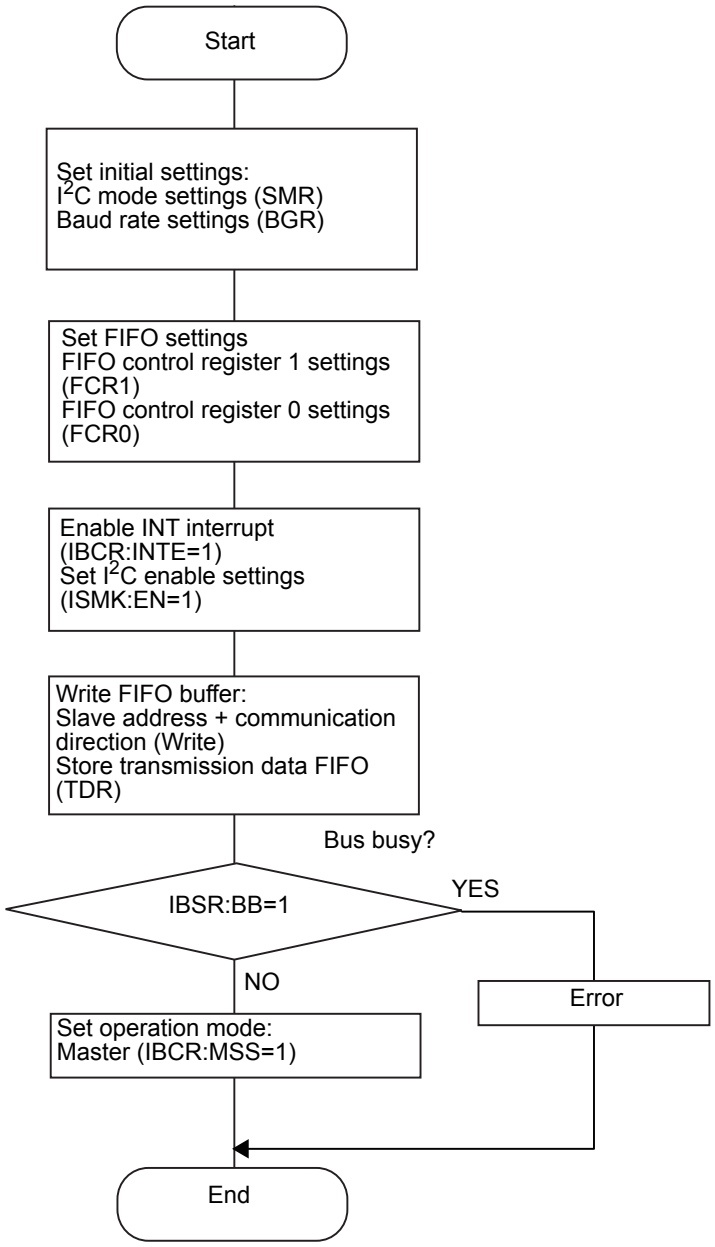
I²C master transmission/ slave reception FIFO communication flow

Figure 24-104. Master Transmission Main Settings



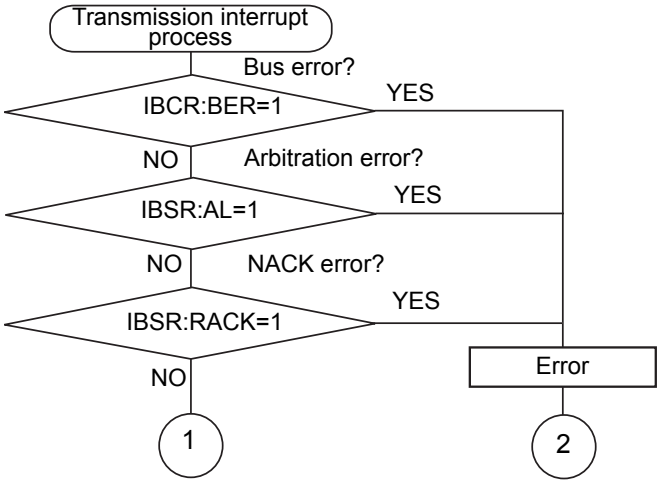
ISMK: 7-bit slave address mask register (ISMK)

Figure 24-105. Master Transmission Initial Settings



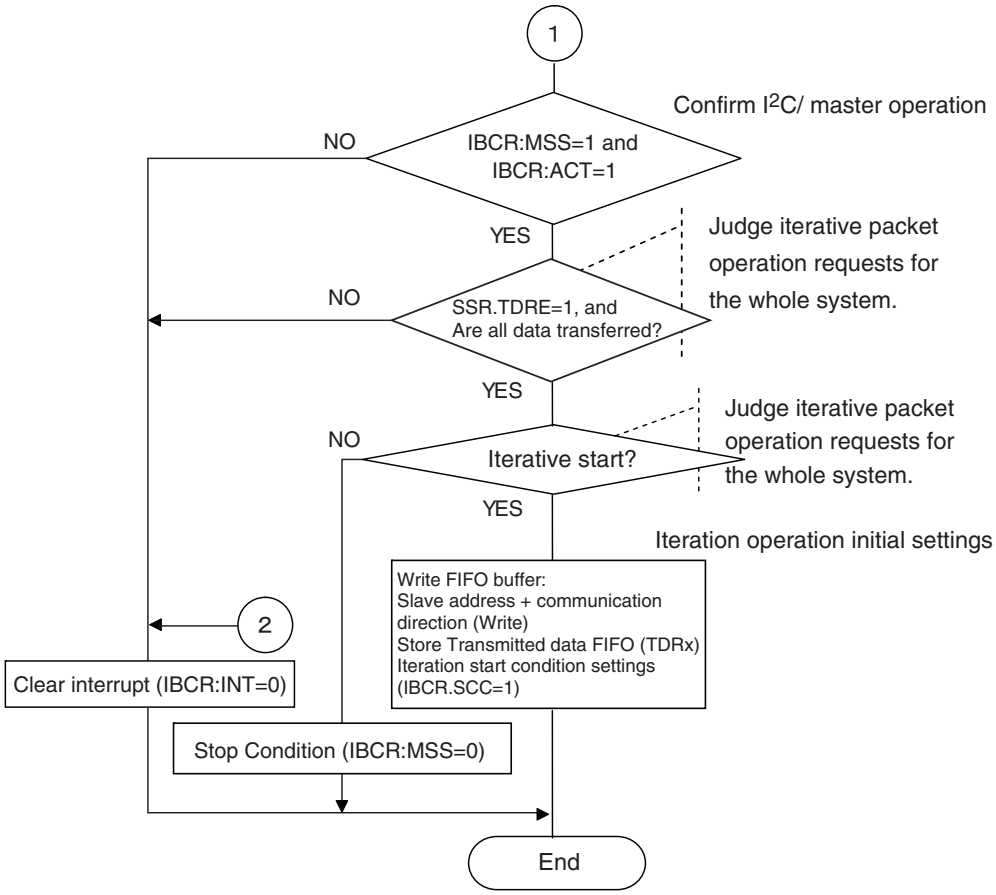
SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
IBCR: I^2C bus control register (IBCR)
IBSR: I^2C bus status register (IBSR)
TDR: Transmitted data register (TDR)

Figure 24-106. Master Transmission Interrupt Process



IBCR: I2C bus control register (IBCR)
IBSR: I2C bus status register (IBSR)

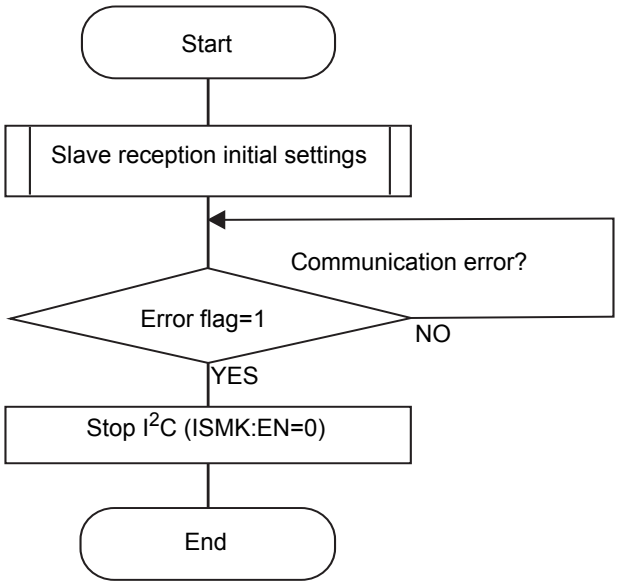
* For actual error handling, please judge each status error flag and handle each error by considering your systems.



IBCR: I2C bus control register (IBCR)
TDR: Transmitted data register (TDR)
SSR: Serial status register (SSR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

Figure 24-107. Slave Reception Main Settings



ISMK: 7-bit slave address mask register (ISMK)

Figure 24-108. Slave Reception Initial Settings

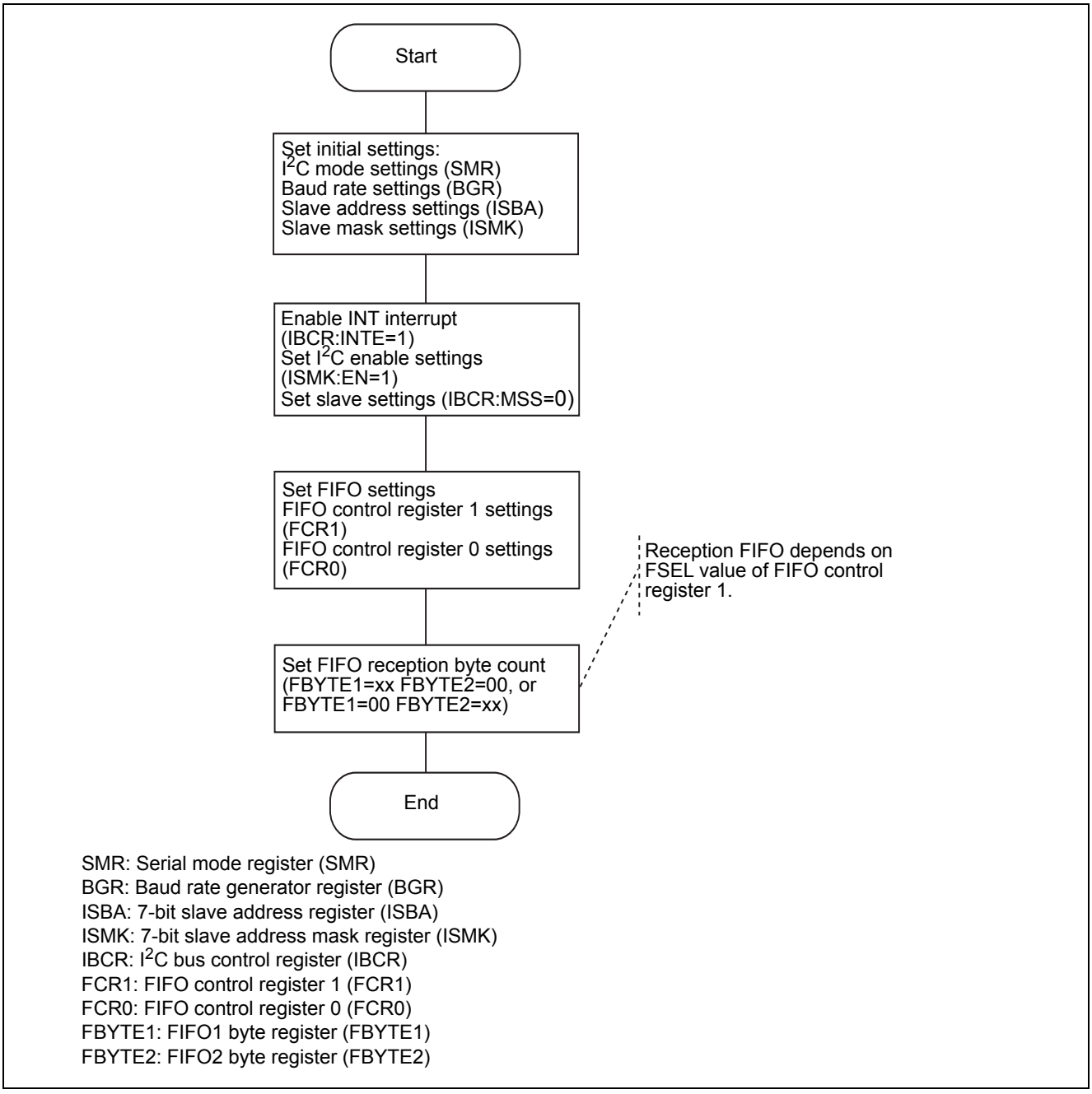
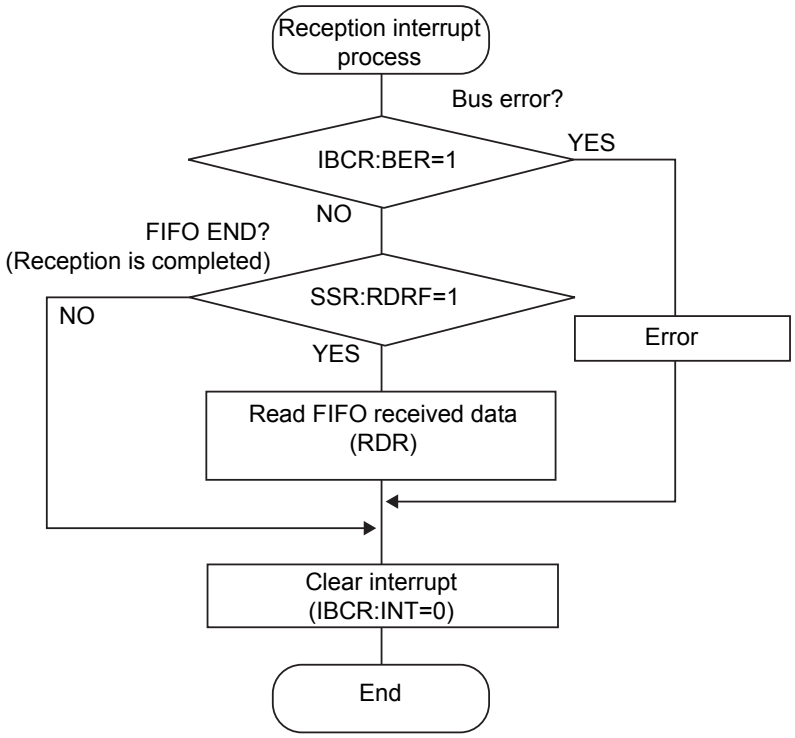


Figure 24-109. Slave Reception Interrupt Process



IBSR: I^2C bus status register (IBSR)
SSR: Serial status register (SSR)
RDR: Received data register (RDR)
IBCR: I^2C bus control register (IBCR)

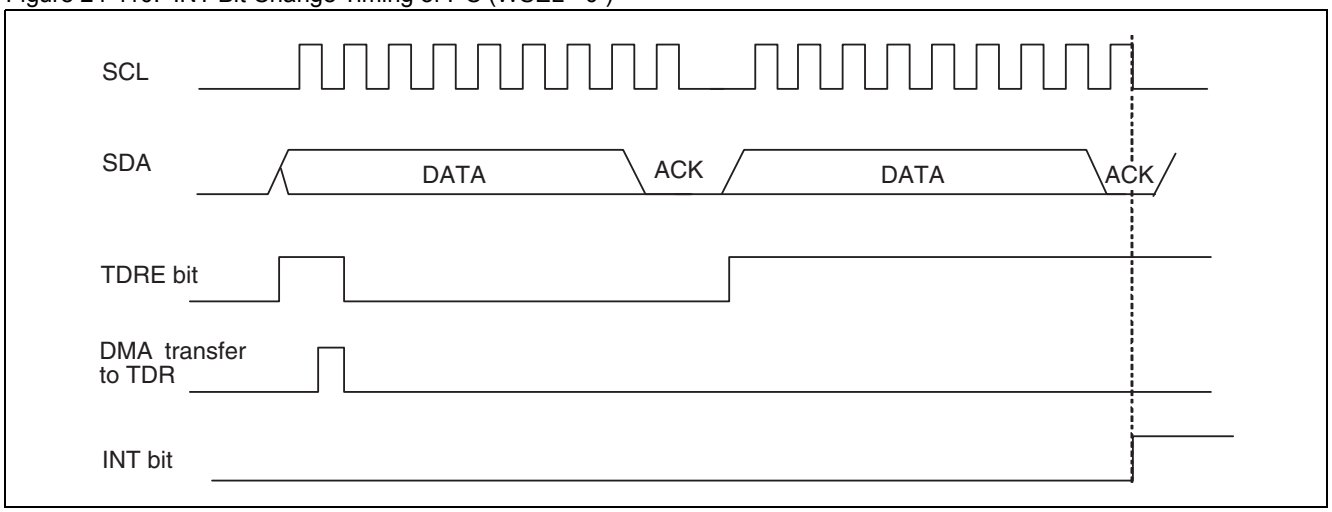
* For actual error handling, please judge each status error flag and handle each error by considering your systems.

24.24 Notes on I²C Mode

The notes for when you use the I²C mode are shown below.

- FIFO cannot be used for requesting DMA transfer with a channel with FIFO. Please set as FIFO operation disable.
- To request a DMA transfer request, set the block size of DMA to one time.
- When master reception and slave reception are selected, it is required to use two channels for DMA; one is used for DMA transfer to receive data and the other one is used for DMA transfer to send dummy data.
- In I²C mode, if there is no valid data in transmission register (TDR), and transmission data empty flag bit (TDRE) is "1", the interrupt flag (INT) becomes "1" as shown in Figure 1 when the data on I²C bus for 9 bits (WSEL=0) or for 8 bits (WSEL=1) is transmitted. When the interrupt flag (INT) becomes "1" during DMA transfer, DMA transfer cannot be continued unless clearing the bit to "0" by software. (Common to master transmission, slave transmission, mater reception, and slave reception.)

Figure 24-110. INT Bit Change Timing of I²C (WSEL= 0)



To perform DMA transfer in I²C mode, since the specification is as shown above, such operations listed below are required for performing DMA transfer to TDR before the interrupt flag (INT) becomes "1". Below operations are possible to perform to prioritize DMA transfer of I²C.

- Use DMA which has a higher priority (channel number is small). It is enabled to use by fixing the priority setting bit (AT=0).
- Set the value of DMA-halt by interrupt level bit as small as possible (LVL4-LVL0 bit in DILVR register).

- In case of writing the transmission data to transmission data register (TDR) by DMA transfer after transmission data empty flag (SSR: TDRE) becomes "1", or writing the data by software confirming the transmission data empty flag (SSR:TDRE), transmission data empty flag (SSR:TDRE) may not become "0". Therefore, the transmission data should be written before SCL in ACK field falls. There are no restrictions on writing the transmission data by software after the interrupt flag (IBCR:INT) becomes "1".
  When performing DMA transfer or sending the data by software confirming the transmission data empty flag (SSR:TDRE), please follow below procedures if the data cannot be written before SCL in ACK field falls.
  - Setting

    Set the timing of interrupt flag (IBCR:INT) becoming "1" to the 8th bit (WSEL=1).
  - Procedures

    To transmit or receive data by master, the following procedures are required. To transmit or receive data by slave, it is not required to perform the following.

    1. Write the first byte (slave address) to the transmission data register by software.
    2. Set to 8-bit for wait selection (IBCR:WSEL="1" write) at the same time that master is started (IBCR:MSS="1" write).
    3. After sending the first byte, the interrupt flag (IBCR:INT) becomes "1". Write the second byte to transmission data register (TDR) by software after confirming ACK response (IBSR:RACK="0"). Set the DMAC, and activate DMA transfer, then write "0" to interrupt flag (IBCR:INT).
    4. After transmission and reception are completed, terminate the master (IBCR:MSS="0" write) or reboot (IBCR:SCC="1" write).

25. USB Clock Generating Part

This chapter explains the functions and operations of the USB clock generating part.

25.1 Overview

25.2 Configuration

25.3 Registers

25.4 Explanation of Operations and Setting Procedure Examples

25.1 Overview

The USB clock generates the clock used for USB device operations.

Overview

The USB clock generating part has the following functions:

- Enabling/Stopping of USB clock output
- PLL macro oscillation clock division
- USB clock selection
- Stopping of the USB clock in standby mode (main timer mode, watch mode, or stop mode)

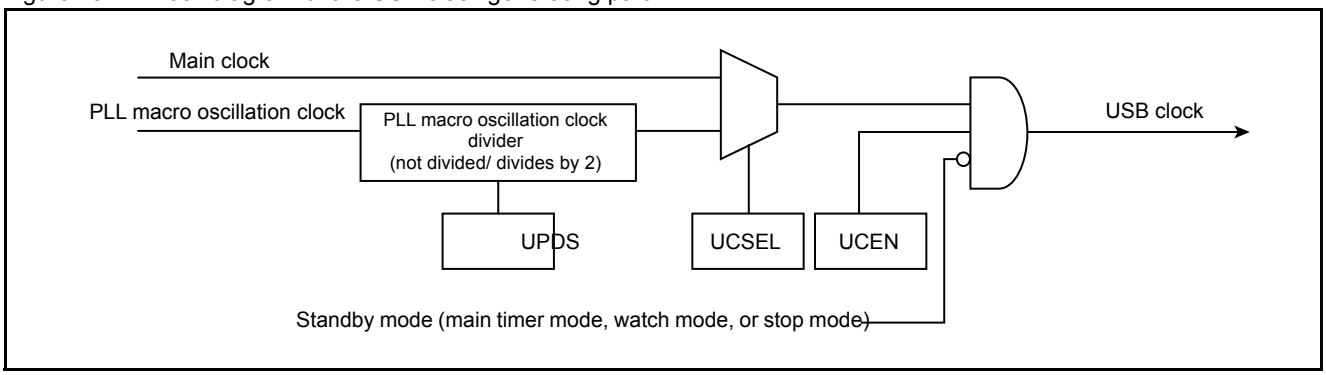
25.2 Configuration

This section explains the configuration of the USB clock generating part.

Block diagram of the USB clock generating part

Figure 25-1 is a block diagram of the USB clock generating part.

Figure 25-1. Block diagram of the USB clock generating part



25.3 Registers

This section explains the configuration and functions of registers used with the USB clock generating part.

Registers of the USB clock generating part

Table 25-1 lists the registers of the USB clock generating part.

Table 25-1. Registers of the USB clock generating part

Abbreviated Register Name	Register Name	Reference
UCCR	USB clock configuration register	25.3.1

25.3.1 USB Clock Configuration Register (UCCR)

This register controls the USB clock.

Figure 25-2 shows the bit configuration of the USB clock configuration register (UCCR).

Figure 25-2. Bit configuration of the USB clock configuration register (UCCR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	UPDS	UCSEL	UCEN
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	X	X	X	X	X	0	0	1

R/W: Read/Write

X: Undefined

[bit7 to bit3]: Reserved bits

In case of writing	Write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2]: UPDS (PLL macro oscillation clock division select bit)

This bit selects whether to divide the PLL macro oscillation clock.

Written Value	Explanation
0	Sets no division.
1	Sets that the clock be divided by 2.

Notes:

- If USB clock output is enabled (UCEN=1), this bit cannot be changed.
- This bit and the UCEN bit must not be changed at the same time.

[bit1]: UCSEL (USB clock select bit)

This bit specifies the main clock or PLL macro oscillation clock as the USB clock.

Written Value	Explanation
0	Main clock
1	PLL macro oscillation clock

Notes:

- If USB clock output is enabled (UCEN=1), this bit cannot be changed.
- This bit and the UCEN bit must not changed at the same time.

[bit0]: UCEN (USB clock output enable bit)

This bit controls USB clock output.

Written Value	Explanation
0	Disables USB clock output.
1	Enables USB clock output.

Note: This bit and the UPDS bit or UCSEL bit must not be changed at the same time.

25.4 Explanation of Operations and Setting Procedure Examples

This section explains USB clock generation control. The section also provides a setting procedure example.

Overview

The USB clock generation control part requires that operations be performed for the following:

- Register settings
- Main clock selection
- PLL macro oscillation clock selection

Register settings

To set the USB clock configuration register (UCCR) to enable USB clock output, follow the procedure below.

1. Disable USB clock output (UCEN=0).
2. Set the UPDS bit (PLL macro oscillation clock division select bit) and UCSEL bit (USB clock select bit).
3. Enable USB clock output (UCEN=1).

Notes:

- Do not perform steps 1 and 2 or steps 2 and 3 at the same time.
- Make these settings while the USBEN bit of the USB enable register (USBEN) is "0".

Main clock selection

Writing "0" to the UCSEL bit specifies the main clock as the USB clock. When specified, the main clock must be oscillating at 48 MHz.

PLL macro oscillation clock selection

Writing "1" to the UCSEL bit specifies the PLL macro oscillation clock as the USB clock. When specified, the PLL macro oscillation clock must have a frequency of 96 MHz. If the PLL macro oscillation clock is 96 MHz, set the PLL macro oscillation clock to be divided by 2 (UPDS=1). shows the relationship.

Table 25-2. Relationship between the USB clock selection and the UPDS/UCSEL bit

	UPDS	UCSEL
Main clock frequency of 48 MHz	-	0
Setting prohibited	0	1
PLL macro oscillation clock frequency of 96 MHz	1	1

Notes:

- The USB clock stops in standby mode (main timer mode, watch mode, or stop mode).
- It is prohibited to set UCSEL=1 and UPDS=0 since PLL macro oscillation clock is 96MHz to 100MHz.

26. DMA Transfer Request Selector

This chapter explains the functions and operations of the DMA Transfer Request selector.

26.1 Overview

26.2 Configuration

26.3 Registers

26.1 Overview

If DREQ is used as a DMA transfer request, the DMA Transfer Request selector passes DREQ from the USB function/HOST to the DMA controller (DMAC). It also returns DACK from the DMA controller (DMAC) to the appropriate macro.

Overview

The DMA Transfer Request selector passes DREQ from the USB function/HOST to the appropriate channel of the DMA controller (DMAC).

It also returns DACK from the DMA controller (DMAC) to the appropriate macro determined from a comparison with the used channel signal for a connection to the corresponding DACK pin. If the USB function/HOST is used, an acknowledgment corresponding to the last transfer is used as a DMA transfer end signal for the connection.

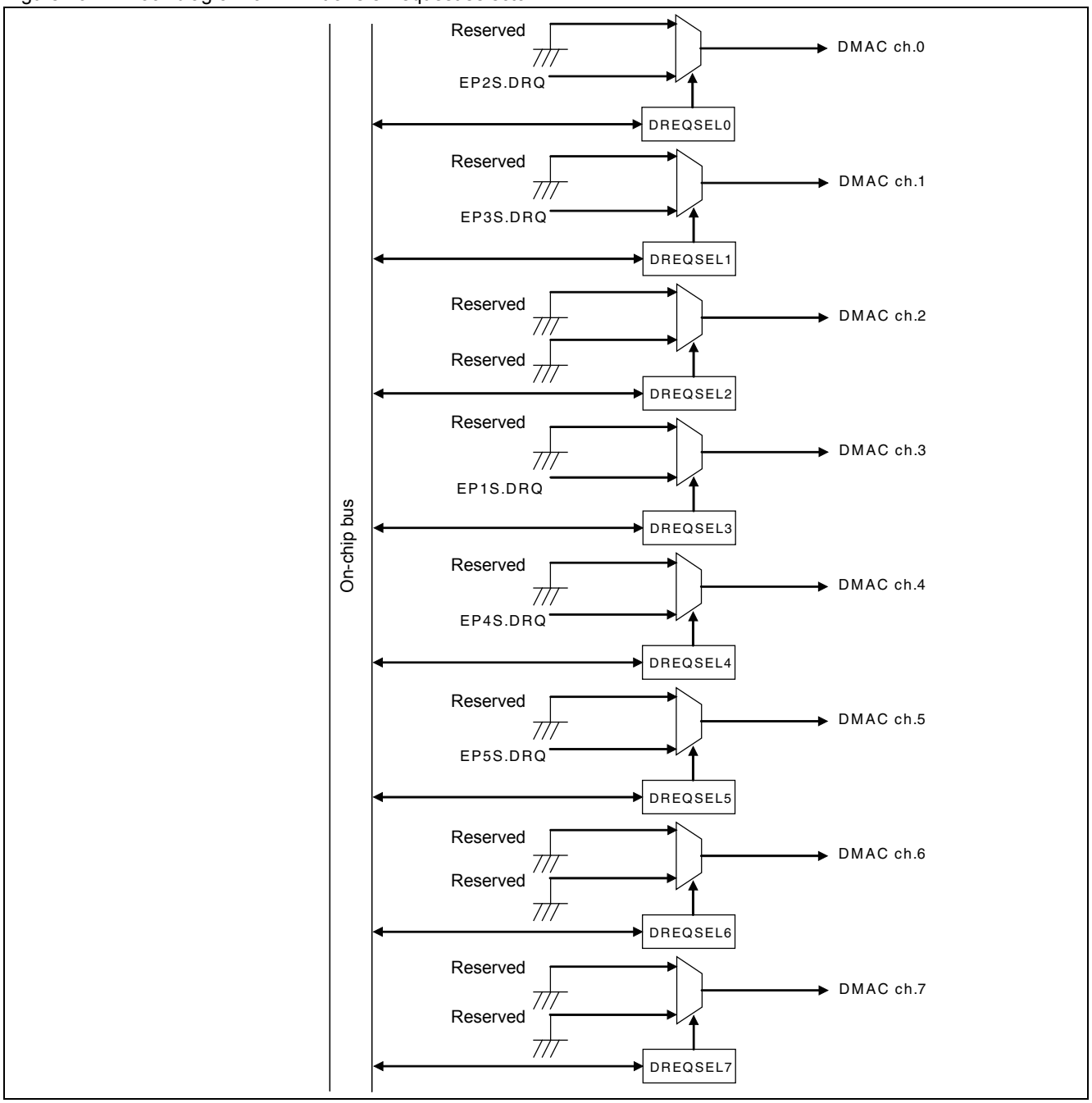
26.2 Configuration

The configuration of DMA Transfer Request is shown below:

Block diagram of DMA transfer request selector

Figure 26-1 shows the block diagram of DMA transfer diagram.

Figure 26-1. Block diagram of DMA transfer request selector



26.3 Registers

This section explains the configuration and functions of registers used by the DMA Transfer Request selector.

Registers of DMA Transfer Request selector

Table 26-1 lists the registers of the DMA Transfer Request selector.

Table 26-1. Registers of the DMA Transfer Request selector

Abbreviated Register Name	Register Name	Reference
DREQSEL	DREQ select register	26.3.1

26.3.1 DREQ Select Register (DREQSEL)

This register selects DREQ.

Figure 26-2 shows the bit configuration of the DREQ select register (DREQSEL).

Figure 26-2. Bit configuration of the DREQ select register (DREQSEL)

bit	7	6	5	4	3	2	1	0
	DREQSEL7	DREQSEL6	DREQSEL5	DREQSEL4	DREQSEL3	DREQSEL2	DREQSEL1	DREQSEL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	1	1	0	1	1

R/W: Read/Write

Note: Be sure to access this register in units of bytes or halfwords.

[bit7 to bit0]: DREQSEL7 to DREQSEL0 (DREQ select bits)

The bit numbers of DREQSEL7 to DREQSEL0 correspond to channel numbers of the DMA controller (DMAC). DREQ is set for each channel.

Channel	DREQ Select Bit	Written Value	Explanation
0	DREQSEL0	0	Reserved
		1	EP2S.DRQ (64 bytes) (interrupt, bulk transfer)
1	DREQSEL1	0	Reserved
		1	EP3S.DRQ (64 bytes) (interrupt, bulk transfer)
2	DREQSEL2	0	Reserved
		1	Reserved
3	DREQSEL3	0	Reserved
		1	EP1S.DRQ (256 bytes) (interrupt, bulk transfer)
4	DREQSEL4	0	Reserved
		1	EP4S.DRQ (64 bytes) (interrupt, bulk transfer)
5	DREQSEL5	0	Reserved
		1	EP5S.DRQ (64 bytes) (interrupt, bulk transfer)
6	DREQSEL6	0	Reserved
		1	Reserved
7	DREQSEL7	0	Reserved
		1	Reserved

Note: If the written value is reserved, "0" is output.

27. USB Function

This chapter explains the functions and operations of the USB function.

27.1 Overview

27.2 Configuration

27.3 Registers

27.4 Explanation of Operations and Setting Procedure Examples

27.1 Overview

The USB function is the interface that supports the Universal Serial Bus (USB) communication protocol. The Full Speed transfer speed (12 Mbps) is supported.

Overview

The USB function supports the USB communication protocol. Basic protocol operations (handshake) are supported by the hardware, and only the processing of USB communication data is required for implementing USB communication.

The USB function has the following features:

- Support of the Full Speed (12 Mbps)
- Automatic response for the device status
- Automatic generation and checking for Bit Stripping, Bit Stuffing, CRC5, and CRC16
- Toggle check using a data synchronization bit
- Automatic response to all standard commands except the Get/SetDescriptor and SynchFrame commands (For the Get/SetDescriptor and SynchFrame commands, processing similar to that of class vendor commands can be executed.)
- Class vendor commands are received as data, and the firmware can respond to these commands.
- Support of up to six endpoints (Endpoint 0 is provided for the fixed use of control transfers.)
- Two transfer data buffers included in each endpoint (One of the two buffers included in endpoint 0 is dedicated for IN transfers, and the other for OUT transfers.)
- Support of automatic data transfer mode using DMA (except for the buffers in endpoint 0)

Note: In case of using USB function, use on-chip bus clock (HCLK) with 13 MHz or over.

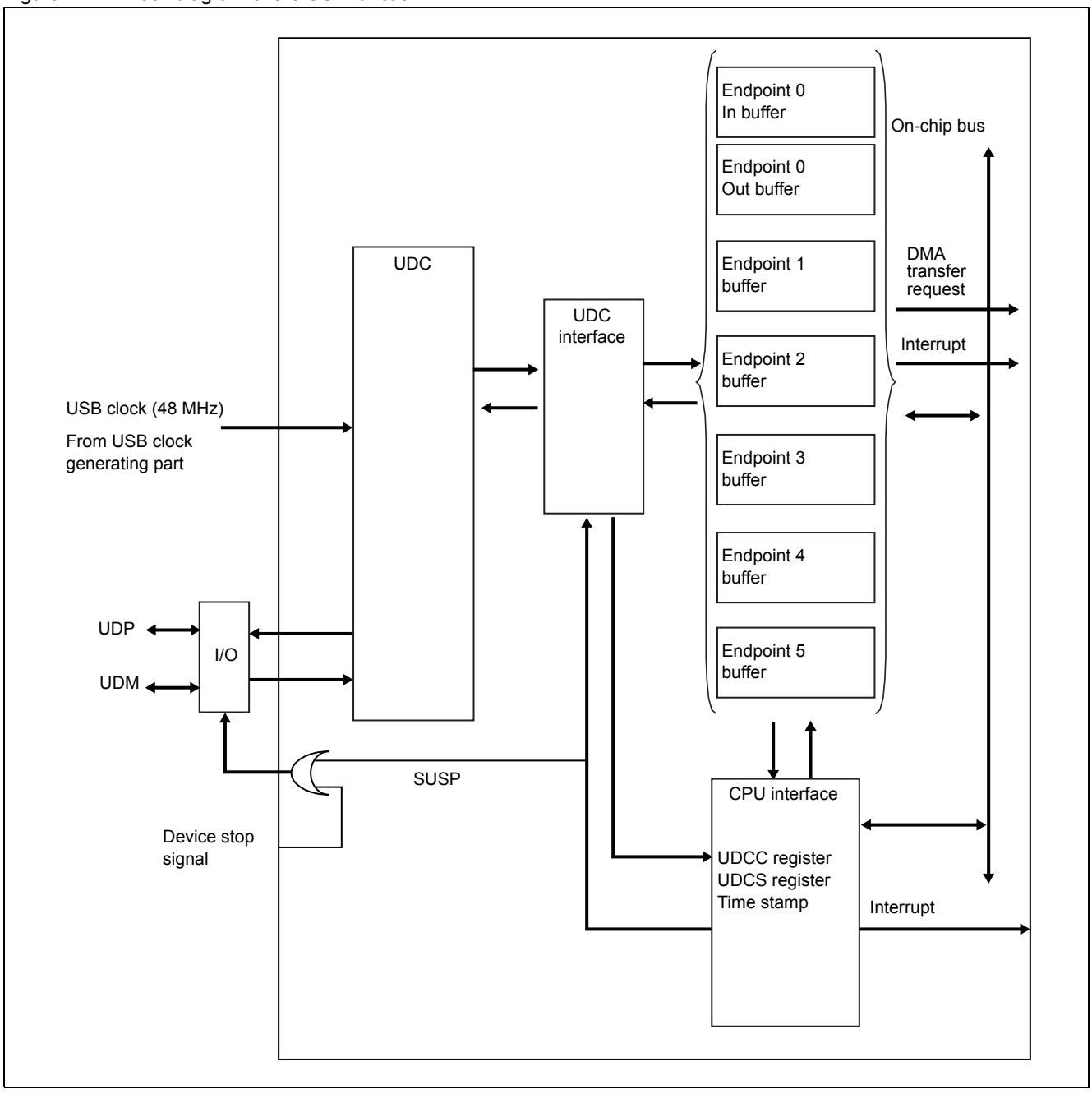
27.2 Configuration

This section shows the configuration of the USB function.

Block diagram of the USB function

Figure 27-1 is a block diagram of the USB function.

Figure 27-1. Block diagram of the USB function



27.3 Registers

This section explains the configuration and functions of the registers used by the USB function.

USB function register list

Table 27-1 lists the registers of the USB function.

Table 27-1. Registers of the USB function

Abbreviated Register Name	Register Name	Reference
UCCR	USB Clock Setting Register	25.3.1
USBSEL	USB Selection Register	27.3.1
USBEN	USB Enable Register	27.3.2
UDCC	UDC control register	27.3.3
EP0C	EP0 control register	27.3.4
EP1C	EP1 control register	27.3.5
EP2C	EP2 control register	27.3.5
EP3C	EP3 control register	27.3.5
EP4C	EP4 control register	27.3.5
EP5C	EP5 control register	27.3.5
TMSP	Time stamp register	27.3.6
UDCS	UDC status register	27.3.7
UDCIE	UDC enable interrupt request register	27.3.8
EP0IS	EP0I status register	27.3.9
EP0OS	EP0O status register	27.3.10
EP1S	EP1 status register	27.3.11
EP2S	EP2 status register	27.3.11
EP3S	EP3 status register	27.3.11
EP4S	EP4 status register	27.3.11
EP5S	EP5 status register	27.3.11
EP0DTH	EP0 data register upper	27.3.12
EP0DTL	EP0 data register lower	27.3.12
EP1DTH	EP1 data register upper	27.3.12
EP1DTL	EP1 data register lower	27.3.12
EP2DTH	EP2 data register upper	27.3.12
EP2DTL	EP2 data register lower	27.3.12
EP3DTH	EP3 data register upper	27.3.12
EP3DTL	EP3 data register lower	27.3.12
EP4DTH	EP4 data register upper	27.3.12
EP4DTL	EP4 data register lower	27.3.12
EP5DTH	EP5 data register upper	27.3.12
EP5DTL	EP5 data register lower	27.3.12

Register bit updating timing UDCC.RST dependency

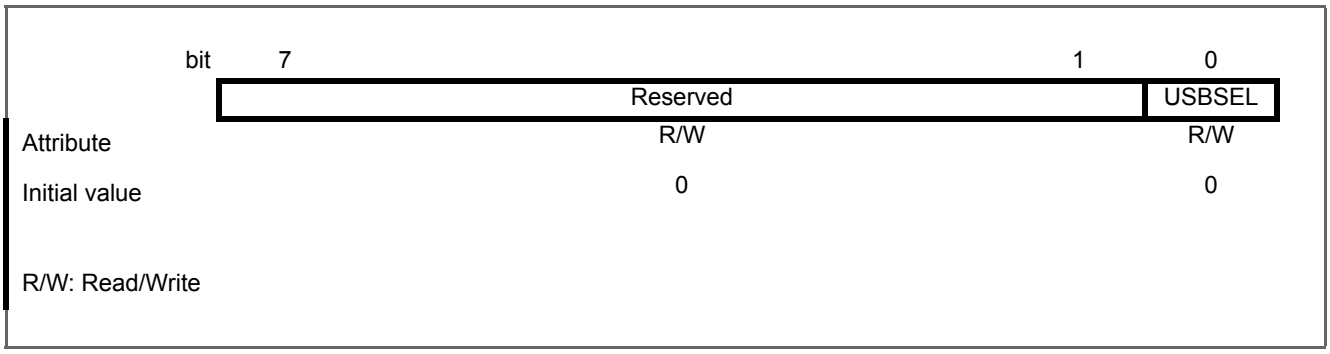
	Register	Bit
Register bit to be updated at UDCC.RST=1	UDCC	HCONX, RFBK, PWC
	EP0C	PKS0
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
	EP3C	EPEN, TYPE, DIR, PKS3
	EP4C	EPEN, TYPE, DIR, PKS4
	EP5C	EPEN, TYPE, DIR, PKS5
Register bit to be initialized at UDSS.RST=1 (Update the register bit at UDCC.RST=0)	EP0IS	BFINI, DRQI
	EP0OS	BFINI, DRQO, SPK
	EP1S	BFINI, DRQ, SPK
	EP2S	BFINI, DRQ, SPK
	EP3S	BFINI, DRQ, SPK
	EP4S	BFINI, DRQ, SPK
	EP5S	BFINI, DRQ, SPK
	TMSP	TMSP
	UDCS	SUSP, SOF, BRST, WKUP, SETP, CONF
	UDCIE	SUSPIE, SOFIE, BRSTIE, WKUPIE, CONFN, CONFIE
Register bit not to be affected by UDCC.RST.	UDCC	RESUME, USTP
	EP0C	STAL
	EP1C	DMAE, NULE, STAL
	EP2C	DMAE, NULE, STAL
	EP3C	DMAE, NULE, STAL
	EP4C	DMAE, NULE, STAL
	EP5C	DMAE, NULE, STAL
	EP1DTH/L	BFDT
	EP2DTH/L	BFDT
	EP3DTH/L	BFDT
	EP4DTH/L	BFDT
	EP5DTH/L	BFDT

27.3.1 USB Selection Register (USBSEL)

This register selects USB.

Figure 27-2 shows the bit configuration of the USB selection register.

Figure 27-2. Bit Configuration of USB Selection Register.



Note: Access this register by byte or half-word.

[bit7 to bit1]: Reserved bit

In case of writing	Write "0".
In case of reading	"0" is read.

[bit0]: USBSEL (USB selection bit)

This bit selects USB.

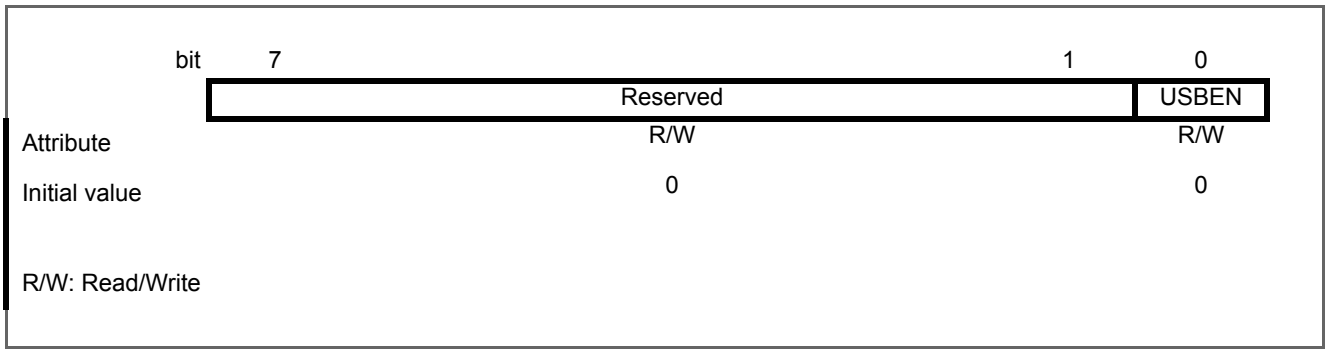
Written Value	Explanation
0	Uses USB function/HOST.
1	The setting is prohibited.

27.3.2 USB Enable Register (USBEN)

This register enables the usage of USB.

Figure 27-3 shows the bit configuration of USB enable register.

Figure 27-3. Bit Configuration of USB Enable Register



Note: Access this register by byte.

[bit7 to bit1]: Reserved bit

In case of writing	Write "0".
In case of reading	"0" is read.

[bit0]: USBEN (USB enable bit)

Enables the use of USB function/HOST.

Written Value	Explanation
0	Disable the use of USB function/HOST.
1	Enable the use of USB function/HOST.

Note: Write "1" in this bit to use USB function/HOST.

27.3.3 UDC Control Register (UDCC)

This register controls the UDC core circuit.

Figure 27-4 shows the bit configuration of the UDC control register (UDCC).

Figure 27-4. Bit configuration of the UDC control register (UDCC)

bit	7	6	5	4	3	2	1	0
	RST	RESUM	HCONX	USTP	Reserved	Reserved	RFBK	PWC
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	1	0	0	0	0	0

R/W: Read/Write

[bit7]: RST (Function reset bit)

This bit individually resets the USB function. This reset is equivalent to the system reset of the chip. The bit is used to reset the USB function when the host is connected via a cable. The initial value of this bit is "1" and this bit can be reset. To cancel the reset condition, write "0" to this bit.

Written Value	Explanation
0	Cancels the reset condition of the USB function.
1	Resets the USB function.

Note: This bit initializes the relevant bits in the time stamp register (TMSP), UDC status register (UDCS), and UDC enable interrupt request register (UDCIE) at one time. Also, it initializes the BFINI bit in each of the status registers (EP0IS, EO0OS, EP1S to EP5S) at one time. After making the initial setting, clear this bit (the BFINI bits are not cleared), and then clear the BFINI bits of the endpoints used.

[bit6]: RESUM (Resume set bit)

If remote wakeup is enabled (the DEVICE_REMOTE_WAKEUP bit has been set by the host with the SET_FEATURE command) and the suspended state is set, writing "1" in this bit sets the USB device to start the RESUM. To release the RESUM, write "0" in this bit.

Written Value	Explanation
0	Releases the resume start instruction from the USB device.
1	Issues the RESUM start instruction from the USB device.

Note: Write "1" in this bit only when the USB device is in the suspended state and remote wakeup is in the enabled state.

[bit5]: HCONX (Host connection bit)

This bit is used so that the USB device recognizes the connection with the host or hub.

Written Value	Explanation
0	Connected with the host or hub.
1	Disconnected from the host or hub.

Notes:

- Even when the external pull-up resistor is set to ON and the connection from the host or hub is recognized, the bus reset and command of the USB bus are ignored if this bit is "1".
- Change this bit when "RST" of bit7 in UDC Control Register (UDCC) is "1".

[bit4]: USTP (UDC stop bit)

This bit stops the clock of the USB operation block. When no USB operation is to be performed, "1" is written to the bit to stop the clock of the USB operation block, and this can reduce power dissipation.

Written Value	Explanation
0	Sets normal mode.
1	Stops the clock of the USB operation block.

Notes:

- Before stop mode is set, write "1" in this bit. To cancel stop mode, write "0" in the SUSP bit in the UDC status register (UDCS), and then write "0" in the USTP bit.
- When the full speed function is used, write "0" in this bit three cycles after writing "1" in the RST bit, so that this bit is definitely reset. This bit and the RST bit can be cleared at the same time.

[bit3, bit2]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit1]: RFBK (Rate feed back mode bit)

This bit specifies a data toggle mode for interrupt transfers.

Written Value	Explanation
0	Selects the alternating data toggle mode. The data PID is toggled when transfer is completed successfully.
1	Selects the data toggle mode. The data PID is unconditionally toggled.

Note: Change this bit when RST of bit7 in CDC Control Register (UDCC) is "1".

[bit0]: PWC (Power control bit)

This bit specifies an operating power mode (self-powered or bus-powered) for the USB function. This information is reflected in the GetStatus standard command.

Written Value	Explanation
0	Uses bus-powered as the operating power source.
1	Uses self-powered as the operating power source.

Note: Change this bit when RST of bit7 in UDC Control Register is "1".

27.3.4 EP0 Control Register (EP0C)

This register controls endpoint 0.

Figure 27-5 shows the bit configuration of the EP0 control register (EP0C).

Figure 27-5. Bit configuration of the EP0 control register (EP0C)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Reserved	Reserved	STAL	Reserved
Attribute	-	-	-	-	-	-	R/W	-
Initial value	X	X	X	X	0	0	0	0

bit	7	6						0
	Reserved	PKS0						
Attribute	-	R/W						
Initial value	0	1000000						

R/W: Read/Write

-: Undefined

X: Undefined

[bit15 to bit12]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit11, bit10]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit9]: STAL (Stall EP0 set bit)

This bit specifies the STALL state (STALL response) for endpoint 0.

Written Value	Explanation
0	Releases the endpoint from the STALL state.
1	Sets the STALL state.

Note: While this bit is "1", the STALL response is continued for the host. After "0" is written to this bit, a normal SETUP packet is received and the device returns from the STALL state.

[bit8, bit7]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit6 to bit0]: PKS0 (Packet size EP0 set bit)

These bits specify the maximum number of transfer bytes in one packet. The maximum number of transfer bytes that can be specified in one packet for endpoint 0 is 64 bytes, which is common to IN and OUT transfers.

Writing "08_H" specifies 8 bytes to be transferred, and writing "40_H" specifies 64 bytes to be transferred.

Notes:

- Set these bits when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the EP0I/O status register (EP0I/OS) are "1". Do not change the value of these bits during a USB operation.
- Do not write a value that is "40_H" or greater or "00_H" in these bits.

27.3.5 EP1 to 5 Control Registers (EP1C to EP5C)

These registers control endpoints 1 to 5.

Figure 27-6 shows the bit configuration of the EP1 to 5 control registers (EP1C to EP5C).

Figure 27-6. Bit configuration of the EP1 to 5 control registers (EP1C to EP5C)

EP1 control register (EP1C)

bit	15	14	13	12	11	10	9	8
	EPEN	TYPE		DIR	DMAE	NULE	STAL	PSK1
Attribute	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Initial value	0	11		0	0	0	0	1

bit	7	0
	PSK1	
Attribute	R/W	
Initial value	00000000	

EP2 to 5 control registers (EP2C to EP5C)

bit	15	14	13	12	11	10	9	8
	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved
Attribute	R/W	R/W		R/W	R/W	R/W	R/W	-
Initial value	0	11		0	0	0	0	0

bit	7	6	0
	Reserved	PSK2 to PSK5	
Attribute	-	R/W	
Initial value	0	1000000	

R/W: Read/Write

-: Undefined

[bit15]: EPEN (Endpoint1 to 5 enable bit)

This bit enables or disables the endpoint. If the endpoint is enabled, it is configured by the host as an endpoint used for the USB function, and the TYPE bit, DIR bit, and PKS bit in this register become valid in configuration information.

Written value	Explanation
0	Disables the endpoint.
1	Enables the endpoint.

Note: Set this bit when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of this bit during a USB operation.

[bit14, bit13]: TYPE (Endpoint type bit)

These bits specify a transfer type supported by the endpoint.

Written Value	Explanation
00	This specification is prohibited.
01	This specification is prohibited.
10	Specifies bulk transfer.
11	Specifies interrupt transfer.

Note: Set these bits when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of these bits during a USB operation.

[bit12]: DIR (Endpoint direction bit)

This bit specifies a transfer direction supported by the endpoint.

Written Value	While in USB Function Operation Mode	While in Host Operation Mode (for Either EP1 or EP2 Only)
0	Specifies an OUT endpoint.	Specifies an IN endpoint.
1	Specifies an IN endpoint.	Specifies an OUT endpoint.

Note: Set this bit when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of this bit during a USB operation.

[bit11]: DMAE (DMA enable bit)

This bit sets or cancels automatic buffer transfer mode. Automatic buffer transfer mode uses DMA to write transfer data to or read transfer data from the transmission/reception buffer, and it automatically transfers as much transmission/reception data as specified by the data transfer count set for DMA, in sync with IN/OUT data requests from the host.

Written Value	Explanation
0	Cancels automatic buffer transfer mode.
1	Sets automatic buffer transfer mode.

Note: If "1" is written to this bit and automatic buffer transfer mode is set, do not access the transmission/reception buffers from the CPU. Also, for data transfer in the OUT direction, specify the multiples of the number specified in a PKS bit in this register as the DMA transfer count.

[bit10]: NULE (NULL enable set bit)

This bit sets or cancels NULL automatic transfer mode. If automatic buffer transfer mode is set (DMAE = 1), NULL automatic transfer mode automatically transfers 0-byte data upon detection of the last packet transfer for a data transfer request in the IN direction.

Written Value	Explanation
0	Cancels NULL automatic transfer mode.
1	Sets NULL automatic transfer mode.

Note: If data transfer is in the OUT direction or automatic buffer transfer mode is not set, the value of this bit does not affect communication.

[bit9]: STAL (STALL set bit)

This bit sets the STALL state (STALL response) for the endpoint or releases it from the STALL state.

Written Value	Explanation
0	Releases the endpoint from the STALL state.
1	Sets the STALL state.

Note: While this bit is "1", the STALL response is continued for the host. When the ClearFeature command is issued from the host after "0" is written to this bit, the device returns from the STALL state.

[bit8, bit7]: EP2 to EP5 reserved bits

These bits in the EP2 to 5 control registers (EP2C to EP5C) are reserved.

In case of writing	Write "0".
In case of reading	"0" is read.

[bit8 to bit0]: PKS1 (Packet size EP1 set bits)

[bit6 to bit0]: PKS2 to PKS5 (Packet size EP2 To Ep5 set bits)

These bits specify the maximum transfer count in one packet. The following table lists the maximum transfer count in a single packet for each of endpoints 1 to 5:

Endpoint	Maximum Transfer Count	Allowable Setting Range
1	256 bytes (an odd number can also be specified)	001_H to 100_H
2 to 5	64 bytes (an odd number can also be specified)	01_H to 40_H

Notes:

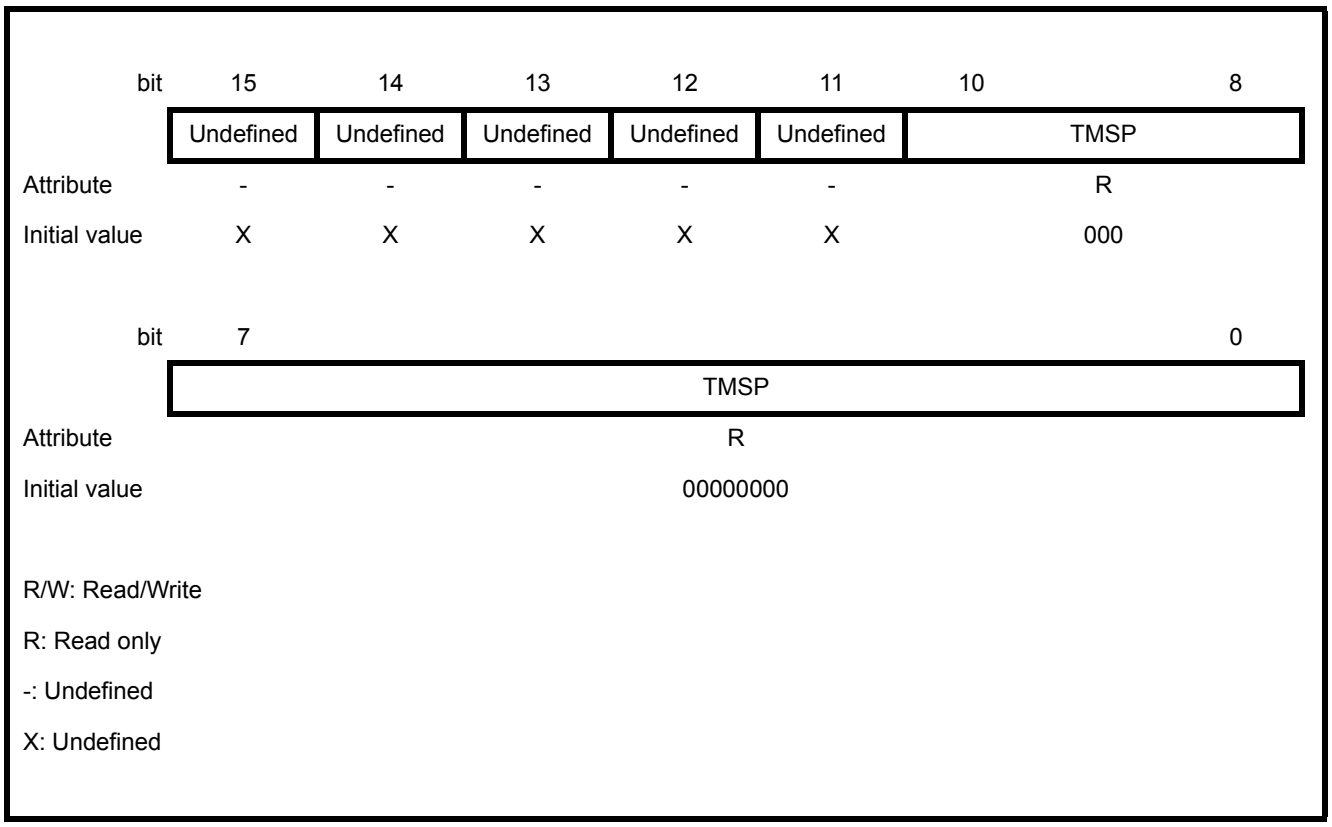
- Set these bits when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of these bits during a USB operation.
- Do not write a value that is equal to or greater than the maximum transfer count ("100_H" or "40_H") or "00" in these bits.
- Set "00" in bit8 and bit7 for endpoints 2 to 5.
- When using automatic buffer transfer mode (DMAE = 1), do not set a value of "0_H" to "2_H" to the relevant endpoint.

27.3.6 Time Stamp Register (TMSP)

This register indicates the frame number as of the SOF packet receipt time.

Figure 27-7 shows the bit configuration of the time stamp register (TMSP).

Figure 27-7. Bit configuration of time stamp register (TMSP)



[bit15 to bit11]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit10 to bit0]: TMSP (Time stamp bits)

These bits indicate the frame number as of the SOF packet reception time. The frame number is updated when an SOF packet is received.

27.3.7 UDC Status Register (UDCS)

This register indicates the bus status concerning USB communication and the command receipt status for a particular command. The bits except the SETP bit are interrupt resources. An interrupt request to the CPU is generated when a corresponding interrupt enable bit is enabled.

Figure 27-8 shows the bit configuration of the UDC status register (UDCS).

Figure 27-8. Bit configuration of the UDC status register (UDCS)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	SUSP	SOF	BRST	WKUP	SETP	CONF
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	0	0	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit5]: SUSP (Suspend bit)

This bit indicates whether the USB function has transferred to the suspended state. The bit is an interrupt resource.

SUSP	In Case of Reading	In Case of Writing
0	The suspended state has not been detected.	The interrupt resource is cleared.
1	The USB function has transferred to the suspended state.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit4]: SOF (Start of frame bit)

This bit indicates whether an SOF packet has been received and the value of the time stamp register (TMSP) has been updated. The bit is an interrupt resource.

SOF	In Case of Reading	In Case of Writing
0	An SOF packet has not been received.	The interrupt resource is cleared.
1	An SOF packet has been received.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit3]: BRST (Bus reset bit)

This bit indicates whether a USB bus reset has been detected. The bit is an interrupt resource.

BRST	In Case of Reading	In Case of Writing
0	No USB bus reset has been detected.	The interrupt resource is cleared.
1	A USB bus reset has been detected.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- When a USB bus reset is detected with this bit, use the BFINI bit in the EP0I status register (EP0IS), the BFINI bit in the EP0O status register (EP0OS) and the BFINI bit in the EP1 to EP5 status register (EP1S to EP5S) to initialize the buffer.

[bit2]: WKUP (Wake up bit)

This bit indicates whether the USB function has returned from the suspended state. The two resources for the return are remote wakeup because "1" is written to the RESUM bit in the UDC control register (UDCC) and wakeup because of a request from the host. This bit is automatically set when the host requests the return. The bit is an interrupt resource.

WKUP	In Case of Reading	In Case of Writing
0	No return request (resume) by the host has been detected.	The interrupt resource is cleared.
1	A return request (resume) by the host has been detected.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- Even when a return request from the host is detected, this bit is not set if "1" is set in the RESUM bit in the UDC control register (UDCC).

[bit1]: SETP (Setup bit)

This bit indicates whether data has been received for the setup stage of a USB control transfer.

SETP	In Case of Reading	In Case of Writing
0	No data has been received for the setup stage of a control transfer.	The resource is cleared.
1	Data has been received for the setup stage of a control transfer.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- This bit is not set for an automatic response for a standard command. The bit is not an interrupt resource.

[bit0]: CONF (Configuration bit)

This bit indicates whether USB function configuration has been completed. The bit is set when the SetConfig USB command has been received correctly. The bit is an interrupt resource.

CONF	In Case of Reading	In Case of Writing
0	SetConfig has not been detected.	The interrupt resource is cleared.
1	SetConfig has been detected.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

27.3.8 UDC Enable Interrupt Request Register (UDCIE)

This register enables interrupts by the interrupt resource of the UDC status register (UDCS) corresponding to a specific bit (except the CONFN bit).

Figure 27-9 shows the bit configuration of the UDC enable interrupt request register (UDCIE).

Figure 27-9. Bit configuration of UDC enable interrupt request register (UDCIE)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	SUSPIE	SOFIE	BRSTIE	WKUPIE	CONFN	CONFIE
Attribute	-	-	R/W	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

[bit7, bit6]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit5]: SUSPIE (SUSP interrupt enable bit)

This bit enables interrupts by the interrupt resource of the SUSP bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the SUSP resource.
1	Enables interrupts by the SUSP resource.

[bit4]: SOFIE (SOF interrupt enable bit)

This bit enables interrupts by the interrupt resource of the SOF bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the SOF resource.
1	Enables interrupts by the SOF resource.

[bit3]: BRSTIE (BRST interrupt enable bit)

This bit enables interrupts by the interrupt resource of the BRST bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the BRST resource.
1	Enables interrupts by the BRST resource.

[bit2]: WKUPIE (WKUP interrupt enable bit)

This bit enables interrupts by the interrupt resource of the WKUP bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the WKUP resource.
1	Enables interrupts by the WKUP resource.

[bit1]: CONFN (Configuration number bit)

This bit indicates the configuration number. The bit is updated when the interrupt resource of the CONF bit in the UDC status register (UDCS) is set.

Read Value	Explanation
0	The configuration number is 0.
1	The configuration number is 1.

[bit0]: CONFIE (CONF interrupt enable bit)

This bit enables interrupts by the interrupt resource of the CONF bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the CONF resource.
1	Enables interrupts by the CONF resource.

27.3.9 EP0I Status Register (EP0IS)

This register indicates the status regarding IN-direction transfers of endpoint 0.

Figure 27-10 shows the bit configuration of the EP0I status register (EP0IS).

Figure 27-10. Bit configuration of the EP0I status register (EP0IS)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQIIE	Undefined	Undefined	Undefined	DRQI	Undefined	Undefined
Attribute	R/W	R/W	-	-	-	R/W	-	-
Initial value	1	0	X	X	X	1	X	X

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Attribute	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X

R/W: Read/Write

-: Undefined

X: Undefined

[bit15]: BFINI (Buffer initial bit)

This bit initializes the transfer data transmit buffer. The bit automatically becomes "1" when "1" is written to the RST bit in the UDC control register (UDCC). If the RST bit is "1", write "0" in the BFINI bit only after writing "0" in the RST bit.

Written Value	Explanation
0	Cancels initialization of the transmit buffer.
1	Initializes the transmit buffer.

Note: The transmit buffer and the DRQI bit are initialized when "1" is written to this bit. Before initializing the transmit buffer, first verify that "1" is written to the DRQI bit or DRQO bit in the EP0O status register (EP0OS) and there is no access from the host, and then write "1" in the STAL bit in the EP0 control register (EP0C) as necessary.

[bit14]: DRQIIE (Data request in interrupt enable bit)

This bit enables interrupts by the interrupt resource of the DRQI bit.

Written Value	Explanation
0	Disables interrupts by the DRQI resource.
1	Enables interrupts by the DRQI resource.

[bit13 to bit11]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit10]: DRQI (Data request in bit)

This bit indicates whether an IN packet transfer from the host for endpoint 0 has been completed correctly and the data has been read from the transmit buffer; in other words, it indicates whether the next transmit data can be written to the buffer. The bit is an interrupt resource.

DRQI	In Case of Reading	In Case of Writing
0	Transmit data cannot be written.	The interrupt resource is cleared.
1	Transmit data can be written.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit9 to bit0]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

27.3.10 EP0O Status Register (EP0OS)

This register indicates the status regarding OUT-direction transfers of endpoint 0.

Figure 27-11 shows the bit configuration of the EP0O status register (EP0OS).

Figure 27-11. Bit configuration of EP0O status register (EP0OS)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQOIE	SPKIE	Undefined	Undefined	DRQO	SPK	Reserved
Attribute	R/W	R/W	R/W	-	-	R/W	R/W	-
Initial value	1	0	0	X	X	0	0	0

bit	7	6						0
	Reserved	SIZE						
Attribute	-	R						
Initial value	0	X						

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

[bit15]: BFINI (Buffer initial bit)

This bit initializes the transfer data receive buffer. The bit automatically becomes "1" when "1" is written to the RST bit in the UDC control register (UDCC). If the RST bit is "1", write "0" in the BFINI bit only after writing "0" in the RST bit.

Written Value	Explanation
0	Cancels initialization of the receive buffer.
1	Initializes the receive buffer.

Note: The receive buffer, DRQO bit, and SPK bit are initialized when "1" is written to this bit. Before initializing the reception buffer, first verify that "1" is written to the DRQO bit or DRQI bit in the EP0I status register (EP0IS) and there is no access from the host, and then write "1" in the STAL bit in the EP0 control register (EP0C) as necessary.

[bit14]: DRQOIE (Data request out interrupt enable bit)

This bit enables interrupts by the interrupt resource of the DRQO bit.

Written Value	Explanation
0	Disables interrupts by the DRQO resource.
1	Enables interrupts by the DRQO resource.

[bit13]: SPKIE (SPK interrupt enable bit)

This bit enables interrupts by the interrupt resource of the SPK bit.

Written Value	Explanation
0	Disables interrupts by the SPK resource.
1	Enables interrupts by the SPK resource.

[bit12, bit11]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit10]: DRQO (Data request out bit)

This bit indicates whether an OUT packet transfer from the host for endpoint 0 has been completed correctly and the data has been written to the reception buffer; in other words, it indicates whether the received data can be read from the buffer. The bit is an interrupt resource.

DRQO	In Case of Reading	In Case of Writing
0	Received data cannot be read.	The interrupt resource is cleared.
1	Received data can be read.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- Clear this bit by writing "0" in the bit after reading data from the reception buffer. However, write "0" in the bit only if "1" has been written to it.
- If "1" is held in this bit, the reception buffer is not updated. The buffer can be updated while "0" is written to the bit. If the bit is "1" when an OUT packet is requested, the NAK response is automatically returned to the host.

[bit9]: SPK (Short packet bit)

This bit indicates that packet (including 0-byte packets) of the transfer data received correctly from the host has not reached the maximum packet count specified in the PKS bit in the EP0 control register (EP0C). The bit is an interrupt resource.

SPK	In Case of Reading	In Case of Writing
0	Data of the max. packet count have been received.	The interrupt resource is cleared.
1	Data of less than the max. packet count have been received.	Ignored

Note: When a read-modify-write instruction is used, "1" is read.

[bit8, bit7]: Reserved bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit6 to bit0]: SIZE (Packet size bits)

These bits indicate the number of data bytes written to the reception buffer after the completion of an OUT packet transfer of endpoint 0. The bits are updated to valid values when the interrupt resource of the DRQO bit in the EP0O status register (EP0OS) is set. For example, when 8 bytes of data are written, these bits indicate "08_H", and when 64 bytes of data are written, these bits indicate "40_H".

27.3.11 EP1 to 5 Status Registers (EP1S to EP5S)

These registers indicate the status regarding endpoints 1 to 5.

Figure 27-12 shows the bit configuration of the EP1 to 5 status registers (EP1S to EP5S).

Figure 27-12. Bit configuration of EP1 to 5 status registers (EP1S to EP5S)

EP1 status register (EP1S)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	SIZE1
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	R
Initial value	1	0	0	X	0	0	0	X

bit	7	0
	SIZE1	
Attribute	R	
Initial value	X	

EP2 to 5 status registers (EP2S to EP5S)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	-
Initial value	1	0	0	X	0	0	0	0

bit	7	6	0
	Reserved	SIZE 2 to SIZE 5	
Attribute	-	R	
Initial value	0	X	

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

[bit15]: BFINI (Endpoint1 to 5 enable bit)

This bit initializes the transfer data transmit/receive buffers. The bit automatically becomes "1" when "1" is written to the RST bit in the UDC control register (UDCC). If the RST bit is "1", write "0" in the BFINI bit only after writing "0" in the RST bit.

Written Value	Explanation
0	Cancels initialization of the transmission/reception buffers.
1	Initializes the transmission/reception buffers.

Note: The transmission/reception buffers of endpoints 1 to 5 are configured as double buffers. Writing "1" in this bit initializes the double buffers, the DRQ bit, and the SPK bit at the same time. Before initializing the transmission/reception buffers, first verify that the DRQ bit has become "1", the BUSY bit has become "0", and there is no access from the host, and then write "1" in the STAL bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) as necessary.

[bit14]: DRQIE (Data request interrupt enable bit)

This bit enables interrupts by the interrupt resource of the DRQ bit.

Written Value	Explanation
0	Disables interrupts by the DRQ resource.
1	Enables interrupts by the DRQ resource.

Note: When using automatic buffer transfer mode, make the setting of DMA to enable the transfer, and then write "1" in the DRQIE bit.

[bit13]: SPKIE (SPK interrupt enable bit)

This bit enables interrupts by the interrupt resource of the SPK bit.

Written Value	Explanation
0	Disables interrupts by the SPK resource.
1	Enables interrupts by the SPK resource.

[bit12]: Reserved bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit11]: BUSY (Busy flag bit)

This bit indicates that the host is writing data to or reading data from the transmission/reception buffer. The bit is automatically updated.

Read Value	Explanation
0	There is no access from the host.
1	The host is writing or reading data.

Note: If "1" is indicated by this bit and the DRQ bit, a buffer different from the one accessed by the CPU or DMA is being accessed by the host. Usually, the control with the BUSY bit is not required. Before initializing the transmission/reception buffer by using the BFINI bit, verify that the DRQ bit is "1", the BUSY bit is "0", and there is no access from the host, and then write "1" in the STAL bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C).

[bit10]: DRQ (Data request bit)

This bit indicates that a packet transfer of one of endpoints 1 to 5 has been completed correctly, and data processing is required. The bit is an interrupt resource.

DRQ	In Case of Reading	In Case of Writing
0	The data cannot be written/read from the transmission/ reception buffer.	The interrupt resource is cleared.
1	A packet transfer has been completed correctly.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- If automatic buffer transfer mode is not used, write "0" in this bit to clear it after data is written to or read from the transmission/reception buffer. The buffer being accessed is switched when the bit is cleared. After DRQ bit is cleared, DRQ of "0" might not be read. If IN is set for the transfer direction, this bit is "1", and the bit is cleared without the data in the buffer being written, data with a length of 0 bytes will be set as a result. If "1" is written to the DIR bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) in the initial settings, the DRQ bit of the relevant endpoint also becomes "1" at the same time. Also, if this bit is "0", do not write "0" in it.

[bit9]: SPK (Short packet bit)

This bit indicates that the transfer data from the host has been received normally but the number of transfer bytes in individual packets (including 0-byte packets) is less than the max. packet count specified per packet in the PKS bit in the relevant register among EP1 to 5 control registers (EP1C to EP5C). The bit is an interrupt resource.

SPK	In Case of Reading	In Case of Writing
0	The data of the max. packet count have been received.	The interrupt resource is cleared.
1	The data less than the max. packet count have been received.	Ignored

Notes:

- When a read-modify-write instruction is used, "1" is read.
- If data transfer is in the IN direction, this bit does not change.

[bit8, bit7]: EP2 to EP5 reserved bits

These bits in the EP2 to 5 status registers (EP2S to EP5S) are reserved bits.

In case of writing	Ignored
In case of reading	"0" is read.

[bit8 to bit0]: SIZE1 (Packet size bits)

[bit6 to bit0]: SIZE2 to SIZE 5 (Packet size bits)

These bits indicate the number of data bytes written to the reception buffer after the completion of an OUT packet transfer of one of endpoints 1 to 5. The bits are updated to valid values when the relevant DRQ interrupt resource is generated.

The following table lists the maximum transfer count of each of endpoints 1 to 5:

Endpoint	Maximum Transfer Count	Indication Range
1	256 bytes	000_H to 100_H
2 to 5	64 bytes	00_H to 40_H

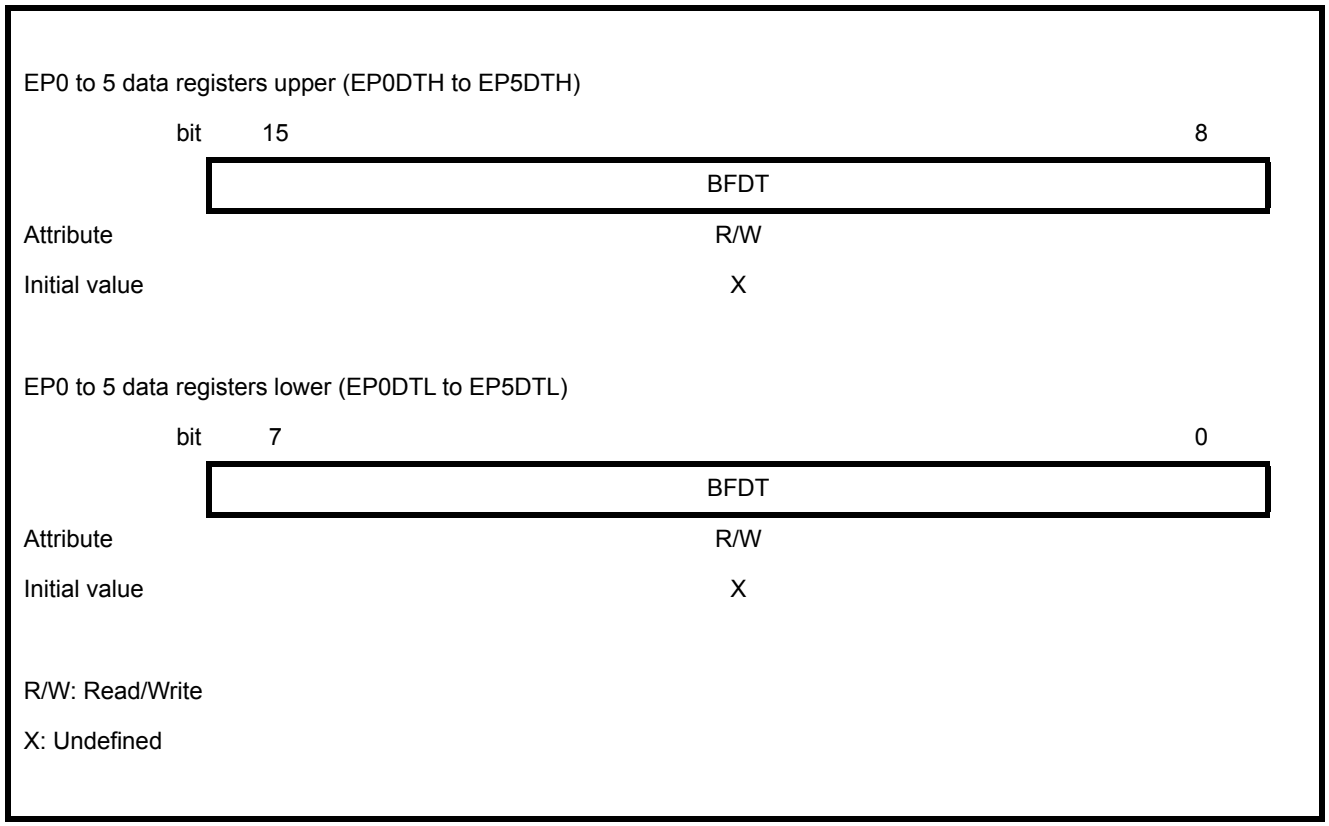
Note: These bits indicate the number of data bytes written to the buffer by the host for an OUT-direction transfer. The read value has no meaning for an IN-direction transfer.

27.3.12 EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL)

These registers are used for read access or write access of the transmit/receive buffers holding transfer data regarding end-points 0 to 5.

Figure 27-13 shows the bit configuration of the EP0 to 5 data registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL).

Figure 27-13. Bit configuration of the EP0 to 5 data registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL)

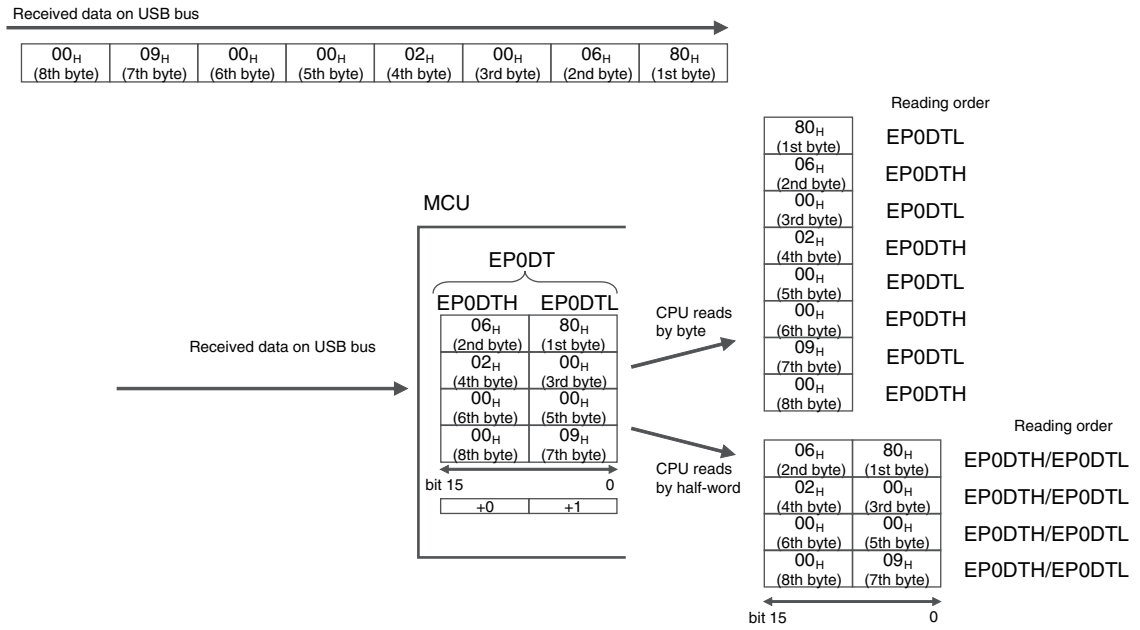


[bit15 to bit0]: BFDT (Buffer data bit)

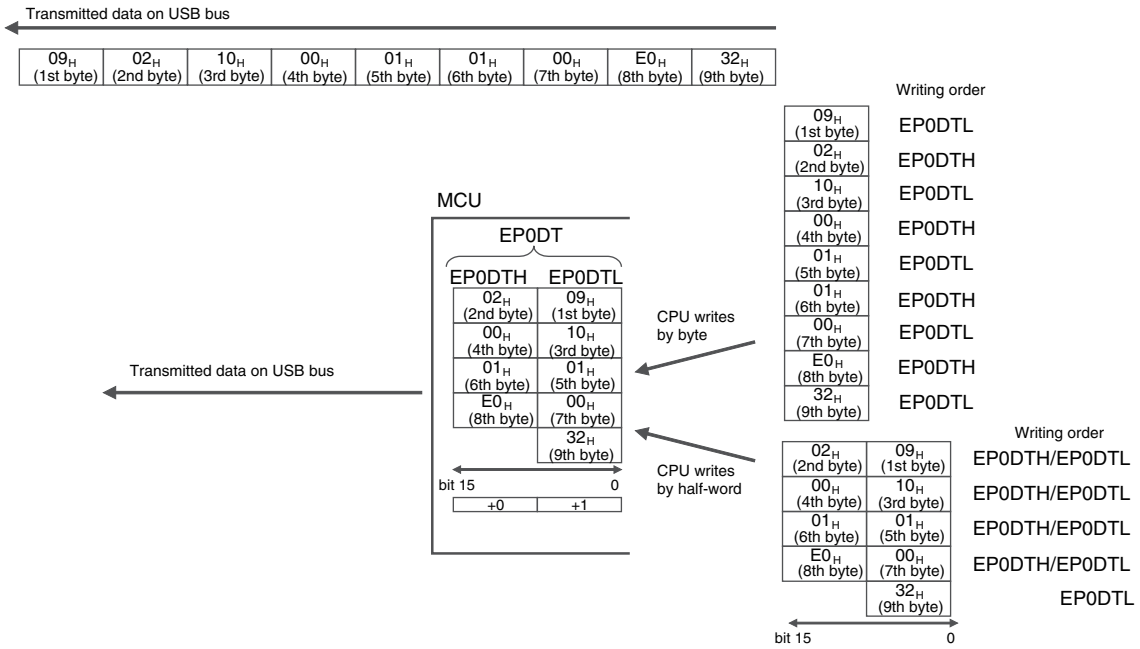
These registers are used to read data from or write data to the transmit/receive buffer of each endpoint.

Note: EP0 to EP5 data registers(EP0DTH to EP5DTH/EP0DTL to EP5DTL) are little endian. The order for storing data to these registers is from the lower data (EPxDTL) to the upper data (EPxDTH).
(x=1 to 5)
The CPU access to EP0 to EP5 data registers can be implemented by byte and half-word and by byte, the lower data (EPxDTL) is accessed first and then the upper data (EPxDTH) is accessed.
The access is continued for the lower data and the upper data alternately.
For half-word access, execute the endian transformation for the transferred data. The access to these registers with bit operation instruction is prohibited.

(Example) When transferring SETUP of get descriptor



(Example) When transferring IN of get descriptor



The CPU access to EP0 to EP5 data registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) are implemented only by half-word access. Execute the endian transformation for the transferred data (see "Automatic buffer transfer mode" in 27.4.6 DMA Transfer Function).

27.4 Explanation of Operations and Setting Procedure Examples

This section explains operations of the USB function and also shows examples of procedures for setting the operating state are shown.

Overview

The USB function performs bidirectional packet transfer between a host controller that supports the USB communication protocol. The host and USB devices are connected and configured using enumeration, and communicate by using device drivers for different types of transfer.

Note: To use the USB function, set the USB clock generation block to enable USB clock output while USB operations are disabled by a USB enable register (USBEN) setting (USBEN = 0), and then enable USB operations (USBEN = 1).

This section uses enumeration in an example to explain operations of the USB communication between the host and a USB device, and then shows operations of registers and USB packets to provide an understanding of all the processing.

- Enumeration processing

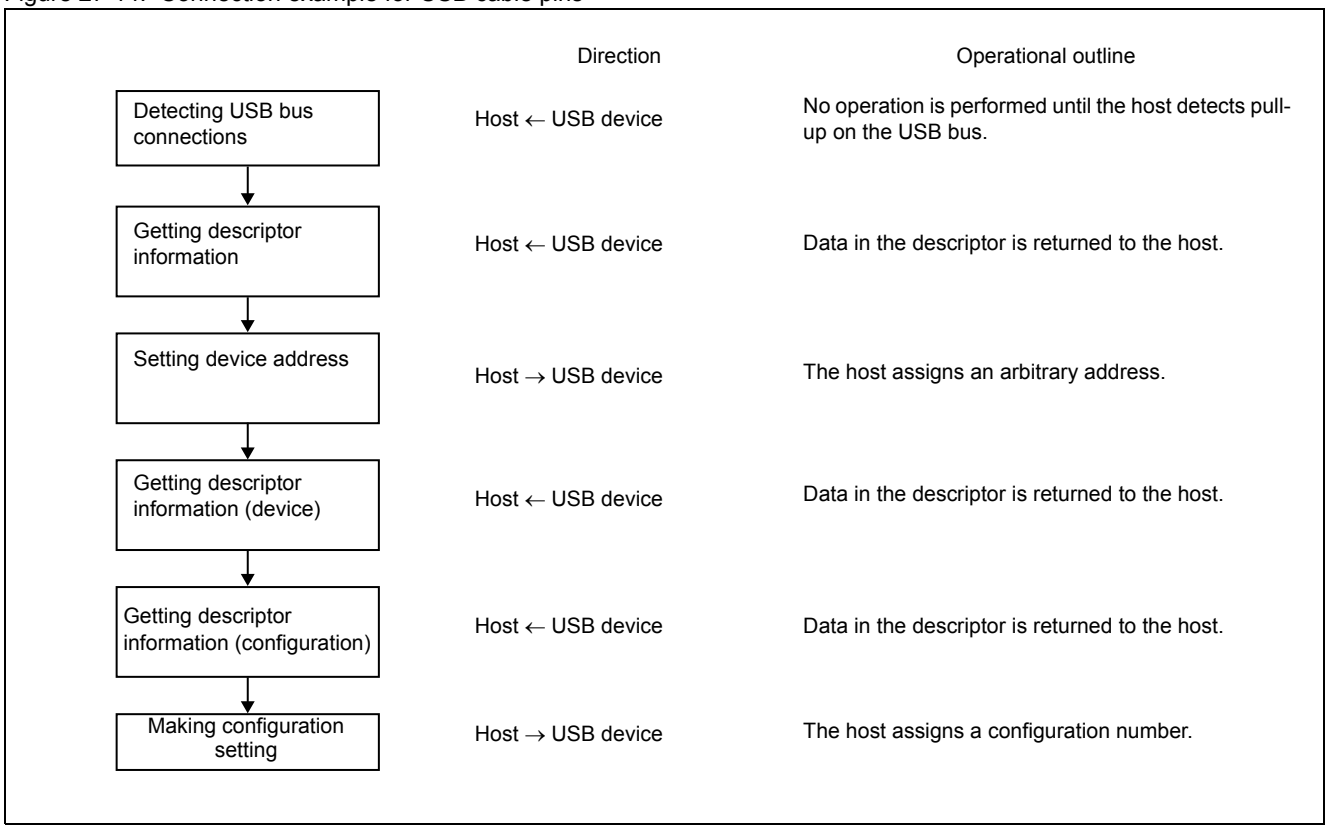
  Enumeration processing establishes the initial connection between host and the USB device during a USB operation. The host uses a USB control transfer (a type of USB transfer) to check the types of devices that have a USB connection (the procedure is specified in the USB specifications). Endpoint 0 among the six endpoints is used for this purpose (USB specification).

  To use endpoint 1 to endpoint 5 for the operation, receive the following operations on USB bus:

  1. USB bus reset
  2. Address setting by SET_Address
  3. Configuration setting by SET_Config

  Figure 27-14 shows a connection example for USB cable pins.

Figure 27-14. Connection example for USB cable pins



- Detecting connections

  The USB device notifies the host.

  The host monitors the two signal lines on the USB bus (D+, D-). If any of the signals is at the "H" level, the host recognizes that a device is connected.

  For details of the procedure for using a self-powered USB device, see "27.4.1 Detection of Connections and Disconnections". To use a bus-powered USB device, follow the instructions explained in "Making initial settings for registers and starting operation" below.

- Making initial settings for registers and starting operation

  The following procedure is an example for making initial settings for USB function registers.

  1. Make settings (packet size, etc.) for endpoint 0 by using the EP0 control register (EP0C).
  2. Set the EPEN bits, DIR bits, TYPE bits, etc. of endpoints 1 to 5 by using the EP1 to 5 control registers (EP1C to EP5C).
  3. Clear the RST bit in the UDC control register (UDCC).
  4. Clear the BFINI bits in the EP0I status register (EP0IS), EP0O status register (EP0OS), and EP1 to 5 status registers (EP1S to EP5S).
  5. Clear the HCONX bit in the UDC control register (UDCC).

- Resetting the USB bus

  The USB device core is initialized when the host generates the bus reset condition, but the conditions of the register and the buffer are not initialized.

  For the USB device, perform the following operations in the order shown (this processing is not required for resetting the bus first after a USB connection is established):

  1. Use the BFINI bit in the EP0I status register (EP0IS), the BFINI bit in the EP0O status register (EP0OS) and the BFINI bit in the EP1 to EP5 status register (EP1S to EP5S) to initialize the buffer.
  2. Restore control of the firmware to the point immediately prior to the start of enumeration.

- Getting descriptor information

  When the host sends a request to the USB device, it sends data to the host. The communication takes place over three stages: the setup stage, the data stage, and the status stage.

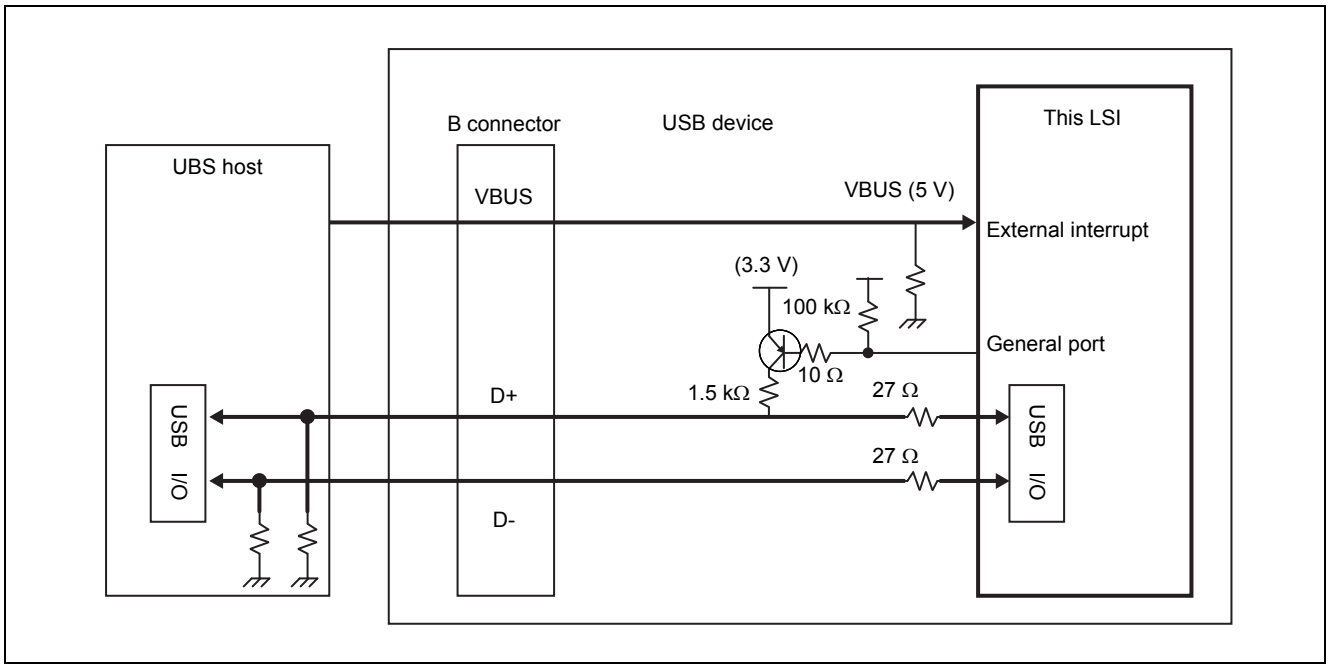
  1. Setup stage
     In this stage, packets are checked to verify that they were received normally from the host, and the received information is decoded to determine what the command is. Also, the descriptor information to be returned in the next data stage is prepared in the transmit buffer.
  2. Data stage
     Data is checked to verify that the host sent it normally.
  3. Status stage
     The host transfers a packet containing no data, and termination processing is executed.

27.4.1 Detection of Connections and Disconnections

This section explains the detection of USB host connections and disconnections.

By connecting the external interrupt pin to the VBUS pin of the USB connector and connecting a pull-down resistor to it, a disconnection from the USB host can be detected. Figure 27-15 shows an example of connections to the D+, D-, and VBUS pins of the USB connector.

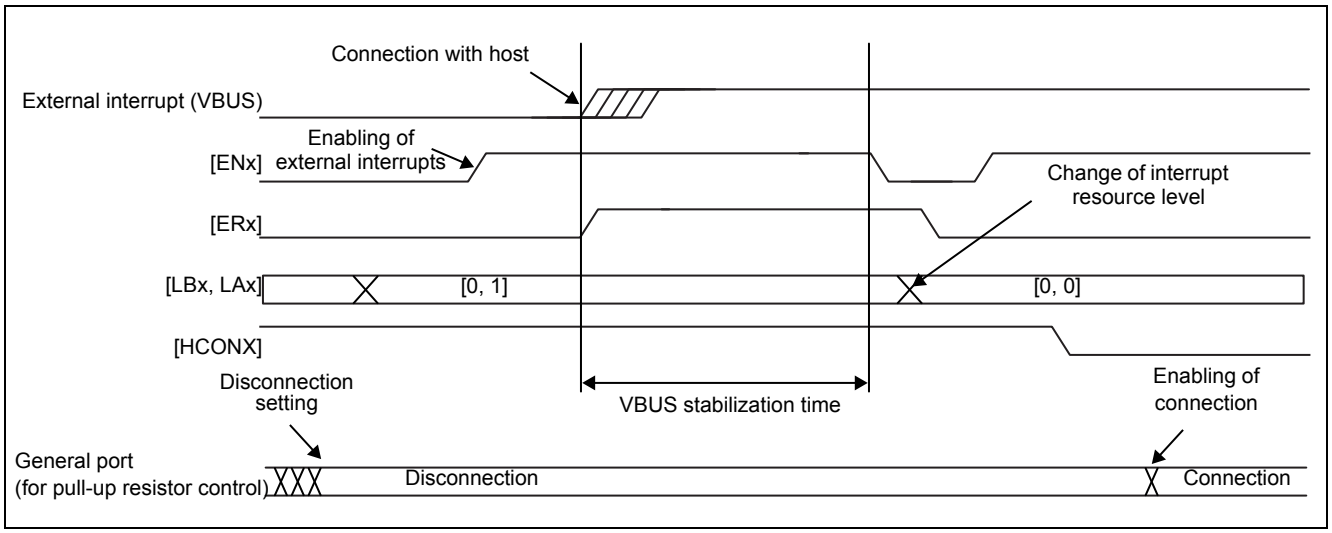
Figure 27-15. USB system configuration example



Connection detection operation

Figure 27-16 shows the operation at the connection detection time.

Figure 27-16. Operation at the connection detection time



The USB device recognizes and processes the connection to the host in the following sequence:

1. The general port for pull-up resistor control is set on the pull-up resistor disconnection side.
2. The level of the external interrupt resource connected to VBUS is set to "H" level detection, and interrupts are enabled.
3. The connection of the USB host in "H" level detection at the external interrupt pin is recognized, which is followed by a wait for the VBUS stabilization time.
4. External interrupts are disabled once.

   The external interrupt resource level is changed to "L" level detection, and the interrupt resource is cleared to enable external interrupts again.
5. Initial settings are made (all initialization is performed including that for the USB function registers).

   For details, see "Making initial settings for registers and starting operation" in this "Overview" section.
6. After the HCONX bit in the UDC control register (UDCC) is cleared, the general port pull-up resistor control is set on the pull-up resistor connection side to connect the pull-up resistor of D+ is connected.

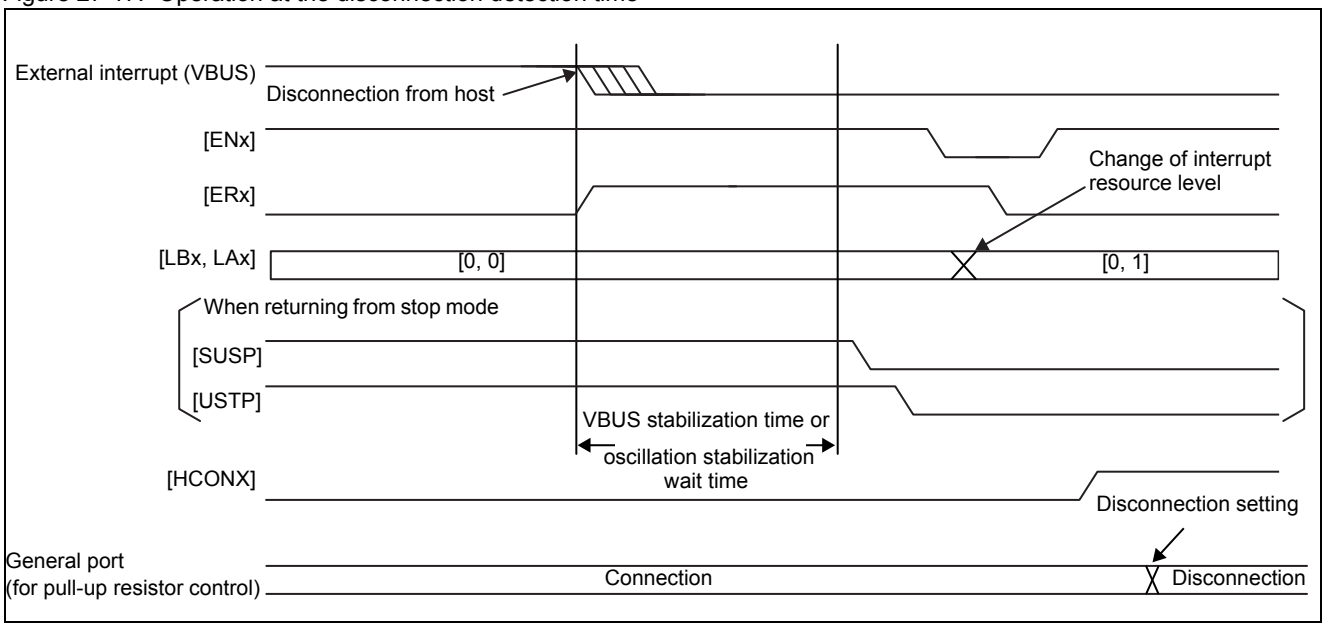
   Even if the pull-up resistor is not controlled, clear the HCONX bit.

Note: If an external noise filter is connected to the external interrupt pin, the program need not wait for the above VBUS stabilization time.

Disconnection detection operation

Figure 27-17 shows the operation at the disconnection detection time.

Figure 27-17. Operation at the disconnection detection time



The USB device recognizes and processes the disconnection from the host in the following sequence:

1. The "L" level of the external interrupt pin connected to VBUS is detected, and the USB host is recognized as disconnected.
2. After the return from stop mode and the wait for the oscillation stabilization wait time, the SUSP bit in the UDC status register (UDCS) and the USTP bit in the UDC control register (UDCC) are cleared in this order.

   In any mode other than stop mode, there is a wait for the VBUS stabilization time.
3. External interrupts are disabled once.

   The external interrupt resource level is changed to the "H" level detection, and the interrupt resource is cleared to enable external interrupts again.
4. After "1" is written to the HCONX bit in the UDC control register (UDCC), the general port for pull-up resistor control is set on the pull-up resistor disconnection side to disconnect the pull-up resistor of D+.

   Even if the pull-up resistor is not controlled, write "1" in the HCONX bit.

Note: If an external noise filter is connected to the external interrupt pin, the program need not wait for the above VBUS stabilization time.

27.4.2 Register Operation for a Command Response

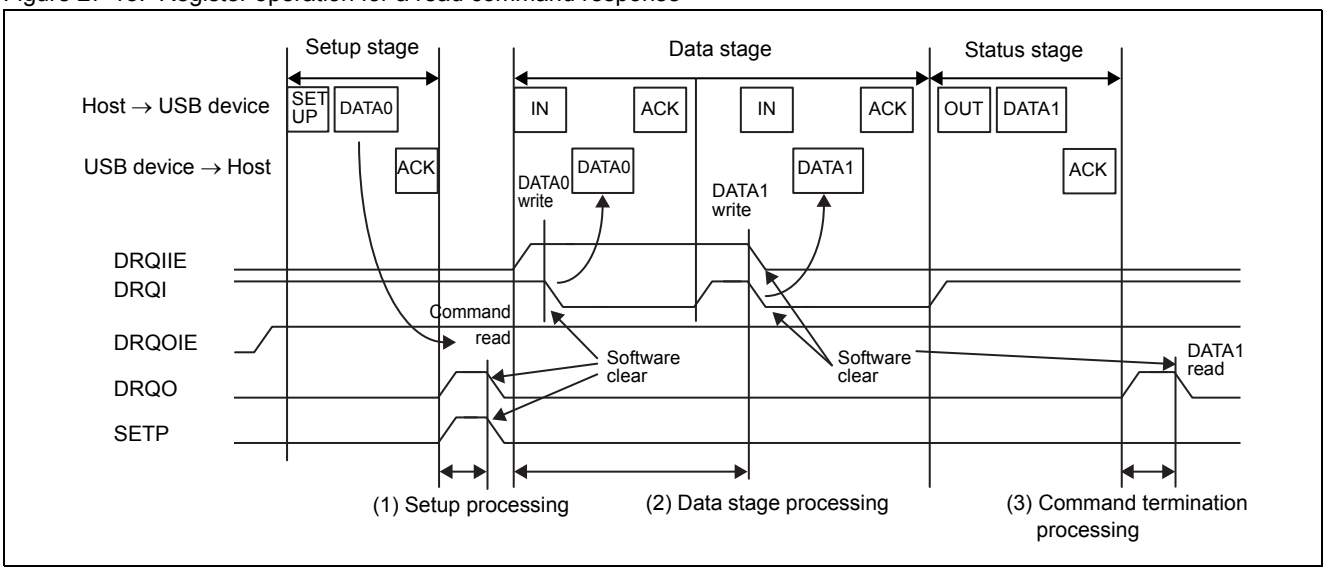
This section explains how to process USB packets through the operation and control of basic registers (architecture).

Firmware processing for a CPU interrupt is executed for each handshake. This is equivalent in meaning to executing the processing in each stage of individual packets.

Register operation for a read command response

Figure 27-18 shows the register operation for a read command response (in the case of the GetDescripter or SynchFrame command or any of the class vendor commands).

Figure 27-18. Register operation for a read command response



- Setup processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" when data is received for the setup stage. CPU interrupt processing begins when the DRQO bit becomes "1". The SETP bit in the UDC status register (UDCS) is verified. If the SETP bit is "1", as many commands as necessary in the receive buffer are read (it is not always necessary to read all 8 bytes) and decoded, various setup processing is executed, the SETP bit and DRQO interrupt resource bit are cleared, and then the processing returns.

- Data stage post-processing

  If the command decoding results indicate that the data stage is in the IN direction, "1" is written to the DRQIE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S) to enable interrupts (which is the initial value of the DRQI bit in the EP0I status register (EP0IS), which is an interrupt resource, so "1" is just written to the DRQIE bit). Then, a CPU interrupt is used to transfer transmit data to the transmit buffer. The DRQI interrupt resource bit is cleared after the transfer is completed, and then the processing returns.

  The DRQI bit becomes "1" at the completion of the IN-direction data packet processing. CPU interrupt processing is entered when the DRQI bit becomes "1", and transmit data is transferred to the transmit buffer in preparation for the next data packet. The DRQI interrupt resource bit is cleared after the transfer is completed, and then the processing returns.

  Note: This USB function is not compatible with the newly added commands for USB 2.0; for GetDescripter command, respond with USB 1.1.

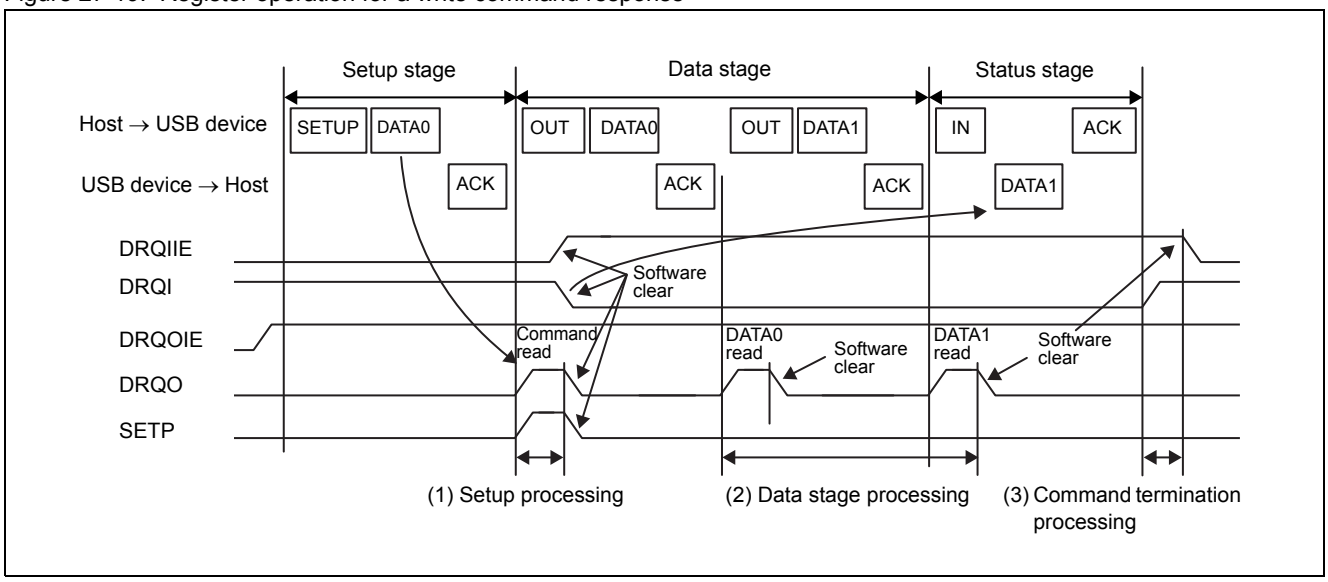
- Command termination processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" upon completion of the OUT-direction status stage. CPU interrupt processing is entered when the DRQO bit becomes "1", then the received data count is confirmed to be 0, the DRQO interrupt resource bit is cleared in preparation for the next setup stage, and then the processing returns.

Register operation for a write command response

Figure 27-19 shows the register operation for a write command response (in the case of the SetDescripter command or any of the class vendor commands).

Figure 27-19. Register operation for a write command response



- Setup processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" when data is received for the setup stage. CPU interrupt processing begins when the DRQO bit becomes "1". The SETP bit in the UDC status register (UDCS) is verified. If the SETP bit is "1", as many commands as necessary in the receive buffer are read (it is not always necessary to read all 8 bytes) and decoded, and then various setup processing is executed.

  In preparation for a 0-byte response in the status stage, without writing any data in the transmit buffer, "0" is written to the DRQI bit in the EO0I status register (EP0IS) to clear it (because the initial value of the PRQI interrupt resource bit is "1"). For confirmation of the normal end of the status stage, "1" is written to the DRQIIE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S). Also, the SETP bit and DRQO interrupt resource bit are cleared, and then there is a return from the interrupt.

- Data stage post-processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" upon completion of the OUT-direction data stage. CPU interrupt processing begins when the DRQO bit becomes "1". First, the SIZE bit in the EP0O status register (EP0OS) is verified, and DMA is activated with a specified data count corresponding to the received data count, or CPU read is used to read data from the receive buffer. Then, the DRQO interrupt resource bit is cleared, and then there is a return from the interrupt.

- Command termination processing

  The DRQI bit in the EP0I status register (EP0IS) becomes "1" upon completion of the IN-direction status stage. CPU interrupt processing begins when the DRQI bit becomes "1", and the normal end of the status stage is confirmed. Then, the DRQI interrupt resource bit is cleared, and then the processing returns.

27.4.3 STALL Response and Release

For Endpoint0 and For Endpoints 1 to 5, this section explains STALL response and release procedures.

STALL response and release procedures for Endpoint0

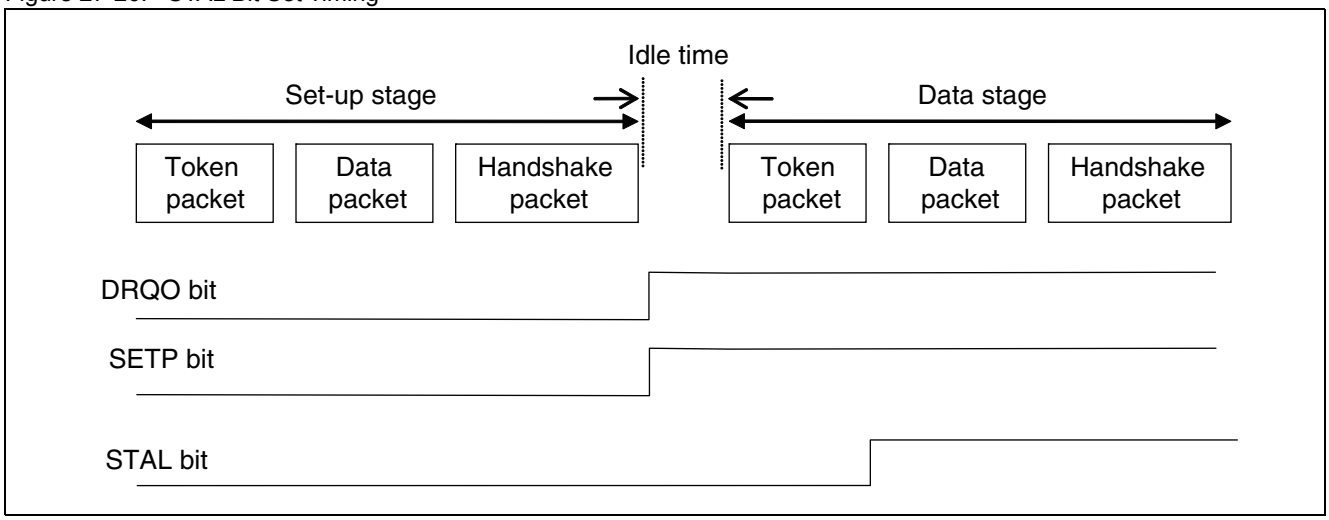
STALL response and release procedures for Endpoint0 are executed with STAL bit of EP0 Control Register (EP0C).

- Set timing of STAL bit

  For STALL response, interprets the command at detecting SETP bit of "1" (DRQO bit = 1 for interrupt) that indicates the set-up stage of control transfer. (See Figure 27-20)

  After setting STAL bit, clear interrupt cause (DRQO bit).

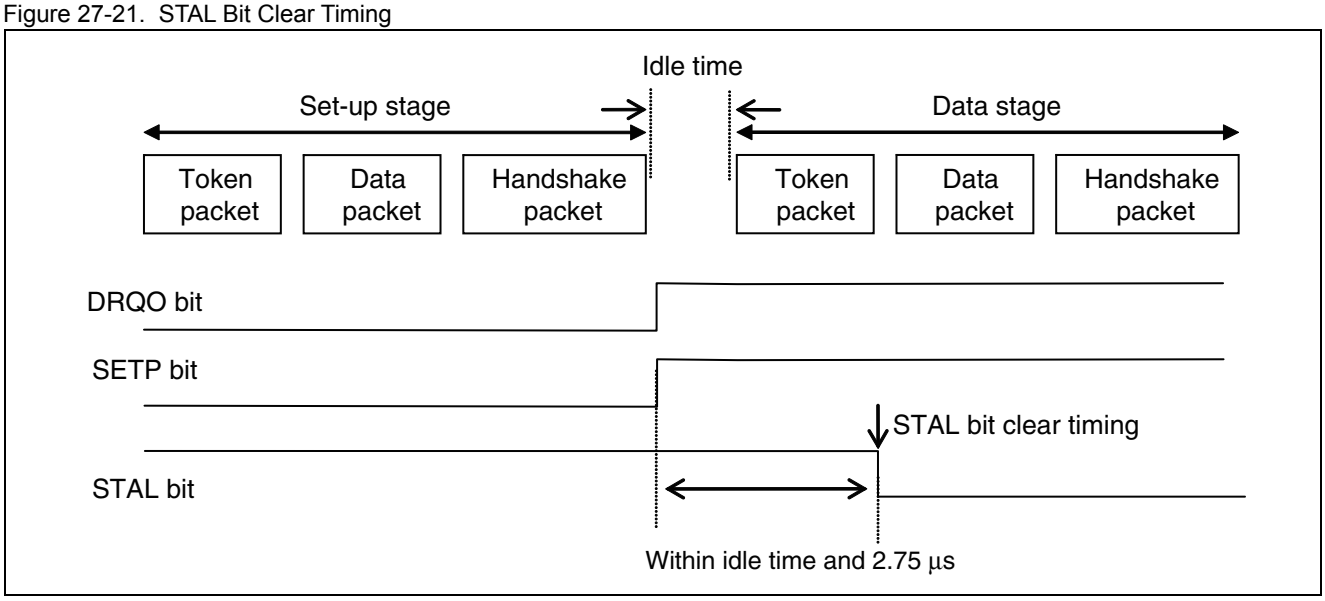
Figure 27-20. STAL Bit Set Timing



- STAL Bit Clear Timing

  For STALL release, clears STAL bit at detecting SETP bit of "1" (DRQO bit = 1 for interrupt) that indicates the set-up stage of control transfer, and sets STAL bit if the STALL response is required. (See Figure 27-21)

Figure 27-21. STAL Bit Clear Timing



For STALL response release (STAL bit clear), clear STAL bit the period between the time when STEP bit of "1" (DRQ0 bit= 1 for interrupt) is detected and the time when the data packet transmission/reception of the next data stage is started. The period between the time when DRQ0 becomes "1" and the time when STAL bit is cleared is as follows: (Transfer speed: at Full speed of 12Mbps) When STAL bit is not cleared in the following period, execute STAL response with the handshake of data stage.

The period between the time when DRQ0 BIT of "1" is detected and the time when STAL bit is cleared: within idle time + 2.75 μs

Note: When idle time is the shortest period of 2-bit transfer time, the above period is within about 2.9 μs.

If the STAL bit clear cannot executed within the above period, take an appropriate countermeasures such as lengthening of the idle time with a driver of USB HOST.

STALL response /release of Endpoints 1 to 5

STALL response /release of Endpoints 1 to 5 are controlled with Control registers of EP1 to EP5 and internal condition bit

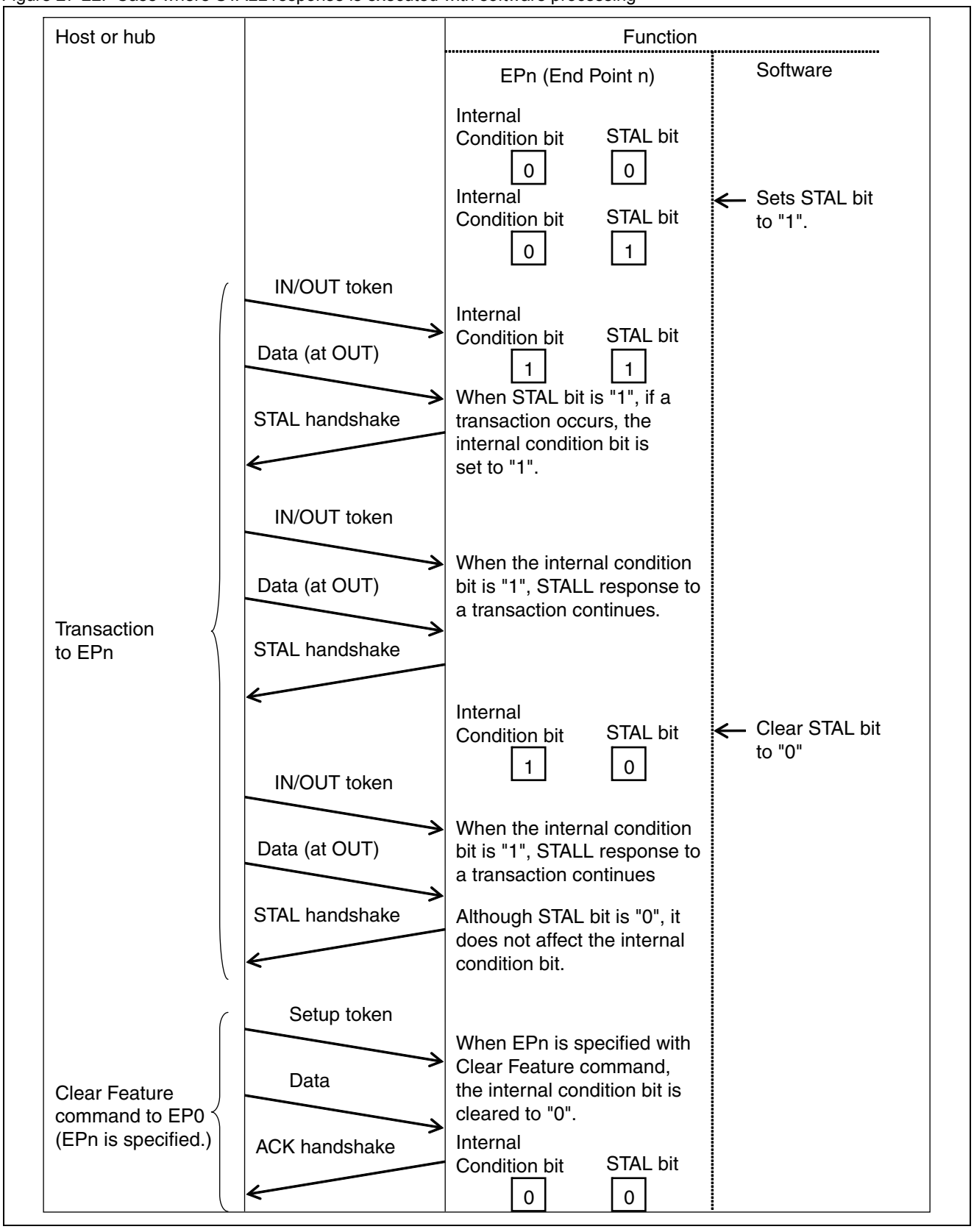
- To execute STAL response with software

  The procedures to execute STAL response with software are shown in Figure 27-22. To execute STAL response, set STAL bit of the relevant endpoint with software. In this time, the internal condition bit does not change. Furthermore, when a host
  generates a transaction to the endpoint where STAL bit is set, hardware would automatically set the internal condition bit of the relevant endpoint and gives STALL response to the host.

  Once the internal condition bit is set, the internal condition bit has been set and continues STAL response until Clear Feature command is issued from the host despite of the clearing of STAL bit.

  As long as the STAL bit is set, STAL bit response continues even if the internal condition bit is cleared with Clear Feature command because the internal condition bit is set every time a transaction to the relevant endpoint occurs. Therefore, to release the STAL response, be sure to clear STAL bit and the internal condition bit with the Clear Feature command.

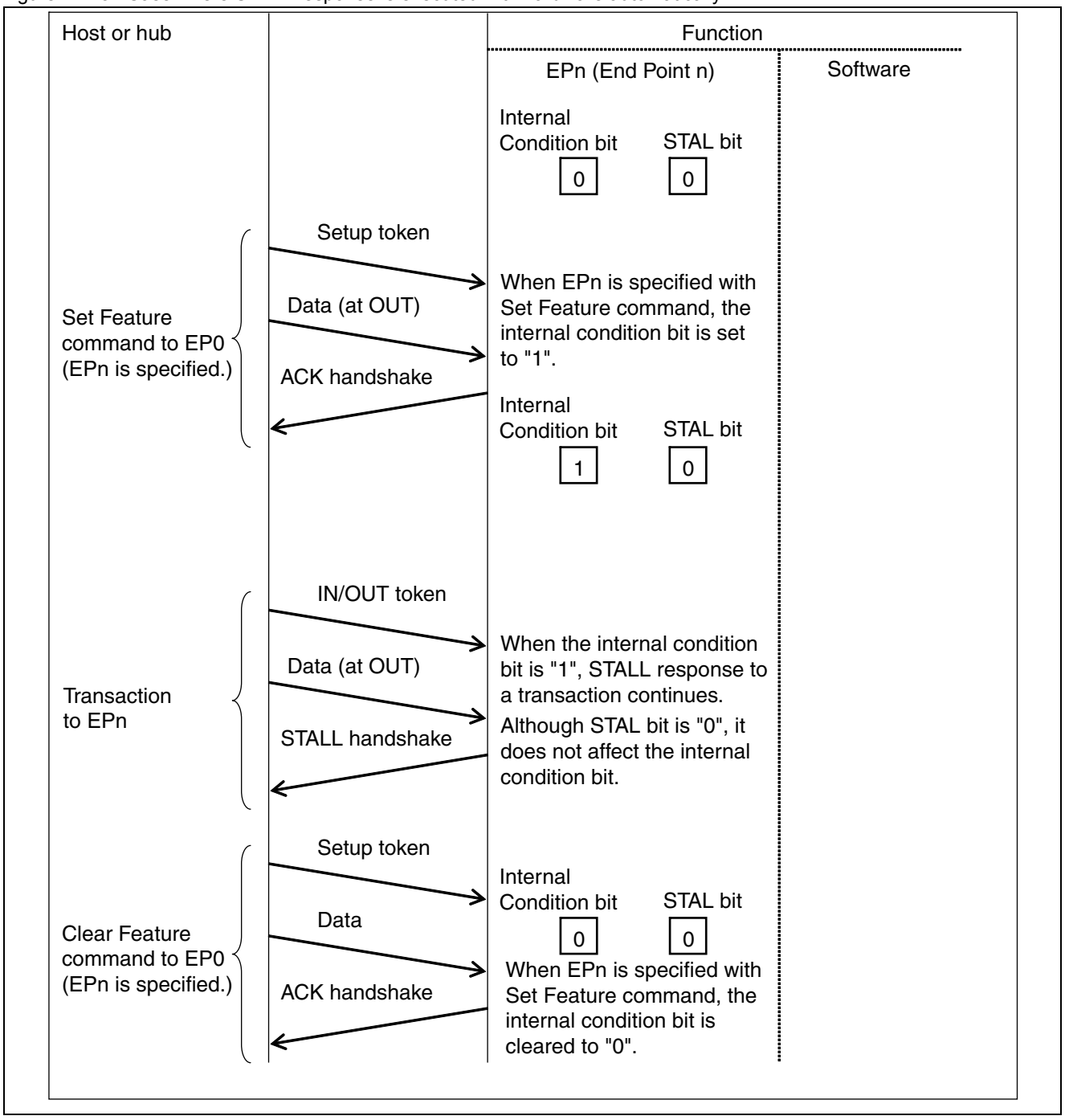
Figure 27-22. Case where STALL response is executed with software processing



- To automatically execute STALL response with hardware.

  The procedures to execute STAL response with hardware are shown in Figure 27-23. When STALL response is set with Set Feature command, the hardware would automatically set the internal condition bit of the relevant endpoint and gives the STALL response regardless of the STAL bit. Once the internal condition bit is set, the internal condition bit has been held until Clear Feature command is issued from the host to clear the internal condition bit regardless of STAL bit. After the relevant bit is cleared with Clear Feature command, STAL bit is referenced. Therefore, to clear STAL response, be sure to clear the internal condition bit with Clear Feature command.

Figure 27-23. Case where STALL response is executed with hardware automatically



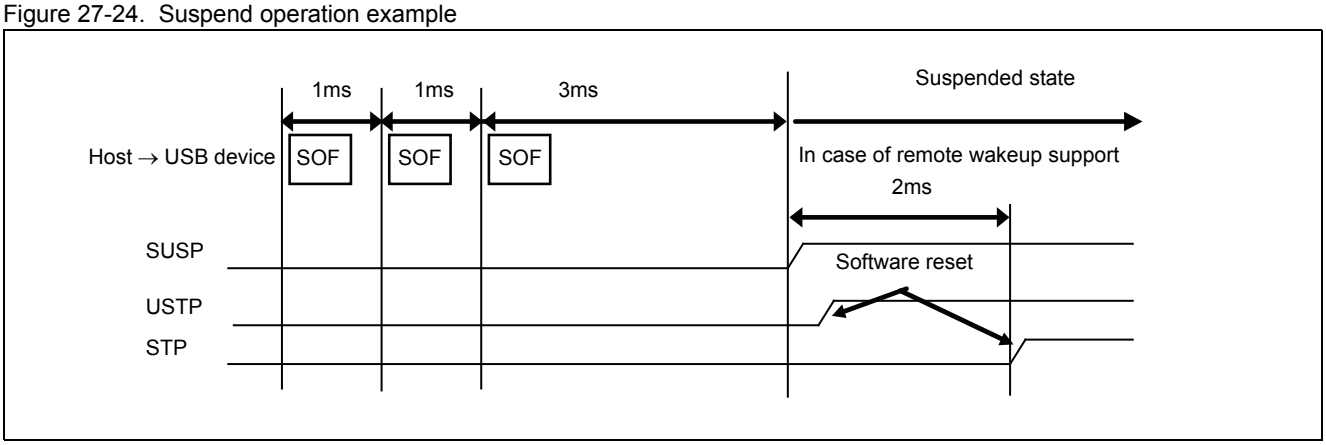
27.4.4 Suspend Function

This section explains the operation that begins at the time of USB device transition to the suspended state and continues until the USB device enters standby mode (watch mode or stop mode). In a bus-powered configuration, set the power dissipation of the USB device in the suspended state at 500 μA or less.

The SUSP bit in the UDC status register (UDCS) becomes "1" when the USB device core detects the suspended state.

Figure 27-24 shows the suspend operation.

Figure 27-24. Suspend operation example



- Suspend processing

  When no operation has occurred for a period equal to or longer than 3 ms on the USB bus, the USB function detects the suspend condition, the SUSP bit in the UDC status register (UDCS) becomes "1", and the interrupt resource is set. If the device supports remote wakeup, there is a wait for 2 ms from this point in time (preventing remote wakeup during this period), and standby mode (watch mode or stop mode) is set.

27.4.5 Wakeup Function

The USB protocol has the following two methods for the USB device transition from the suspended state to the wakeup state:

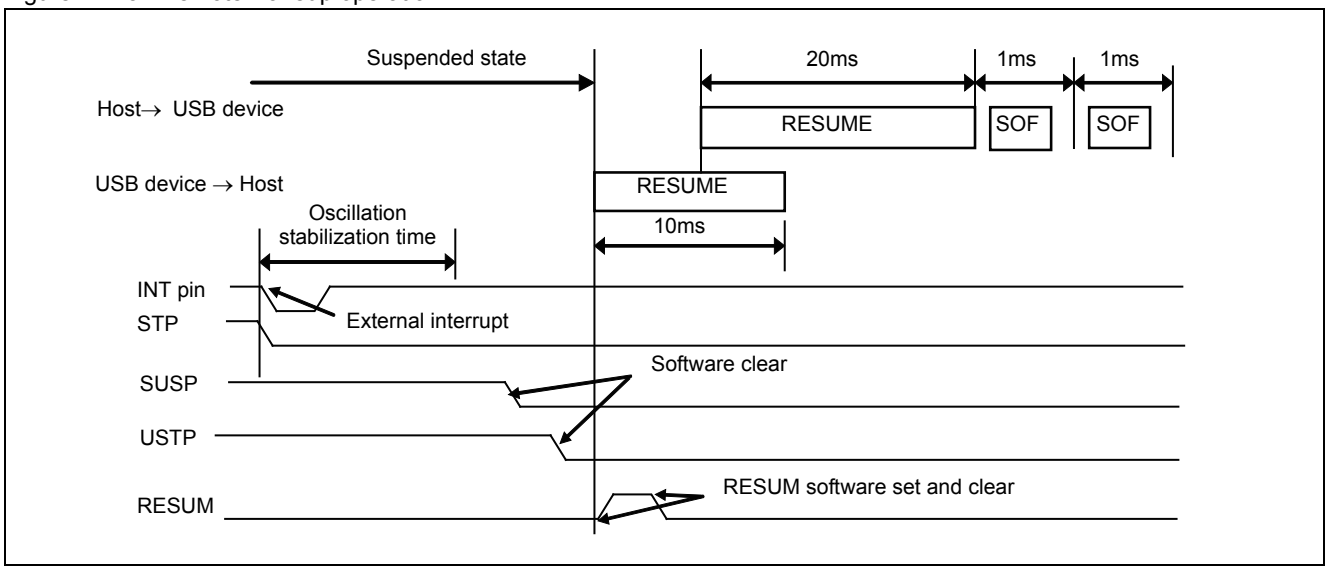
- Remote wakeup by the USB device
- Wakeup by the host

This section explains their operations.

Remote wakeup operation

Figure 27-25 shows the remote wakeup operation.

Figure 27-25. Remote wakeup operation



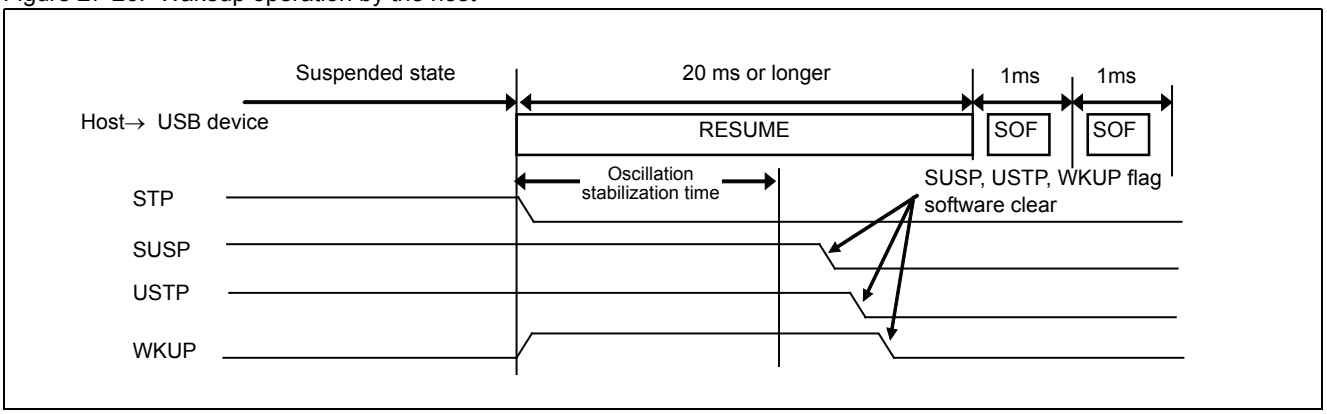
The USB device must perform the following procedure:

1. Use an external interrupt to make the USB device return from stop mode.
2. Write "0" in the SUSP bit of the UDC status register (UDCS) to clear it.
3. Write "0" in the USTP bit of the UDC control register (UDCC) to clear it.
4. Write "1" in the RESUM bit in the UDC control register (UDCC).
5. Write "0" in the RESUM bit in the UDC control register (UDCC) to clear it.

Wakeup operation by the host

Figure 27-26 shows the wakeup operation by the host.

Figure 27-26. Wakeup operation by the host



The USB device must perform the following procedure:

1. Set an oscillation stabilization time that does not exceed 10 ms.
2. Write "0" in the SUSP bit of the UDC status register (UDCS) to clear it.
3. Write "0" in the USTP bit of the UDC control register (UDCC) to clear it.
4. Write "0" in the WKUP bit in the UDC status register (UDCS), which is the interrupt resource, to clear it.

27.4.6 DMA Transfer Function

Data communicated by the USB function can be transferred by DMA between the transmission/reception buffer and built-in RAM. One of the following two modes can be selected for DMA transfer:

- Packet transfer mode
- Automatic buffer transfer mode

This section explains the two DMA transfer modes.

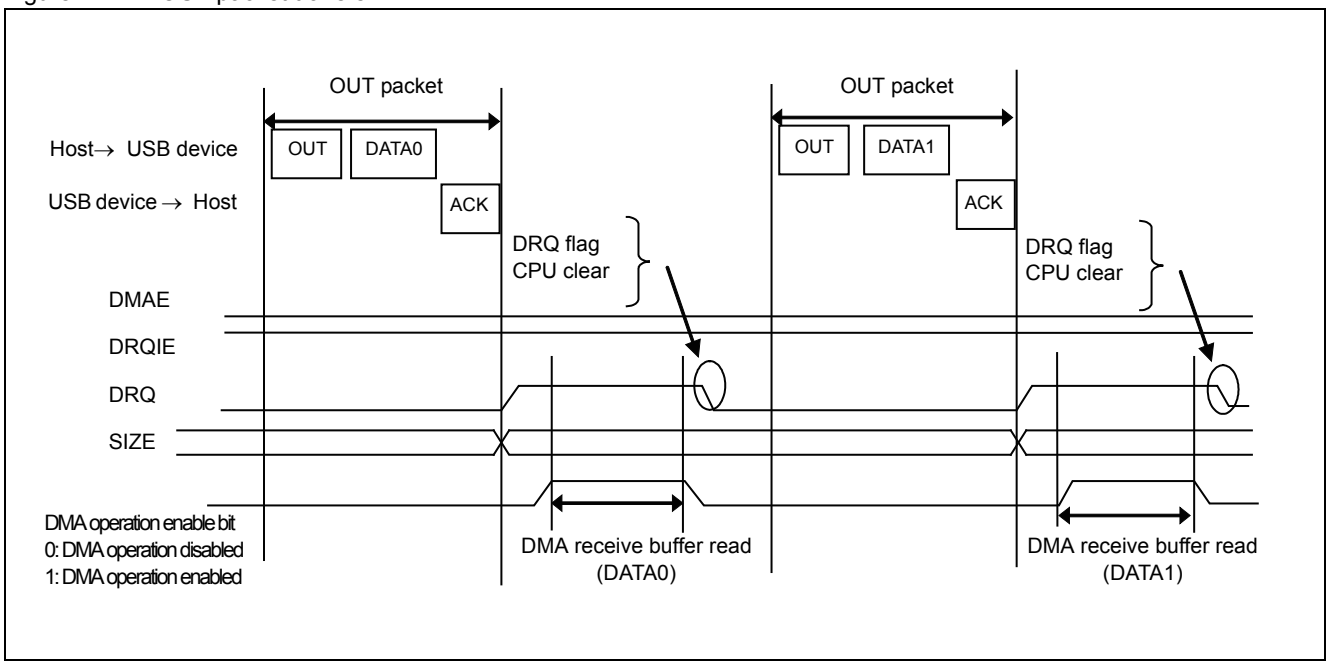
Packet transfer mode

In packet transfer mode, the transfer count is set in DMA for each packet, and the interrupt resource is cleared after a DMA transfer is completed. In this mode, the buffers of endpoints 1 to 5 can be accessed. Set DREQ with DREQ selection register (DREQSEL) to use DMA.

- OUT-direction operation

  Figure 27-27 shows the buffer access timing in an OUT-direction transfer (host to USB device).

Figure 27-27. OUT packet transfer



For an OUT-direction transfer, the USB device must perform the following procedures:

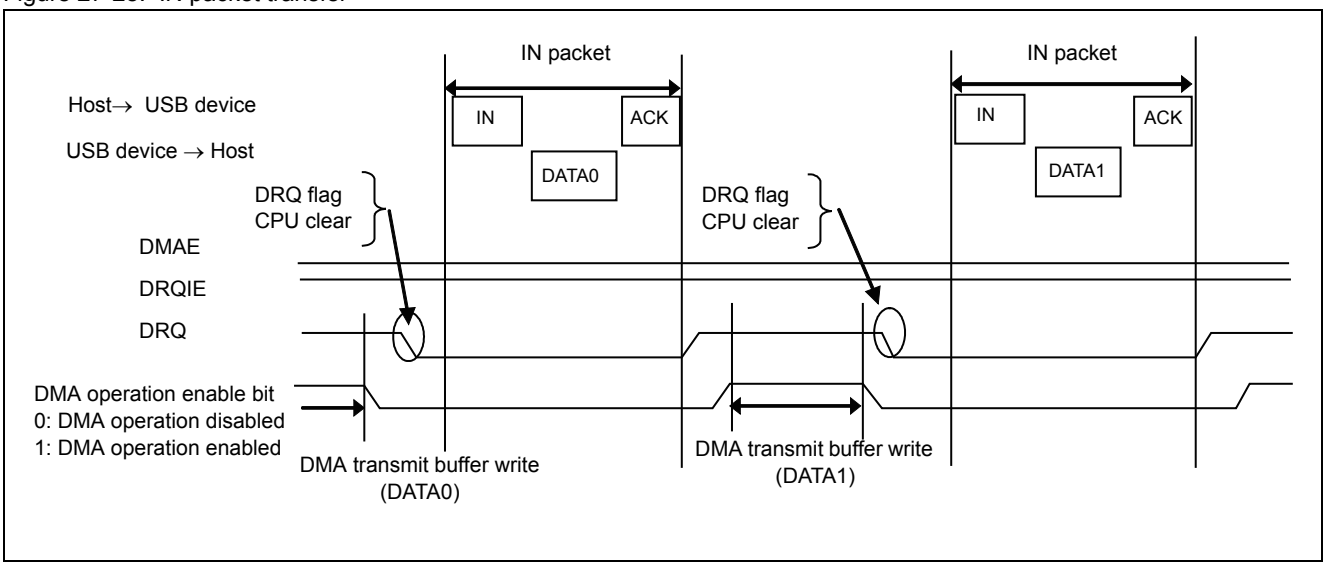
1. Verify the transfer data count after the DRQ flag becomes "1" and interrupt processing begins.
2. Set the block-size-related DMA registers and a transfer count corresponding to the amount of data to be transferred, and enable DMA to start the transfer.
3. Clear the following flags (bits) of registers after the transfer:
   a. DRQ bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S)
   b. Interrupt resource flag of the relevant register among the DMA channel status registers (DCSR0 to DCSR7) of the DMA controller (DMAC)

   Return from the interrupt processing.

- IN-direction operation

  Figure 27-28 shows the buffer access timing in an IN-direction transfer (USB device to host).

Figure 27-28. IN packet transfer



For an IN-direction transfer, the USB device must perform the following procedure:

1. After the DRQ flag becomes "1" and interrupt processing begins, set the block-size-related DMA registers and a transfer count corresponding to the amount of data to be transferred in the next IN packet, and enable DMA to start the transfer.
2. Clear the following flags (bits) of registers after the transfer:
   a. DRQ bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S)
   b. Interrupt resource flag of the relevant register among the DMA channel status registers (DCSR0 to DCSR7) of the DMA controller (DMAC)

   Return from the interrupt processing.

Automatic buffer transfer mode

In automatic buffer transfer mode, specified data count is transferred at one time according to the setting.

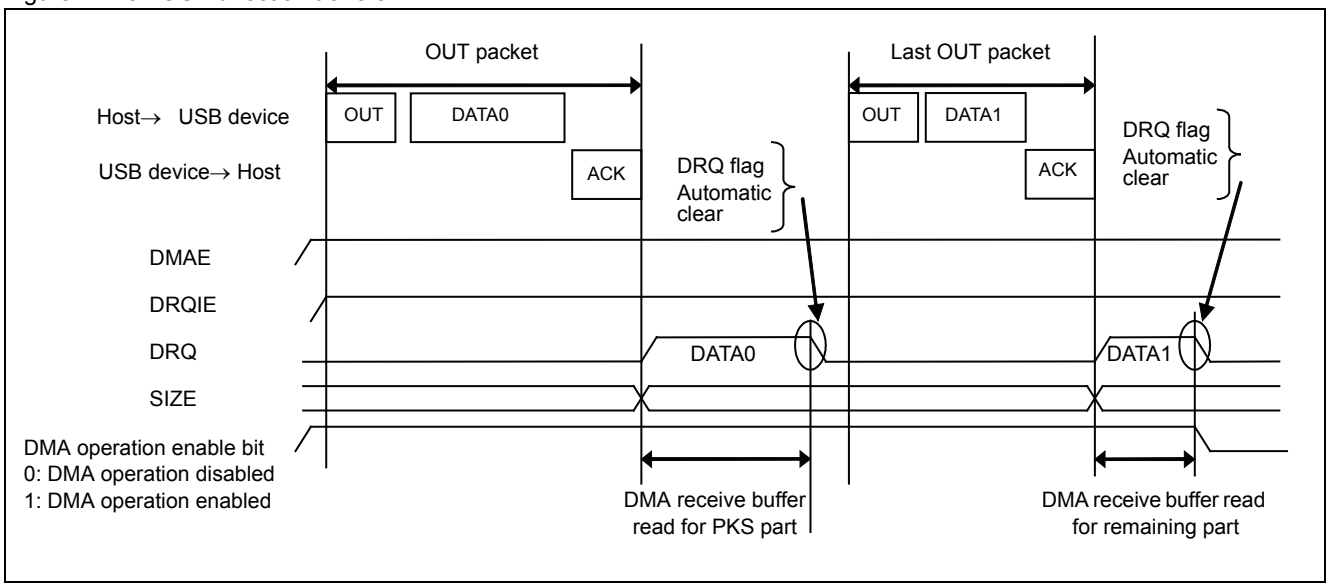
In this transfer mode, even-number bytes can be transferred. To transfer the odd-number bytes, CPU transfer processing is necessary. Before using DMA transfer, set DREQ with DREQ selection register.

Set the total transfer data count in DMA beforehand, and also set the transfer enable bit. If automatic buffer transfer mode is enabled (DMAE = 1) when the DRQ flag is set after a transfer from the host, data count specified by the PKS bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) is transferred. Then, the interrupt resource is automatically cleared. After the transfer from the host, similar processing is repeated as many times as the specified transfer data count set in DMA. During this processing, no setting by the CPU is required, and this mode transfers data at one time according to the setting. To perform the subsequent transfer in a case where a CPU interrupt is entered after the last data transfer, make DMA controller (DMAC) settings again, enable DMA, and return from the interrupt processing. Since the USB function is used with the setting of DMAE = 1 in this mode, only buffer access for endpoints 1 to 5 is valid.

- OUT-direction operation

  Figure 27-29 shows the buffer access timing in an OUT-direction transfer (host to USB device).

Figure 27-29. OUT-direction transfer



For an OUT-direction transfer, the USB device must perform the following procedures:

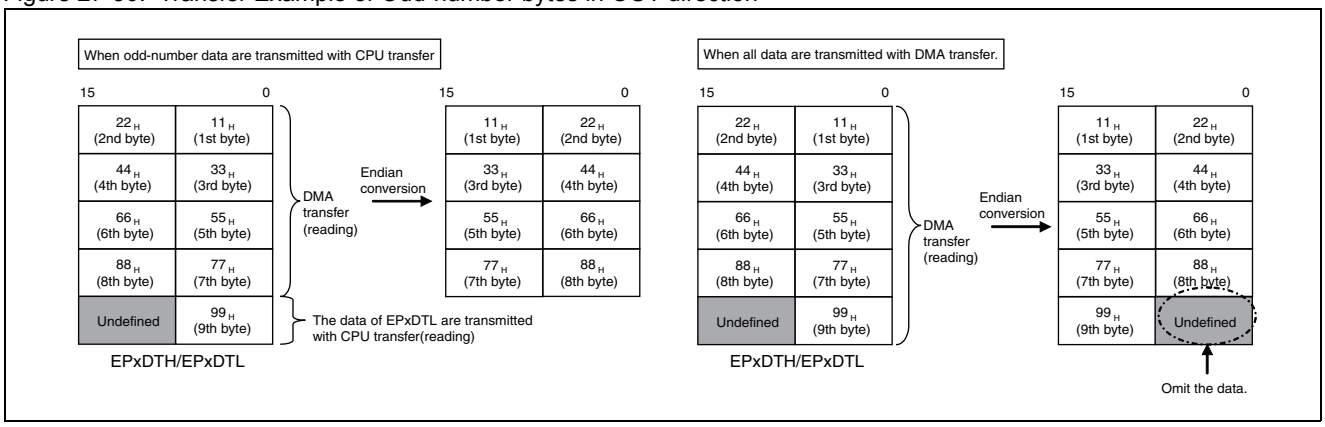
1. Set the block-size-related DMA registers and a transfer count corresponding to the total amount of data to be transferred, and enable DMA to start the transfer.
2. Write "1" in the DMAE bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) to set automatic buffer transfer mode, and then write "1" in the DRQIE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S) to enable interrupts.
3. Clear the relevant interrupt resource flag in the DMA channel status register (DCSR0 to DCSR7) of the DMA controller (DMAC) after the transfer, and make the DMAC setting again as necessary.

   Return from the interrupt processing.

For transmitting odd-number data with DMA transfer, the following two methods are available.

- For the last data, execute the CPU transfer and read the lower byte (EPxDTL).
- Transmit all data and one byte data together with DMA transfer and omit the last data after endian conversion.

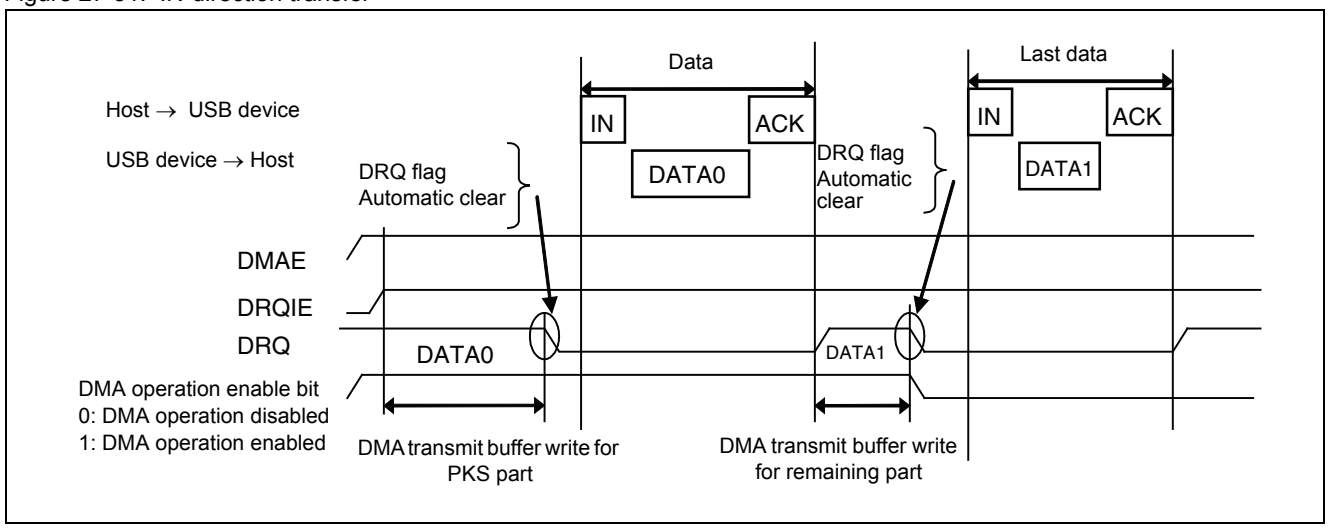
Figure 27-30. Transfer Example of Odd-number bytes in OUT direction



- IN-direction operation

  Figure 27-31 shows the buffer access timing in an IN-direction transfer (USB device to host).

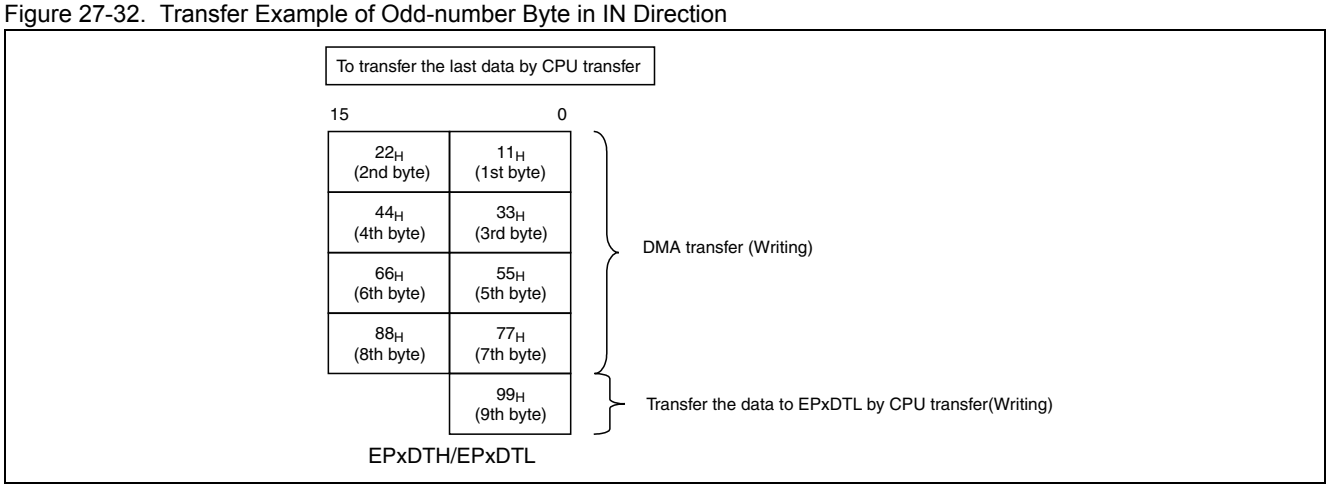
Figure 27-31. IN-direction transfer



For an IN-direction transfer, the USB device must perform the same operations as for an OUT-direction transfer. Referring to "OUT-direction operation", execute the processing in the same sequence.

For transmitting odd-number data with DMA transfer, transfer the last data with CPU transfer and write them in the lower byte (EPxDTL).

Figure 27-32. Transfer Example of Odd-number Byte in IN Direction

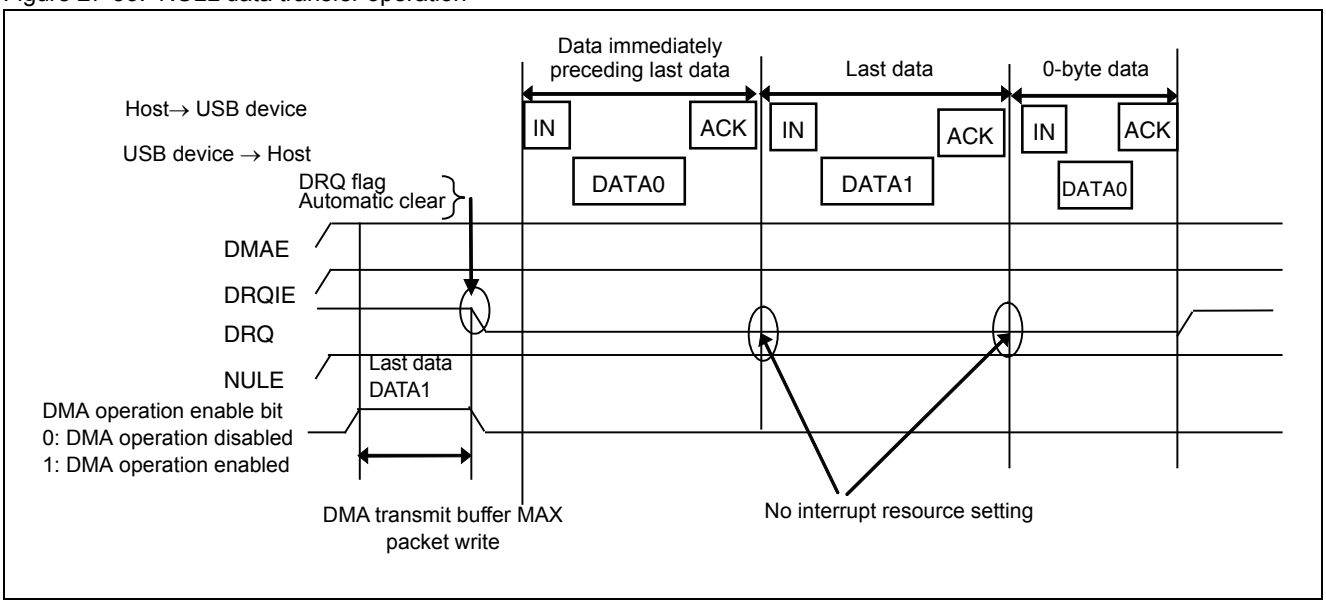


27.4.7 NULL Transfer Function

This section explains NULL transfer mode. In NULL transfer mode, if the number of data bytes transferred from the USB function reaches the max. packet count in the last packet transferred, a 0-byte transferred is automatically performed in the transfer of the next packet.

In NULL transfer mode, a transfer is performed under the following conditions: automatic buffer transfer mode is set (DMAE = 1 in the relevant register among the EP1 to 5 control registers (EP1C to EP5C)), and an IN-direction data transfer request has been generated. Under these conditions, if the DMA data count for the last data write becomes 0 when DMA write with the maximum transfer bytes is completed, a 0-byte data transfer is automatically set upon receipt of the last IN-direction data transfer request from the host. Then, 0-byte data is automatically transmitted for the next IN-direction data transfer request. After the last data is written to the buffer with DMA and until the 0-byte data is read from the host, the DRQ interrupt request flag is not set. Figure 27-33 shows the buffer access timing.

Figure 27-33. NULL data transfer operation



Note: The NULL transfer function is valid only for an IN transfer from the USB device to the host.

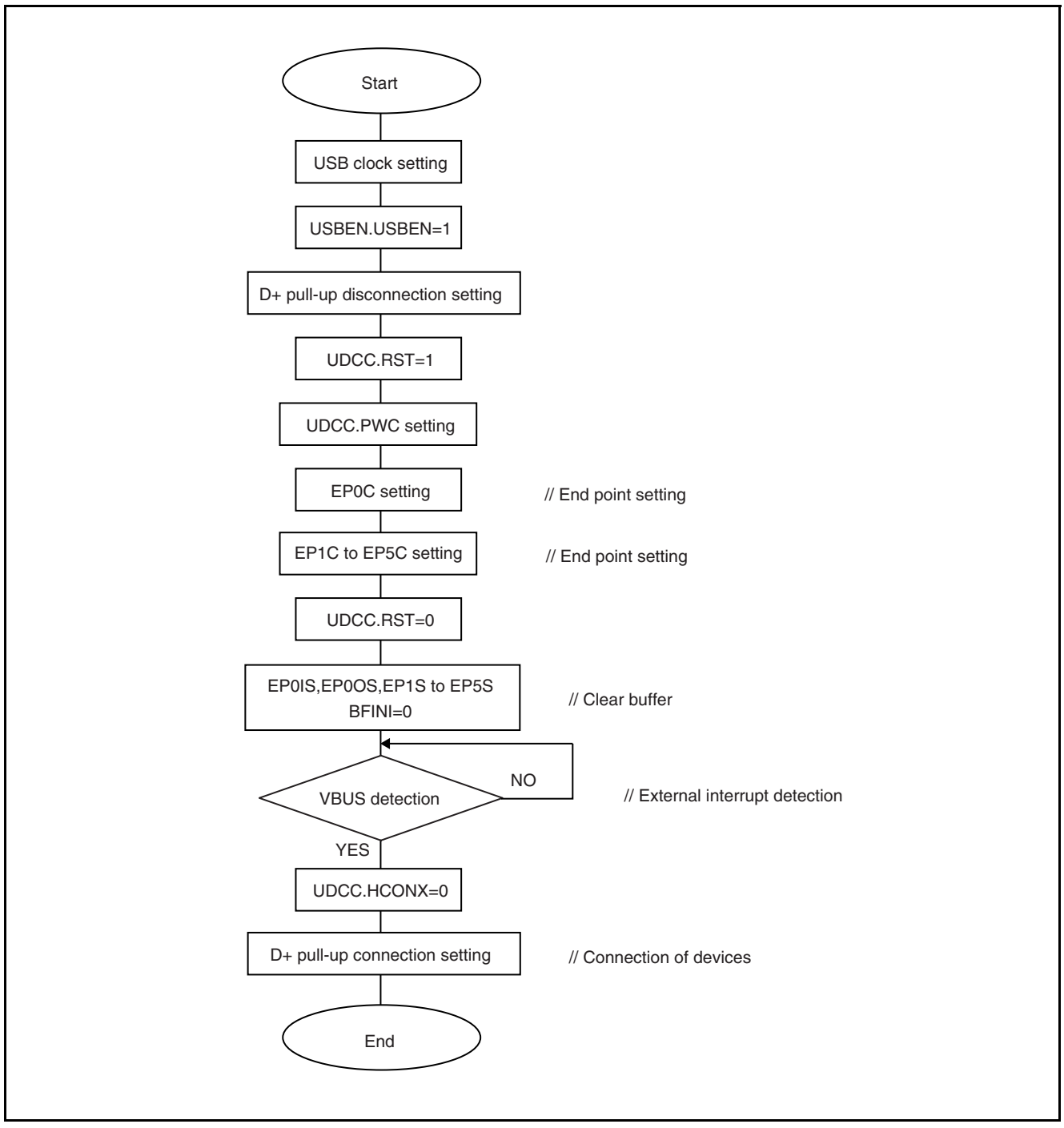
The USB device must perform the following procedure:

1. Write "1" in the DMAE bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C). Automatic buffer transfer mode is set.
2. Write "1" in the DRQIE bit and NULE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S). Interrupts are enabled, and NULL automatic transfer mode is set.

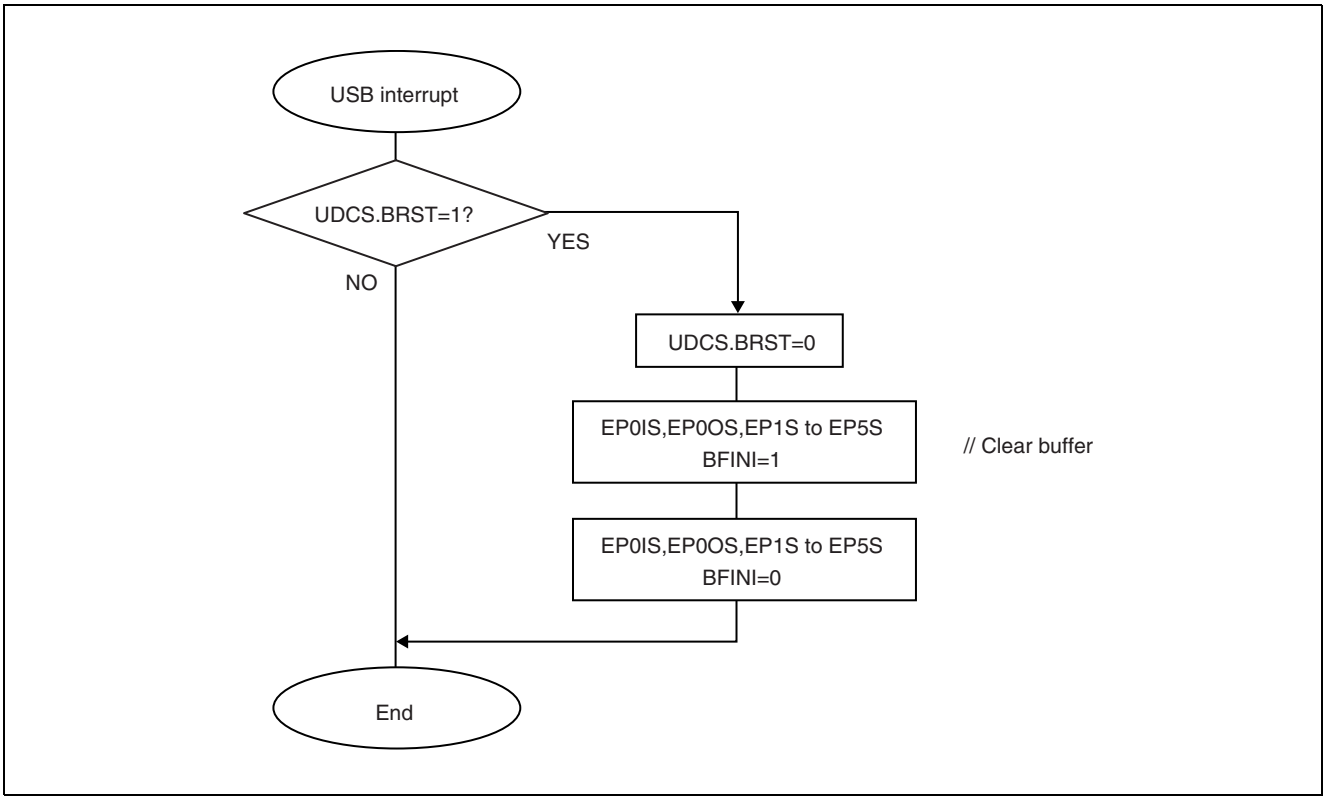
27.4.8 Software Control Examples

This section shows flowcharts for initialization, a bus reset, CPU transfers, packet transfer mode (IN/OUT), and automatic buffer transfer mode (IN/OUT).

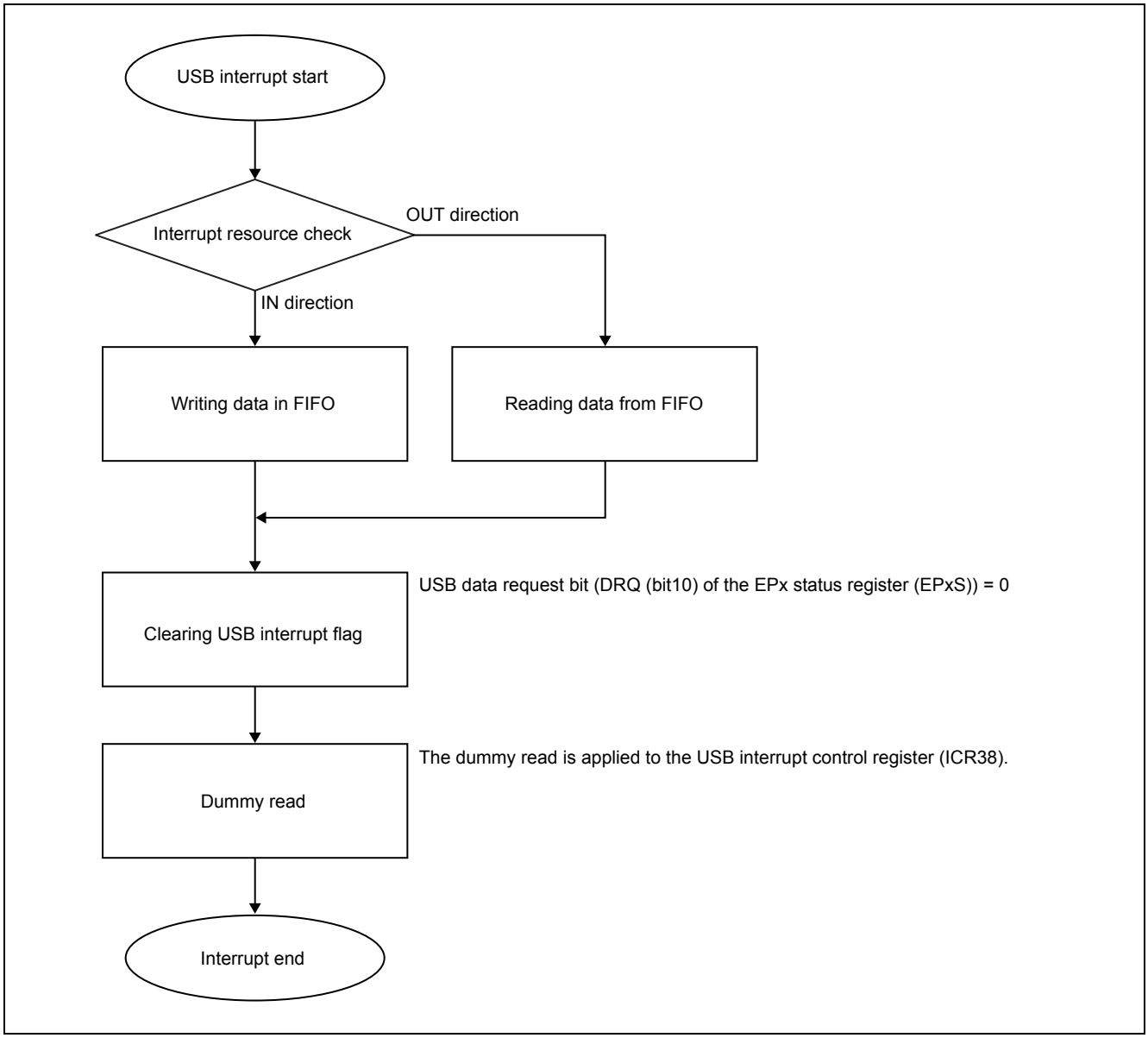
Initialization



Bus reset

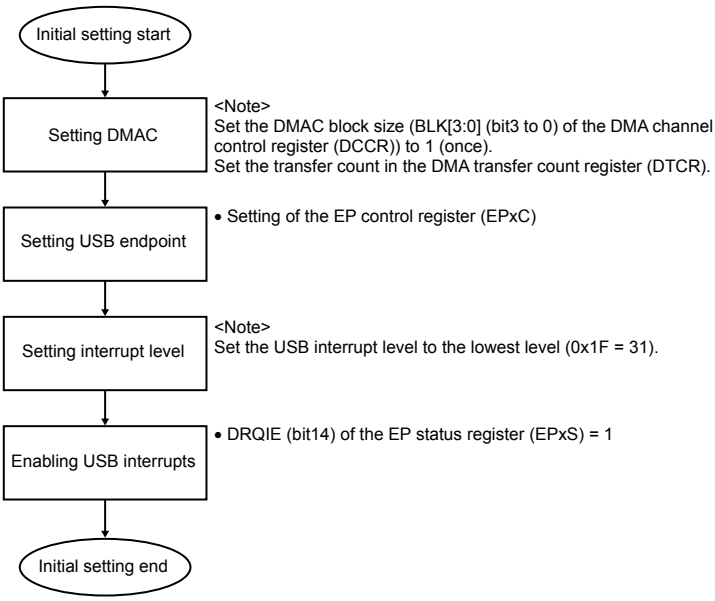

USB interrupt
UDCS.BRST=1?
YES
NO
UDCS.BRST=0
EP0IS,EP0OS,EP1S to EP5S
BFINI=1
// Clear buffer
EP0IS,EP0OS,EP1S to EP5S
BFINI=0
End

Control example for a CPU transfer

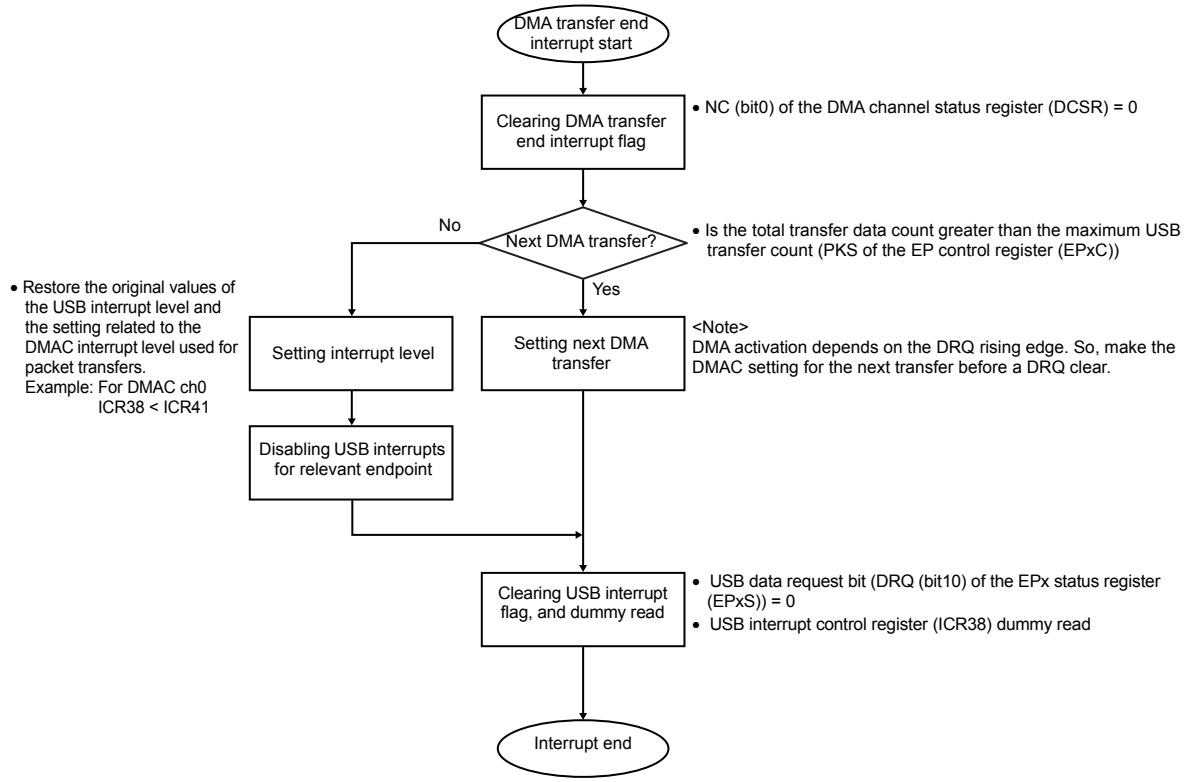


Control example for IN-direction packet transfer mode

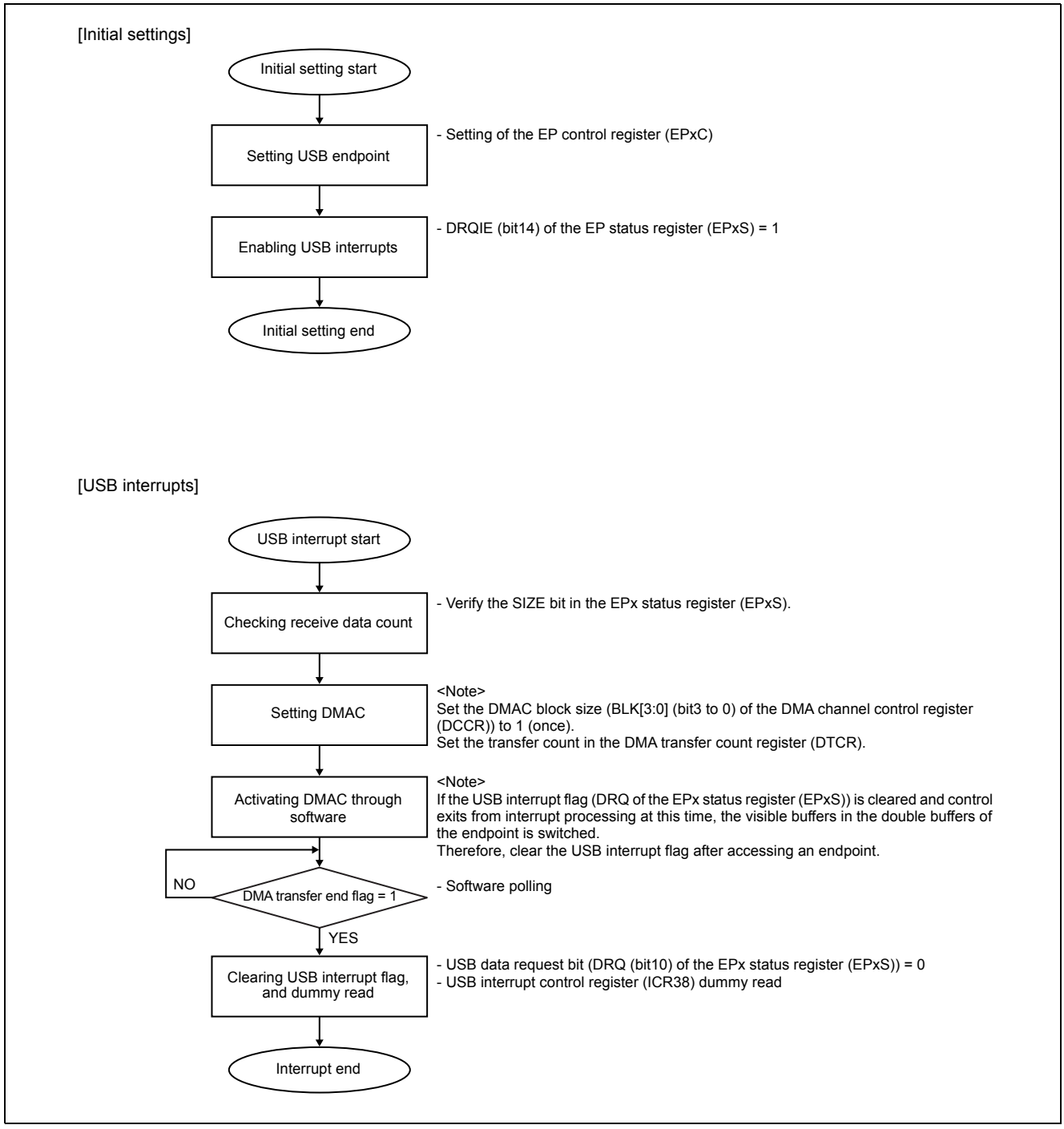
[Initial settings]



[DMA transfer end interrupts]

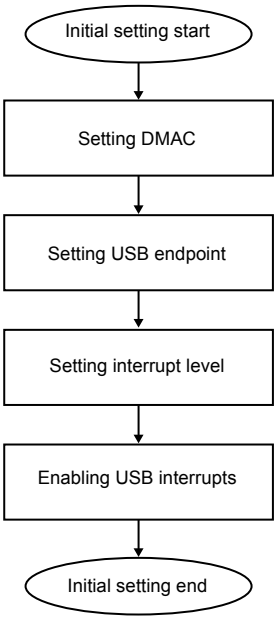


Control example for OUT-direction packet transfer mode



Control example for an IN-direction automatic transfer with a specified data count

[Initial settings]



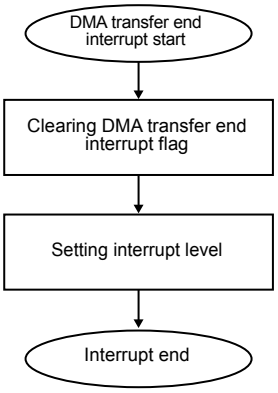
<Note>
Set the DMAC block size (BLK[3:0] (bit3 to 0) of the DMA channel control register (DCCR)) to 1 (once).
Set the total transfer count in the DMA transfer count register (DTCR).
ST (bit15) of the DMA channel control register (DCCRn) = 0
Set "1" in DT (bit11).

- Setting of the EP control register (EPxC)
  DMAE (bit11) of the EP control register (EPxC) = 1

<Note>
Set the USB interrupt level to the lowest level (0x1F = 31).

- DRQIE (bit14) of the EP status register (EPxS) = 1

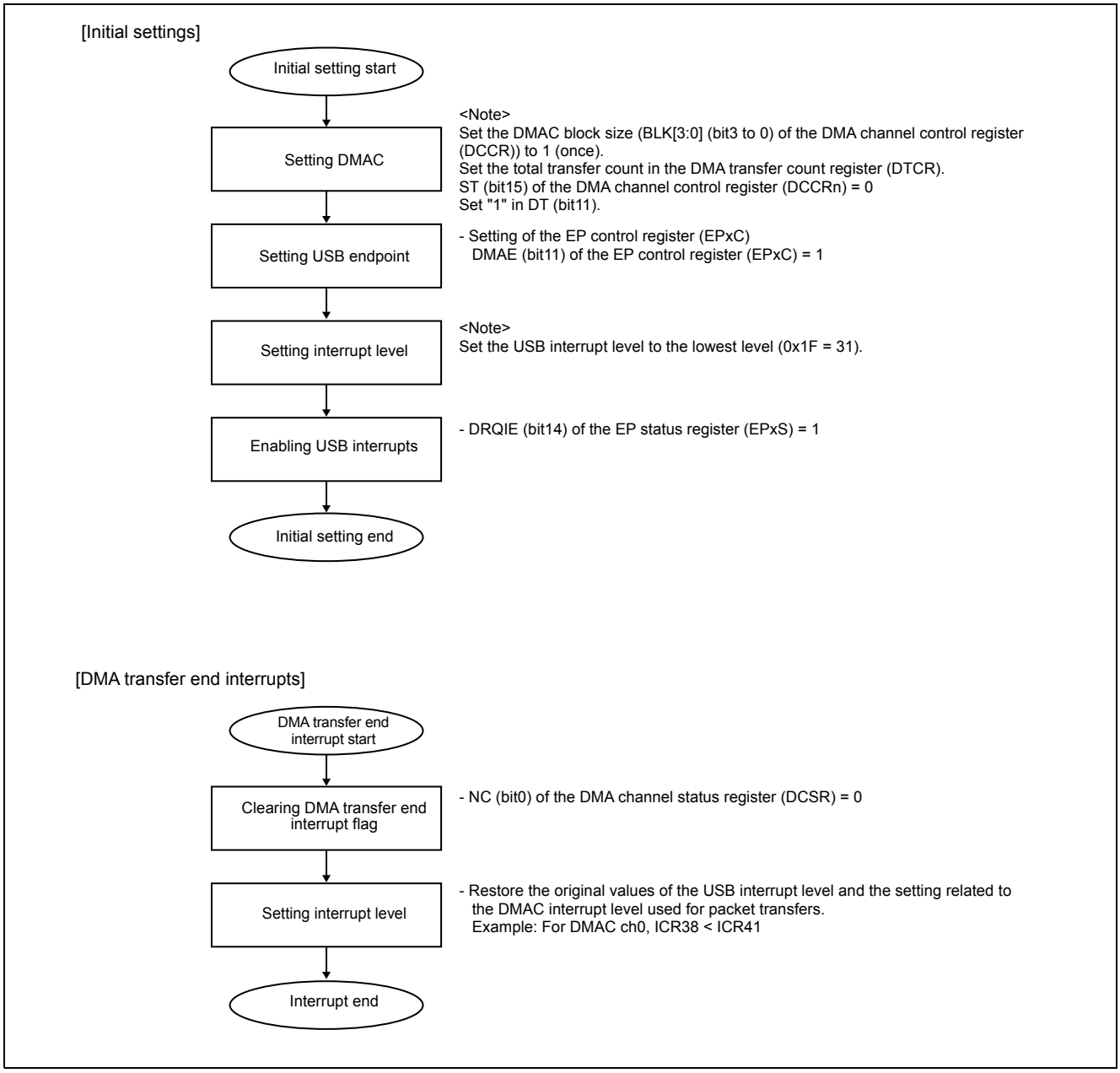
[DMA transfer end interrupts]



- NC (bit0) of the DMA channel status register (DCSR) = 0

- Restore the original values of the USB interrupt level and the setting related to the DMAC interrupt level used for packet transfers.
  Example: For DMAC ch0, ICR38 < ICR41

Control example for an OUT-direction automatic transfer with a specified data count



28. USB Host

This chapter explains the functions and operations of USB HOST.

28.1 Overview

28.2 Configuration

28.3 Registers

28.4 Explanation of Operations and Setting Procedure Examples

28.1 Overview

USB HOST is a function that ensures that a minimum of host operations are performed to enable the system to transfer data with USB devices.

Overview

USB HOST has the following features:

- Automatic detection/support of full-speed transmission
- Automatic detection of connection/disconnection of USB devices
- Support for reset transmission to the USB bus
- Support of IN/OUT/SETUP/SOF tokens
- Automatic detection of a handshake packet for IN token processing (except in the STALL state)
- Automatic detection of a handshake packet for OUT token processing
- Support of a maximum packet length of up to 256 bytes
- Support for different types of error (CRC error, toggle error, timeout error, etc.)
- Support for a wakeup function
- Cypress original USB HOST function. It also operates as a USB function by switching the operation mode.

Note: In case of using USB HOST, use On-chip bus clock (HCLK) with 13 MHz or over.

- Specification outline of USB HOST

  Table 28-1 shows the specification outline of USB HOST.

Table 28-1. Specification Outline of USB HOST

		HOST
HUB support		O[1]
Transfer	Bulk transfer	O
	Control transfer	O
	Interrupt transfer	O
	Isochronous transfer	O
Transfer speed	Low speed	X
	Full speed	O
Support of PRE packets		X
Support of SOF packets		O
Errors	CRC error	O
	Toggle error	O
	Timeout	O
	Maximum size of packet < Received data	O
Detection of connection/disconnection of USB devices		O
Detection of transfer speed		O

O: Supported
X: Not supported
[1]: Supports Full Speed only and the first layer of the HUB.

28.2 Configuration

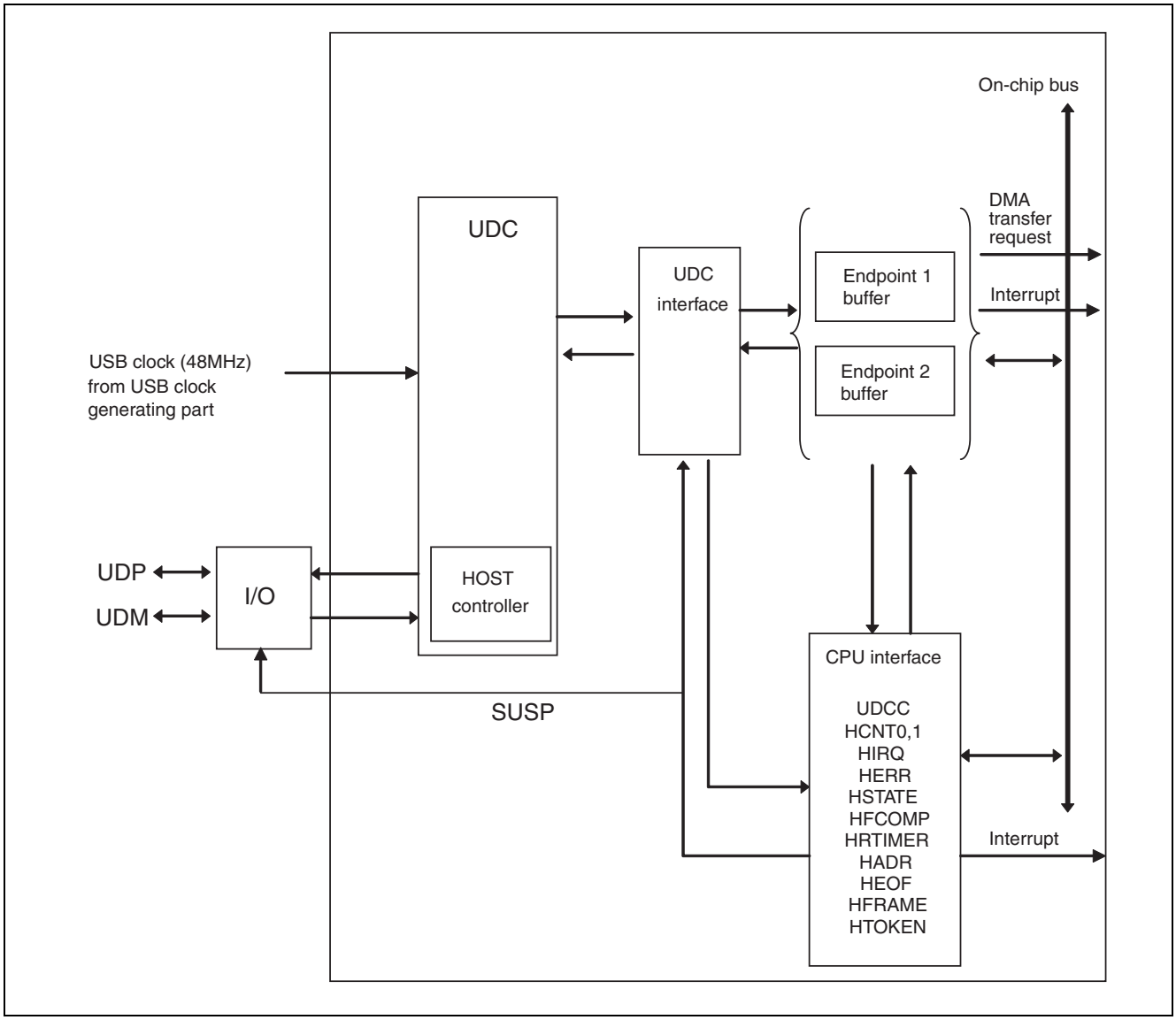
USB HOST consists of the following blocks:

- CPU I/F
- Buffer
- UDC I/F
- USB bus reset control part
- Reception control part
- Transmission control part
- Host transmission control part

Block diagram of USB HOST

Figure 28-1 is a block diagram of USB HOST.

Figure 28-1. Block Diagram of USB HOST



28.3 Registers

This section explains the configuration and functions of registers used by the USB HOST.

USB HOST register list

Table 28-2 lists the registers of USB HOST.

Table 28-2. USB HOST Registers

Abbreviated Register Name	Register Name	Reference
UCCR	USB clock setting register	25.3.1
USBSEL	USB selection register	27.3.1
USBEN	USB enable register	27.3.2
UDCC	UDC control register	27.3.3
EP1C	EP1 control register	27.3.5
EP2C	EP2 control register	27.3.5
EP1S	EP1 status register	27.3.11
EP2S	EP2 status register	27.3.11
EP1DTH	EP1 data register upper	27.3.12
EP1DTL	EP1 data register lower	27.3.12
EP2DTH	EP2 data register upper	27.3.12
EP2DTL	EP2 data register lower	27.3.12
HCNT0	Host control register 0	28.3.1
HCNT1	Host control register 1	28.3.1
HIRQ	Host interrupt register	28.3.2
HERR	Host error status register	28.3.3
HSTATE	Host status register	28.3.4
HFCOMP	SOF interrupt FRAME comparison register	28.3.5
HRTIMER	Retry timer setting register	28.3.6
HADR	Host address register	28.3.7
HEOF	EOF setting register	28.3.8
HFRAME	FRAME setting register	28.3.9
HTOKEN	Host token endpoint register	28.3.10

List of register bits update timing UDCC. RST dependence

Table 28-3. List of Register Bits Update Timing UDCC. RST Dependence

	Register	Bit
Register bits that have to be updated when UDCC.RST=1	HCNT0	HOST
	HSTATE	CLKSEL
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
Register bits that are initialized when UDCC.RST=1. (Update these bits when UDCC.RST=0.)	HCNT0	URST
	HIRQ	TCAN, RWKIRQ, URIRQ, CMPIRQ, CNNIRQ, DIRQ, SOFIRQ
	HERR (all bits)	LSTSOF, RERR, TOUT, CRC, TGERR, STUFF, HS
	HSTATE	SOFBUSY, SUSP
	HFRAME	FRAME0, FRAME1
	HTOKEN (all bits)	TGGL, TKNEN, ENDPT
	EP1S	BFINI, DRQ, SPK
	EP2S	BFINI, DRQ, SPK
Register bits that are not affected by UDCC.RST.	HCNT0	RWKIRE, URIRE, CMPIRE, CNNIRE, DIRE, SOFIRE
	HCNT1	SOFSTEP, CANSEL, RETRY
	HIRQ	CNNIRQ, DIRQ
	HFCOMP	HFRAMECOMP
	HSTATE	TMODE, CSTAT
	HRTIMER0 to HRTIMER2	RTIMER0, RTIMER1, RTIMER2
	HADR	Address
	HEOF	EOF0, EOF1

28.3.1 Host Control Register 0, 1 (HCNT)

This register sets the USB operation mode and interrupts.

Figure 28-2 shows the bit configuration of host control register 0, 1 (HCNT).

Figure 28-2. Bit Configuration of Host Control Register 0, 1 (HCNT)

Host Control Register 1

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	Reserved	Reserved	Reserved	SOFSTEP	CANCEL	RETRY
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	1
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×	×	×	×	×	×	×	×

Host Control Register 0

bit	7	6	5	4	3	2	1	0
	RWKIRE	URIRE	CMPIRE	CNNIRE	DIRE	SOFIRE	URST	HOST
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
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×	×	×	×	×	×	❍	×

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15 to bit11]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).

[bit10]: SOFSTEP (SOF step bit)

This bit is a SOF interrupt generation select bit.

If the bit is set to "1", an interrupt flag for SOF (HIRQ.SOFIRQ) is set to "1" every time that SOF processing is executed.

If the bit is set to "0", the setting value of the SOF interrupt FRAME comparison register (HFCOMP) is compared with the lower 8-bit of the frame number for SOF, if the comparison result is a match, the interrupt flag for SOF (HIRQ.SOFIRQ) is set to "1".

Written Value	Explanation
0	Generates an interrupt request according to the HFCOMP setting.
1	Generates an interrupt request during SOF processing.

Notes:

- If SOF token (TKNEN="001_B") is processed according to the host token end point register (HTOKEN), the interrupt flag for SOF (HIRQ. SOFIRQ) does not become "1" not depending on the setting of this register.
- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit9]: CANCEL (Token cancel enable bit)

This bit is a token cancellation enable bit.

If "1" is set to this bit and writing a token to be processed in the host token end point register (HTOKEN) in the EOF area (area is set by an EOF setting register) cancels the process of the token. If "0" is set to this bit, writing the token to be processed does not cancel the process of the token. Cancellation of the token can be known by reading the TCAN bit in the host interrupt register (HIRQ).

Written Value	Explanation
0	Continues with the token.
1	Cancels the token.

Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit8]: RETRY (Retry enable bit)

This bit is a RETRY enable bit.

If an error such as the NAK error or other error (such as HERR. RERR="1", HERR. TOUT="1", HERR. CRC="1", HERR. TGERR="1" or HERR. STUFF="1") occurs and this bit is set to "1", the operation is retried. The operation is retried for the period specified in the retry timer register (HRTIMER).

Written Value	Explanation
0	Does not retry the operation.
1	Retries the operation.

The bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit7]: RWKIRE (Remote wake up interrupt request enable bit)

This bit is a resume interrupt enable bit.

If "1" is set to this bit and changing the RWKIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit and changing the RWIRQ bit in host interrupt register (HIRQ) to "1" does not generate an interrupt.

Written Value	Explanation
0	Disables interrupts after a reactivation.
1	Enables interrupts after a reactivation.

Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit6]: URIRE (USB bus rest interrupt request enable bit)

This bit is a bus reset interrupt enable bit.

If "1" is set to this bit and changing the URIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit and changing the URIRQ bit in host interrupt register (HIRQ) to "1" does not generate an interrupt.

Written Value	Explanation
0	Disables interrupts after a USB bus reset.
1	Enables interrupts after a USB bus reset.

Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit5]: CMPIRE (Completion interrupt request enable bit)

This bit is a token completion interrupt enable bit.

If "1" is set to this bit, changing the CMPIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the CMPIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts at the completion time of a token.
1	Enables interrupts at the completion time of a token.

Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit4]: CNNIRE (Connection interrupt request enable bit)

This bit is a device connection detection interrupt enable bit.

If "1" is set to this bit, changing the CNNIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the CNNIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts when a USB device is connected.
1	Enables interrupts when a USB device is connected.

Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit3]: DIRE (Disconnection interrupt request enable bit)

This bit is a device disconnection detection interrupt enable bit.

If "1" is set to this bit, changing the DIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the DIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts when a USB device is disconnected.
1	Enables interrupts when a USB device is disconnected.

Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit2]: SOFIRE (Start of frame interrupt request enable bit)

This bit is a SOF interrupt enable bit.

If "1" is set to this bit, changing the SOFIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the SOFIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts when SOF is transmitted.
1	Enables interrupts when SOF is transmitted.

Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit1]: URST (USB bus reset bit)

This bit is a bus reset bit.

If "1" is set to this bit, a bus reset is processed. "1" is indicated by the bit during the reset operation, and it changes to "0" when the reset is completed. Setting "0" to this bit does not operate anything.

Written Value	Explanation
0	Keeps the USB bus in its current state.
1	Resets the USB bus.

Notes:

- Nothing is operated by setting "1" to this bit while the RST bit in the UDC control register (UDCC) is "1".
- Do not write "1" to this bit when the SUSP bit in the host status register (HSTATE) is "1" or when processing of a token is in progress.
- Do not update the host control registers (HCNT0, HCNT1) while this bit is "1".

[bit0]: HOST (Host mode bit)

This bit is a host mode bit.

If "1" is set to this bit, the device is operated in HOST mode. If "0" is set to this bit, the device is operated in function mode.

Written Value	Explanation
0	Function mode is set.
1	Host mode is set.

Notes:

- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- Update this bit only while the RST bit in the UDC control register (UDCC) is "1".
- If operation mode is switched by this bit, it cannot be switched soon. To verify whether the mode is switched by reading this bit.
- To switch from host mode to function mode, verify that the below conditions are satisfied and set "1" to the RST bit in the UDC control register (UDCC) before switching.
  - The SOFBUSY bit in the host status register (HSTATE) is "0".
  - The TKNEN bit in the host token endpoint register (HTOKEN) is "000_B".
  - The SUSP bit in the host status register (HSTATE) is "0".
- To switch from function mode to host mode, and set on the pull-up resistor disconnection side of the general port for pull-up resistor control to disconnect the host or HUB.

28.3.2 Host Interrupt Register (HIRQ)

This register indicates the interrupt request flags for USB HOST. Except the TCAN bit, these bits enable generation of interrupt requests according to the settings of the interrupt enable bits in host control registers 0 and 1 (HCNT0, HCNT1).

Figure 28-3 shows the bit configuration of the host interrupt select register (HIRQ).

Figure 28-3. Bit Configuration of the Host Interrupt Register (HIRQ)

Host Interrupt Register (HIRQ)

bit	7	6	5	4	3	2	1	0
	TCAN	Reserved	RWKIRQ	URIRQ	CMPIRQ	CNNIRQ	DIRQ	SOFIRQ
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
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	❍	❍	❍	❍	❍	×	×	❍

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit7]: TCAN (Token cancel flag bit)

This bit is a token cancel flag.

This bit indicates that the processing of token was canceled according to the CANCEL bit in the host control register 1 (HCNT1) when this bit is changed to "1". If "0" is written to this bit, it indicates that the processing of the token wan not canceled. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

TCAN	In case of reading	In case of writing
0	No such token has been canceled.	Clears this bit.
1	Such a token has been canceled.	Ignored.

Notes:

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- No interrupt request is generated with this bit. To process with an interrupt, verify whether the token was cancelled during the interrupt process with SOF.

[bit6]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).

[bit5]: RWKIRQ (Remote wake up interrupt request bit)

This bit is a remote wake up completion flag.

When this bit changes to "1", it indicates that the remote operation has been completed. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the RWKIRE bit in host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

RWKIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by a reactivation.	Clears interrupt factor.
1	An interrupt request has been generated by a reactivation.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit4]: URIRQ (USB bus reset interrupt request bit)

This bit is a bus reset completion flag.

When this bit changes to "1", it indicates that a reset to a USB bus has been completed. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the URIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

URIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by a USB bus reset.	Clears interrupt factor.
1	An interrupt request has been generated by a USB bus reset.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit3]: CMPIRQ (Completion interrupt request bit)

This bit is a token completion flag.

When this bit changes to "1", it indicates that a token has been completed. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the CMPIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

CMPIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by the completion of a token.	Clears interrupt factor.
1	An interrupt request has been generated by the completion of a token.	Ignored.

Notes:

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- The bit does not change to "1" when the TCAN bit in the host interrupt register (HIRQ) changes to "1".

[bit2]: CNNIRQ (Connection interrupt request bit)

This bit is a device connection detection flag.

When this bit changes to "1",it indicates that connection of the device has been detected. Changing this bit to "0" does not mean anything. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the CNNIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

CNNIRQ	In case of reading	In case of writing
0	No interrupt request is generated by the detection of the connection with a USB device.	Clears interrupt factor.
1	An interrupt request has been generated by the detection of the connection with a USB device.	Ignored.

Notes:

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- The connection of the device is detected in function mode also.

[bit1]: DIRQ (Disconnection interrupt request bit)

This bit is a device disconnection detection flag.

When this bit changes to "1", it indicates that disconnection of the device has been detected. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the DIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

DIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by the detection of the disconnection of a USB device.	Clears interrupt factor.
1	An interrupt request has been generated by the detection of the disconnection of a USB device.	Ignored.

Notes:

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- The disconnection of the device is detected in function mode also.

[bit0]: SOFIRQ (Start of frame interrupt request bit)

This bit is a SOF starting flag.

When this bit changes to "1", it indicates that the SOF token process has been started. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the SOFIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

SOFIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by the start of an SOF token.	Clears interrupt factor.
1	An interrupt request has been generated by the start of an SOF token.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

28.3.3 Host Error Status Register (HERR)

This register indicates whether an error occurred during data transmission/reception in host mode.

Figure 28-4 shows the bit configuration of the host error status register (HERR).

Figure 28-4. Bit Configuration of the Host Error Status Register (HERR)

bit	15	14	13	12	11	10	9	8
	LSTSOF	RERR	TOUT	CRC	TGERR	STUFF	HS	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	11	
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	❍	❍	❍	❍	❍	❍	❍	

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15]: LSTSOF (Lost SOF bit)

This bit is a lost SOF flag.

When this bit changes to "1", and trying to process the SOF token in host mode, it indicates that the SOF token cannot be processed because processing of another token was in progress. It indicates that a lost SOF error has not been detected if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

LSTSOF	In case of reading	In case of writing
0	No SOF processing error.	Clears this bit.
1	SOF processing error.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit14]: RERR (Receive error bit)

This bit is a reception error flag.

When this bit changes to "1", it indicates that the received data is greater than the specified maximum number of packets in host mode. If the reception error is detected, TOUT bit of this register also changes to "1". If this bit is "0", it indicates that the error has not occurred. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

RERR	In case of reading	In case of writing
0	No reception error has occurred.	Clears this bit.
1	A maximum packet reception error has occurred.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit13]: TOUT (Time OUT error bit)

This bit is a timeout error flag.

When this bit changes to "1", it indicates that no response to a token was obtained from the device within the given period in host mode. If this bit is "0", it indicates that the timeout error has not occurred. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

TOUT	In case of reading	In case of writing
0	No timeout has occurred.	Clears this bit.
1	A timeout has occurred.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit12]: CRC (CRC error bit)

This bit is a CRC error flag.

When this bit changes to "1", it indicates that a CRC error has been detected in host mode. If this bit is "0", it indicates that the CRC error has not been detected. If a CRC error has occurred, TOUT bit of this register also changes to "1". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

CRC	In case of reading	In case of writing
0	No CRC error has occurred.	Clears this bit.
1	A CRC error has occurred.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit11]: TGERR (Toggle error bit)

This bit is a toggle error flag.

When this bit changes to "1", it indicates that the received toggle was not a match in host mode. If this bit is "0", it indicates that the toggle error has not been detected. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

TGERR	In case of reading	In case of writing
0	No toggle error has occurred.	Clears this bit.
1	A toggle error has occurred.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit10]: STUFF (Stuffing error bit)

This bit is a stuffing error flag.

When this bit changes to "1", it indicates that a stuffing error has been detected. If this bit is "0", it indicates that the stuffing error has not been detected. If the stuffing error has been detected, TOUT bit of this register also changes to "1". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

STUFF	In case of reading	In case of writing
0	No stuffing error has occurred.	Clears this bit.
1	A stuffing error has occurred.	Ignored.

Note: This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit9, bit8]: HS (Hand shake status bits)

These bits are hand shake status flags.

These bits indicate the handshake status for transmission/reception.

NULL is indicated when no handshake has occurred caused by an error or when an SOF token is completed with the TKNEN bit in host token endpoint register (HTOKEN).

The bits are updated at the transmission/reception completion time.

Table 28-4. Handshake

bit9	bit8	Handshake Status
0	0	ACK
0	1	NAC
1	0	STALL
1	1	NULL

Note: These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".

28.3.4 Host Status Register (HSTATE)

This register indicates the USB circuit status, such as for the connection with a USB device and the transfer mode. Note that the CLKSEL bit is valid in function mode also.

Figure 28-5 shows the bit configuration of the host status register (HSTATE).

Figure 28-5. Bit Configuration of the Host Status Register (HSTATE)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	CLKSEL	SOFBUSY	SUSP	TMODE	CSTAT
Attribute	-	-	R/W	R/W	R/W	R/W	R	R
Initial value	X	X	0	1	0	0	1	0
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-	-	×	×	❍	❍	×	×

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit7, bit6]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit5]: Reserved bit

This bit is a reserved bit. Always write "0" to this bit. "0" is read when reading.

[bit4]: CLKSEL (Clock select bit)

This bit specifies the USB operation clock.

Read Value	Explanation
0	The setting is prohibited.
1	A clock for full speed.

Notes:

- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- Update this bit only while the RST bit in the UDC control register (UDCC) is "1".
- Always set "1" to this bit. Do not set "0" to this bit.
- Use on-chip bus clock (HCLK) with 13 MHz or over.

[bit3]: SOFBUSY (SOF busy bit)

This bit is a SOF busy flag.

This bit changes to "1", when a SOF token is processed in the host token endpoint register (HTOKEN), and it indicates whether the SOF timer is active. If this bit is "0", it indicates that SOF timer is stopped. While SOF timer is operated, SOF timer can be stopped by writing "0" to this bit. When "1" is written to this bit, the writing is ignored.

SOFBUSY	In case of reading	In case of writing
0	The SOF timer is stopped.	Stops the SOF timer.
1	The SOF timer is active.	Ignored.

Notes:

- This bit is initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- To stop SOF timer, writing "0" to this bit does not stop SOF timer soon. Reading this bit can verify whether it is stopped.

[bit2]: SUSP (Suspend bit)

This bit is a suspend setting bit.

Writing "1" to the bit sets the suspended state.

If "0" is written to the bit when it is "1" or if the USB bus changes to the k-state, the suspended state is canceled and the RWKIRQ bit in the host interrupt register (HIRQ) changes to "1".

Table 28-5. Suspend setting

SUSP	Operation
Writing "1"	Sets the suspended state.
Writing "0" when the bit is "1"	Sets the resume state.
Other	The state does not change.

Notes:

- This bit becomes an initial value when the RST bit in the UDC control register (UDCC) is "1".
- Do not write "1" to this bit during a USB operation (USB bus reset, data transmission/reception during SOF timer operation).
- Do not stop the USB clock in host mode, even in the suspended state.
- If the setting of this bit is changed, the change will not be reflected to the desired state. The change can be verified by reading this bit.

[bit1]: TMODE (Transmission mode bit)

This bit is a transfer mode flag.

When this bit is "1", it indicates that the full speed device is connected. When this bit is "0", it indicates that the low speed device is connected. It is valid when the CSTAT bit in host status register (HSTATE) is "1".

Read Value	Explanation
0	Low speed (This series does not support the low speed.)
1	Full speed

Notes:

- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- Use on-chip bus clock (HCLK) with 13MHz or over.
- This series does not support the low speed.

[bit0]: CSTAT (Connect status bit)

This bit is a connection status flag.

When this bit is "1", it indicates that the device is connected. When this bit is "0", it indicates that the device is disconnected.

Read Value	Explanation
0	The USB device is disconnected.
1	The USB device is connected.

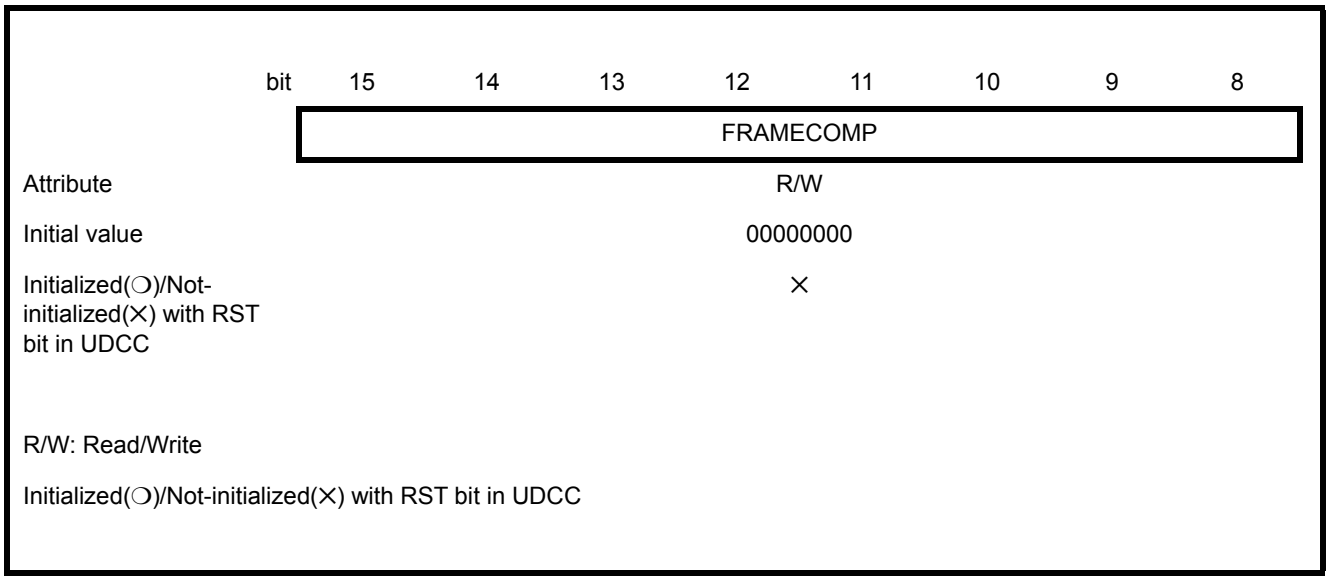
Note: This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

28.3.5 SOF Interrupt FRAME Comparison Register (HFCOMP)

This register specifies the data that is compared with the lower 8-bit of the frame number for SOF token processing. The register is compared with the lower 8-bit of the frame number, when the SOFSTEP bit in the host control register 0 (HCNT0) is "0". If the SOFIRE bit is "1", that is, the comparison result is a match, the SOFIRQ bit in the host interrupt register (HIRQ) changes to "1" at the start of SOF transmission, generating an interrupt request.

Figure 28-6 shows the bit configuration of the SOF interrupt FRAME comparison register (HFCOMP).

Figure 28-6. Bit Configuration of the SOF Interrupt FRAME Comparison Register (HFCOMP)



[bit15 to bit8]: FRAMECOMP

These bits are FRAME comparison data.

This register specifies the data that is compared with the lower 8-bit of the frame number for SOF token processing.

When the SOFSTEP bit in the host control register 0 (HCNT0) is "0", this register is compared with the frame number for SOF token processing. If the comparison result is a match, "1" is set to the SOFIRQ bit of host interrupt register (HIRQ).

When the SOFSTEP bit in the host control register 0 (HCNT0) is "0", the setting of this register is invalid.

Note: These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

28.3.6 Retry Timer Setting Register (HRTIMER)

This register sets the retry interval for tokens.

Figure 28-7 shows the bit configuration of the retry timer setting register (HRTIMER).

Figure 28-7. Bit Configuration of the Retry Timer Setting Register (HRTIMER)

Retry Timer Setting Register (HRTIMER)

bit	7(23)	6(22)	5(21)	4(20)	3(19)	2(18)	1(17)	0(16)
	Reserved						RTIMER2	
Attribute	-						R/W	
Initial value	X						00	
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-						×	

bit	15	14	13	12	11	10	9	8
	RTIMER1							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×							

bit	7	6	5	4	3	2	1	0
	RTIMER0							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×							

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit23 to bit18]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit17 to bit0]: RTIMER2 to RTIMER0

These bits set the retry interval for tokens. The retry timer is activated by the start of processing of a token, decrementing the timer by 1 at the 1-bit transfer clock (12 MHz at full speed). Once the retry timer reaches 0, it will be completed after the token is processed. If a retry is made for a token in the EOF area, the retry timer stops until SOF processing is completed. The value of the stopped timer is decremented by 1 after the SOF processing is completed.

Note: These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

28.3.7 Host Address Register (HADR)

This register is used for the address field at the token transmission time.

Figure 28-8 shows the bit configuration of the host address register (HADR).

Figure 28-8. Bit Configuration of the Host Address Register (HADR)

bit	15	14	13	12	11	10	9	8
	Reserved	Address						
Attribute	-	R/W						
Initial value	X	0000000						
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-	×						

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15]: Reserved bit

This bit is a reserved bit. It is undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit14 to bit8]: Address

These bits are address bits.

These bits specify the token address.

Note: These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

28.3.8 EOF Setting Register (HEOF)

This register sets the period in which token processing is disabled prior to SOF token processing. In a comparison of the SOF timer value and the data in this register, if the SOF timer value is smaller than this data and a processing request is issued for an IN token, OUT token, or SETUP token, the requested token is processed after the processing of the SOF token. This is done to avoid concurrent processing of SOF tokens generated by the hardware and other tokens. This register has the time unit of 1-bit transfer time.

Figure 28-9 shows the bit configuration of the EOF setting register (HEOF).

Figure 28-9. Bit Configuration of the EOF Setting Register (HEOF)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	EOF1					
Attribute	-	-	R/W					
Initial value	X	X	000000					
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-	-	×					

bit	7							0
	EOF0							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×							

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15, bit14]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit13 to bit0]: EOF1, EOF0 (End of frame bits)

These bits are EOF bits.

These registers set the period in which token processing is disabled prior to SOF token processing. Set these registers with a margin with 1 packet length or longer. These registers have the time unit of 1-bit transfer time.

For example, suppose that MAXPKT = 64 bytes at full speed, then a value of "$2C9_H$" will be set in these bits according to the following formula:

(Token_length + packet_length + header + CRC) × 7/6 + Turn_around_time

= (34 bits + 546 bits) × 7/6 + 36 bits

= 712.7 bits

Note: These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

28.3.9 FRAME Setting Register (HFRAME)

This register sets the frame number for SOF token processing. If SOF activation is set in the TKNEN bit in the host token end-point register (HTOKEN), the SOF timer is activated and then SOF is automatically transmitted every 1 ms. The register is automatically incremented by 1 every time that SOF processing is completed.

Figure 28-10 shows the bit configuration of the FRAME setting register (HFRAME).

Figure 28-10. Bit Configuration of the FRAME Setting Register (HFRAME)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	Reserved	Reserved	Reserved	FRAME1		
Attribute	-	-	-	-	-	R/W		
Initial value	X	X	X	X	X	000		
Initialized(❍)/Not-initialized(✕) with RST bit in UDCC	-	-	-	-	-	❍		

bit	7	6	5	4	3	2	1	0
	FRAME0							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(✕) with RST bit in UDCC	❍							

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(✕) with RST bit in UDCC

[bit15 to bit11]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit10 to bit0]: FRAME1, FRAME0

These bits are frame setting bits.

They specify the number of frames of SOF.

Notes:

- These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".
- Before setting SOF to the TKNEN bit in the host token endpoint register (HTOKEN), specify the number of frames in this register.
- Do not write to these bits while SOF token processing is in progress and the SOFBUSY bit in the host status register (HSTATE) is "1".

28.3.10 Host Token Endpoint Register (HTOKEN)

This register sets the toggle, an endpoint, and tokens.

Figure 28-11 shows the bit configuration of the host token endpoint register (HTOKEN).

Figure 28-11. Bit Configuration of the Host Token Endpoint Register (HTOKEN)

bit	7	6	5	4	3	2	1	0
	TGGL	TKNEN			ENDPT			
Attribute	R/W	R/W			R/W			
Initial value	0	000			0000			
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	❍	❍			❍			

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit7]: TGGL (Toggle bit)

This bit is a toggle bit.

This bit specifies the data toggle. In transmission, toggle data is transmitted according to the setting of this bit. In reception, received toggle data and the toggle data indicated by this bit are compared to detect any errors.

Written value	Explanation
0	DATA0
1	DATA1

Notes:

- This bit becomes an initial value when the RST bit in the UDC control register (UDCC) is "1".
- Set this bit when the TKNEN bit in host token endpoint register (HTOKEN) is "000_B".

[bit6 to bit4]: TKNEN (Token enable bits)

These bit are token enable bits.

Tokens are transmitted according to the setting of these bits. The bits change to "000_B" after the operation is completed, and the CMPIRQ bit in the host interrupt register (HIRQ) changes to "1". If the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated.

The settings of the TGGL and ENDPT bits in SOF token processing are ignored.

Table 28-6. Token Setting

bit6	bit5	bit4	Operation
0	0	0	The token is not transmitted.
0	0	1	A SETUP token is transmitted.
0	1	0	An IN token is transmitted.
0	1	1	An OUT token is transmitted.
1	0	0	An SOF token is transmitted.
1	0	1	Isochronous IN is transmitted.
1	1	0	Isochronous OUT is transmitted.
1	1	1	Reserved (The setting is prohibited.)

Notes:

- These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".
- PRE packets are not supported.
- Do not set these bits to "100_B" when the SOFBUSY bit in the host status register (HSTATE) is "1".
- Set to host mode before writing to these bits.
- In cases of issuing the token again after an interrupt flag (CMPIRQ) becomes "1", write to these bits after waiting for 3 or more cycles of the USB transfer clock (12 MHz at full speed).
- In the disconnected state (CSTAT="0" in HSTATE), the token is not processed even if the bits are written.

[bit3 to bit0]: ENDPT (Endpoint bits)

These bits are endpoint bits.

These bits set the endpoint used to transmit/receive data to/from a USB device.

Note: These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".

28.4 Explanation of Operations and Setting Procedure Examples

This section explains operations of USB HOST. Also, examples of procedures for setting the operating state are shown.

28.4.1 Connecting a USB Device

This section explains how to detect the connections with external USB devices through software.

Setting the HOST function

To allow the USB HOST to operate, disable USB operation (USBEN = 0) in the USB enable register (USBEN), enable USB clock output in the USB clock generating part setting, and then enable USB operation (USBEN = 1).

Then, write "1" to the HOST bit in host control register 0 (HCNT0) for operation as a host.

Disconnected/Connected state of an external USB device

When an external USB device is disconnected, the USB HOST UDP/UDM pins are at the "L" level as determined through a pull-down resistor. At this time, the CSTAT bit in the host status register (HSTATE) is "0", and the TMODE bit is undefined. If the external USB device is connected again, the CSTAT bit in the host status register (HSTATE) changes to "1".

Detecting the connections with external USB devices

When detecting the connections with external USB devices, the CNNIRQ bit in the host interrupt register (HIRQ) changes to "1", and the CNNIRE bit in the host control register 0 (HCNT0) is set to "1", and a device connection interrupt request is generated. To clear this interrupt request, write "0" to the CNNIRQ bit in the host interrupt register (HIRQ). To detect the connections with USB devices through polling instead of interrupts, write "0" to the CNNIRE bit in host control register 0 (HCNT0), and write a program for confirming that the CNNIRQ bit in the host interrupt register (HIRQ) changes to "1".

Obtaining a transfer speed and selecting the clock for a connected USB device

To obtain an available transfer speed for a connected USB device after its connection was detected, see the value of the TMODE bit in the host status register (HSTATE). shows the relationship between transfer speeds and the TMODE bit in the host status register (HSTATE).

Table 28-7. Relationship Between Transfer Speeds for a Connected USB Device and TMODE

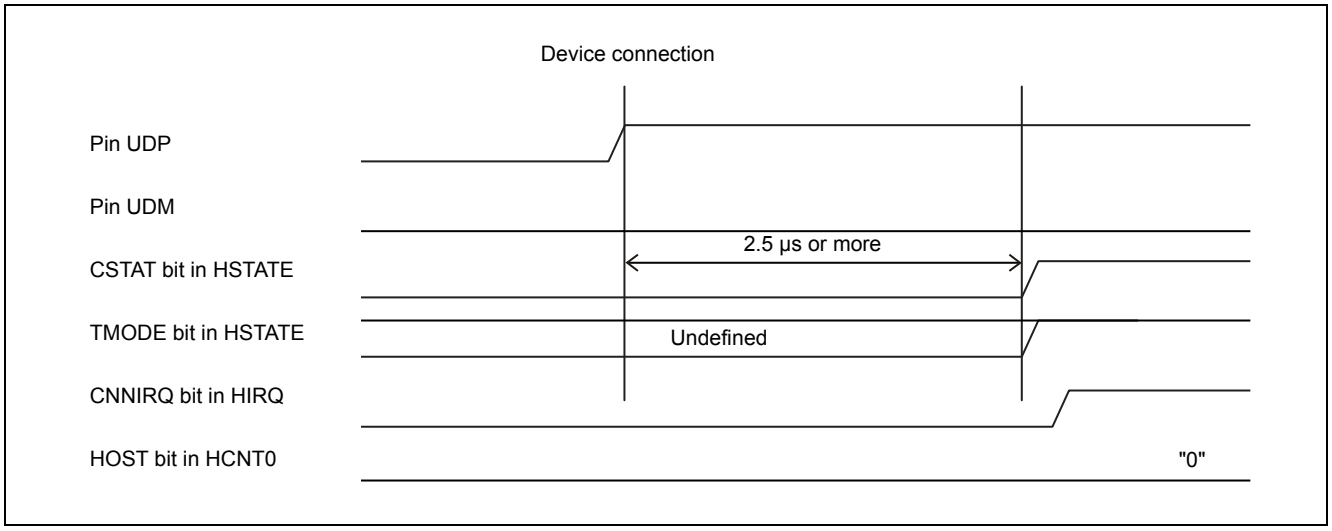
Transfer Speed for Connected USB Device	TMODE
Full speed	1
Low speed	0

Note: This series does not support the low speed.

After obtaining the transfer speed for the external USB device, if the RST bit in the UDC control register (UDCC) is "1", update the CLKSEL bit in the host status register (HSTATE) according to the obtained transfer speed.

Figure 28-12 shows an example of the timing in detection of the connection with a full-speed device.

Figure 28-12. Example of the Timing in Detection of the Connection with a Full-speed Device (HOST bit in HCNT0 = 0)



Notes:

- The CSTAT bit in the host status register (HSTATE) changes to "1" after 2.5 µs elapses after the external USB device is connected.
- The TMODE bit and CSTAT bit in the host status register (HSTATE) are updated irrespective of the value of the HOST bit in host control register 0 (HCNT0). CNNIRQ bit and DIRQ bit of host interrupt register (HIRQ) become "1" if the conditions are satisfied.

28.4.2 Resetting the USB Bus

This section explains how to reset the USB bus in host mode.

Resetting the USB bus

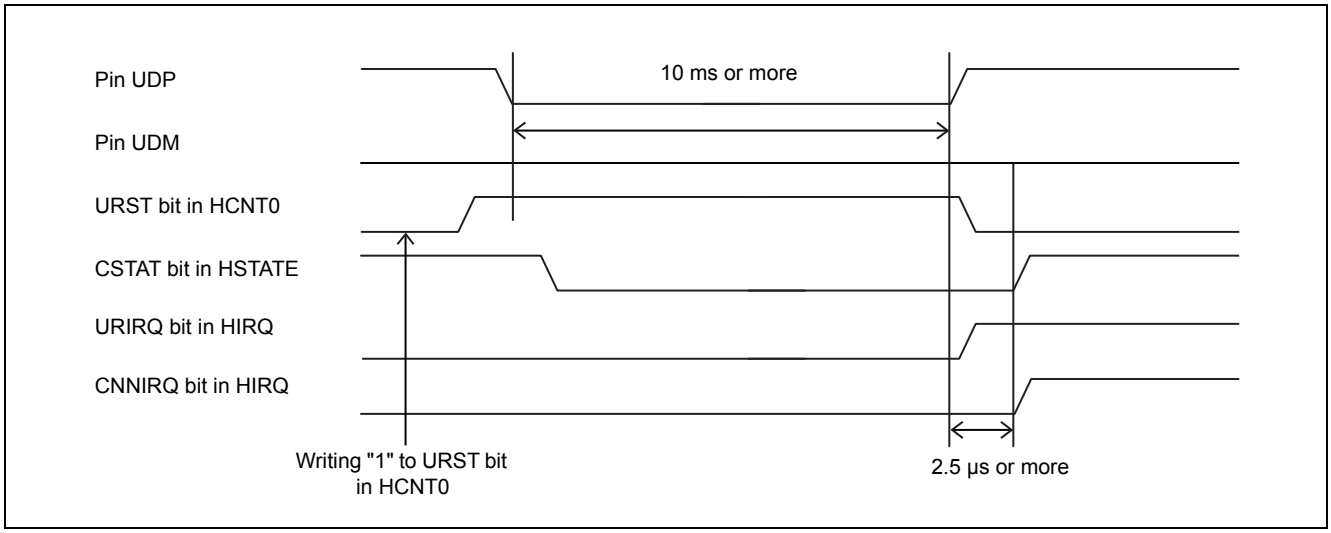
Writing "1" to the URST bit in host control register 0 (HCNT0) in host mode results in SE0 being transmitted for 10 ms or more to reset the USB bus. The URST bit in host control register 0 (HCNT0) changes to "0" at the end of the USB bus reset. At this point, if the URIRE bit in host control register 0 (HCNT0) is "1", an interrupt request is generated, and the URIRQ bit in the host interrupt register (HIRQ) changes to "1". To clear the interrupt request, write "0" to the URIRQ bit in the host interrupt register (HIRQ).

Note the following points about resetting the USB bus:

1. Before resetting the USB bus, confirm that the CSTAT bit in the host status register (HSTATE) is "1" and the USB device is connected.
2. When the USB bus is reset, the CSTAT bit in the host status register (HSTATE) changes to "0", indicating the disconnected state. The DIRQ bit in the host interrupt register (HIRQ) does not change to "1" at this time.
3. After resetting the USB bus, compare the CLKSEL bit and TMODE bit in the host status register (HSTATE). If they do not match, update the CLKSEL bit so that they match. Update this bit only while the RST bit in the UDC control register (UDCC) is "1".
4. After resetting the USB bus, please process the token after confirming that the USB device is connected by the CNNIRQ bit in the host interrupt register (HIRQ) or the CSTAT bit in the host status register (HSTATE).

Figure 28-13 shows an example of the reset timing for a USB device.

Figure 28-13. Example of the Reset Timing for a USB Device



Note: After resetting the USB bus, the token cannot be issued unless connection of the USB device is detected.

28.4.3 Token Packets

This section explains how to set token packets.

The setting is unnecessary if there is no change in each register (HADR, EP1C, EP2C, HFRAME and HEOF).

How to set token packets

In host mode, the buffer of endpoint 1 and the buffer of endpoint 2 are used as the buffer for transmission/reception. In case of processing either an IN token, OUT token, or SETUP token in host mode, specify the destination addresses in the host address register (HADR). Also, in the PKS bit and DIR bit in the EP1 control register (EP1C) or EP2 control register (EP2C), specify the maximum number of bytes per packet and the transfer direction.

If the DIR bit in the EP1 control register (EP1C) is "1", the buffer of endpoint 1 is used as an OUT-direction buffer, and the buffer of endpoint 2 is used as an IN-direction buffer. In this case, write "0" to the DIR bit in the EP2 control register (EP2C).

If the DIR bit in the EP1 control register (EP1C) is "0", the buffer of endpoint 1 is used as an IN-direction buffer, and the buffer of endpoint 2 is used as an OUT-direction buffer. In this case, write "1" to the DIR bit in the EP2 control register (EP2C).

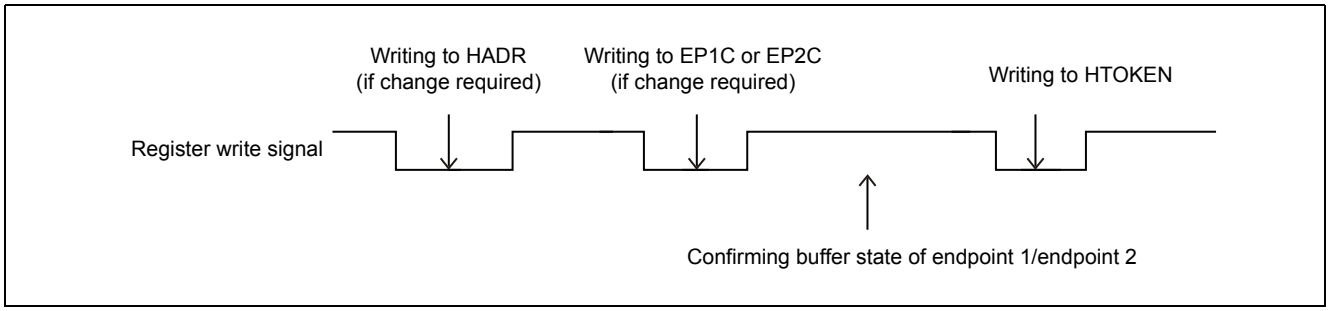
To process the token, follow the following procedures for the settings.

1. Set the DIR bit and PKS bit in the EP1 control register (EP1C) and EP2 control register (EP2C).
2. If the endpoint n (n=1 or 2) to be used is used in an OUT-direction, write the transmitted data in the buffer of the endpoint n (n=1 or 2), and set "0" to the DRQ bit in the EPn status register (EpnS:n=1 or 2). If the direction is in IN-direction, read the DRQ bit of the EPn status register (EPnS:n=1 or 2), and confirm that "0" is written.
3. Set the target endpoint, token and toggle data in the host token endpoint register (HTOKEN).

The USB circuit transmits token packets in the order of Sync, token, address, endpoint, CRC5, and EOP according to the specified tokens. (Sync, CRC5, and EOP are automatically transmitted.) After the end of each packet, the CMPIRQ bit in the host interrupt register (HIRQ) changes to "1", and TKNEN in the host token endpoint register (HTOKEN) becomes "000_B". At this point, if the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated. To clear the interrupt request, write "0" to the CMPIRQ bit in the host interrupt register (HIRQ).

Figure 28-14 shows an example of setting the registers for processing of IN/OUT/SETUP tokens.

Figure 28-14. Example of Setting the Registers for Processing of IN/OUT/SETUP Tokens



For an SOF token, writing an EOF time and frame number to the EOF setting register (HEOF) and FRAME setting register (HFRAME), respectively, and writing the SOF token code to the TKNEN bit in the host token endpoint register (HTOKEN) results in Sync, the SOF token, the frame number, CRC5, and EOP being transmitted. At this time, the SOFBUSY bit in the host status register (HSTATE) changes to "1", and the FRAME setting register (HFRAME) is incremented by 1. Also at this time, CMPIRQ in the host interrupt register (HIRQ) changes to "1", and the TKNEN bit in the host token endpoint register (HTOKEN) clears to "000_B". At this point, if the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated. Subsequently, interrupt requests by CMPIRQ will not be generated for automatically generated SOF. To clear the interrupt request by the end of the token, write "0" to the CMPIRQ bit in the host interrupt register (HIRQ).

While the SOFBUSY bit in the host status register (HSTATE) is "1", SOF is automatically transmitted every 1 ms. The SOFBUSY bit in the host status register (HSTATE) changes to "0" under the following conditions (conditions to stop SOF):

- "0" is written to the SOFBUSY bit in the host status register (HSTATE).
- The USB bus is reset (by writing "1" to the URST bit in host control register 0 (HCNT0)).
- "1" is written to the SUSP bit in the host status register (HSTATE).
- The USB device is disconnected (the CSTAT bit in HSTATE is "0").

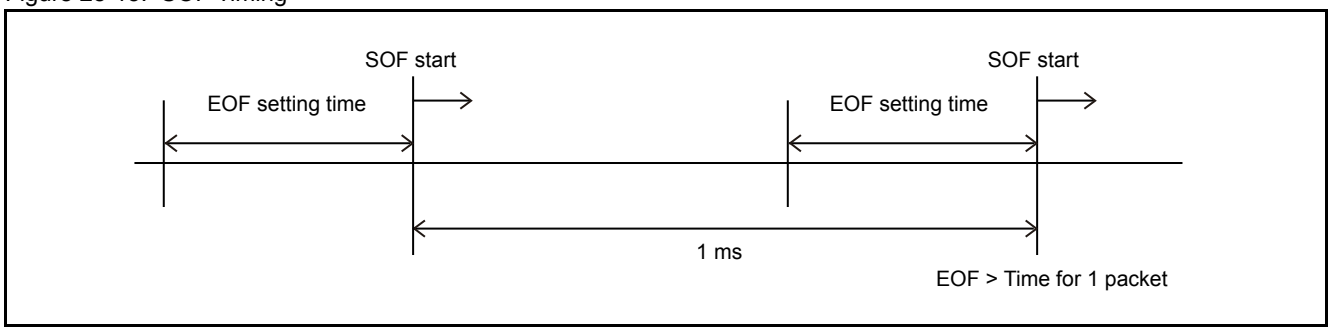
Before switching from host mode to function mode, be sure to write "0" to the SOFBUSY bit in the host status register (HSTATE) and confirm that the SOFBUSY bit has changed to "0" and the TKNEN bit of host token endpoint register (HTOKEN) is "000_B" and confirm that the SUSP bit in host status register (HSTATE) is "0" and set the RST bit in UDC control register (UDCC) to "1".

To change the SOFBUSY bit in the host status register (HSTATE) back to "1", process the SOF token again.

If the TKNEN bit in the host token endpoint register (HTOKEN) is written between the EOF setting time and SOF start time, the EOF setting register (HEOF) will wait until SOF is processed before processing the specified token in order to avoid concurrent processing of the SOF token and other tokens. The EOF setting register (HEOF) has a 1-bit unit of time. For example, suppose that "10_H" is written to EOF bit in the EOF setting register (HEOF), then the value will be 16 × 1/12 MHz = 1333.3 ns in full speed mode. If the specified EOF setting time is shorter than the time for one packet, SOF processing may overlap with processing of another token. In such cases, the LSTSOF bit in the host error status register (HERR) changes to "1", and SOF will not be processed. If LSTSOF in the host error status register (HERR) changes to "1", increase the time in the data for the EOF setting register. (See "28.3.8 EOF Setting Register (HEOF)".)

Figure 28-15 shows the SOF timing.

Figure 28-15. SOF Timing



28.4.4 Data Packets

This section explains how to transfer data packets.

A data packet is transferred in the following procedure after a token packet is transmitted.

- In transmission
  1. Sync is automatically transmitted.
  2. If the TGGL bit in the host token endpoint register (HTOKEN) is "0", DATA0 is transmitted; if it is "1", DATA1 is transmitted.
  3. To transmit all transmitted data, select the buffer of endpoint 1 if the DIR bit in the EP1 control register (EP1C) is "1", or select the buffer of endpoint 2 if the DIR bit is "0".
  4. 16-bit CRC is transmitted.
  5. 2-bit EOP is transmitted.
  6. 1-bit J State is transmitted.
- In reception
  1. Receive Sync.
  2. Toggle data is received and compared with the TGGL bit in the host token endpoint register (HTOKEN).
  3. To sort the received data when the toggle data and the TGGL bit are found to match, select the buffer of endpoint 2 if the DIR bit in the EP1 control register (EP1C) is "1", or select the buffer of endpoint 1 if the DIR bit is "0".
  4. 16-bit CRC is verified when EOF is received.

If the HOST bit in host control register 0 (HCNT0) is "1", write inverted values as the data in the DIR bits in the EP1 control register (EP1C) and EP2 control register (EP2C). For example, if the DIR bit in the EP1 control register (EP1C) is "0", write "1" to the DIR bit in the EP2 control register (EP2C).

28.4.5 Operation of a Handshake Packet

This section explains the operation of a handshake packet that notifies the transmitter/receiver of the host state.

The handshake packet determines whether the receiving side is in a state where it is capable of properly receiving data and the receiving side transmits ACK, NAK, or STALL. If the USB circuit receives the handshake packet, the type of the received handshake packet is written to the HS bit in the host error status register (HERR). If the handshake packet is transmitted, the type of the transmitted handshake packet is written to the HS bit in the host error status register (HERR).

28.4.6 Retry Function

If an error such as the NAK error or CRC error occurs at the packet end time and the RETRY bit in host control register 1 (HCNT1) is "1", the operation is retried for the period specified in the retry timer register (HRTIMER).

This section explains operations of the retry function.

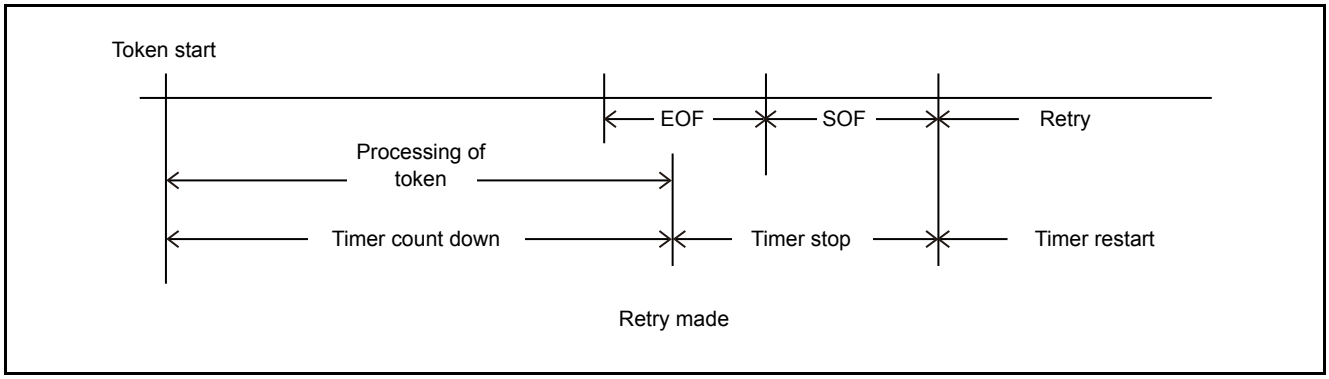
If an error other than the STALL error or device disconnection error (the HS bit of the host error status register (HERR) is "01_B", RERR bit is "1", TOUT bit is "1", CRC bit is "1", or STUFF bit is "1") occurs during transfer and the RETRY bit in host control register 1 (HCNT1) is "1", the operation for the token involved is retried. The retries are ended under the following conditions:

- "0" is written to the RETRY bit in host control register 1 (HCNT1).
- A retry timer value of 0 is detected.
- The SOFIRQ bit in host interrupt register (HIRQ) becomes "1" by SOF.
- ACK is detected.
- The disconnection of the device is detected.

The retry timer is activated at the start of token transfer, and it counts down using a 1-bit transfer clock. However, if a retry is made in the EOF area, the counting is stopped. If the SOF token ends with "0" in the SOFIRQ bit in the host interrupt register (HIRQ), the timer resumes from the "stopped" timer value. At the point where the retry timer changes to "0", the packet transfer ends and the CMPIRQ bit of the host interrupt register (HIRQ) is set to "1".

Figure 28-16 shows operations of the retry timer with "0" in the SOFIRQ bit in the host interrupt register (HIRQ).

Figure 28-16. Retry Timer Operations (SOFIRQ in the Host Interrupt Register (HIRQ) = 0)



Completion information for the completed packet is written to each register when the retry operation is completed.

28.4.7 SOF Interrupts

This section explains operations of SOF interrupts.

If the SOFSTEP bit of the host control register 1 (HCNT1) is "0", the SOF interrupt FRAME comparison register (HFCOMP) is compared with the lower 8-bit of the frame number for SOF token, if the comparison result is a match, the SOFIRQ bit of the host interrupt register (HIRQ) is set to "1" at transmission of SOF.

If the SOFSTEP bit of the host control register 1 (HCNT1) is "1", SOFIRQ bit of host interrupt register (HIRQ) is set to "1" every time that SOF is transmitted.

If the SOFIRE bit of host control register 0 (HONT0) is "1", and changing the SOFIRQ bit of host interrupt register (HIRQ) to "1" generates an interrupt. SOF processing by the host token end point register (HTOKEN) does not change the SOFIRQ bit of the host interrupt register (HIRQ) to "1".

Figure 28-17 and Figure 28-18 show examples of operations of SOF interrupts.

Figure 28-17. Example of SOF Interrupt Operations (SOFSTEP in Host Control Register 1 (HCNT1) = 1)

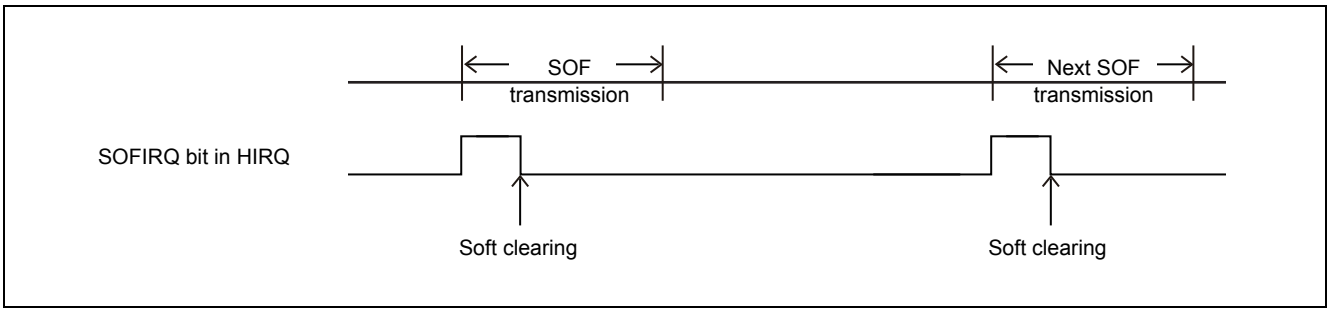
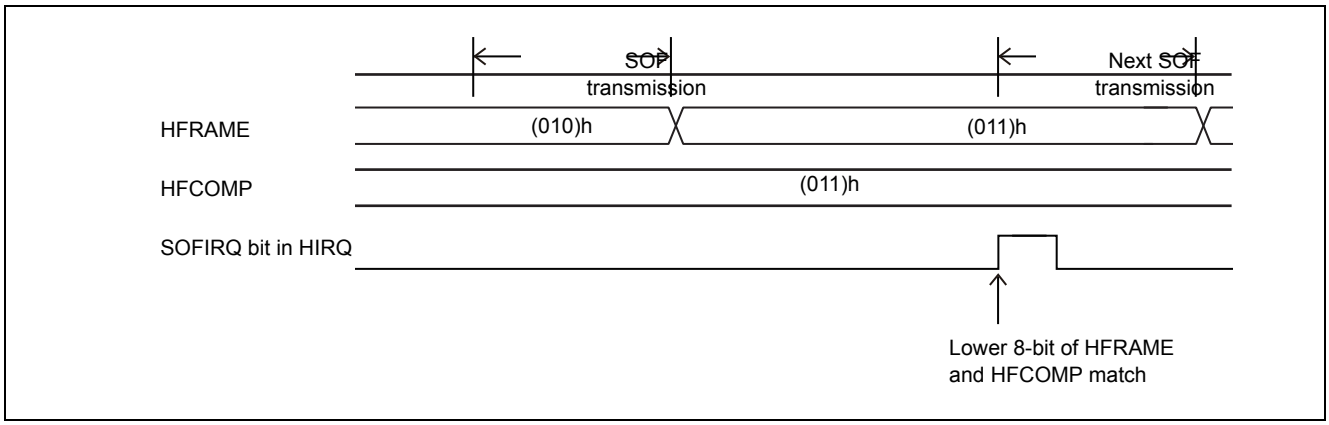


Figure 28-18. Example of SOF Interrupt Operations (SOFSTEP in Host Control Register 1 (HCNT1) = 0)



If "1" is written to the CANCEL bit in host control register 1 (HCNT1), a token other than an SOF token is set in the host token endpoint register (HTOKEN) in the EOF area, and when the SOFIRQ bit in the host interrupt register (HIRQ) changes to "1" at the next SOF transmission, that token will not be processed and the TKNEN bit in the host token endpoint register (HTOKEN) will be cleared to "000_B". In this case, the CMPIRQ bit in the host interrupt register (HIRQ) does not change to "1". Cancellation of the token can be known from the TCAN bit in the host interrupt register (HIRQ) when the SOFIRQ bit changes to "1". To process the token again, write "0" to the TCAN bit in the host interrupt register (HIRQ), and write the token to the TKNEN bit in the host token endpoint register (HTOKEN).

If "0" is written to the CANCEL bit in host control register 1 (HCNT1), the token specified in the host token endpoint register (HTOKEN) is processed after SOF transmission.

Figure 28-19 shows an example of token cancel operations, and Figure 28-20 shows an example of token operations.

Figure 28-19. Example of Token Cancel Operations (CANCEL in Host Control Register 1 (HCNT1) = 1)

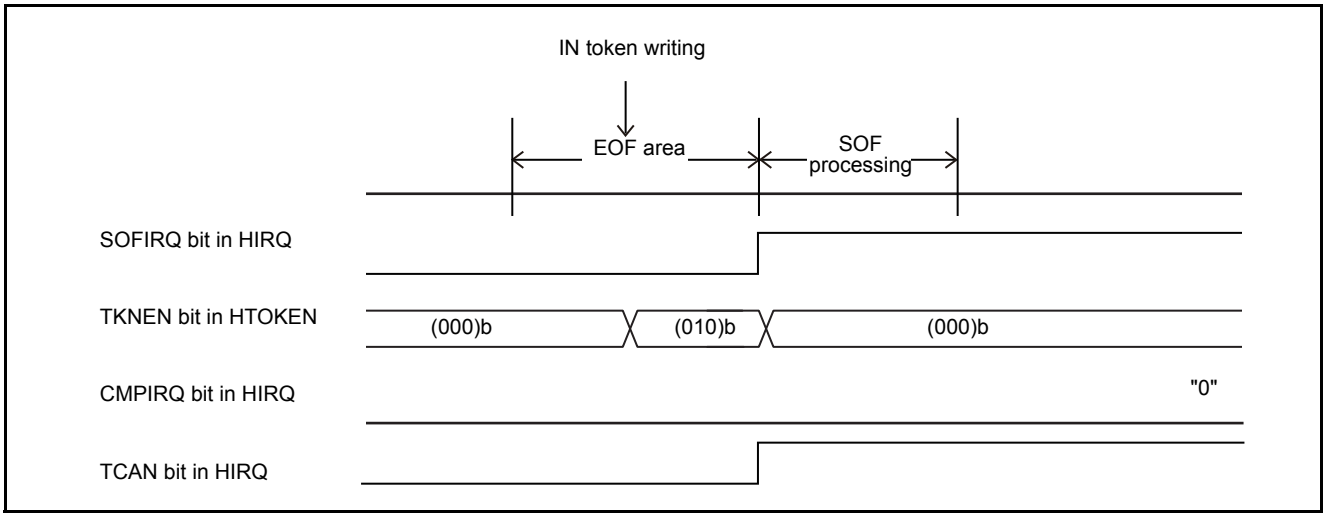
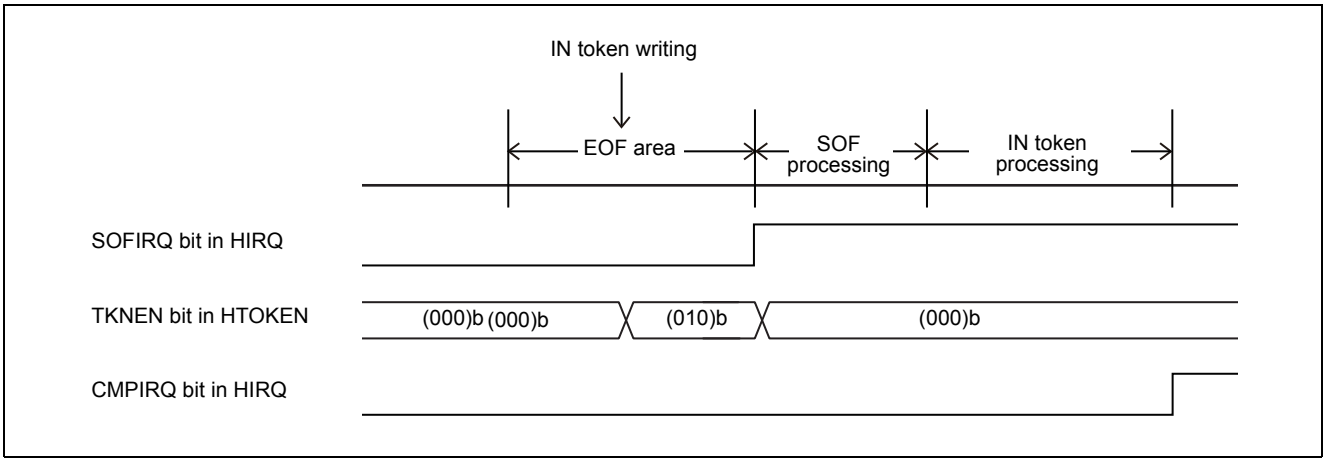


Figure 28-20. Example of Token Operations (CANCEL in Host Control Register 1 (HCNT1) = 0)



28.4.8 Error Status

USB HOST supports various error information.

This section explains error statuses.

Stuffing error

When 6 consecutive "1" bits are detected, 1-bit of "0" should be inserted. However, when 7 consecutive "1" bits are detected, a stuffing error occurs and the STUFF bit in the host error status register (HERR) changes to "1". To clear the STUFF bit, write "0" to it. If the next token is processed without clearing the STUFF bit, the bit is updated at the time that the next token is completed.

Toggle error

The toggle data for a data packet and the TGGL bit in the host token endpoint register (HTOKEN) are compared for an IN token, and if they do not match, a toggle error occurs and the TGERR bit in the host error register (HERR) changes to "1". To clear the TGERR bit in the host error register (HERR), write "0" to the TGERR bit. If the next token is processed without clearing the TGERR bit, the bit is updated at the time that the next token is completed.

CRC error

Data for a received data packet and CRC for an IN token are calculated using a polynomial expression for CRC, "$G(X) = X^{16}+X^{15}+X^{2}+1$", and a remainder other than "$800d_H$" indicates that a CRC error has occurred, which causes the CRC bit in the host error register (HERR) to change to "1". To clear the CRC bit in the host error register (HERR), write "0" to the CRC bit. If the next token is processed without clearing the CRC bit, the bit is updated at the time that the next token is completed.

Time out error

When a data packet or handshake has not been input within the given period, SE0 is detected in received data, or a stuffing error is detected, TOUT bit in the host error status register (HERR) changes to "1". To clear the TOUT bit in the host error register (HERR), write "0" to the TOUT bit. If the next token is processed without clearing the TOUT bit, the bit is updated at the time that the next token is completed.

Receive error

The received packet size is the value in the PKS bit in the EP1 control register (EP1C) when EP1 is used as the receive buffer, and the value in the PKS bit in the EP2 control register (EP2C) when EP2 is used as the receive buffer. If more data is received than the received packet size, the RERR bit in the host error status register (HERR) changes to "1". To clear the RERR bit in the host error register (HERR), write "0" to the RERR bit. If the next token is processed without clearing the RERR bit, the bit is updated at the time that the next token is completed.

28.4.9 Packet End

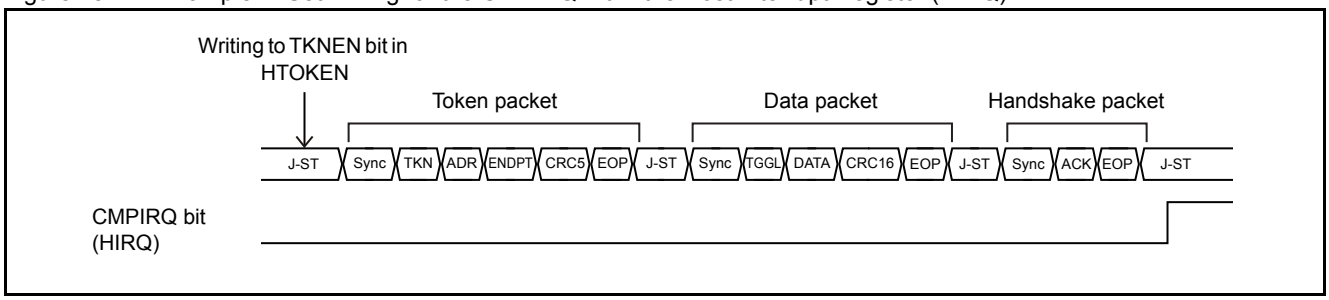
This section explains the operations at the packet end time.

The CMPIRQ bit in the host interrupt register (HIRQ) changes to "1" at the end of a packet in USB HOST. At this point, when the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated.

An interrupt flag is generated at the end of a packet.

Figure 28-21 shows an example of the packet end timing when the TKNEN bit in the host token endpoint register (HTOKEN) is "001_B", "010_B", or "011_B" (SETUP token, IN token, or OUT token).

Figure 28-21. Example 1: Set Timing for the CMPIRQ Bit in the Host Interrupt Register (HIRQ)



J-ST: J State

TKN: Token

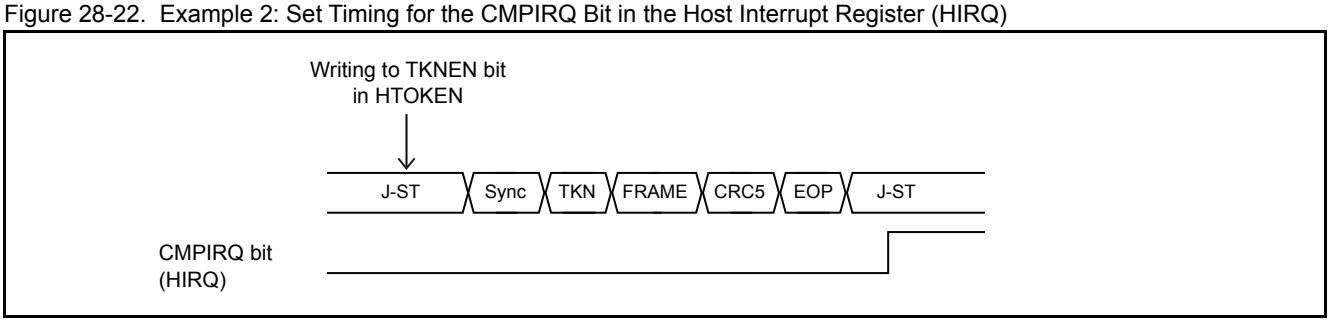
ADR: Address

ENDPT: Endpoint

TGGL: Toggle

Figure 28-22 shows an example of the packet end timing when the TKNEN bit in the host token endpoint register (HTOKEN) is "100_B" (SOF token).

Figure 28-22. Example 2: Set Timing for the CMPIRQ Bit in the Host Interrupt Register (HIRQ)



28.4.10 J-ST: J State

TKN: Token

FRAME: Frame number

28.4.11 Suspend and Resume

USB HOST supports suspend and resume.

This section explains the suspend and resume operations.

Suspend operation

Writing "1" to the SUSP bit in the host status register (HSTATE) places the USB bus in a high impedance state, stopping the circuit blocks that do not require a clock and suspending the USB circuit. The SUSP bit in the host status register (HSTATE) changes to "1" when the USB circuit enters the suspended state.

Do not specify suspend during a USB bus reset when the SOFBUSY bit in the host status register (HSTATE) is "1" and during data transmission or reception. Also, do not stop the clock feed to USB in the suspended state. The following operations can stop the clock:

- Transition to stop mode or watch mode
- Writing of "0" to the UCEN bit in the USB clock configuration register (UCCR)

Resume operation

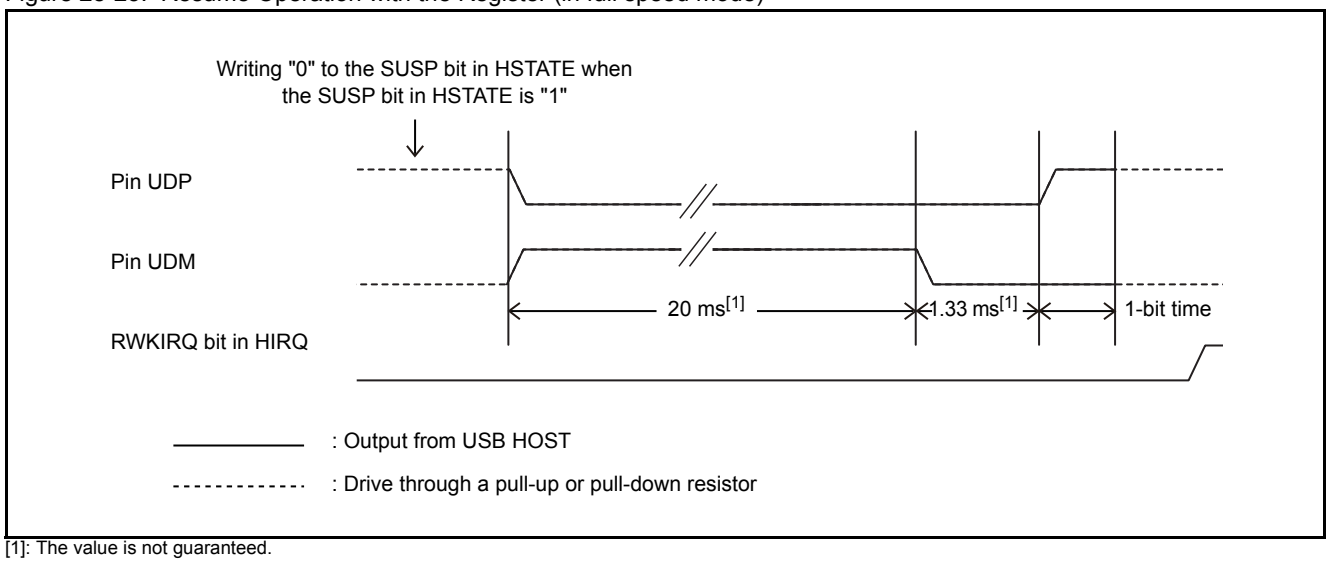
If any of the following conditions are satisfied, the resume operation from the suspended state is started:

1. "0" is written to the SUSP bit in the host status register (HSTATE).
2. UDP/UDM pins in the K State are detected.
3. The disconnection of the device is detected.
4. The connection of the device is detected.

If the RWKIRQ bit in the host interrupt register (HIRQ) changes to "1", tokens can be issued.

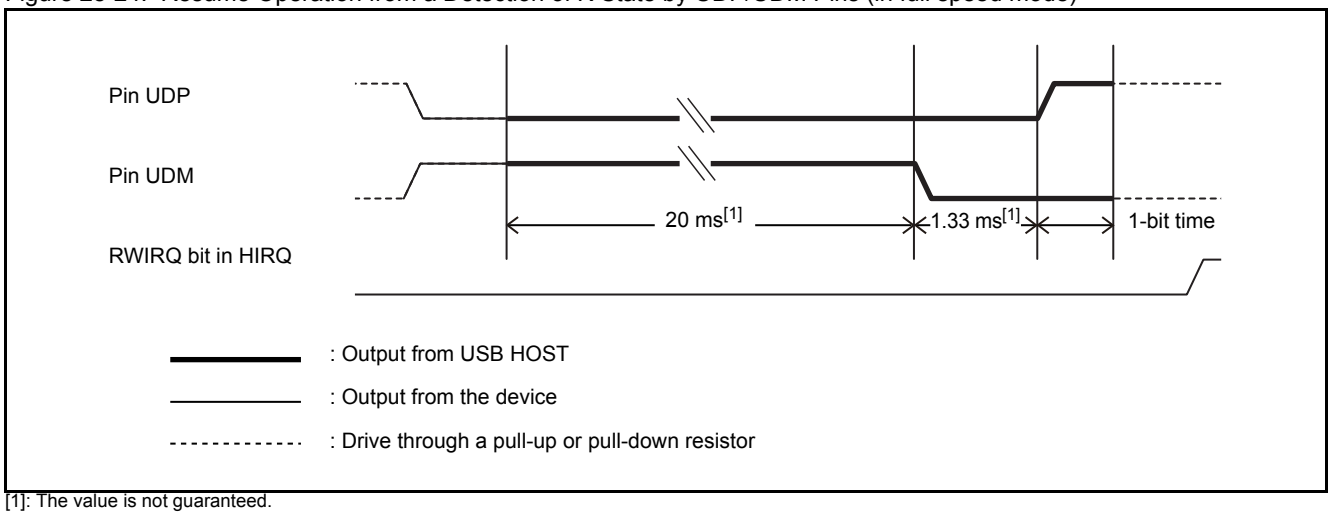
Figure 28-24, Figure 28-25, and Figure 28-26 show the resume operations caused by conditions 1, 2, 3, and 4 respectively.

Figure 28-23. Resume Operation with the Register (in full speed mode)



[1]: The value is not guaranteed.

Figure 28-24. Resume Operation from a Detection of K State by UDP/UDM Pins (in full speed mode)



[1]: The value is not guaranteed.

Figure 28-25. Resume Operation for a Disconnected Device

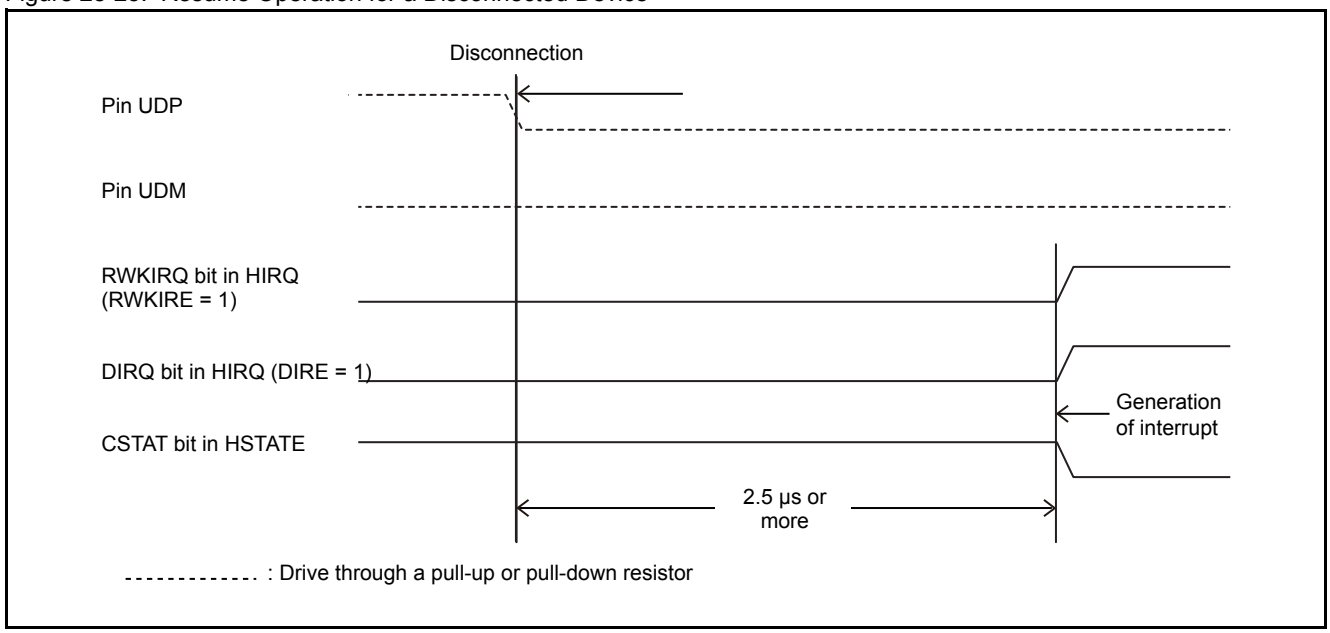
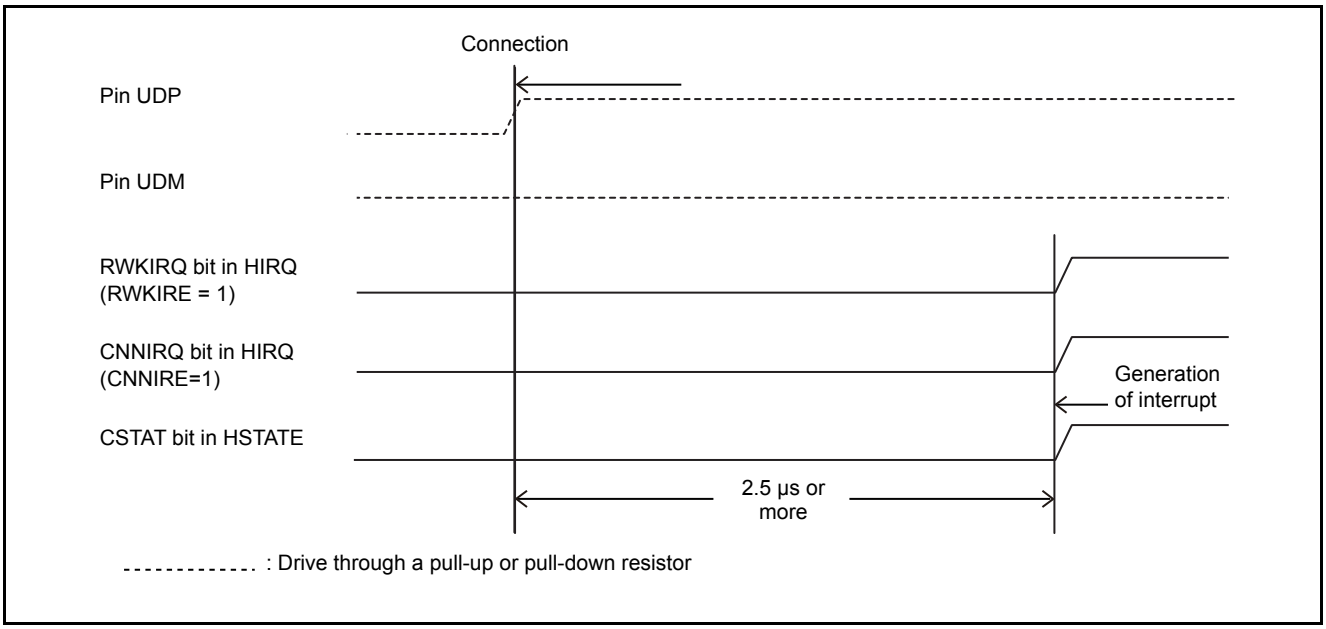


Figure 28-26. Resume Operation for a Connected Device



28.4.12 Disconnecting a USB Device

This section explains the operation of disconnecting a USB device.

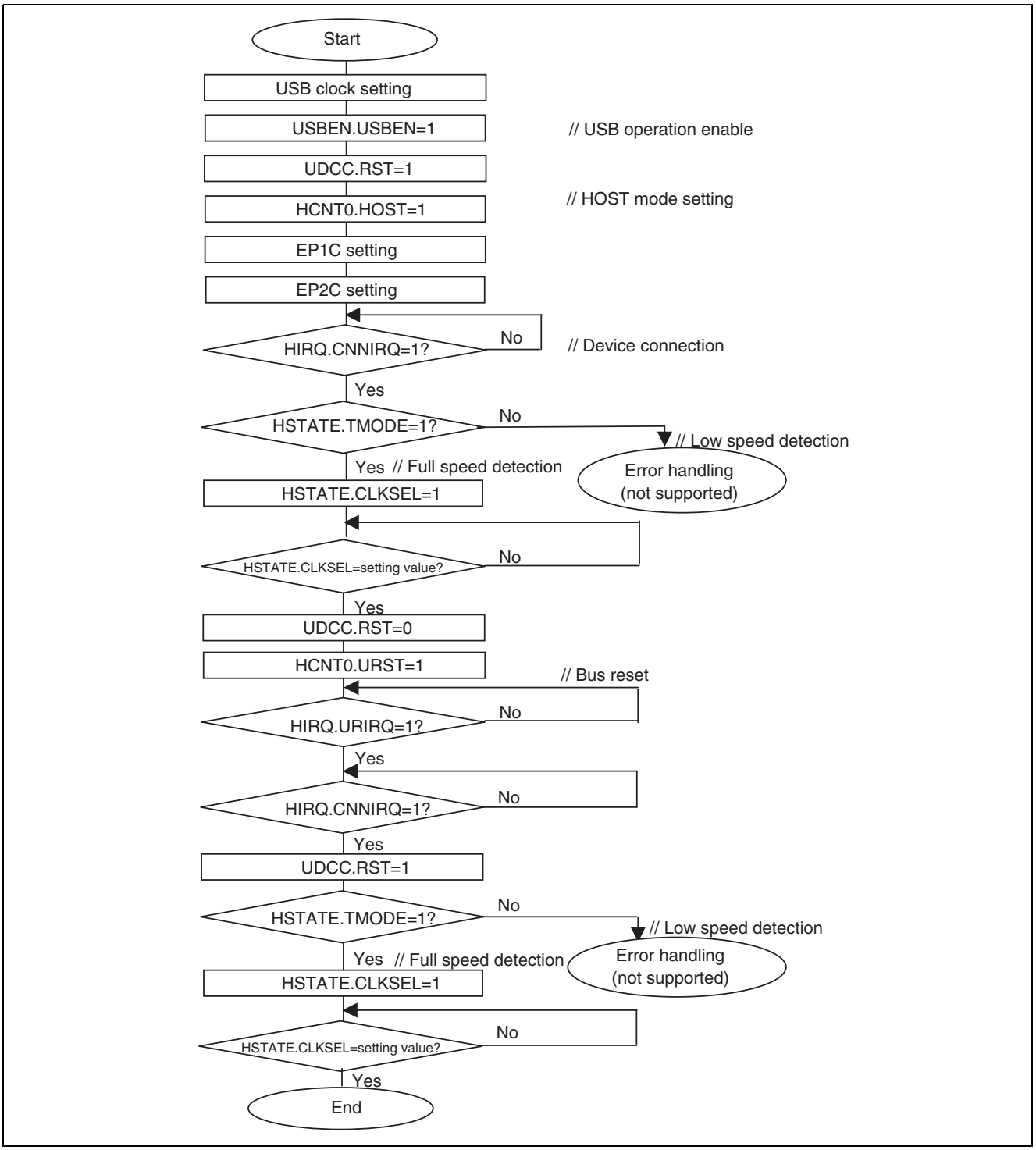
If both the UDP/UDM pins are at the "L" level for 2.5 µs or more, the USB device is judged disconnected, causing the CSTAT bit in the host status register (HSTATE) to change to "0", and the DIRQ bit in host interrupt register (HIRQ) changes to "1". If the DIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated. To clear this interrupt request, write "0" to the DIRQ bit in the host interrupt register (HIRQ).

When the USB bus is reset, the USB device is judged disconnected, which causes the CSTAT bit in the host status register (HSTATE) to change to "0", but the DIRQ bit in the host interrupt register (HIRQ) does not change to "1".

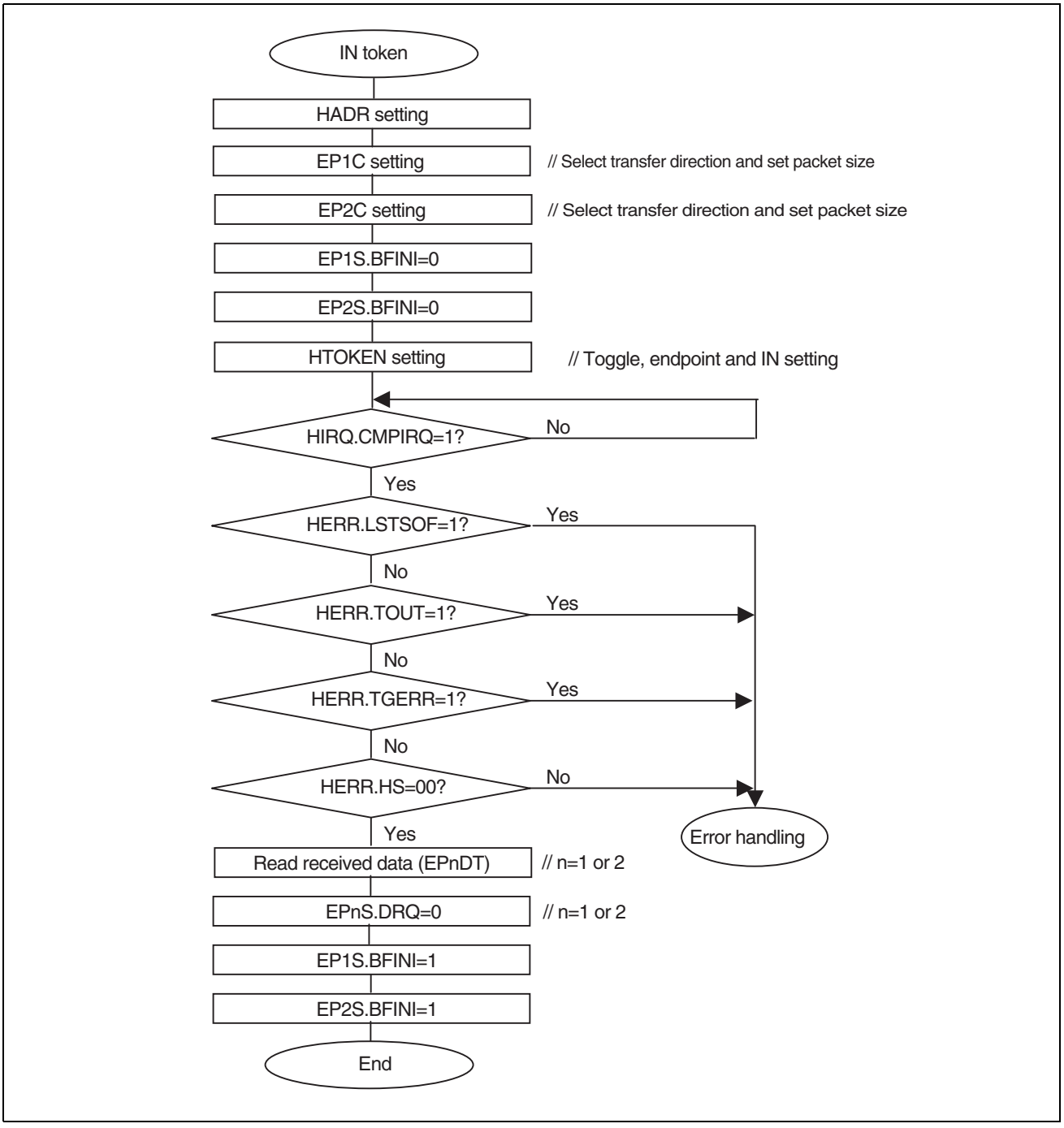
28.4.13 Flowcharts for tokens in operation with USB HOST

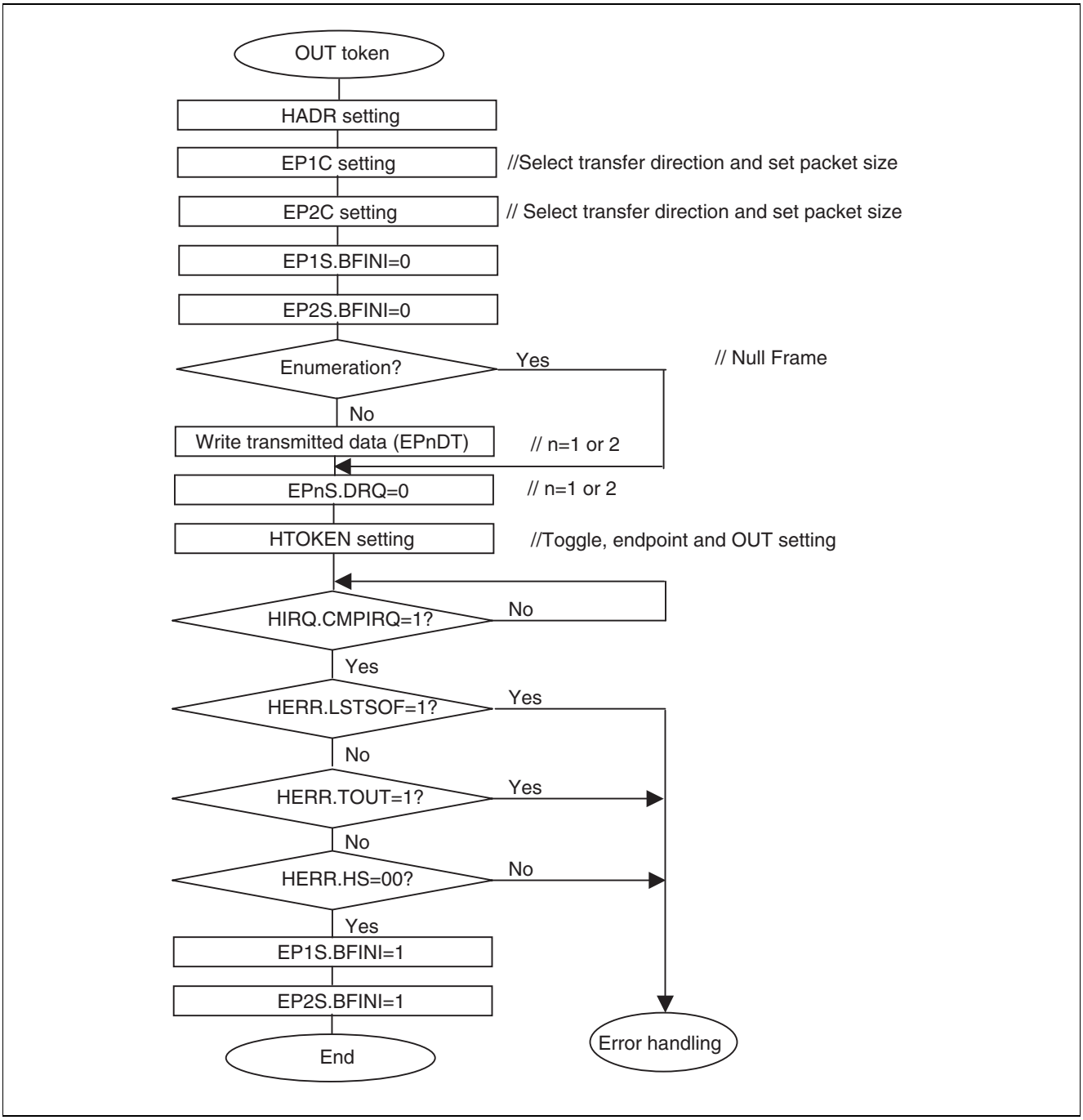
The following figures are flowcharts for tokens in operation with USB HOST.

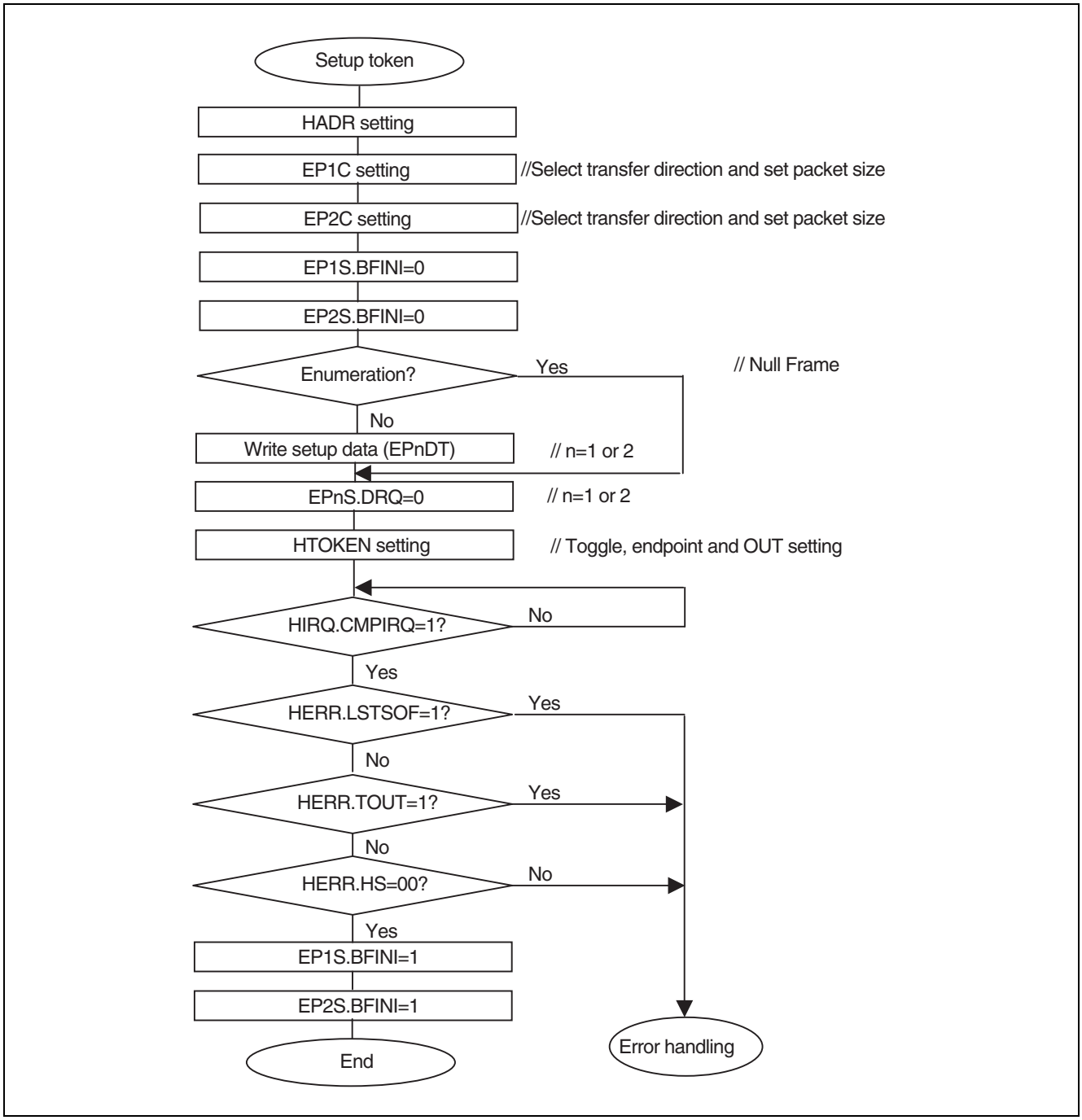
Initialization and device detection



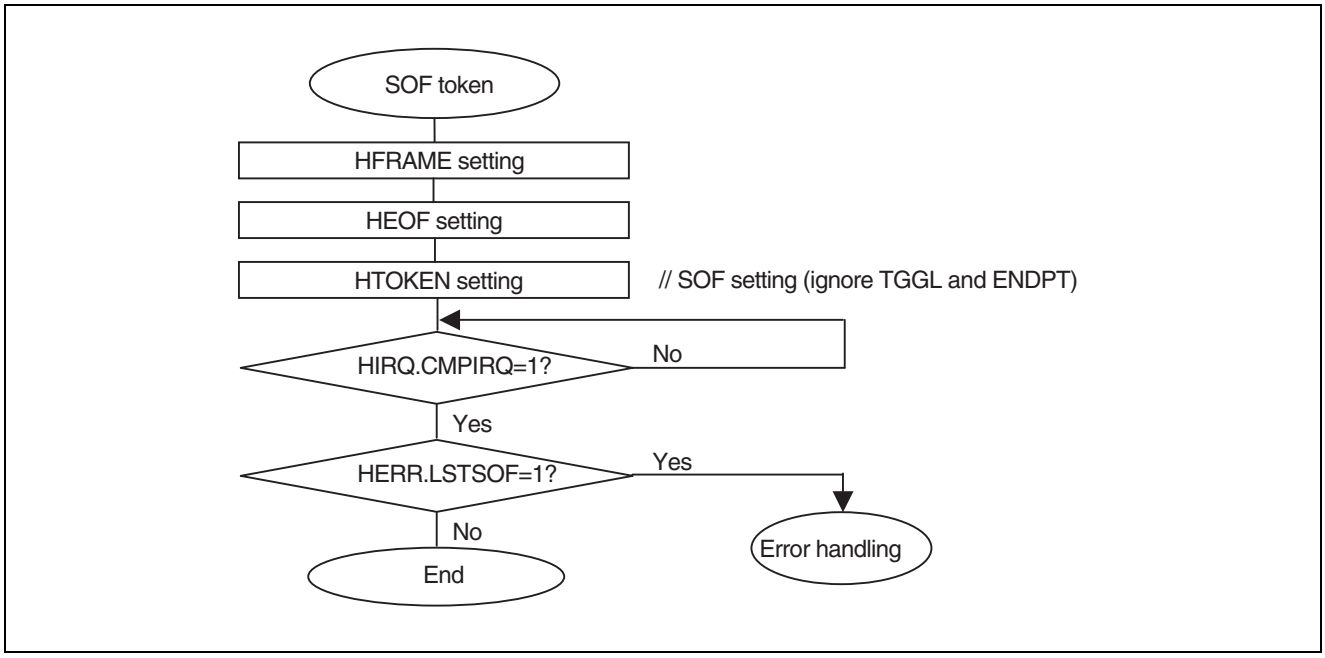
IN, OUT, SETUP tokens




OUT token
HADR setting
EP1C setting
//Select transfer direction and set packet size
EP2C setting
// Select transfer direction and set packet size
EP1S.BFINI=0
EP2S.BFINI=0
Enumeration?
Yes
// Null Frame
No
Write transmitted data (EPnDT)
// n=1 or 2
EPnS.DRQ=0
// n=1 or 2
HTOKEN setting
//Toggle, endpoint and OUT setting
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
HERR.TOUT=1?
Yes
No
HERR.HS=00?
No
Yes
EP1S.BFINI=1
EP2S.BFINI=1
End
Error handling


Setup token
HADR setting
EP1C setting
//Select transfer direction and set packet size
EP2C setting
//Select transfer direction and set packet size
EP1S.BFINI=0
EP2S.BFINI=0
Enumeration?
Yes
// Null Frame
No
Write setup data (EPnDT)
// n=1 or 2
EPnS.DRQ=0
// n=1 or 2
HTOKEN setting
// Toggle, endpoint and OUT setting
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
HERR.TOUT=1?
Yes
No
HERR.HS=00?
No
Yes
EP1S.BFINI=1
EP2S.BFINI=1
End
Error handling

SOF token


SOF token
HFRAME setting
HEOF setting
HTOKEN setting
// SOF setting (ignore TGGL and ENDPT)
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
Error handling
End

29. Remote Control Reception

This chapter describes the functions and operation of the remote control reception with HDMI-CEC and ACK auto-response.

29.1 Overview of Remote Control Reception

29.2 Remote Control Reception Registers

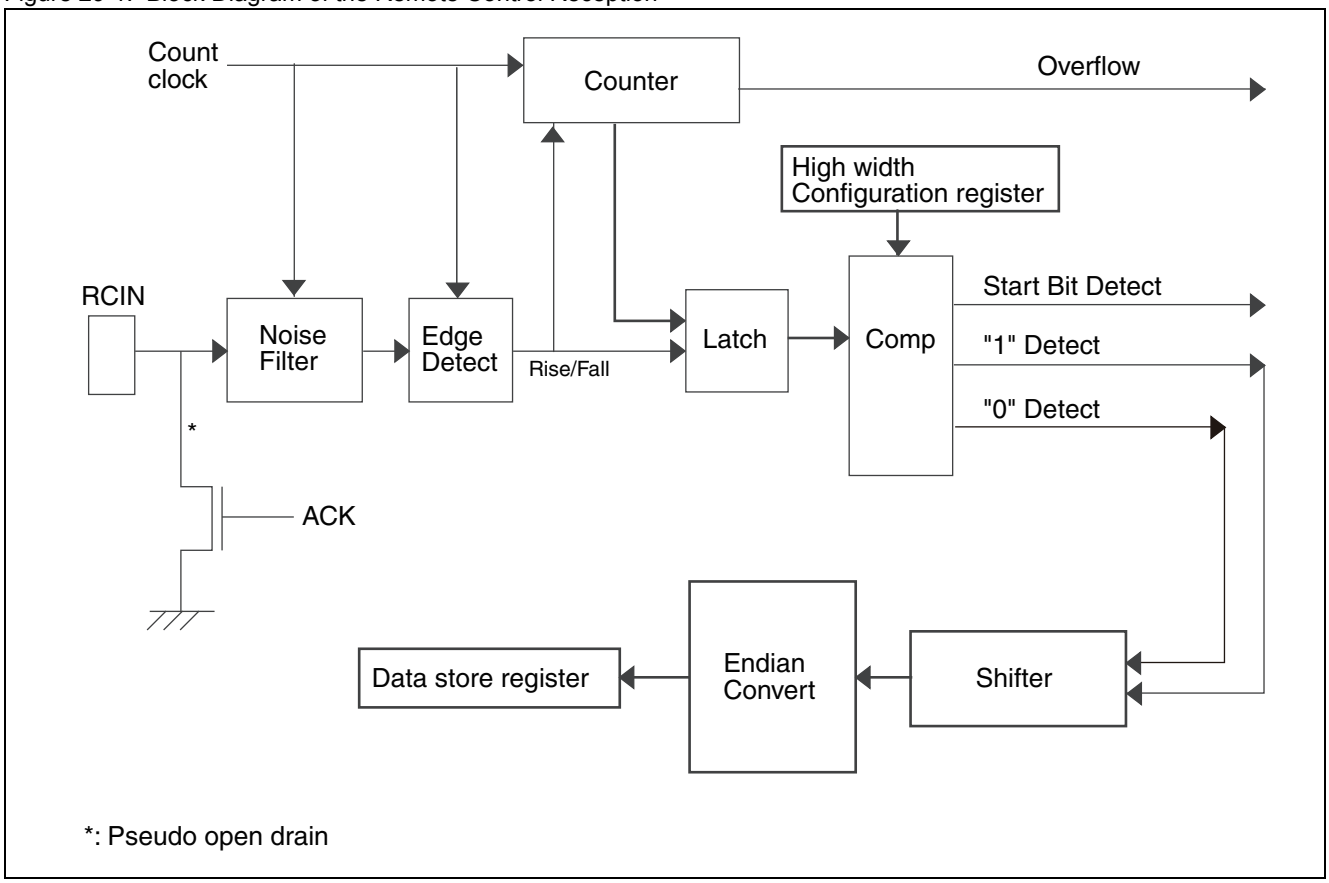
29.3 Description of Remote Control Reception Operation and Example Configuration Procedures

29.1 Overview of Remote Control Reception

This product is equipped with a remote control reception function with HDMI-CEC reception and ACK auto-response.

Block Diagram of the Remote Control Reception

Figure 29-1. Block Diagram of the Remote Control Reception



29.2 Remote Control Reception Registers

This section describes the configuration and functionality of the registers used by the remote control reception.

Remote control reception register configuration

Table 29-1 shows a list of the remote control reception registers.

Table 29-1. List of Remote Control Reception Registers

Address	Register Abbreviation	Register Name
000020_H	RCCR	Remote Control Reception Control Register[1]
000021_H	RCST	Remote Control Reception Interrupt Control Register[1]
000022_H	RCSHW	Start Bit High Width Configuration Register[1]
000023_H	RCDAHW	High Width Configuration Register A[1]
000024_H	RCDBHW	High Width Configuration Register B[1]
000025_H	Reserved	
000026_H	RCADR1	Device Address Configuration Register 1[1]
000027_H	RCADR2	Device Address Configuration Register 2[1]
000028_H	RCDTHH	Data Store Register HH
000029_H	RCDTHL	Data Store Register HL
$00002A_H$	RCDTLH	Data Store Register LH
$00002B_H$	RCDTLL	Data Store Register LL
$00002C_H$	RCCKD	Clock Divider Register[2]
$00002D_H$		

[1]: Only accessible by 8-bit access
[2]: Only accessible by 16-bit access

29.2.1 Remote Control Reception Control Register (RCCR)

Figure 29-2 shows the bit configuration of the Remote Control Reception Control Register (RCCR).

Remote control reception control register (RCCR)

Figure 29-2. Bit Configuration of the Remote Control Reception Control Register (RCCR)

bit	7	6	5	4	3	2	1	0
	THSEL	Reserved	Reserved	Reserved	ADRCE	MOD1	MOD0	EN
Attribute	R/W							
Initial value	0	–	–	–	0	0	0	0

R/W: Readable/writable

Note: This register is only accessible by 8-bit access.

[bit 7]: THSEL

Threshold value selection bit.
The initial value is "0".
Selects the criteria for judging "0" or "1" based on High Width Configuration Register A/B

Condition	THSEL	
	0	1
W > Width A W < Width B	"0" data	"1" data
W > Width A W ≥ Width B	"1" data	"0" data

[bit 6 to bit 4]: Reserved

When written	Ignored.
When read	"0" is read.

[bit 3]: ADRCE

Address compare enable bit.
The initial value is "0" (compare disabled). Setting this bit to "1" enables comparisons between the received address and device address.
When comparisons are enabled, ACK/OVF interrupts only occur when the addresses match.
In CEC mode, an ACK response is returned when the addresses match. In the case of a broadcast address, the address is treated as a match but an ACK response is not performed.
Set this to "0" when not in SIRCS mode or HDMI-CEC mode.

[bit 2, bit 1]: MOD1, MOD0

Sets the operating mode of the remote control reception.

MOD1	MOD0	Function
0	0	SIRCS mode [Initial value]
0	1	Setting prohibited
1	0	NEC/AEHA mode (repeat signal not supported)
1	1	HDMI-CEC mode

When not in SIRCS mode (i.e., when MOD1=1), the input signal is inverted internally for processing.
The High width comparison is then applied to the Low width.

[bit 0]: EN

Operation enabled bit.
The remote control reception begins operating when this bit is set to "1".
The initial value is "0" (stopped).
Do not change the following Configuration Registers or bits while this bit is "1" (running).

THSEL bit, ADRCE bit, or MOD bit of the RCCR register
OVFSEL bit of the RCST register
RCSHW, RCDAHW, RCDBHW, RCADR1, RCADR2, or RCCKD register

29.2.2 Remote Control Reception Interrupt Control Register (RCST)

Figure 29-3 shows the bit configuration of the Remote Control Reception Interrupt Control Register (RCST).

Remote control reception interrupt control register (RCST)

Figure 29-3. Bit Configuration of the Remote Control Reception Interrupt Control Register (RCST)

bit	7	6	5	4	3	2	1	0
	STIE	ACKIE	OVFIE	OVFSEL	ST	ACK	EOM	OVF
Attribute	R/W							
Initial value	0							

R/W: Readable/writable

Note: This register is only accessible by 8-bit access.

[bit 7]: STIE

Start bit interrupt enable bit.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

[bit 6]: ACKIE

ACK interrupt enable bit.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

This bit is only effective in CEC mode.

[bit 5]: OVFIE

Counter overflow interrupt enable bit.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

This interrupt only occurs when an overflow occurs after a start bit is detected.
This interrupt does not occur when a start bit has not been detected.

[bit 4]: OVFSEL

Overflow detection condition selection bit.

Value	Description
0	An overflow occurs when the counter counts 128 clocks.
1	An overflow occurs when the counter counts 256 clocks.

[bit 3]: ST

Start bit detected indicator bit.

Value	Description
0	Start bit not detected
1	Start bit detected

Cleared by writing "0".
When the STIE bit is "1", an interrupt occurs when a start bit is detected.

[bit 2]: ACK

ACK detected indicator bit.

Value	Description
0	ACK not detected
1	ACK detected

Cleared by writing "0".
When the ACKIE bit is "1", an interrupt occurs when an ACK is detected.
If address comparison is enabled, the interrupt occurs only when the addresses are matched.
This bit is only valid in CEC mode.

[bit 1]: EOM

EOM detected indicator bit.

Value	Description
0	EOM not detected
1	EOM detected

Cleared by writing "0".
This bit is only valid in CEC mode.

[bit 0]: OVF

Counter overflow detected indicator bit.

Value	Description
0	Counter overflow not detected
1	Counter overflow detected

If address comparison is enabled, an interrupt only occurs if the addresses match.
Cleared by writing "0".
In SIRCS mode, the OVF flag is not set until the second byte is received.

29.2.3 Device Address Configuration Register 1, 2 (RCADR1, RCADR2)

Figure 29-4 shows the bit configuration of Device Address Configuration Register 1 and 2 (RCADR1, RCADR2)

Device address configuration register 1, 2 (RCADR1, RCADR2)

Figure 29-4. Bit Configuration of Device Address Configuration Register 1, 2 (RCADR1, RCADR2)

bit	7	6	5	4	3	2	1	0
	Reserved			RCADR1, 2				
Attribute	R/W							
Initial value	–	–	–	0	0	0	0	0

R/W: Readable/writable

Note: These registers are only accessible by 8-bit access.

[bit 7 to bit 5]: Reserved bits

When written	Ignored.
When read	"0" is read.

[bit 4 to bit 0]: RCADR1, 2

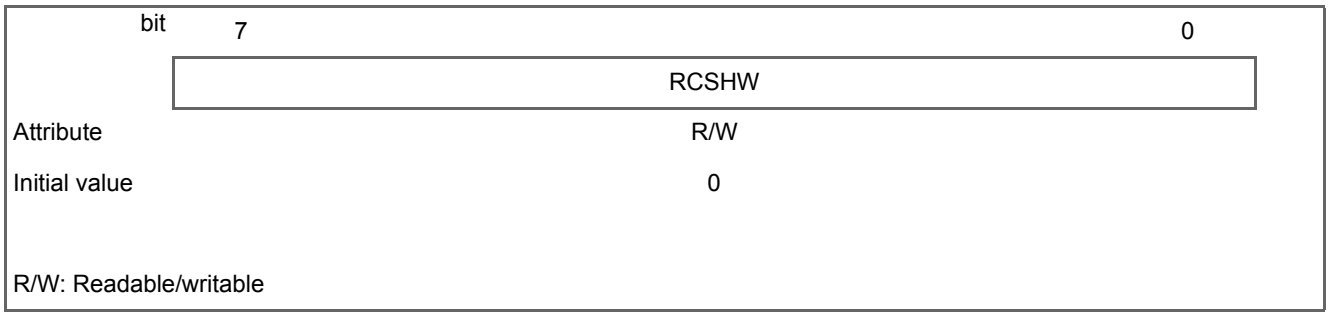
These registers set the address of the device (receiver).
The address configured in this register is compared to the device address received from the remote control and the HDMI-CEC destination.
When in HDMI-CEC mode, do not set this register to $0F_H$ (the broadcast address).

29.2.4 Start Bit High Width Configuration Register (RCSHW)

Figure 29-5 shows the bit configuration of the Start Bit High Width Configuration Register (RCSHW).

Start bit high width configuration register (RCSHW)

Figure 29-5. Bit Configuration of the Start Bit High Width Configuration Register (RCSHW)



Note: This register is only accessible by 8-bit access.

This register sets the High duration of the start bit.
A start bit is detected when the width of the received High exceeds the set value.
If the High width of the received signal is less than the set value, a start bit is not detected and the receiver re-enters the start bit High detection wait state.
When OVFSEL=0, set RCSHW ≤ 127 (a value that does not exceed the overflow detection).

29.2.5 High Width Configuration Register A (RCDAHW)

Figure 29-6 shows the bit configuration of the High Width Configuration Register A (RCDAHW).

High width configuration register A (RCDAHW)

Figure 29-6. Bit Configuration of the High Width Configuration Register A (RCDAHW)

bit	7	0
	RCDAHW	
Attribute	R/W	
Initial value	0	

R/W: Readable/writable

Note: This register is only accessible by 8-bit access.

This is the A register for setting the High duration of the start bit.
Set the value of this register such that 2 ≤ RCDAHW < RCDBHW.
Furthermore, when in CEC mode, set RCDAHW < 46 (less than the ACK response pulse width).

29.2.6 High Width Configuration Register B (RCDBHW)

Figure 29-7 shows the bit configuration of the High Width Configuration Register B (RCDBHW).

High width configuration register B (RCDBHW)

Figure 29-7. Bit Configuration of the High Width Configuration Register B (RCDBHW)

bit	7	0
	RCDBHW	
Attribute	R/W	
Initial value	0	

R/W: Readable/writable

Note: This register is only accessible by 8-bit access.

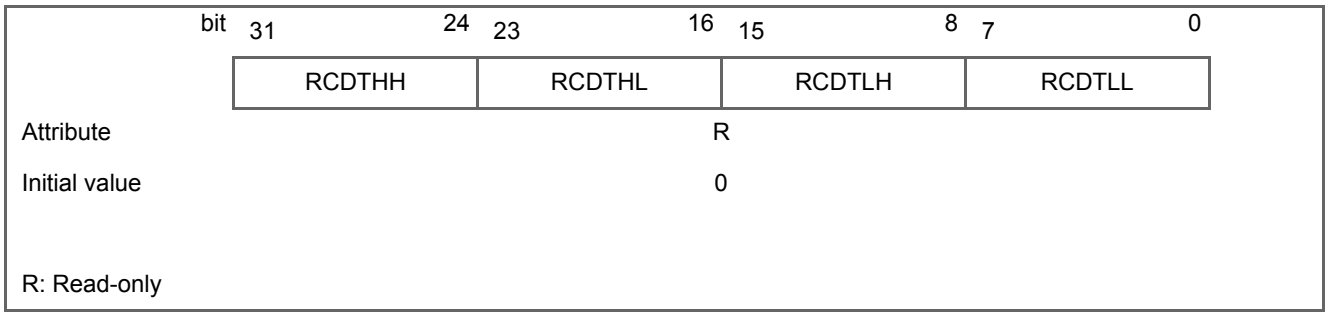
This is the B register for setting the High duration of the start bit.
Do not set this to a value smaller than RCCDAHW.
Always set this register such that RCCDAHW < RCCDBHW < RCSHW.

29.2.7 Data Store Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL)

Figure 29-8 shows the bit configuration of the Data Store Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL).

Data store register (RCDTHH, RCDTHL, RCDTLH, RCDTLL)

Figure 29-8. Bit Configuration of the Data Store Register (RCDTHH, RCDTHL, RCDTLH, RCDTLL)



This register stores received data.
In CEC mode, received data is stored in RCDTHH.
In remote control mode, data is stored in order from RCDTHH for each 8 bits received.
When a counter overflow interrupt occurs, the bits that have been received up to that point are stored packed against the MSB.
When the EN bit of the RCCR register is "0", values read from this register are undefined.
If a signal that exceeds 4 bytes is received, the excess part is ignored and is not updated in the register.

29.2.8 Clock Division Configuration Register (RCCKD)

Figure 29-9 shows the bit configuration of the Clock Division Configuration Register (RCCKD).

Clock division configuration register (RCCKD)

Figure 29-9. Bit Configuration of the Clock Division Configuration Register (RCCKD)

bit	15	13	12	11	0
	Reserved		CKSEL	CKDIV	
Attribute	R/W		R/W	R/W	
Initial value	–		0	0	

R/W: Readable/writable

Note: This register is only accessible by 16-bit access.

[bit 15 to bit 13]: Reserved bits

When written	Ignored.
When read	"0" is read.

[bit 12]: CKSEL

Operating clock selection bit.

Value	Description
0	Selects the clock obtained by dividing the resource clock.
1	Selects the source oscillator clock (32 kHz).

When the sub clock is selected using the CKSEL bit of the Clock Division Configuration Register (RCCKD), write "1" to the SCEN bit of the Clock Source Selection Register (CSELR) to start subclock oscillation.

[bit 11 to bit 0]: CKDIV

Selects the division ratio of the resource clock.
The division ratio is CKDIV + 1.
This can be set from division by 1 (no division) to division by 4096 (no division is performed when CKSEL=1).

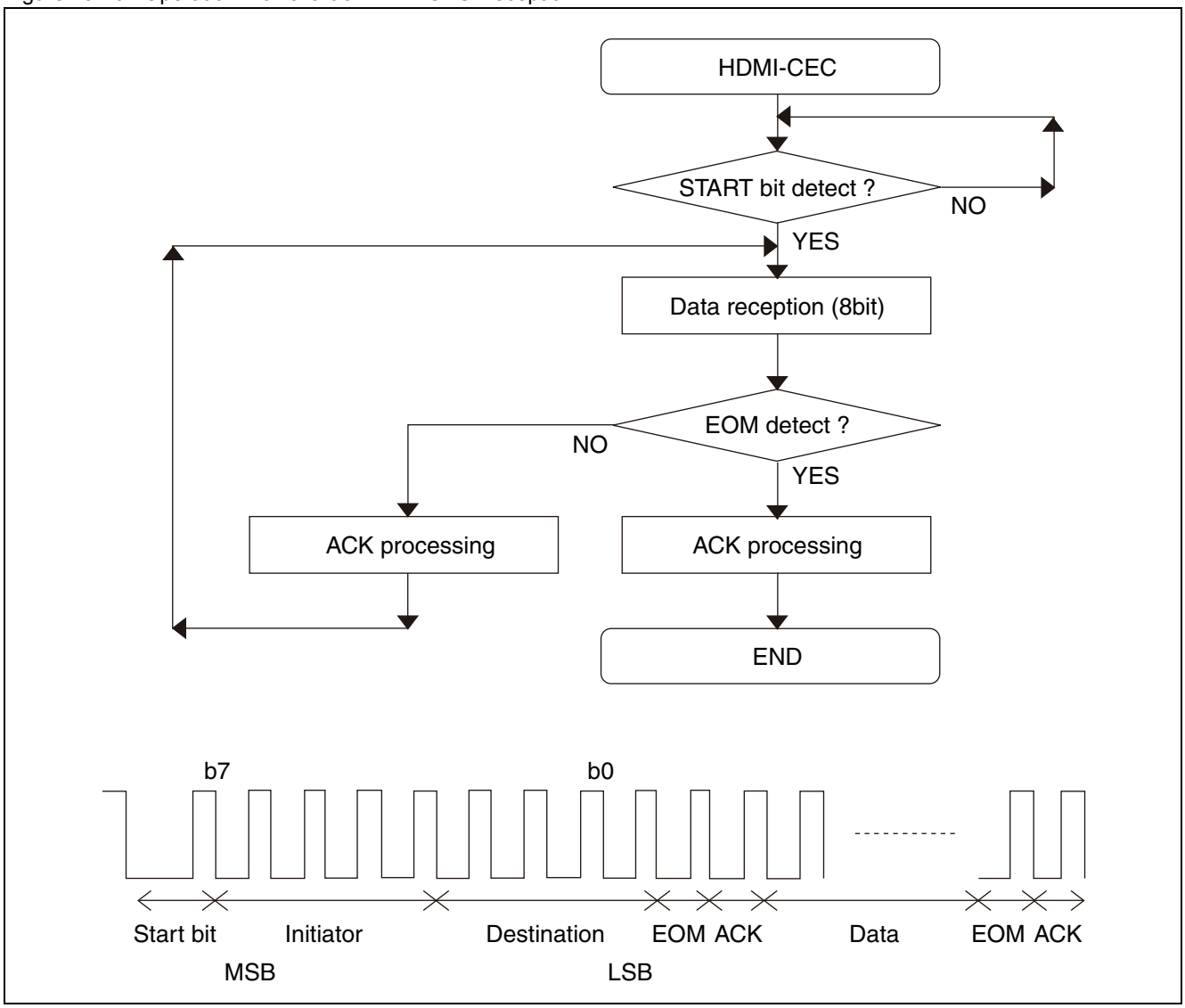
29.3 Description of Remote Control Reception Operation and Example Configuration Procedures

This section describes the operation of the remote control reception and gives examples of the settings for configuring each of the operating states.

Operation flowchart of the remote control reception

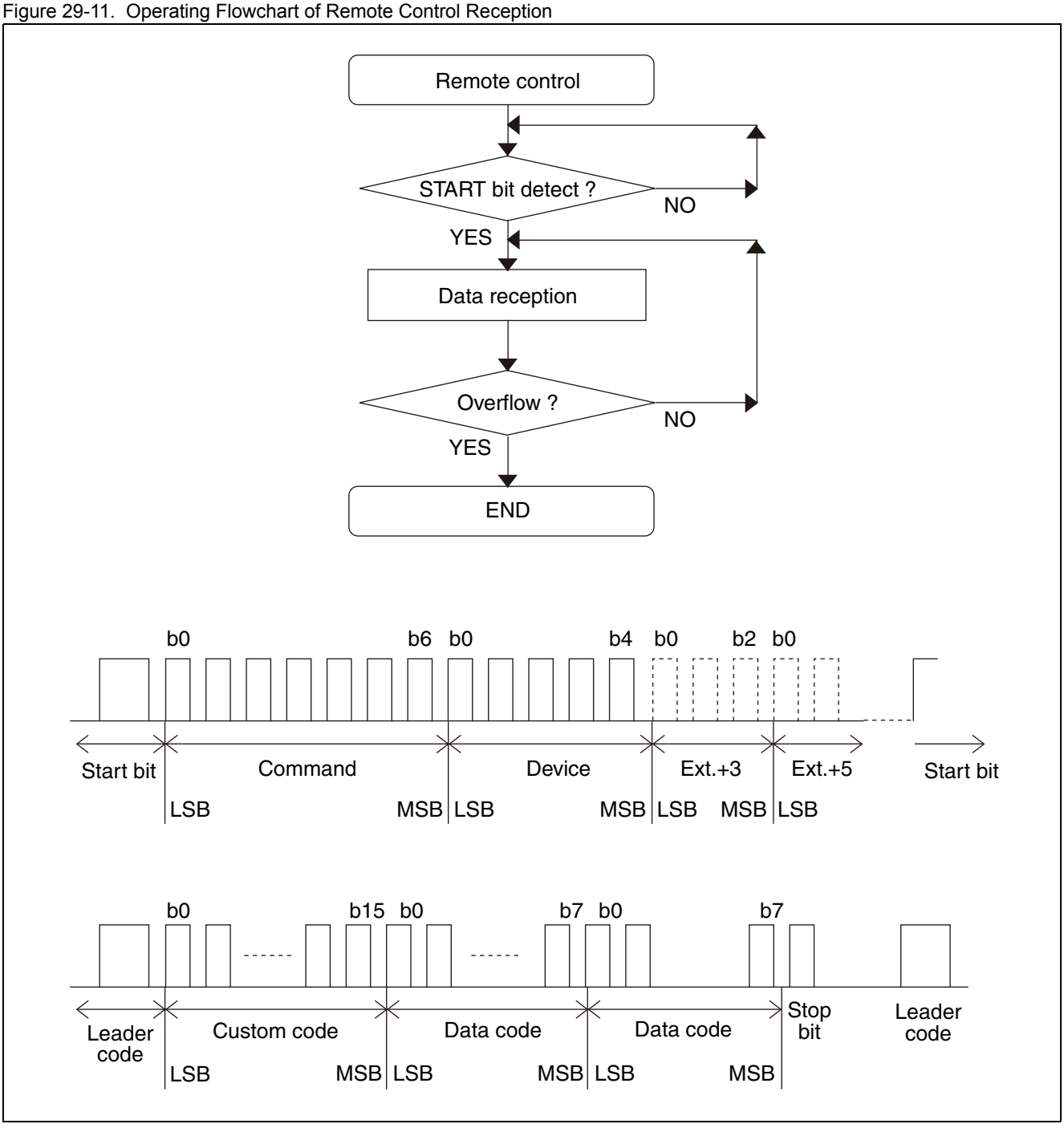
- HDMI-CEC reception

Figure 29-10. Operation Flowchart of HDMI-CEC Reception



- Remote control reception

Figure 29-11. Operating Flowchart of Remote Control Reception



Remote control reception configuration examples

Table 29-2. Example Configuration for HDMI-CEC

Register	Settings	Remarks
Remote Control Reception Control Register	MOD=11, THSEL=1, ADRCE=1	
Remote Control Reception Interrupt Control Register	ACKIE=1, OVFSEL=1, OVFIE=1	(7.8 ms)
Start Bit High Width Configuration Register	114	3.5 ms
High Width Configuration Register A	13	0.4 ms
High Width Configuration Register B	42	1.3 ms

Table 29-3. Example Configuration for Remote Control (SIRCS)

Register	Setting	Remarks
Remote Control Reception Control Register	MOD=00, THSEL=0, ADRCE=1	
Remote Control Reception Interrupt Control Register	ACKIE=0, OVFSEL=0, OVFIE=1	3.9 ms
Start Bit High Width Configuration Register	76	2.3 ms
High Width Configuration Register A	17	0.52 ms
High Width Configuration Register B	37	1.1 ms

Table 29-4. Example Configuration for Remote Control (NEC)

Register	Setting	Remarks
Remote Control Reception Control Register	MOD=10, THSEL=0	
Remote Control Reception Interrupt Control Register	ACKIE=0, OVFSEL=1, OVFIE=1	7.8 ms
Start Bit High Width Configuration Register	144	4.4 ms
High Width Configuration Register A	15	0.46 ms
High Width Configuration Register B	52	1.6 ms

30. OSDC

This chapter describes the OSDC.

30.1 OSDC Specifications

30.2 Display Functions

30.3 Control Functions

30.4 Display Control Commands (Main/OSDC Operation)

30.5 Display Control Commands (Sub Operation)

30.6 Display Control Command Write Conditions and Update Timing

30.7 Font Memory • CPU Read Access

30.1 OSDC Specifications

This chapter describes the specifications of the OSDC.

OSDC specifications

30.1.1 Features

30.1.2 Block Diagram

30.1.1 Features

The OSDC is an on-screen display controller capable of high resolution display of a maximum of 60 columns x 32 lines at a maximum resolution of 32 dots x 32 dots per character. The OSDC has a built-in palette circuit capable of displaying 256 colors from a selection of 65536 colors.

The OSDC also offers a variety of display functions, including main and sub screens, sprites, text and screen background text display, and graphics display functions.

Features

The features of the OSDC are as follows.

- Screen display capacity

  Main screen:Maximum 60 columns × 32 lines (maximum 1920 characters)

  Sub screen:Maximum 60 columns × 32 lines (maximum 1920 characters)

- Character sizes

  L size:[X] 32 dots × [Y] 2 to 32 (configurable in units of 2 dots)

  M size:[X] 24 dots × [Y] 2 to 32 (configurable in units of 2 dots)

  S size:[X] 16 dots × [Y] 2 to 32 (configurable in units of 2 dots)

  - The L, M, and S sizes are configurable for each character.
  - The Y values are configurable per line.

- Character types

  Maximum of 16384 built-in character types[1]
  [1]: Graphics characters use the space of 8 consecutive characters.

- Display modes

Normal characters/graphics characters	(configurable per character)
Trimming display (full perimeter trimming/right trimming/shadow trimming)	(configurable per line)
Character background (solid fill/shaded background/background characters)	(configurable per character)
Italic display	(configurable per character)
Underline display	(configurable per character)
Line background (solid fill/shaded background)	(configurable per line)
Enlarged display (including standard, double width, double height, double height × double width, quadruple width, quadruple height, quadruple height × quadruple width)	(configurable per line)
Blinking display	
Designation of blinking characters	(configurable per character)
Blink mode selection	(configurable per character)
Blink period and duty ratio	(configurable per screen)

- Sprite character display (can only be used with graphics characters)

  Capable of displaying two blocks (consisting of a maximum of 2 horizontal characters × 2 vertical characters) on the main/sub screen. (The display position can be moved in units of 1 dot vertically and horizontally)

- Screen background character display (can only be used with graphics characters)

  Capable of displaying a repeating pattern (consisting of a maximum of 2 × 2 (horizontal × vertical) characters) behind the main/sub screen.

- Display colors

  Character color/background color: (configurable per character) 256 colors from 65536 colors

  Line background color/character trimming color: 256 colors from 65536 colors (configurable per line)

  Screen background color: 256 colors from 65536 colors (configurable per screen)

  Graphics character dot colors: 256 colors from 65536 colors (configurable per dot)

  Shaded background frame color (highlight/shadow): 8 types of 256 colors from 65536 colors (configurable per character)

- Display position control

  Main screen horizontal display position: Configurable in units of 1 dot

  Main screen vertical display position: Configurable in units of 1 dot

  Sub screen horizontal display position: Configurable in units of 1 dot

  Sub screen vertical display position: Configurable in units of 1 dot

  Sprite screen horizontal display position: Configurable in units of 1 dot

  Sprite screen vertical display position: Configurable in units of 1 dot

  Line spacing control: Configurable in units of 2 dots (configurable per line)

- Character/color signal output

  R[4:0], G[5:0], B[4:0]: OSD color digital output signal

  ROUT, GOUT, BOUT: OSD color analog output signal

  VOB: OSD display period signal[1]

  VOA[2:0]: OSD alpha blend display period signal*

  [1]: VOB,VOA[2:0] are common to digital and analog.
  VOA[2:0] outputs an effective data when asserting VOB.

- Interrupt functions

  Line display end detection interrupt

  Vertical sync signal detection interrupt

  VRAM fill end detection interrupt

- Clock frequency

  Digital maximum frequency: 75 MHz

  Analog maximum frequency: 50 MHz

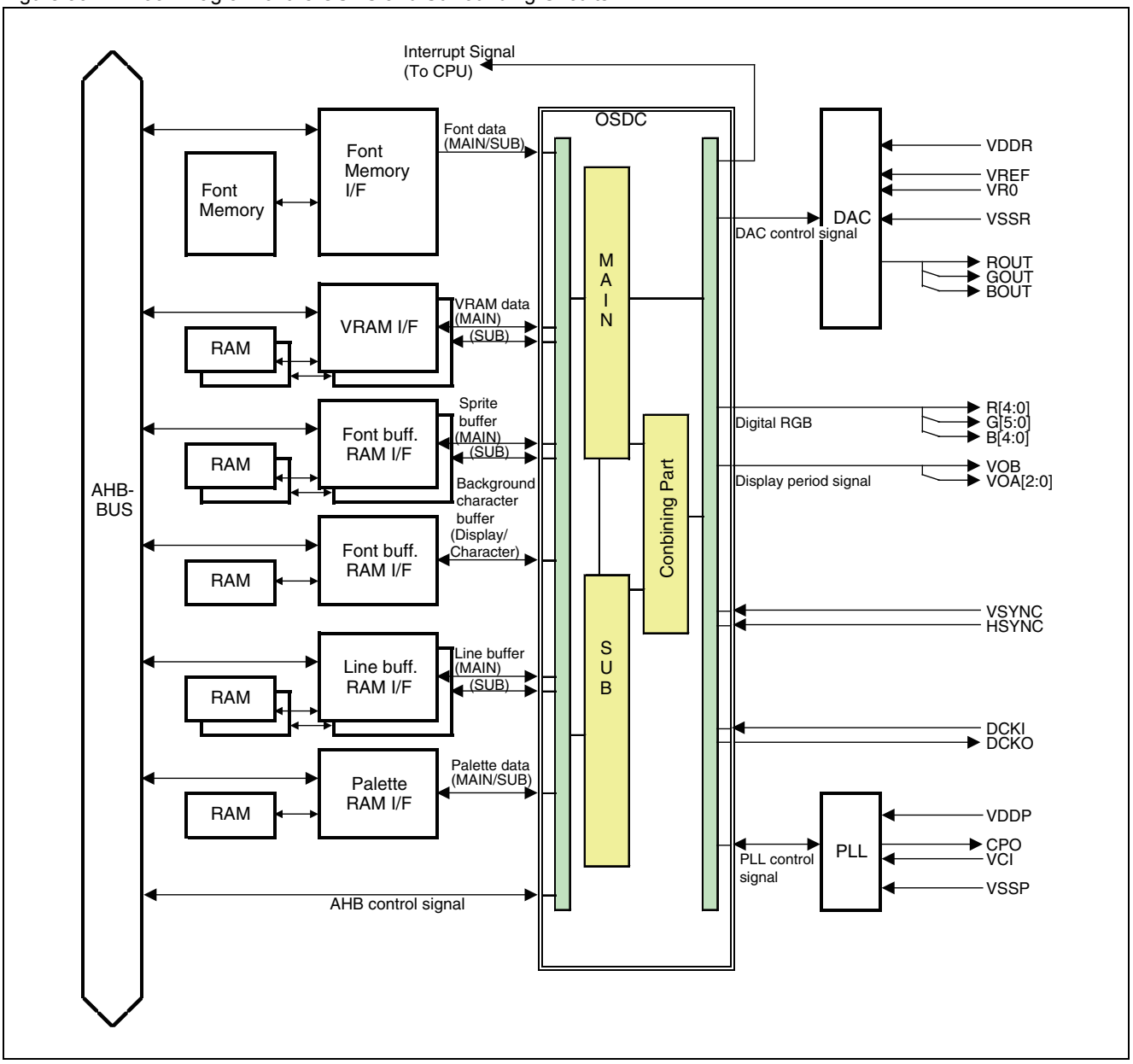
30.1.2 Block Diagram

A block diagram of the OSDC and surrounding circuits is shown below.

Block diagram

Figure 30-1 shows a block diagram of the OSDC and surrounding circuits.

Figure 30-1. Block Diagram of the OSDC and Surrounding Circuits



30.2 Display Functions

This section describes the OSDC display functions.

30.2.1 Screen Configuration
30.2.2 Screen Display Modes
30.2.3 Screen Output Control
30.2.4 Screen Display Position Control
30.2.5 Font Memory Configuration
30.2.6 Display Memory (VRAM) Configuration
30.2.7 Writing to Display Memory (VRAM)
30.2.8 Palette Configuration
30.2.9 Character Display
30.2.10 Character Background Display
30.2.11 Line Background Display
30.2.12 Screen Background Display
30.2.13 Sprite Character Display

30.2.1 Screen Configuration

The display screen is composed of a combination of various display elements.

Screen configuration

Table 30-1shows the order of precedence with which each of the display elements are combined to create the display screen when the main screen is configured as having priority (sub operation: screen output control (command 5-0), MCC=0)

Table 30-1. Screen Elements of the Display Screen (when the main screen is configured as having priority)

	Display screen name		Screen configuration	Display position control
Top layer	Sprite character (main)		One sprite (consisting of a maximum of 2 × 2 characters)	Horizontal/vertical: 1 dot
↕	Sprite character (sub)		One sprite (consisting of a maximum of 2 × 2 characters)	Horizontal/vertical: 1 dot
	Main screen	Characters (+ trimming)	60 columns × 32 lines	Horizontal/vertical: 1 dot
		Character background	60 columns × 32 lines	(Same as main characters)
		Line background	32 lines	Horizontal: Fixed Vertical: (Same as main characters)
	Sub screen	Characters (+ trimming)	60 columns × 32 lines	Horizontal/vertical: 1 dot
		Character background	60 columns × 32 lines	(Same as sub characters)
		Line background	32 lines	Horizontal: Fixed Vertical: (Same as sub characters)
	Main screen background characters		One type (consisting of a maximum of 2 × 2 characters) Full screen or window	Fixed
	Main screen background		Single color. Full screen or window	Fixed
Bottom layer	Sub screen background		Single color. Full screen or window	Fixed

Table 30-2 shows the order of precedence with which each of the display elements are combined to create the display screen when the sub screen is configured as having priority (sub operation: screen output control (command 5-0), MCC=1).

Table 30-2. Screen Elements of the Display Screen (when the sub screen is configured as having priority)

	Display screen name		Screen configuration	Display position control
Top layer	Sprite character (sub)		One sprite (consisting of a maximum of 2 × 2 characters)	Horizontal/vertical: 1 dot
↕	Sprite character (main)		One sprite (consisting of a maximum of 2 × 2 characters)	Horizontal/vertical: 1 dot
	Sub screen	Characters (+ trimming)	60 columns × 32 lines	Horizontal/vertical: 1 dot
		Character background	60 columns × 32 lines	(Same as sub characters)
		Line background	32 lines	Horizontal: Fixed Vertical: (Same as sub characters)
	Main screen	Characters (+ trimming)	60 columns × 32 lines	Horizontal/vertical: 1 dot
		Character background	60 columns × 32 lines	(Same as main characters)
		Line background	32 lines	Horizontal: Fixed Vertical: (Same as main characters)
	Sub screen background		Single color. Full screen or window	Fixed
	Main screen background characters		One type (consisting of a maximum of 2 × 2 characters) Full screen or windowed	Fixed
Bottom layer	Main screen background		Single color. Full screen or window	Fixed

Screen configuration diagram 1

Figure 30-2 shows the screen configuration diagram (overall) when the main screen is configured as having priority, and Figure 30-3 shows the screen configuration diagram (overall) when the sub screen is configured as having priority.

Figure 30-2. Screen Configuration Diagram (Overall When Main Screen Has Priority)

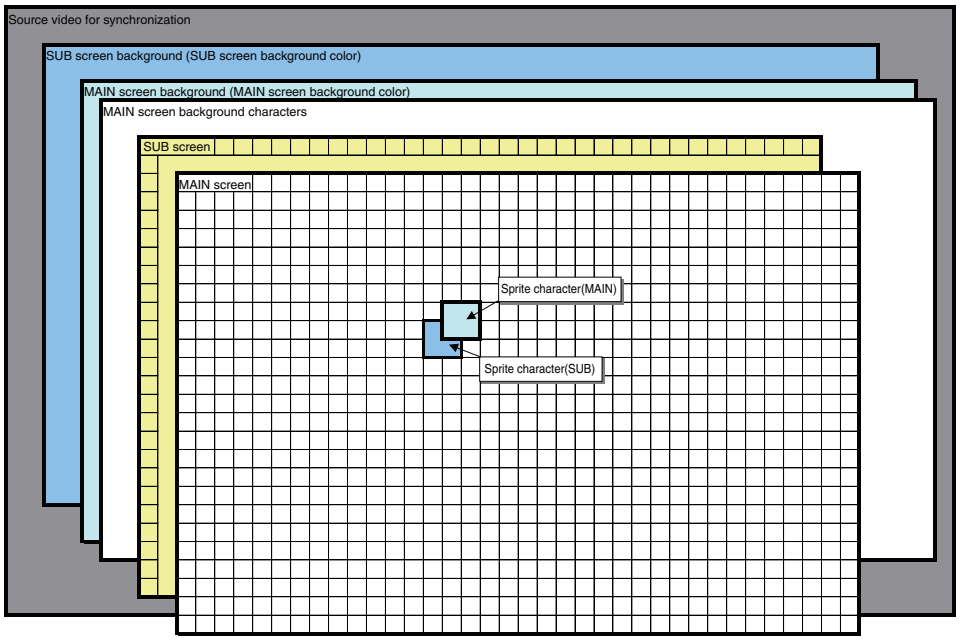
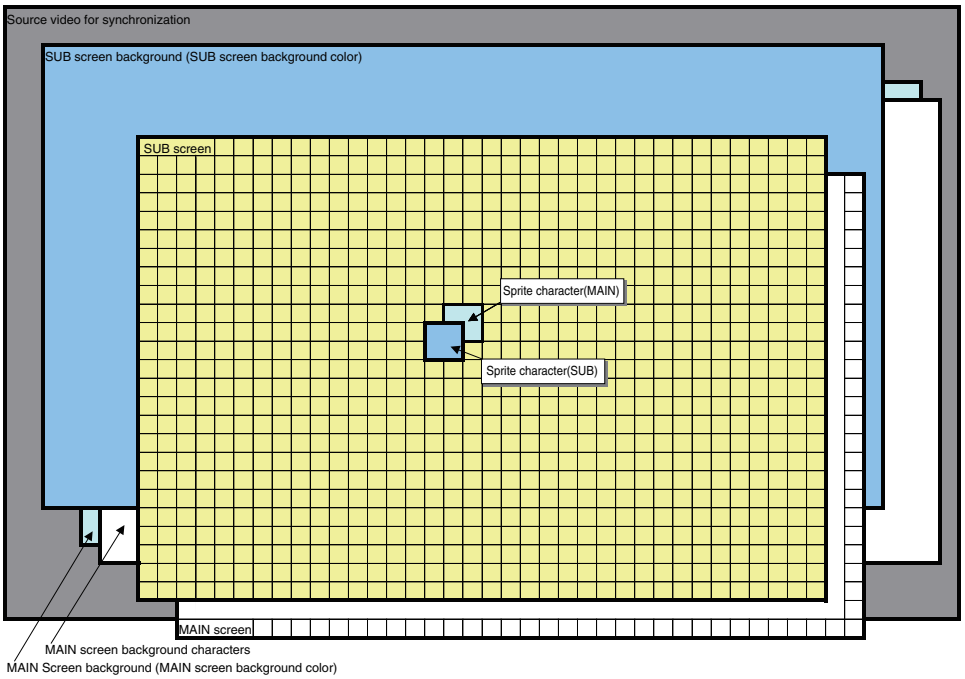


Figure 30-3. Screen Configuration Diagram (Overall When Sub Screen Has Priority)

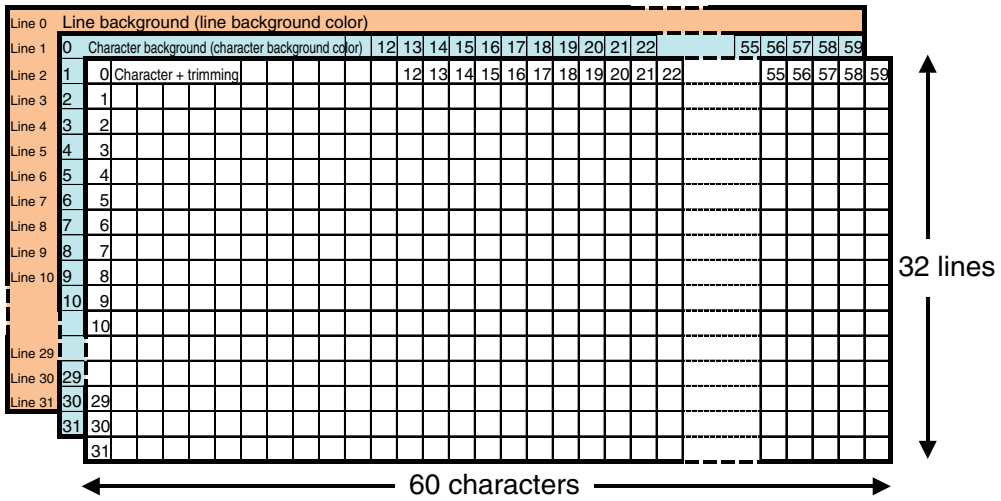


Screen configuration diagram 2

Figure 30-4 shows a diagram of the configuration of the main screen.

Note: Underlines and shaded frame regions of shaded backgrounds (line backgrounds or character backgrounds) are displayed with higher priority than characters. If the shaded frame of a line background and the shaded frame of a character background overlap, the shaded frame of the character background is displayed with higher priority. Furthermore, if a shaded frame and an underline overlap, the shaded frame is displayed with higher priority.

Figure 30-4. Configuration Diagram of the Main Screen

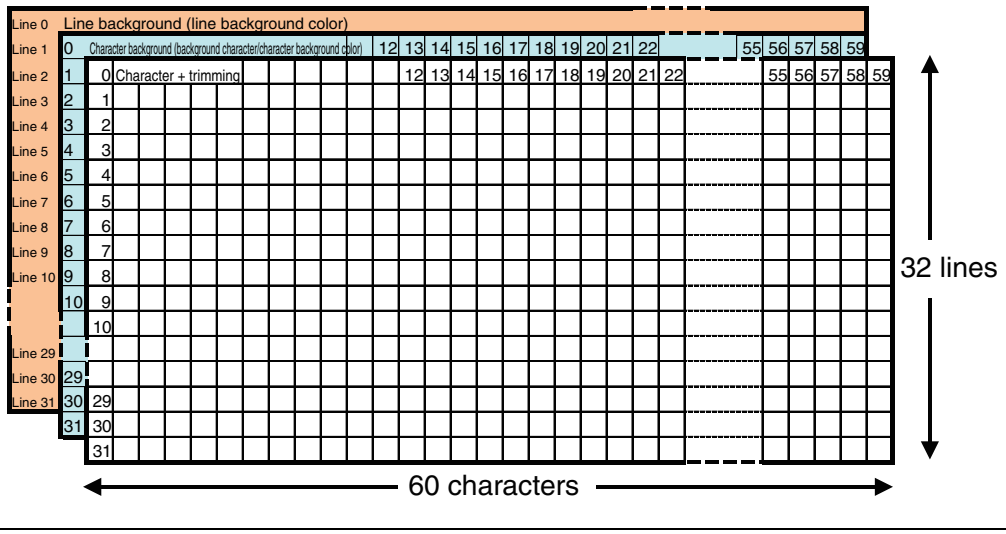


Screen configuration diagram 3

Figure 30-5 shows a diagram of the configuration of the sub screen.

Note: Underlines and shaded frame regions of shaded backgrounds (line backgrounds or character backgrounds) are displayed with higher priority than characters. If the shaded frame of a line background and the shaded frame of a character background overlap, the shaded frame of the character background is displayed with higher priority. Furthermore, if a shaded frame and an underline overlap, the shaded frame is displayed with higher priority.

Figure 30-5. Configuration Diagram of the Sub Screen



30.2.2 Screen Display Modes

Table 30-3 shows the display modes of each of the screen elements.

Screen display modes

Table 30-3. Screen Display Modes (Sheet 1 of 3)

<table>
<tr><th colspan="2">Display screen name</th><th colspan="8">Display mode</th></tr>
<tr><td colspan="2" rowspan="2">Main screen background</td><td colspan="8">Not displayed</td></tr>
<tr><td colspan="8">Displayed (Full screen, or inside or outside of the main screen window)</td></tr>
<tr><td colspan="2" rowspan="4">Main screen background characters</td><td colspan="8">Not displayed</td></tr>
<tr><td colspan="2" rowspan="4">Displayed (Full screen, or inside or outside of the main screen window)
Note: Can only be used with graphics characters</td><td colspan="2" rowspan="4">Character configuration</td><td colspan="4">Composed of 1 character</td></tr>
<tr><td colspan="4">Composed of 2 characters horizontally</td></tr>
<tr><td colspan="4">Composed of 2 characters vertically</td></tr>
<tr><td colspan="4">Composed of 2 × 2 (vertical × horizontal) characters [character background and background character display are disabled]</td></tr>
<tr><td rowspan="6">Main screen</td><td rowspan="4">Line background</td><td colspan="6">Not displayed</td><td rowspan="3">Line spacing</td><td rowspan="3">0 to 14 dots (Above Only, Below Only, or Above and Below)</td></tr>
<tr><td colspan="6">Solid background display</td></tr>
<tr><td colspan="3">Shaded background concaved display (2 dots)</td><td colspan="2" rowspan="2">Control display above or below shaded background frame</td><td>Not displayed</td></tr>
<tr><td colspan="3">Shaded background convexed display (2 dots)</td><td>Displayed</td></tr>
<tr><td rowspan="5">Character background</td><td colspan="8">Not displayed</td></tr>
<tr><td colspan="6">Solid background display</td><td rowspan="4">Extend character background (meaningful when the line spacing setting is not "0")</td><td>Normal</td></tr>
<tr><td colspan="2">Shaded background concaved display (1 to 4 dots)</td><td rowspan="2">Shaded background merge with character on right</td><td>Independent</td><td rowspan="2">Control erasing above/ below shaded background frame</td><td>Displayed</td><td></td></tr>
<tr><td colspan="2">Shaded background convexed display (1 to 4 dots)</td><td>Merged</td><td>Not displayed</td><td>Extended</td></tr>
<tr><td colspan="8">Background character display [Cannot be used when the main screen background character is composed of 2 × 2 (vertical × horizontal) characters][1]</td></tr>
</table>

Table 30-3. Screen Display Modes (Sheet 2 of 3)

Display screen name		Display mode								
Main screen	Characters	Graphic characters								
		Normal characters	Not displayed (whitespace characters)							
			Displayed[2]	Trimming output control	Not displayed					
					Only displayed when no character background	Trimming mode	Full perimeter trimming		Trimming type	1 dot width
					Displayed except when shaded background		Right trimming			
							Shadow trimming	Lower right		2 dot width
					All displayed			Lower right + right		
				Italics output control	Not displayed					
					Italics displayed					
				Underline output control	Not displayed					
					Underline displayed					
Sprite character (main)		Not displayed								
		Displayed **Note:** Can only be used with graphics characters				Character configuration		Composed of 1 character		
								Composed of 2 characters horizontally		
								Composed of 2 characters vertically		
								Composed of 2 × 2 characters (vertical × horizontal)		

Table 30-3. Screen Display Modes (Sheet 3 of 3)

Display screen name		Display mode							
Sub screen background		Not displayed							
		Displayed (Full screen, or inside or outside of the main screen window)							
Sub screen	Line background	Not displayed					Line spacing	0 to 14 dots (Above Only, Below Only, or Above and Below)	
		Solid background display							
		Shaded background concaved display (2 dots)		Control display above or below shaded background frame		Not displayed			
		Shaded background convexed display (2 dots)				Displayed			
	Character background	Not displayed							
		Solid background display					Extend character background (meaningful when the line spacing setting is not "0")	Normal	
		Shaded background concaved display (1 to 4 dots)	Shaded background right character merge	Independent	Shaded background frame top/bottom erase control	Displayed			
		Shaded background convexed display (1 to 4 dots)		Merged		Not displayed		Extended	
		Background character display [Cannot be used when the main screen background character is composed of 2 × 2 (vertical × horizontal) characters][1]							
	Characters	Graphic characters							
		Normal characters	Not displayed (whitespace characters)						
			Displayed[2]	Trimming output control	Not displayed				
					Only displayed when no character background	Trimming mode	Full perimeter trimming	Trimming type	1 dot width
							Right trimming		
					Displayed except when shaded background		Shadow trimming	Lower right	2 dot width
					All displayed			Lower right + right	
				Italic output control	Not displayed				
					Italics displayed				
				Underline output control	Not displayed				
					Underline displayed				
Sprite character (sub)		Not displayed							
		Displayed **Note:** Can only be used with graphics characters			Character configuration		Composed of 1 character		
							Composed of 2 characters horizontally		
							Composed of 2 characters vertically		
							Composed of 2 × 2 characters (vertical × horizontal)		

[1]: Background character display in the character background (character background character display) does not display correctly in either the main screen or sub screen when the screen background character setting in the main screen is set to 2 × 2 character (vertical × horizontal) composition. Therefore, in order to perform character background character display, either do not display the main screen background character display, or use a setting other than 2 × 2 character composition (vertical × horizontal).
(Care is needed because character background character display on the sub screen also has an effect on the main screen background character display and composition.)

[2]: During display, the "trimming output control", "italics output control", and "underline output control" can be controlled independently.

30.2.3 Screen Output Control

Table 30-4 shows the relationship between the control target and the control bits of the screen output control.

Screen output control

Table 30-4. Screen Output Control

Display Screen Control	
Control target	**Control bit name (unit of control)**
Main and sub characters + trimming + character background + line background	DSP (screen)
Main and sub characters + trimming + character background	LDS (line)
Main screen background characters	PDS (screen)
Main and sub screen background color	UDS (screen)
Main and sub sprite characters	SDS (screen)

30.2.4 Screen Display Position Control

The OSDC can independently start and control the display positions of each of the main and sub screens, main screen background characters, main and sub screen background color, and sprite characters (main and sub).

Display position control of the main and sub screen

Vertical display position: Screen display position control (command 5-1) bits Y10 to Y0

Sets the display position relative to the synchronization pulse of the vertical sync signal (VSYNC pin input signal).
(Simultaneously controls the display positions of characters, character backgrounds, and line backgrounds.)
Configurable from 0 to 2047 dots in units of 1 dot.

Horizontal display position: Screen display position control (command 5-1) bits X10 to X0

Sets the display position relative to the synchronization pulse of the horizontal sync signal (HSYNC pin input signal).
(Simultaneously controls the display positions of characters and character backgrounds.)
Configurable from 0 to 2047 dots in units of 1 dot.

Line spacing: Line control data setting 1 (command 3) bits LW2 to LW0

Line control data setting 2 (command 4) LWUEN and LWDEN bits

The line spacing area of characters is configured by LW2 to LW0. LWUEN and LWDEN can be configured as follows in order to further enable control line spacing above and below the characters.

- When LWUEN=1, space equal to the value of the line spacing setting (LW2 to LW0) is reserved above the characters.
- When LWDEN=1, space equal to the value of the line spacing setting (LW2 to LW0) is reserved below the characters.
- When both LWUEN and LWDEN are 1, space equal to the value of the line spacing setting (LW2 to LW0) is reserved both above and below the characters, and when both LWUEN and LWDEN are 0, the value of the line spacing setting (LW2 to LW0) has no effect.

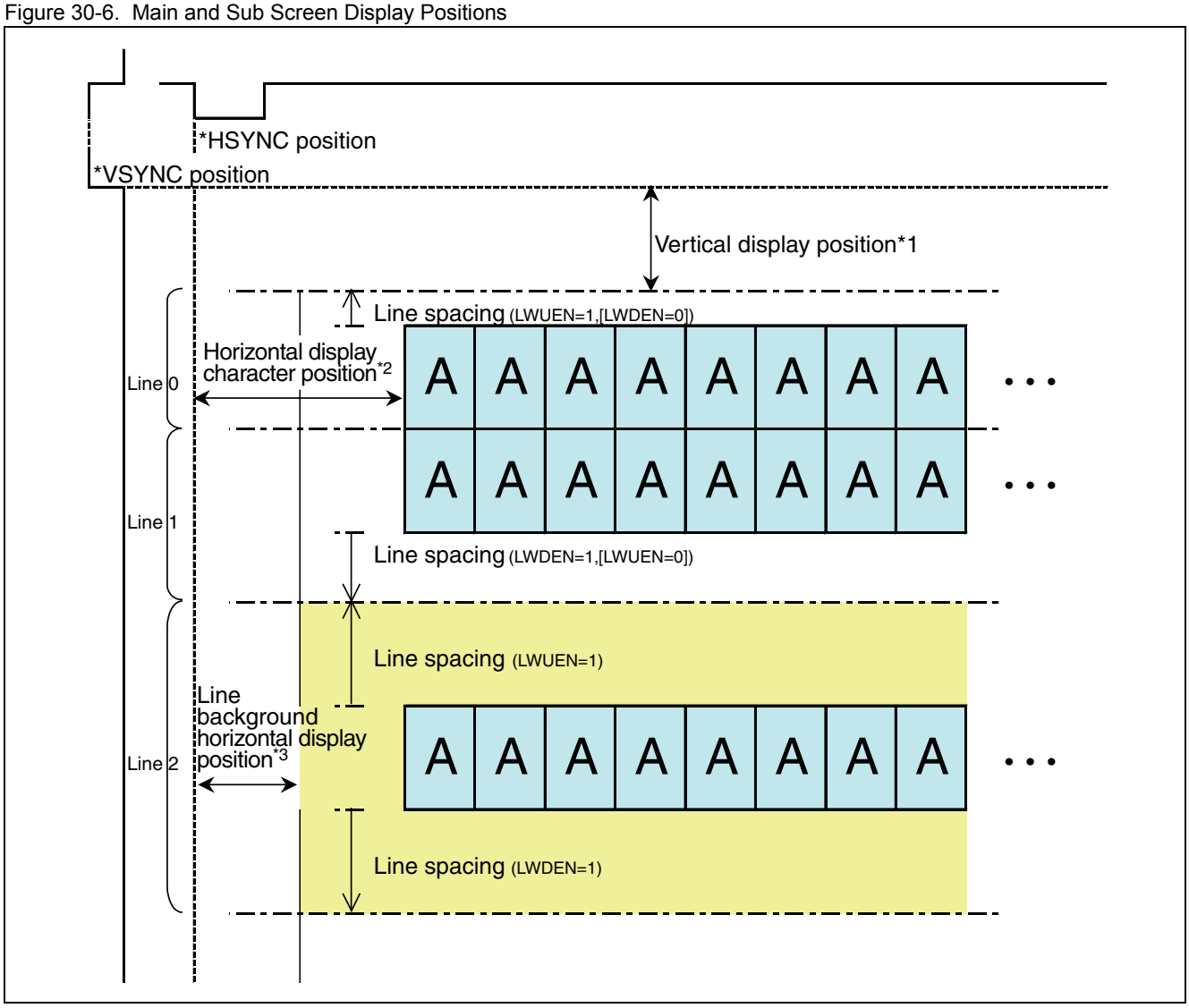
Configurable independently for each line from 0 to 14 dots in units of 2 dots, with spacing Above Only, Below Only, or Above and Below.

Note: When line magnification is specified, the line spacing is displayed magnified.

Note: For lines where the character vertical size is set to 2 dots, the line spacing setting (and LE setting) is prohibited.
Furthermore, attribute display settings for trimming, shaded, italics, and underline are also prohibited.
In order to use each of the attributes, use a font design with a character vertical size of 4 dots or more.

Figure 30-6 shows the display positions of the main and sub screens.

Figure 30-6. Main and Sub Screen Display Positions

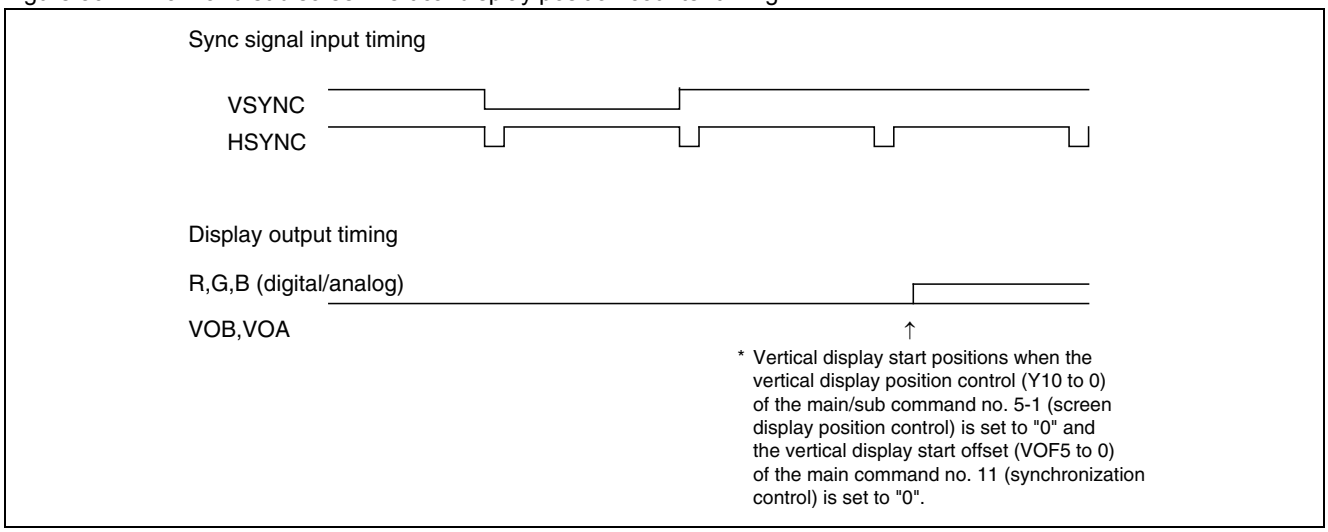


- Vertical display position (see the vertical display position[*1] in Figure 30-6)

  The internal counter for the vertical display position begins counting after the period of 1 Hsync after the vertical sync signal (VSYNC pin input signal) synchronization pulse.

  Figure 30-7 shows the counter timing of the main and sub screen vertical position.

Figure 30-7. Main and sub screen vertical display position counter timing



  The main and sub screen vertical display position starts from the vertical display start offset (main: controlled by command 11 bits VOF5 to 0, see "30.2.4.1 Screen Display Position Offset") setting value + vertical display position control (controlled by command 5-1 bits Y10 to 0) setting value from 1 Hsync period after the vertical sync signal (VSYNC pin input signal) synchronization pulse.

  The setting value calculation for the vertical display position of characters on the main and sub screens is as follows.

  Character vertical display position = Vertical display start offset + Vertical display position control [dots]

- Character horizontal display position (see the character horizontal display position[*2] in Figure 30-6)

  The character horizontal display position is the character display position offset value (see "30.2.4.1 Screen Display Position Offset") + the horizontal display position control setting value from the synchronization pulse significant edge (controlled by the command 13 HE bit) of the horizontal sync signal (HSYNC pin input signal).

  The setting value calculation for the horizontal display position of characters on the main and sub screens is as follows.

  Character horizontal display position = Character display position offset + Horizontal display position control [dots]

- Line background horizontal display position (see the line background horizontal display position[*3] in Figure 30-6)

  The line background horizontal display position is the position after the line background display position offset (see "30.2.4.1 Screen Display Position Offset") from the significant edge of the horizontal sync pulse (controller by command 13 HE bit).

  Remarks:

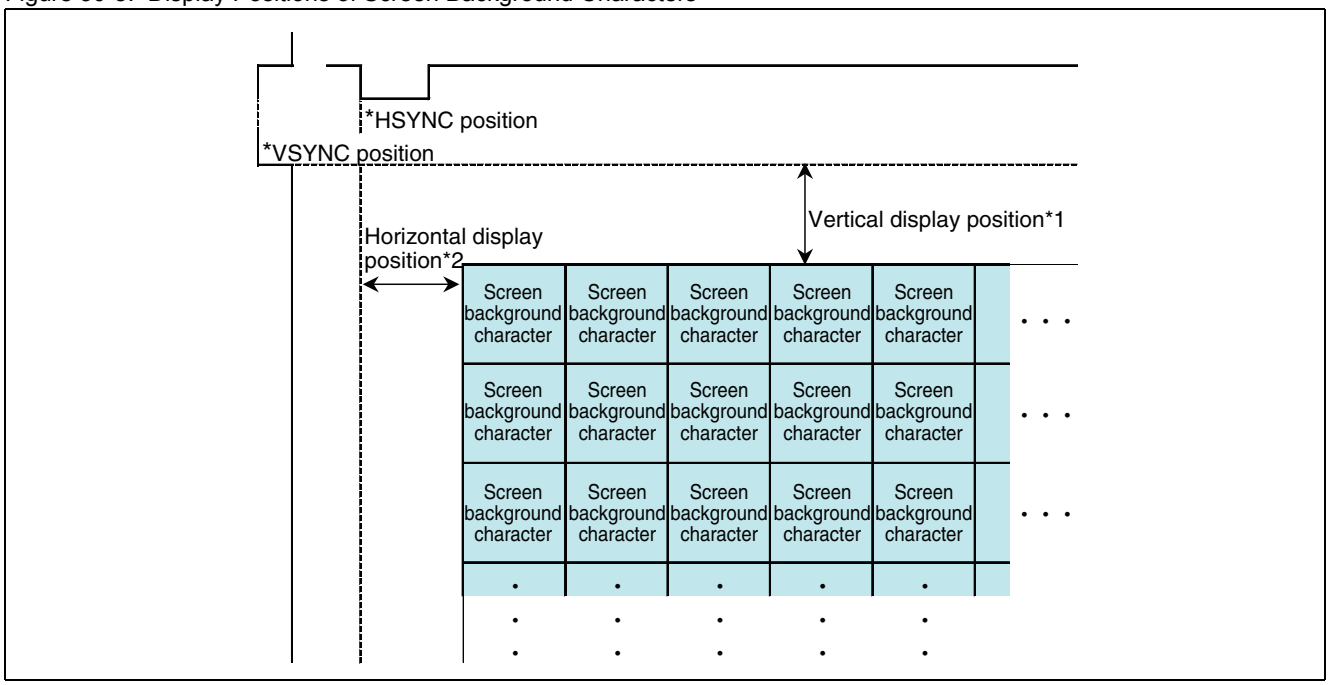
  The line background vertical display position is the same control as the character display position control, and moves the display position at the same time as the main screen and sub screen text.

Note: During the period after the vertical sync signal (VSYNC pin input signal) synchronization pulse and horizontal sync signal (HSYNC pin input signal) synchronization pulse input and before display begins, the display signal output enters the display output OFF state.

Display position control of screen background characters

Figure 30-8 shows the display positions of the screen background characters.

Figure 30-8. Display Positions of Screen Background Characters



- Vertical display position (see the vertical display position*1 in Figure 30-8)

  The vertical display position is the setting value of the vertical display start offset (main: controlled by command 11 bits VOF5 to 0, see "30.2.4.1 Screen Display Position Offset") after the period of 1 Hsync from the vertical sync signal (VSYNC pin input signal) synchronization pulse.

- Horizontal display position (see the horizontal display position*2 in Figure 30-8)

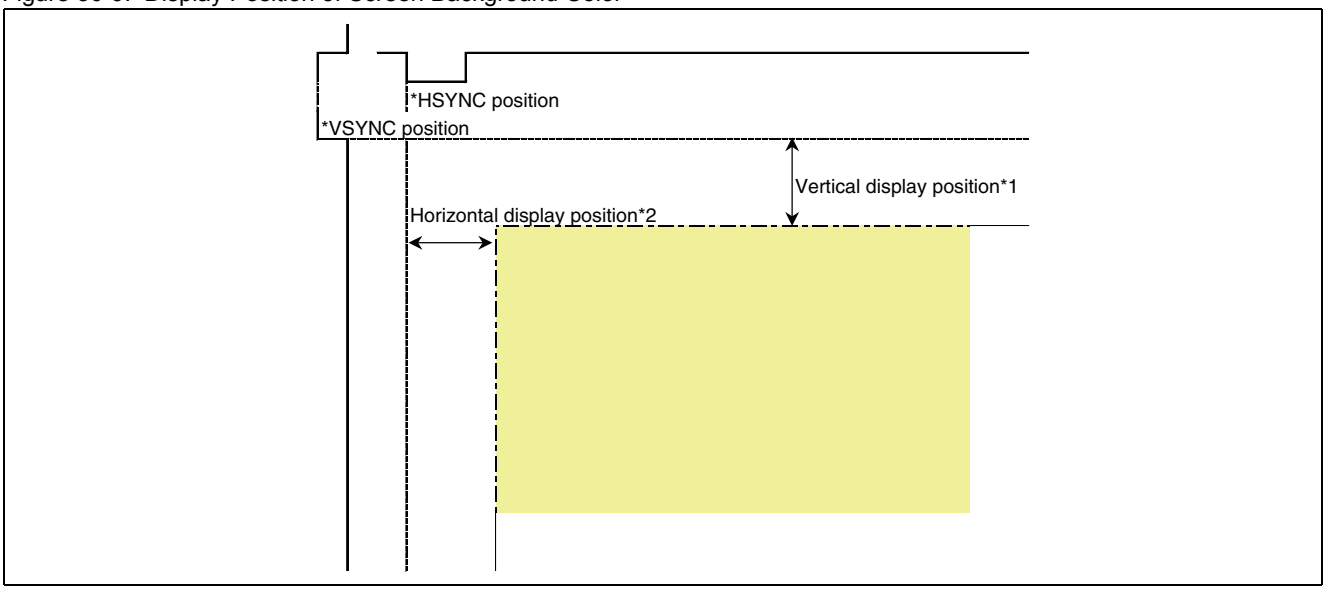
  The horizontal display position is the position of the screen background character display position offset (see "30.2.4.1 Screen Display Position Offset") from the synchronization pulse significant edge (controlled by the command 13 HE bit) of the significant edge of the horizontal sync signal (HSYNC pin input signal) synchronization pulse.

Note: Screen background character display is a function that is only available on the main screen display.
During the period after the vertical sync signal (VSYNC pin input signal) synchronization pulse and horizontal sync signal (HSYNC pin input signal) synchronization pulse input and before display begins, the display signal output enters the display output OFF state.

Display position control of the screen background color

Figure 30-9 shows the display position of the screen background color.

Figure 30-9. Display Position of Screen Background Color

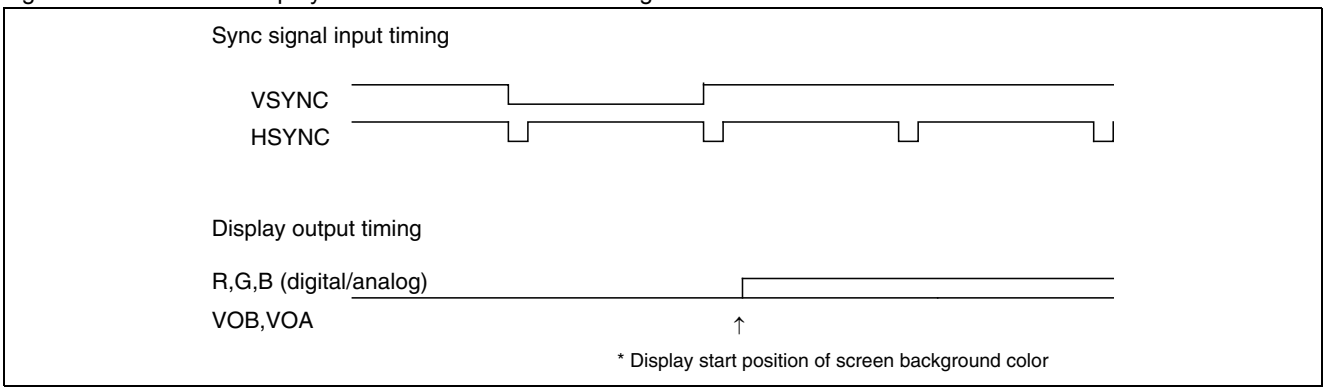


- Vertical display position (see the vertical display position*1 in Figure 30-9)

  The vertical display position is immediately after the vertical sync signal (VSYNC pin input signal) synchronization pulse.

  Figure 30-10 shows the vertical display position of the screen background color.

Figure 30-10. Vertical Display Start Position of Screen Background Color



- Horizontal display position (see the horizontal display position*2 in Figure 30-9)

  The horizontal display position is the position of the screen background color display position offset (see "30.2.4.1 Screen Display Position Offset") from the synchronization pulse significant edge (controlled by the command 13 HE bit) of the significant edge of the horizontal sync signal (HSYNC pin input signal) synchronization pulse.

Note: During the period after the vertical sync signal (VSYNC pin input signal) synchronization pulse and horizontal sync signal (HSYNC pin input signal) synchronization pulse input and before display begins, the display signal output enters the display output OFF state.

Display position control of sprite characters

The OSDC controls the display position of the sprite characters.

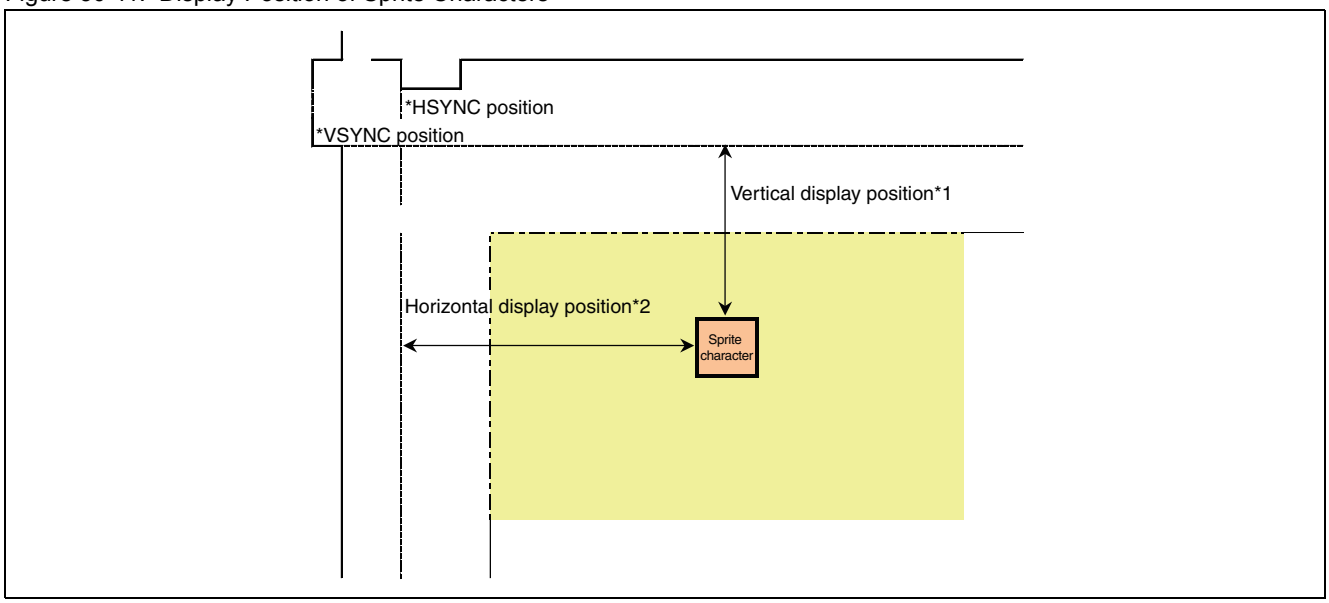
- Vertical display position: Sprite character control 2 (command 9-1) bits SY10 to SY0

  Relative positions are set based on the synchronization pulse of the vertical sync signal (VSYNC pin input signal). Configurable from 0 to 2047 dots in units of 1 dot.
- Horizontal display position: Sprite character control 2 (command 9-1) bits SX11 to SX0

  Relative positions are set based on the synchronization pulse of the horizontal sync signal (HSYNC pin input signal). Configurable from 0 to 4095 dots in units of 1 dot.

Figure 30-11 shows the display position of sprite characters.

Figure 30-11. Display Position of Sprite Characters



- Vertical display position (see the vertical display position*1 in Figure 30-11)

  The sprite vertical display position starts from the vertical display start offset (main: controlled by command 11 bits VOF5 to 0, see "30.2.4.1 Screen Display Position Offset") setting value + sprite character vertical display position (controlled by command 9-1 bits SY10 to 0) setting value from 1 Hsync period (see Figure 30-7) after the vertical sync signal (VSYNC pin input signal) synchronization pulse.

  The setting value calculation for the vertical display position of sprite characters is as follows.

  Sprite vertical display position = Vertical display start offset + Sprite vertical display position control [dots]
- Horizontal display position (see the horizontal display position*2 in Figure 30-11)

  The sprite character horizontal display position is the sprite display position offset value (see "30.2.4.1 Screen Display Position Offset") + the sprite horizontal display start position setting value from the synchronization pulse significant edge (controlled by the command 13 HE bit) of the horizontal sync signal (HSYNC pin input signal).

  The setting value calculation for the horizontal display position of sprite characters is as follows.

  Sprite horizontal display position = Sprite horizontal display position offset + Sprite horizontal display start position [dots]

Note: During the period after the vertical sync signal (VSYNC pin input signal) synchronization pulse and horizontal sync signal (HSYNC pin input signal) synchronization pulse input and before display begins, the display signal output enters the display output OFF state.

30.2.4.1 Screen Display Position Offset

This is a display position offset for each of the main screen, sub screen, main screen background characters, main and sub screen background color, and sprite characters (main and sub). Furthermore, the vertical display position offset setting for the main and sub screens, main screen background characters, and sprite characters (main and sub) can be controlled by command.

Horizontal display position offset

Table 30-5 shows the horizontal display position offset values.

The display position offset is the number of dot clocks from the significant edge (controlled by the command 13 HE bit) of the synchronization pulse of the horizontal sync signal (HSYNC pin input signal).

- The offset values for displaying main and sub screens, sprite characters, and screen background characters are as follows. These values are not mutually related to each other, and each value is fixed offset value.

Table 30-5. Horizontal Display Position Offset Values

Display screen	Offset value
Main and sub sprite characters	63
Main and sub screen (characters)	
Screen background characters	
Main and sub screen (line background)	50
Screen background color	

Note: The unit for offset values is the number of dot clocks.

Vertical display position offset

Table 30-6 shows the vertical display position offset values.

The vertical display position offset is the number of Hsync in the vertical direction from after the period of 1 Hsync after the synchronization pulse of the vertical sync signal (VSYNC pin input signal). If this value is set to a value other than 0, the vertical position control of the main and sub screens, windows, and sprites, and the vertical display position of the screen background characters on main operate using the vertical display start position offset setting value as the origin.

- The main and sub vertical offset values are as follows.

Table 30-6. Vertical Display Start Offset Settings

VOF5 to VOF0	Vertical display start offset value
00_H to $3F_H$	0 to 63

Note: The unit for offset values is Hsync.

The setting values (commands) that enable the vertical display start position offsets are as follows.

- Vertical display position control [command 5-1] (Y10 to Y0)
- Window vertical period end control [command 8-1] (WYE10 to WYE0)
- Window vertical period start control [command 8-1] (WYS10 to WYS0)
- Sprite character vertical display position control [command 9-1] (SY10 to SY0)

30.2.5 Font Memory Configuration

The font memory has capacity for a total of 16384 characters where each character is composed of 32 dots × 32 dots.

- The 16384 characters can be configured arbitrarily by the user.

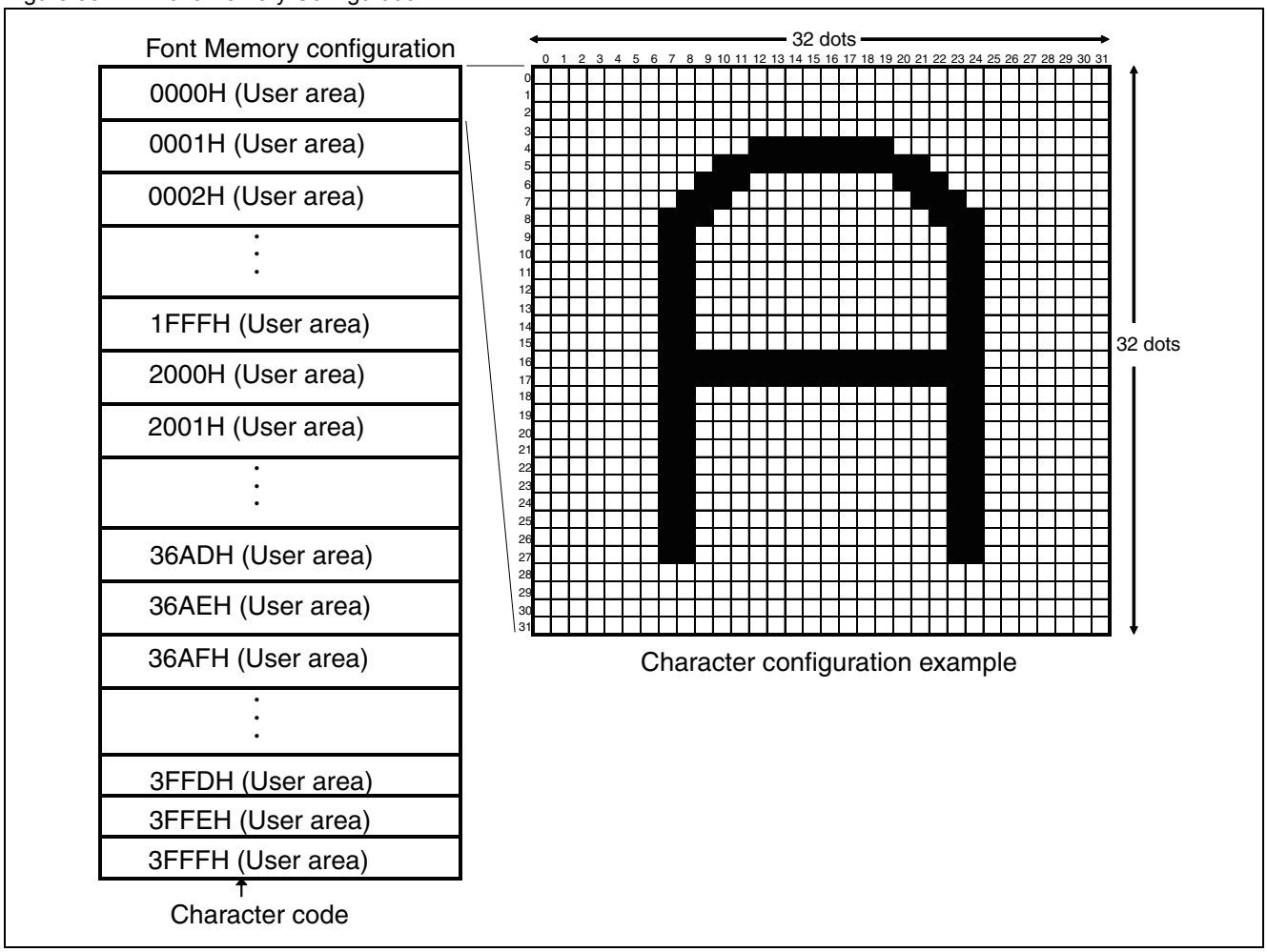
  Note: There is no reserved configuration for whitespace characters. Configure an arbitrary character code as whitespace as necessary.

- Any graphic character or normal character is able to be configured.
  (However, each graphics character uses the data of 8 normal characters, and has a character code that is a multiple of 8 from 0000_H.)

Font memory configuration

Figure 30-12 shows the font memory configuration.

Figure 30-12. Font Memory Configuration



30.2.6 Display Memory (VRAM) Configuration

The display memory is composed of the character RAM for configuring per-character settings and line RAM for configuring per-line settings. (The main screen and sub screen are each composed of separate VRAM.)

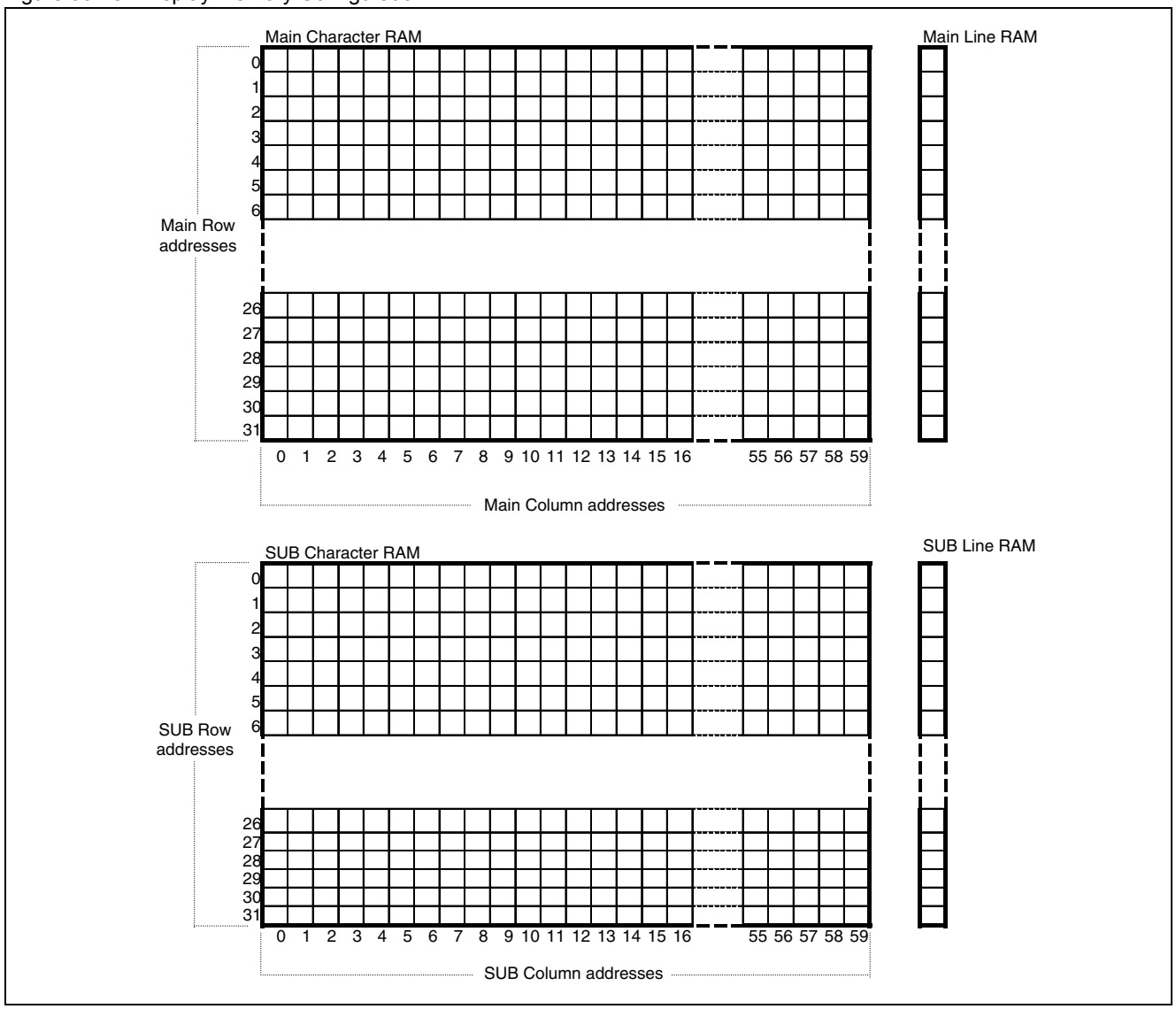
- Character RAM: Main [60 columns × 32 lines (total 1920 characters)], Sub [60 columns × 32 lines (total 1920 characters)]
- Line RAM: Main [32 lines], Sub [32 lines]

Display memory and display screens

Character RAM has a one-to-one correspondence with the displayed characters. Line RAM has a one-to-one correspondence with the displayed lines.

Figure 30-13 shows the display memory configuration.

Figure 30-13. Display Memory Configuration



30.2.7 Writing to Display Memory (VRAM)

Writing to display memory is performed by setting command 0 to 4 to the OSDC command.

- Writing a Single Character to Character RAM
- Batch-Writing Multiple Characters to Character RAM
- Writing to Line RAM

Writing to display memory

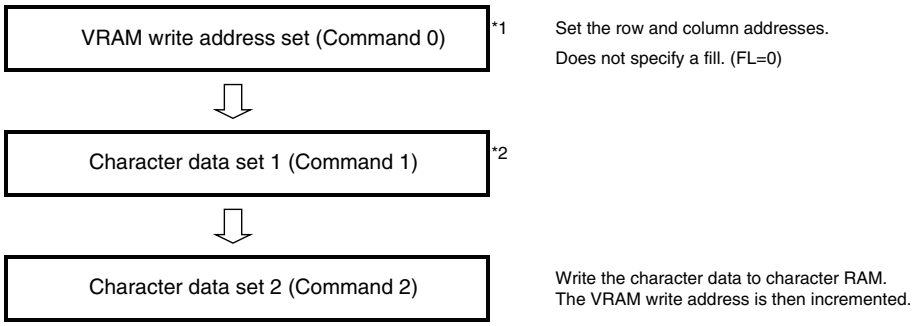
Writing to display memory is performed by issuing an OSDC command to OSDC control command 0 to 4.

Writing a single character to character RAM

An arbitrary character code can be written to an arbitrary address in character RAM.

Figure 30-14 shows the procedure for writing to character RAM.

Figure 30-14. Procedure for Writing to Character RAM



*1: This command can be omitted when writing to the consecutive address following from the previous character RAM write.

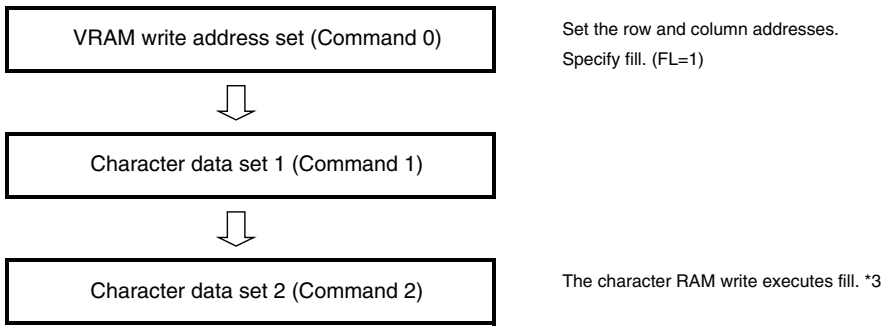
*2: This command can be omitted if the value set by the character data set 1 command has not changed since it was last set.

Batch-writing multiple characters to character RAM (VRAM Fill)

An arbitrary character code can be written to a range of addresses starting from an arbitrary address in character RAM to the last address.

Figure 30-15 shows the procedure for batch-writing (VRAM fill) a single character multiple times into character RAM.

Figure 30-15. Procedure for Batch-writing (VRAM fill) to Multiple Characters in Character RAM



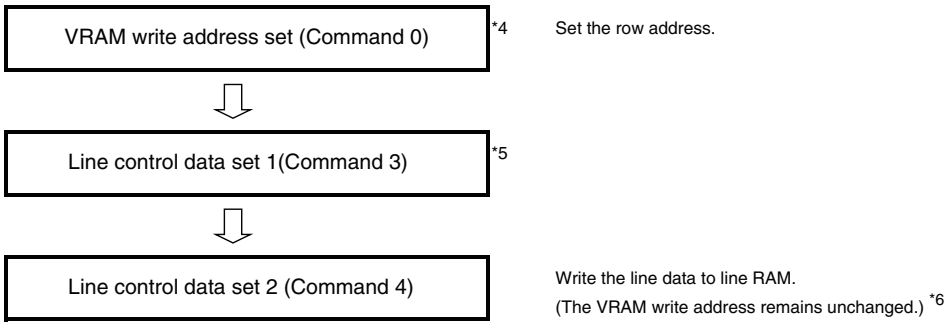
*3: When the VRAM fill ends, the VRAM fill complete interrupt source is set.
The execution time of the VRAM fill depends on the dot clock frequency. Reference values for filling the entire screen are as follows.
When the dot clock is 75 MHz Main: Approx. 0.2 ms, Sub: Approx. 0.2 ms
Issuing commands 1 to 4 while the VRAM fill is executing is prohibited.
If command 0 (FL=0) is issued while the VRAM fill is executing, the VRAM fill is aborted. After the VRAM fill has finished normally or has been aborted, command 0 needs to be reissued to set the VRAM write address.

Writing to line RAM

Arbitrary line data can be written to an arbitrary address in line RAM.

Figure 30-16 shows the procedure for writing to line RAM.

Figure 30-16. Procedure for Writing to Line RAM



*4: There is no line RAM fill function (specifying fill is prohibited).

*5: This command can be omitted if the value set by the line data set 1 command has not changed since it was last set.

*6: The address is not incremented by writing to line RAM.
The line address therefore needs to be set for each line.

30.2.8 Palette Configuration

The palette converts the 8-bit color codes output by the OSDC into 16-bit color codes.

Palette RAM configuration

The color codes specified by the OSDC are converted from 8-bit color codes to RGB color codes with 5 bits, 6 bits, 5 bits (total of 16 bits) by the palette.

Figure 30-17 shows the correspondence between the addresses for configuring the palette and the OSDC color codes.

Figure 30-17. Palette Configuration

OSDC control palette write address (lower 16 bits)		OSDC read palette address (OSDC-set color code*1)		upper palette data (16bit)			lower palette data (16bit)		
			Bit	[31:24]		[23:16]	[15:8]		[7:0]
			Byte address*2	+0		+1	+2		+3
		upper	lower	Bit [31:27]	[26:21]	[20:16]	[15:11]	[10:5]	[4:0]
4200_H	→	01_H	00_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
4204_H	→	03_H	02_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
4208_H	→	05_H	04_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
$420C_H$	→	07	06_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
4210_H	→	09_H	08_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
•		•	•	•	•	•	•	•	•
•		•	•	•	•	•	•	•	•
•		•	•	•	•	•	•	•	•
$43EC_H$	→	$F7_H$	$F6_H$	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
$43F0_H$	→	$F9_H$	$F8_H$	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
$43F4_H$	→	FB_H	FA_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
$43F8_H$	→	FD_H	FC_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)
$43FC_H$	→	FF_H	FE_H	R(5bit)	G(6bit)	B(5bit)	R(5bit)	G(6bit)	B(5bit)

*1: The OSDC configured color codes that are used in palette color code transformations are the following setting values. Character (MC7 to MC0), character background color (MB7 to MB0), trimming color (LF7 to LF0), line background color (LB7 to LB0), shaded background frame color(BH7 to BH0,BS7 to BS0), graphic color control (GF7 to GF0,GC7 to GC0), screen background color (U7 to U0) and graphic character color are displayed.

*2: Although the OSDC control palette write addresses are also given as byte addresses, they are only available as word addresses. (Always perform word writes from addresses xxx0H, xxx4H, xxx8H, etc.)

30.2.9 Character Display

The horizontal and vertical sizes of display characters can be configured.

The specified character data in the font ROM is displayed by clipping the data to the specified size from the top left dot.

Character horizontal size control (per character setting)

Table 30-7 shows the character horizontal sizes of character data set 1 (command 1): bits MS1 and MS0.

Table 30-7. Character Horizontal Size Control

MS1	MS0	Character horizontal size
0	0	S size:16 dots
0	1	M size:24 dots
1	0	L size:32 dots
1	1	(Setting prohibited)

Line character vertical size control (per line setting)

Table 30-8 shows the character vertical sizes of line control data set 1 (command 3): bits LHS3, LHS2, LHS1, and LHS0.

Table 30-8. Character Vertical Size Control

LHS3	LHS2	LHS1	LHS0	Character vertical size
0	0	0	0	2 dots
0	0	0	1	4 dots
0	0	1	0	6 dots
0	0	1	1	8 dots
0	1	0	0	10 dots
0	1	0	1	12 dots
0	1	1	0	14 dots
0	1	1	1	16 dots
1	0	0	0	18 dots
1	0	0	1	20 dots
1	0	1	0	22 dots
1	0	1	1	24 dots
1	1	0	0	26 dots
1	1	0	1	28 dots
1	1	1	0	30 dots
1	1	1	1	32 dots

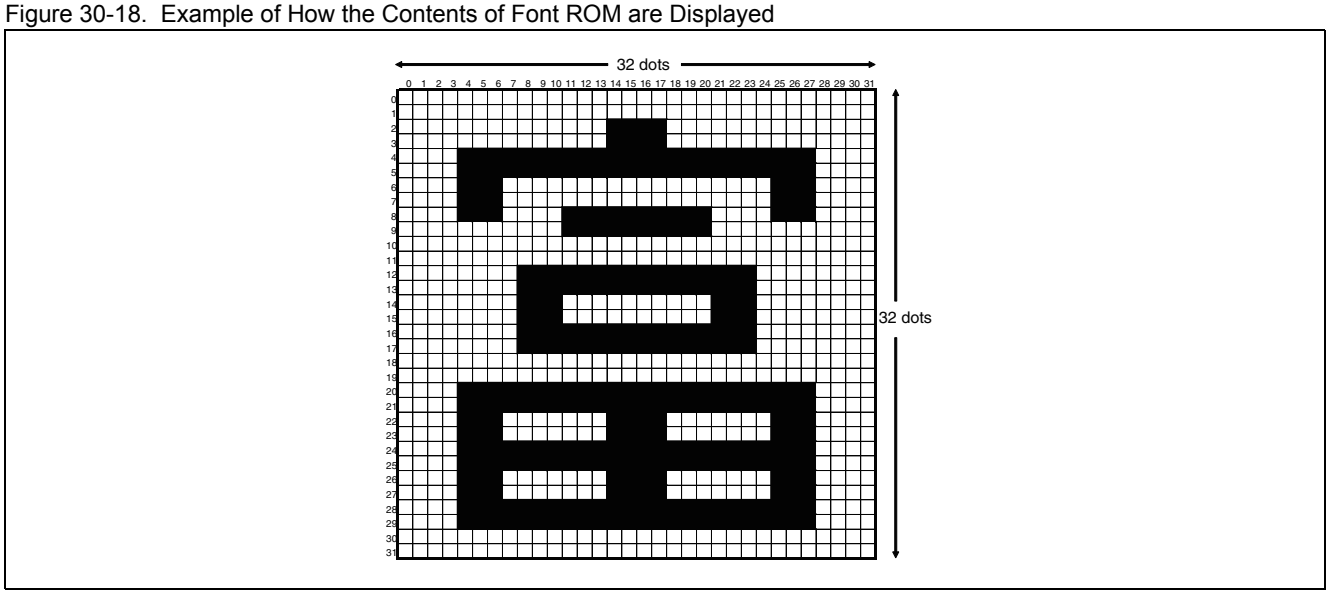
Note: For lines where the character vertical size is set to 2 dots, the line spacing setting (and LE setting) is prohibited. Furthermore, attribute display settings for trimming, shaded, italics, and underline are also prohibited. In order to use each of the attributes, use a font design with a character vertical size of 4 dots or more.

Display examples

- Contents of Font ROM

  Figure 30-18 shows an example how the contents of font ROM are displayed.

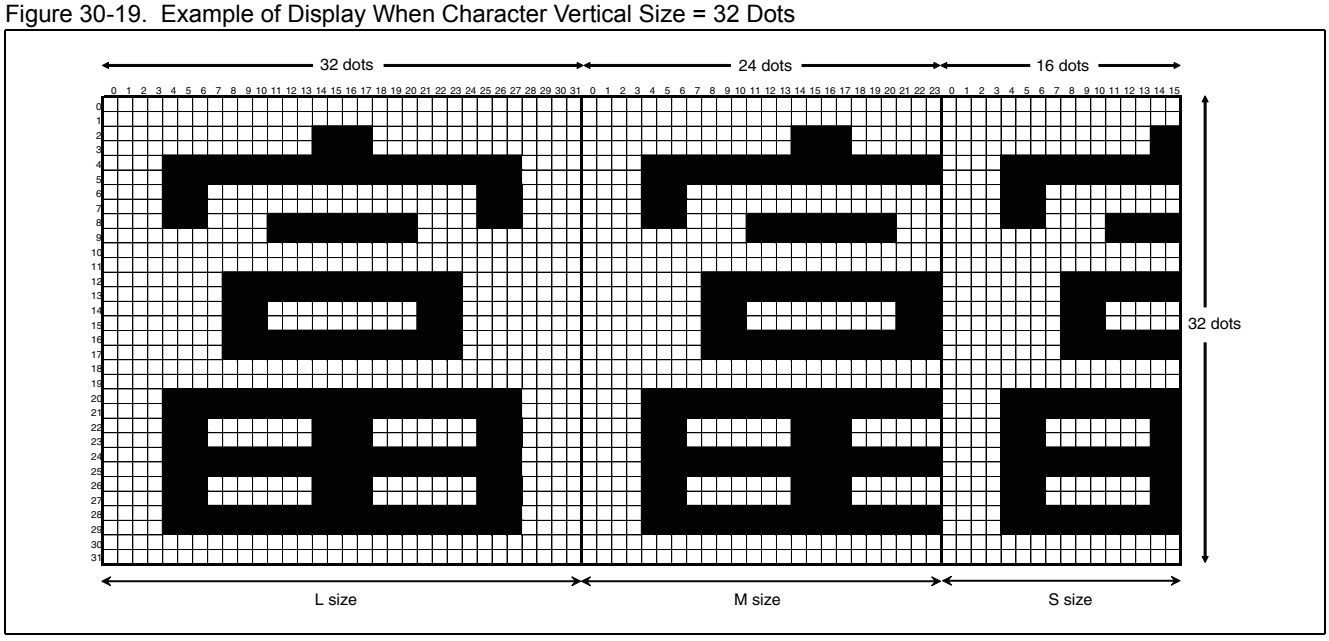
Figure 30-18. Example of How the Contents of Font ROM are Displayed



- Display Example 1 (Character Vertical Size = 32 Dots)

  Figure 30-19 shows examples of how the character is displayed when the character vertical size = 32 dots.

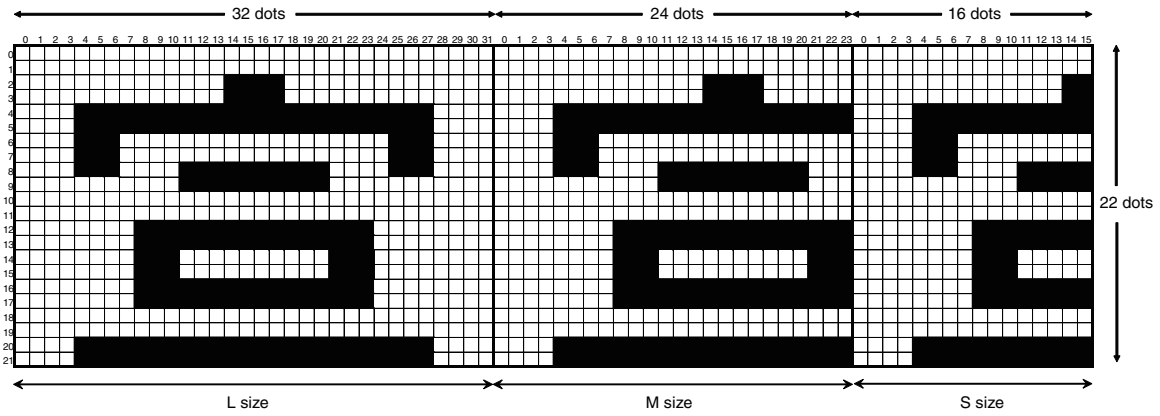
Figure 30-19. Example of Display When Character Vertical Size = 32 Dots



- Display Example 2 (Character Vertical Size = 22 Dots)

  Figure 30-20 shows examples of how the character is displayed when the character vertical size = 22 dots.

Figure 30-20. Example of Display When Character Vertical Size = 22 Dots

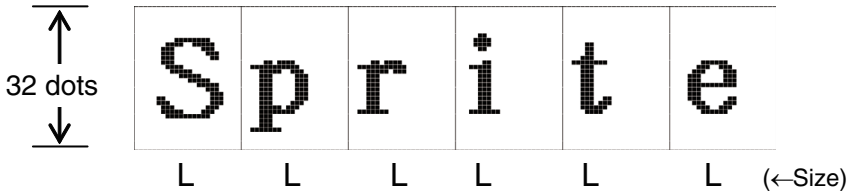


Applied Display Examples

- Example of display using only size L

  Figure 30-21 shows an example of the display when only L size is used.

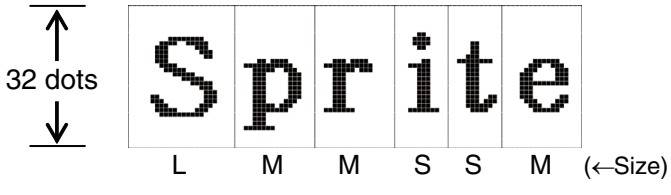
Figure 30-21. Example of Display Using Only Size L



- Example of display using a mixture of L, M, and S sizes

  Figure 30-22 shows an example of the display when using a mixture of L, M, and S sizes.

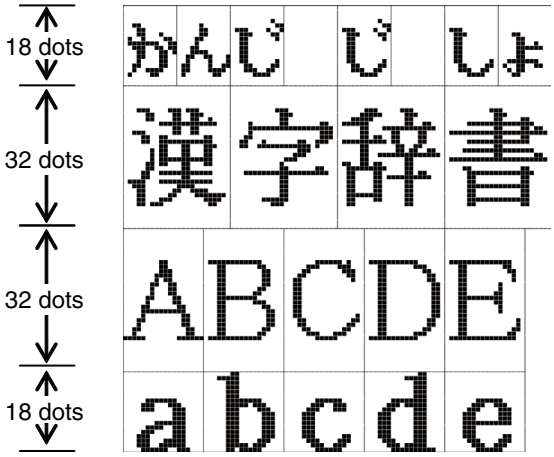
Figure 30-22. Example of Display Using a Mixture of L, M, and S Sizes



- Example of display using L, M, and S sizes with variable vertical sizes

  Figure 30-23 shows an example of the display when using L, M, and S sizes with variable vertical sizes.

Figure 30-23. Example of Display Using L, M, and S Sizes With Variable Vertical Sizes



30.2.9.1 Character Colors

Character colors can be set per-character from 256 colors.

Character colors (configurable per-character from 256 colors)

Character colors can be configured per-character by setting MC7 to MC0 of the character data set 1 (command 1) to the color code.

30.2.9.2 Italic Display

Italic display is a function that displays the character dots tilted. Italic display can be configured per-character.

Italic display control

The italic attribute can be configured per-character using the MIT bit of the character data set 1 (command 1).

Table 30-9. Italic Character Control (Per Character Setting)

MIT	Italic character control
0	Normal character
1	Italic character

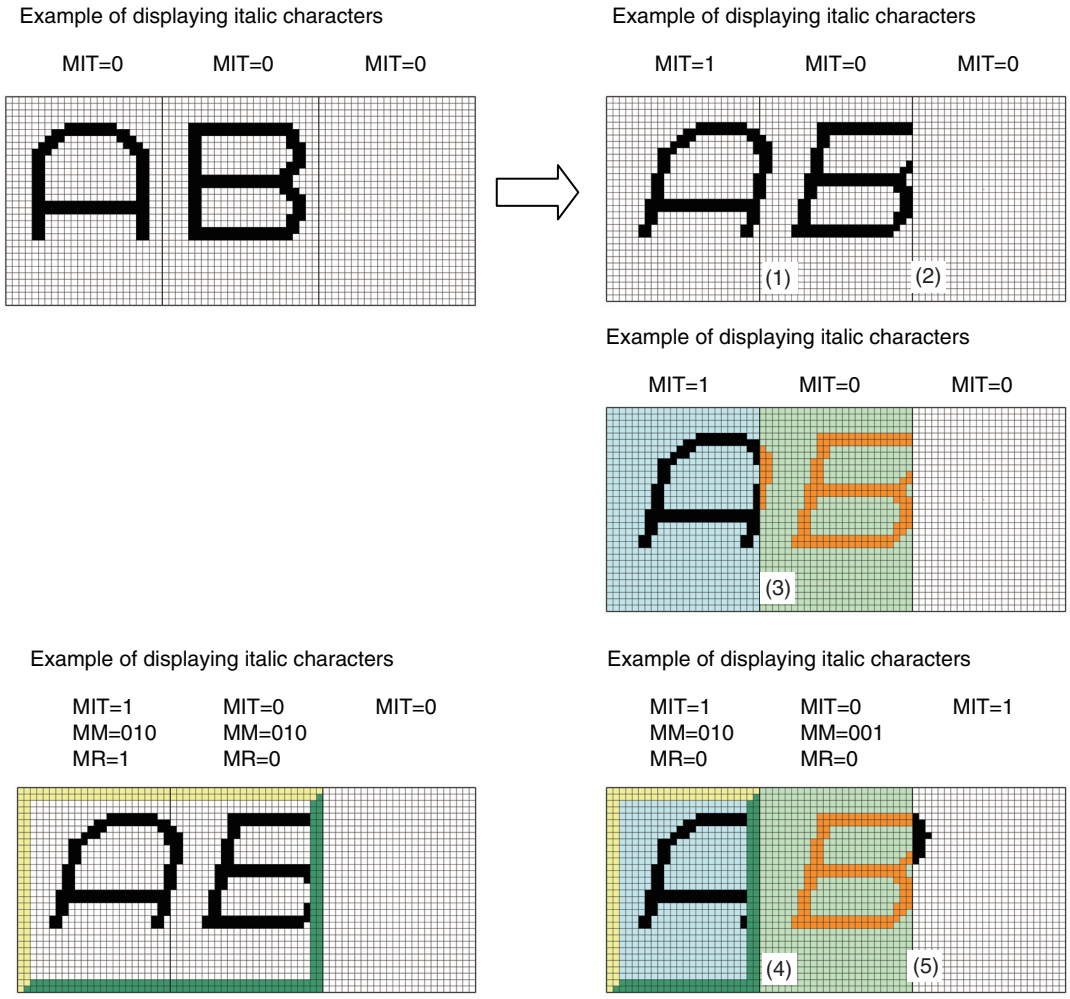
Italic display rules

- The character immediately to the right of a character that has been specified as italic is displayed in italics even if that character is not specified as italic. (See (1) in Figure 30-24)
  Furthermore, even if the character one more position to the right specifies non-italic attributes, italic character dots that extend into that region are cropped. (See (2) in Figure 30-24)
- During italic display, the color of character dots that protrude out from the original character region depend on the color setting of the character region into which the dots protrude. (See (3) in Figure 30-24)
- If the character shaded background has a right shaded frame, the character dots for displaying the italics do not protrude to the right beyond the border. (See (4) in Figure 30-24)
- Italic display only has an effect on normal character dots, and is not applied to the character attributes (character data settings), graphics characters, background character parts, or shaded frames.
- If the character immediately to the right of a character specified as italic is specified as non-italic, and the character one more to the right is also specified as italic, that character is displayed in italics. (See (5) in Figure 30-24)

Display examples

Figure 30-24 shows examples of displaying italic characters.

Figure 30-24. Examples of Displaying Italic Characters



* MM: Character background control (command 1)
MR: Shaded background right character merge control (command 2)

Note: For italics displayed in the 60th column, although the italic character dots protrude into the region on the right like in (5) in the example of (4) in Figure 30-24, the character color of (5) is "00_H".

Origin of italic characters

The starting point of the tilt of italic characters depends on the character vertical size control, and is the bottom left of the 5th dot from the bottom dot, with the dots shifted one dot to the right for each four dots.

Figure 30-25 and Figure 30-26 show the italic conditions.

Figure 30-25. Italic Conditions When Displaying 32 Vertical Dots

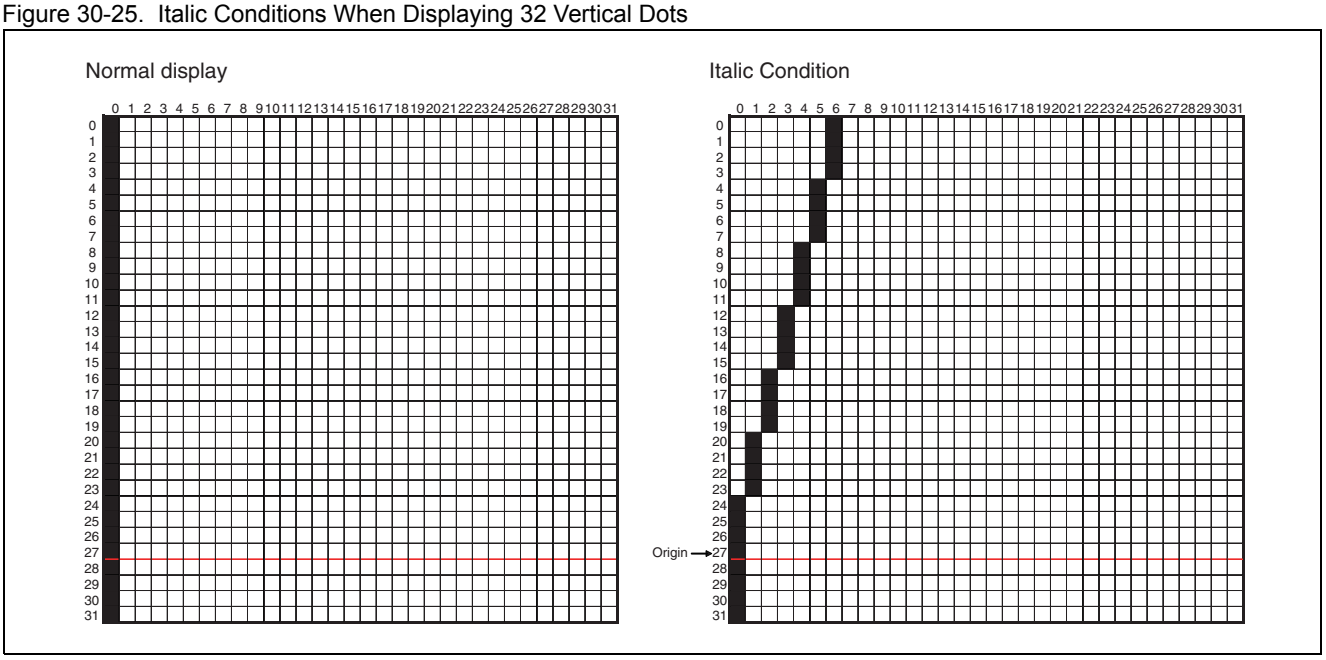
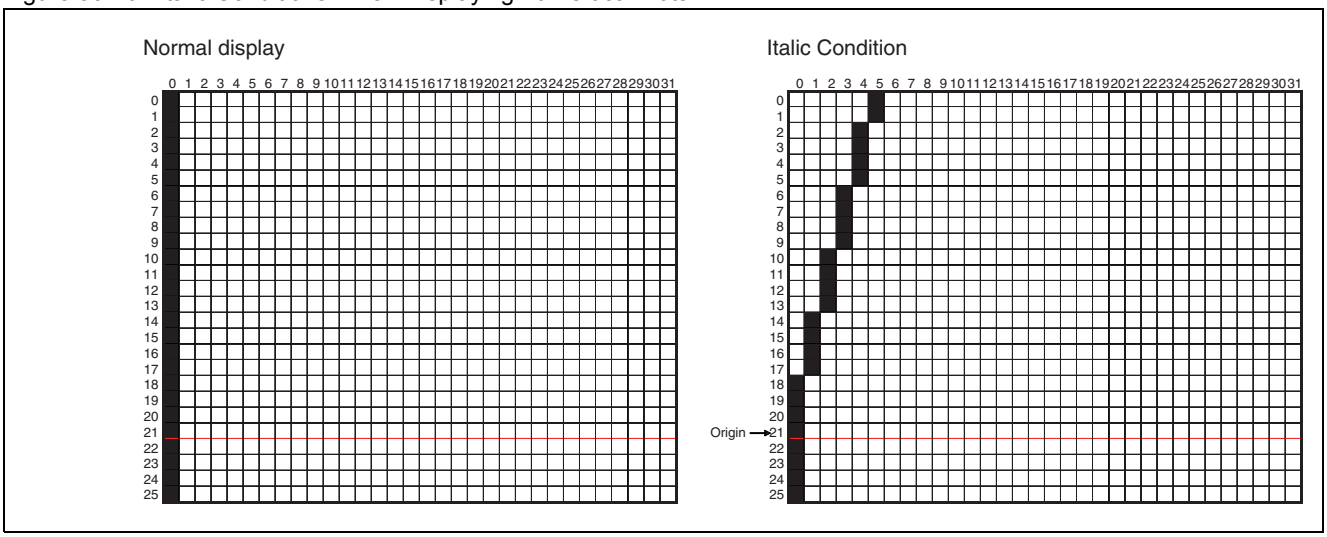


Figure 30-26. Italic Conditions When Displaying 26 Vertical Dots



Note: If italics are displayed on enlarged characters, the italic display is performed on the enlarged character dots.

30.2.9.3 Underline Display

Underline display is a function that displays a horizontal line underneath the characters. Underline display can be configured per-character.

Underline display control

The underline attribute can be configured per-character by setting the MUL bit of the character data set 1 (command 1).

Table 30-10. Underline Display (Per Character Setting)

MUL	Underline control
0	Normal character
1	Underline display

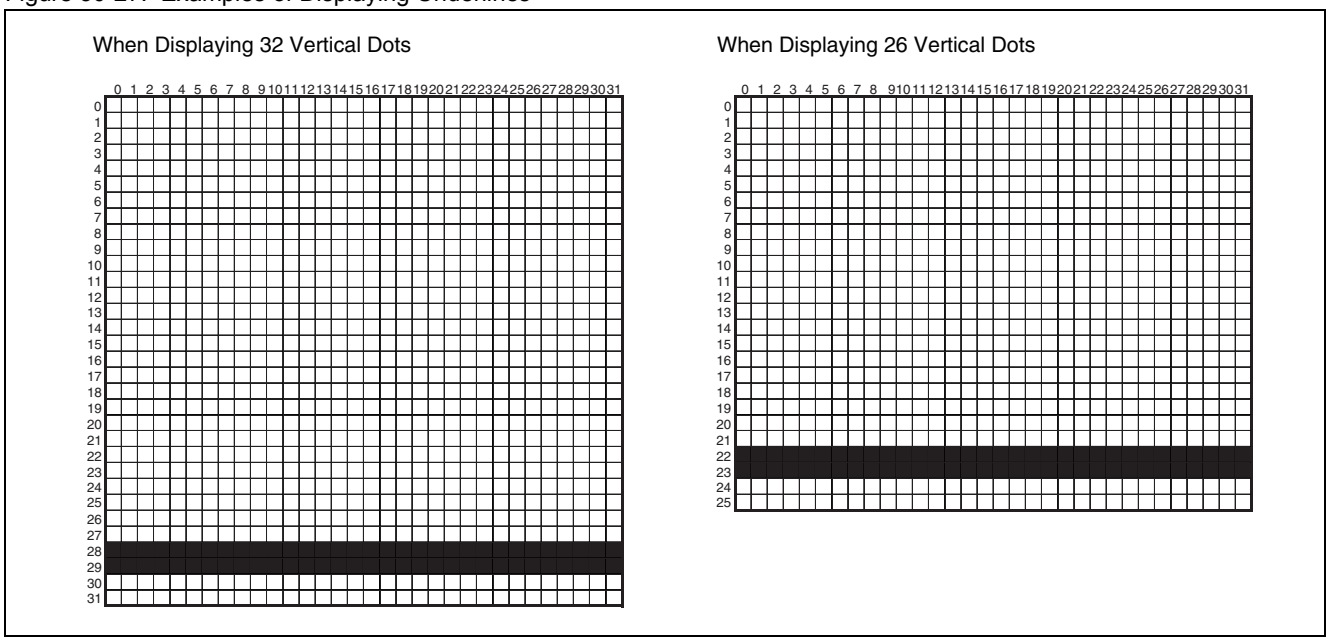
Underline display rules

Underline depends on the character vertical size control, and is displayed as character data in the 3rd and 4th dots in the vertical direction from the bottom dot.

Display examples

Figure 30-27 shows examples of underline display.

Figure 30-27. Examples of Displaying Underlines



Note: Underline display is shown beneath the shaded frame of shaded background. Furthermore, trimming display is not performed on underlines. In all other aspects, the underline is treated the same as character dot data. When using enlarged display, the underline section is also displayed enlarged to match the enlargement, and the underline can also be displayed blinking as part of the character.

30.2.9.4 Character Trimming

Character trimming is a function that displays an trimming around the perimeter of the character dots.

Trimming output control

The presence or absence of the trimming can be controlled by the type of character background. The trimming can be selected from four different types for each line.

Table 30-11 shows the trimming output control of line control data set 1 (command 3): bits LFD and LFC.

Table 30-11. Trimming Output Control (Per Line Setting) (Sheet 1 of 2)

Trimming output control (per line setting)		Character background type (per character setting)				Trimming output
LFD	LFC	MM2	MM1	MM0	Display type	
0	0	0	0	0	Not displayed	No
		0	0	1	Solid background	No
		0	1	0	Shaded background concaved (solid)	No
		0	1	1	Shaded background convexed (solid)	No
		1	0	0	Background character	No
		1	0	1	(Setting prohibited)	–
		1	1	0	Shaded background concaved (background character)	No
		1	1	1	Shaded background convexed (background character)	No
0	1	0	0	0	Not displayed	Yes
		0	0	1	Solid background	No
		0	1	0	Shaded background concaved (solid)	No
		0	1	1	Shaded background convexed (solid)	No
		1	0	0	Background character	No
		1	0	1	(Setting prohibited)	–
		1	1	0	Shaded background concaved (background character)	No
		1	1	1	Shaded background convexed (background character)	No
1	0	0	0	0	Not displayed	Yes
		0	0	1	Solid background	Yes
		0	1	0	Shaded background concaved (solid)	No
		0	1	1	Shaded background convexed (solid)	No
		1	0	0	Background character	Yes
		1	0	1	(Setting prohibited)	–
		1	1	0	Shaded background concaved (background character)	No
		1	1	1	Shaded background convexed (background character)	No

Table 30-11. Trimming Output Control (Per Line Setting) (Sheet 2 of 2)

Trimming output control (per line setting)		Character background type (per character setting)				Trimming output
LFD	LFC	MM2	MM1	MM0	Display type	
1	1	0	0	0	Not displayed	Yes
		0	0	1	Solid background	Yes
		0	1	0	Shaded background concaved (solid)	Yes
		0	1	1	Shaded background convexed (solid)	Yes
		1	0	0	Background character	Yes
		1	0	1	(Setting prohibited)	–
		1	1	0	Shaded background concaved (background character)	Yes
		1	1	1	Shaded background convexed (background character)	Yes

No: Not displayed

Yes: Displayed

Trimming style control

Trimmings are displayed using a combination of four different style settings that are configured per screen and four different control settings that are configured per line.

- Trimming dot control (per screen setting)

  Trimming can be used by selecting the display style for the number of trimming dots (1 dot or 2 dots) using the per-screen trimming dot control.

  Table 30-12 shows the trimming dot control of the screen output control (command 5-0): bit FM0.

Table 30-12. Trimming Dot Control (Per Screen Setting)

Trimming Dot Control	Number of Trimming Dots
FM0	
0	1 dot trimming
1	2 dot trimming

  - 1 dot trimming
    The trimming is displayed composed of a one-dot trimming.
  - 2 dot trimming
    The trimming is displayed composed of a two-dot trimming.

- Shadow trimming control (per screen setting)

  Trimming can be used by selecting the display style from shadow trimming A (lower right) or shadow trimming B (lower right + right) using the per-screen shadow trimming control.

  In order to enable the shadow trimming control, it is a prerequisite that the trimming control in line control data set 1 (command 3) is set to shadow trimming.

  Table 30-13 shows the shadow trimming control of the screen output control (command 5-1): bit FM1.

Table 30-13. Shadow Trimming Control (Per Screen Setting)

Shadow trimming control	Shadow trimming style
FM1	
0	Shadow trimming A (lower right)
1	Shadow trimming B (lower right + right)

  - Shadow trimming A (lower right)
    Characters are displayed by adding an trimming in the lower right direction of the character dots.
  - Shadow trimming B (lower right + right)
    Characters are displayed by adding an trimming in the lower right direction and right direction of the character dots.

- Trimming control (per line setting)

  The trimming can be displayed by choosing from full perimeter, right, or shadow trimming using the per-line trimming control.

  Table 30-14 shows the trimming control of line control data set 1 (command 3): bits LFB and LFA.

Table 30-14. Trimming Control (Per Line Setting)

Trimming control		Trimming output
LFB	LFA	
0	0	Not displayed
0	1	Full perimeter trimming
1	0	Right trimming
1	1	Shadow trimming

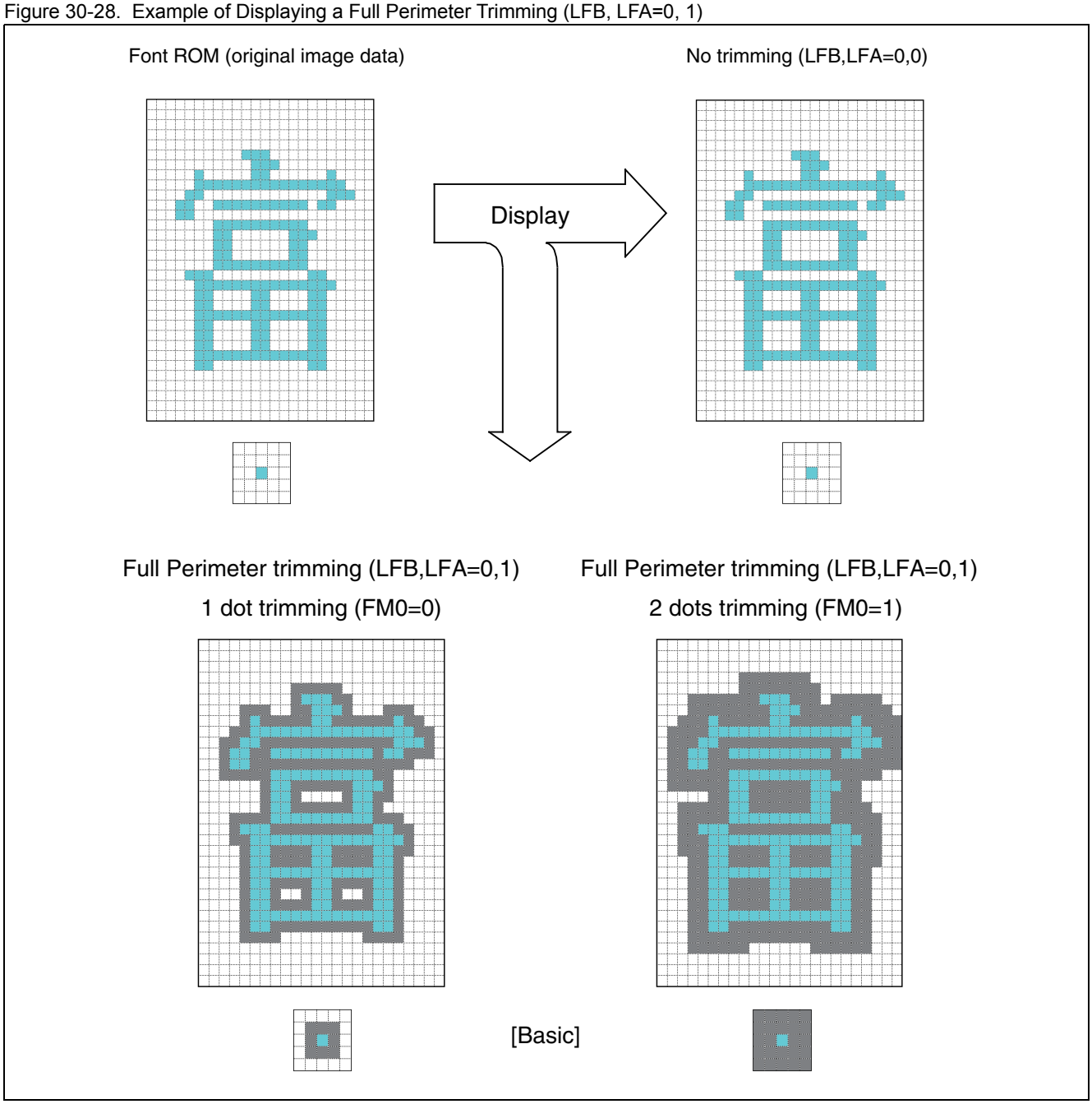
  - Full perimeter trimming
    Characters are displayed by adding an trimming around the full perimeter of the character dots.
  - Right trimming
    Characters are displayed by adding an trimming in the right direction of the character dots.
  - Shadow trimming
    Characters are displayed by adding a shadow trimming to the character dots. The shadow trimming style can be selected from shadow trimming A (lower right) or shadow trimming B (lower right + right) using the screen output control (command 5-1): bit FM1.

Display examples

- Example of displaying a full perimeter trimming (FMB, FMA=0, 1)

  Figure 30-28 shows examples displaying 1-dot and 2-dot trimmings (24 × 32 dots) when full perimeter trimming is selected.

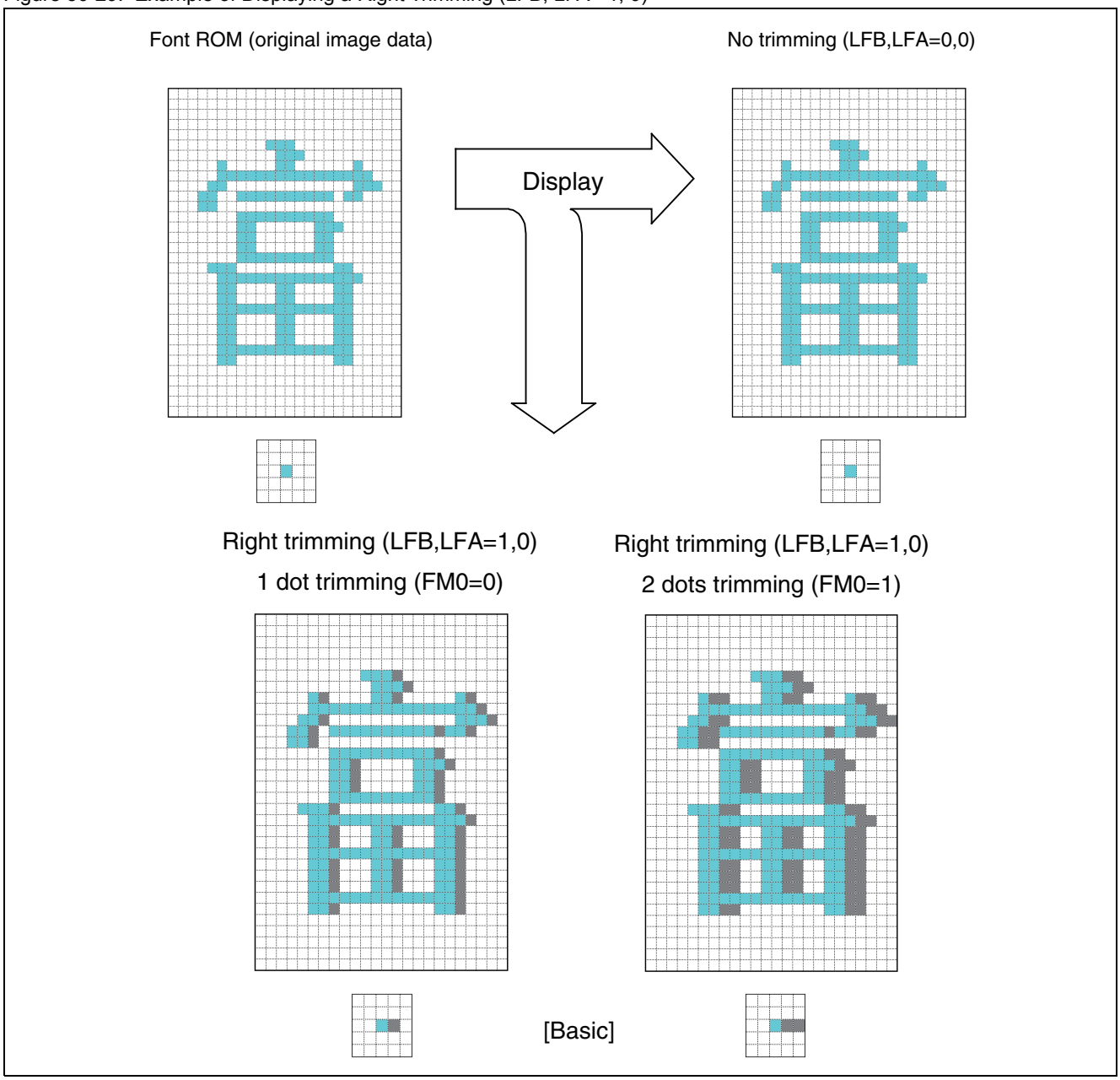
Figure 30-28. Example of Displaying a Full Perimeter Trimming (LFB, LFA=0, 1)



- Example of displaying a right trimming (FMB, FMA= 1, 0)

  Figure 30-29 shows examples displaying 1-dot and 2-dot trimmings (24 × 32 dots) when right trimming is selected.

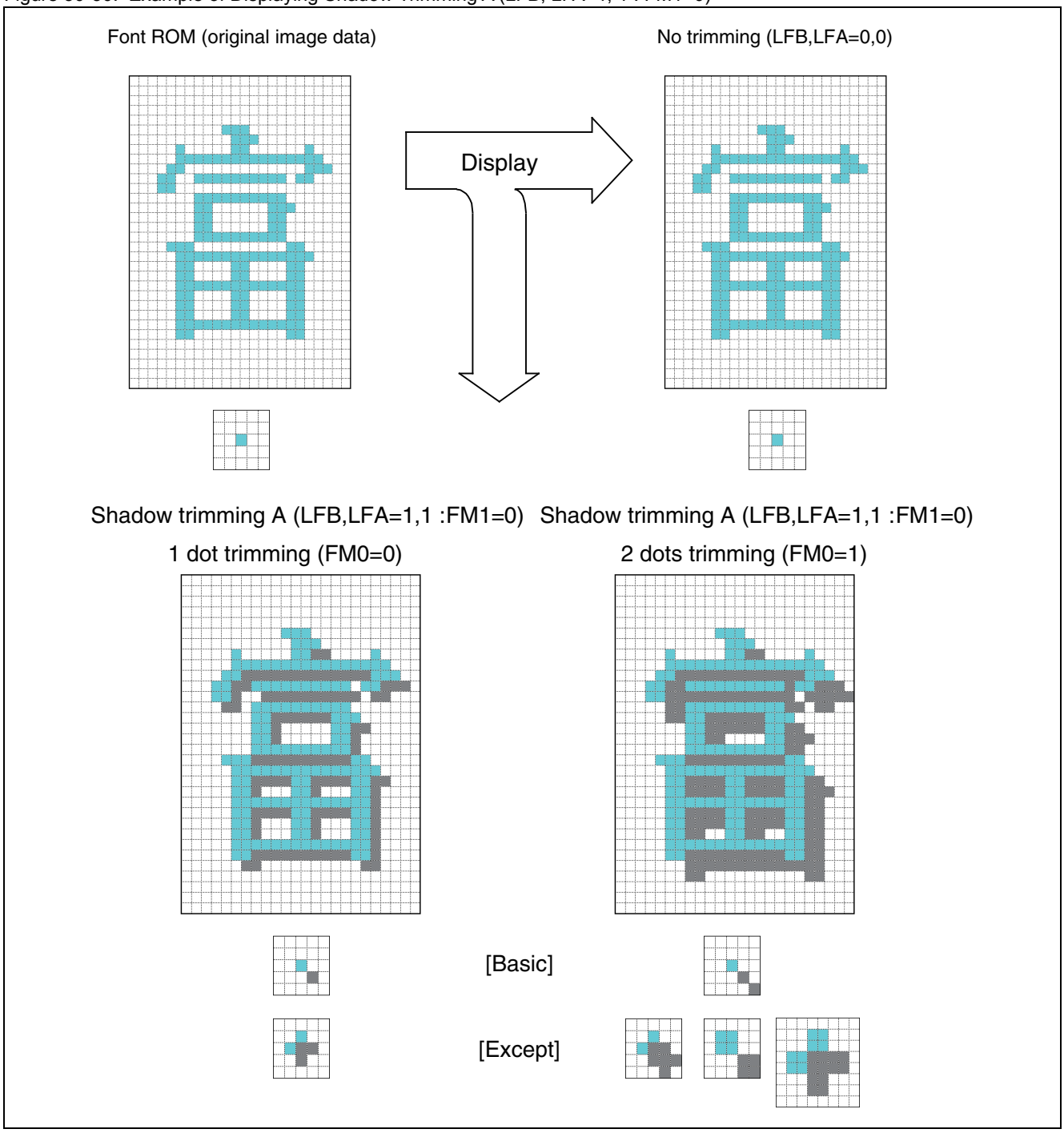
Figure 30-29. Example of Displaying a Right Trimming (LFB, LFA= 1, 0)



- Example of displaying shadow trimming A (FMB, FMA=1, 1 : FM1=0)

  Figure 30-30 shows examples displaying 1-dot and 2-dot trimmings (24 × 32 dots) when shadow trimming A (lower right) is selected.

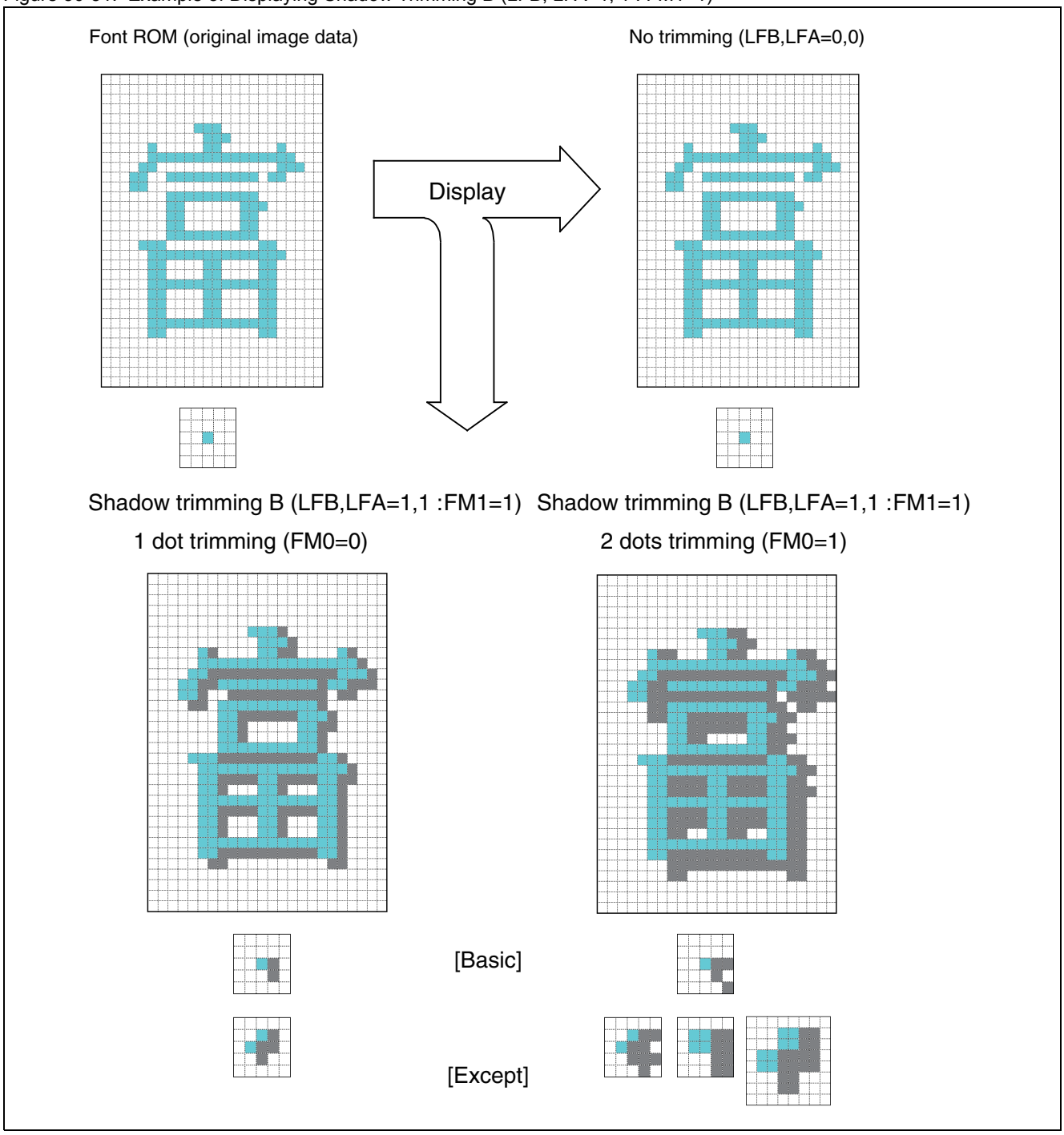
Figure 30-30. Example of Displaying Shadow Trimming A (LFB, LFA=1, 1 : FM1=0)



- Example of displaying shadow trimming B (FMB, FMA=1, 1 : FM1=1)

  Figure 30-31 shows examples displaying 1-dot and 2-dot trimmings (24 × 32 dots) when shadow trimming B (lower right + right) is selected.

Figure 30-31. Example of Displaying Shadow Trimming B (LFB, LFA=1, 1 : FM1=1)



Trimming color

The trimming color can be configured per-line by setting bits LF7 to LF0 of the line control data set 1 (command 3) to the color code.

Trimming display rules

- The display of trimming dots that protrude into the regions of the characters above and below is only performed in regions corresponding to the line spacing setting. (If there is no line spacing, trimming dots that extend outside of the range of the line are not processed. Furthermore, trimming dots that extend over the upper and lower shaded frame when using a shaded background display are not displayed.)
- The display of trimming dots in the left and right neighboring character regions is only possible when the character and the neighboring character have the same character background type. (However, concaved display and convexed display for shaded backgrounds are treated as the same background type.)
- The display of trimming dots for the leftmost or rightmost character outside of the character region is only possible when the character background type of the edge character is no character background.
- When line enlargement control is selected (when the LGY1, LGY0, LGX1, and LGX0 bits of line control data set 2 (command 4) are set to a value other than "0, 0, 0, 0"), the trimming dots are displayed for the character data after performing enlarged display processing. (Enlarged display is not performed on the trimming dots themselves.)
- Trimming dots are not added to graphics characters.
- Trimming dots are not added to character background character parts.
- Trimming dots are not added to underlines.
- Processing to display trimmings in the vertical and horizontal directions is performed on the character dot data corresponding to the character horizontal size (L, M, or S) and line character vertical size (2 to 32 dots). (For example, for a font design where there is a character dot on the 19th vertical line, if the line character vertical size is set to 18 dots, the dot on the 19th vertical line is not referenced when processing the trimming of the 18th vertical line.)

Note: In order to check the condition of the trimming display for each font (full perimeter, right, shadow trimming), check the font design using the corresponding OSDC pattern editor (PEDWIN2).

30.2.9.5 Line Enlarged Display

Line enlarged display is a function that controls the display size of characters, character backgrounds, and line backgrounds (including the line spacing). Enlarged display can be configured separately or simultaneously in the vertical and horizontal directions.

Although the enlargement control applies to the character shaded frame region of shaded backgrounds, it does not apply to trimming display regions. Furthermore, the display positions of lines following a line where line vertical enlarged display is specified are moved down.

Line enlargement control (per line setting)

Table 30-15 and Table 30-16 show the line enlargement control of line control data set 2 (command 4): bits LGX1, LGX0, LGY1, and LGY0.

Table 30-15. Line Enlargement Control (Per Line Setting)

LGY1	LGY0	Display size
0	0	Normal size
0	1	Double-height size
1	0	(Setting prohibited)
1	1	Quadruple-height size

Table 30-16. Line Enlargement Control (Per Line Setting)

LGX1	LGX0	Display size
0	0	Normal size
0	1	Double-width size
1	0	(Setting prohibited)
1	1	Quadruple-width size

Examples of line enlarged display

Figure 30-32 to Figure 30-35 show examples of line enlarged display for display sizes of normal, double-width, quadruple-width, double-height, quadruple-height, double width and height, and double-height/quadruple-width.

Figure 30-32. Examples of Line Enlarged Display

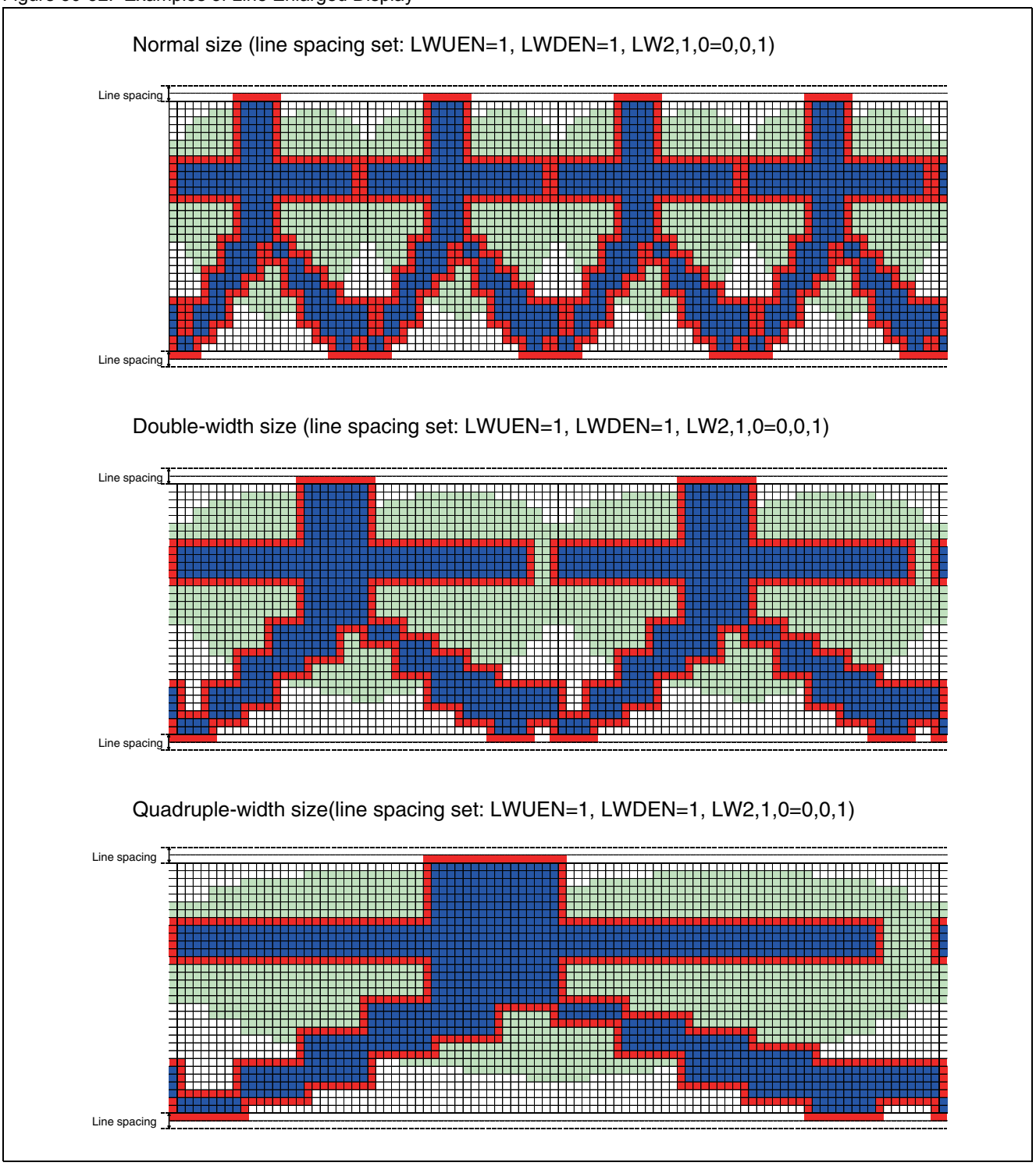


Figure 30-33. Examples of Line Enlarged Display (Continued)

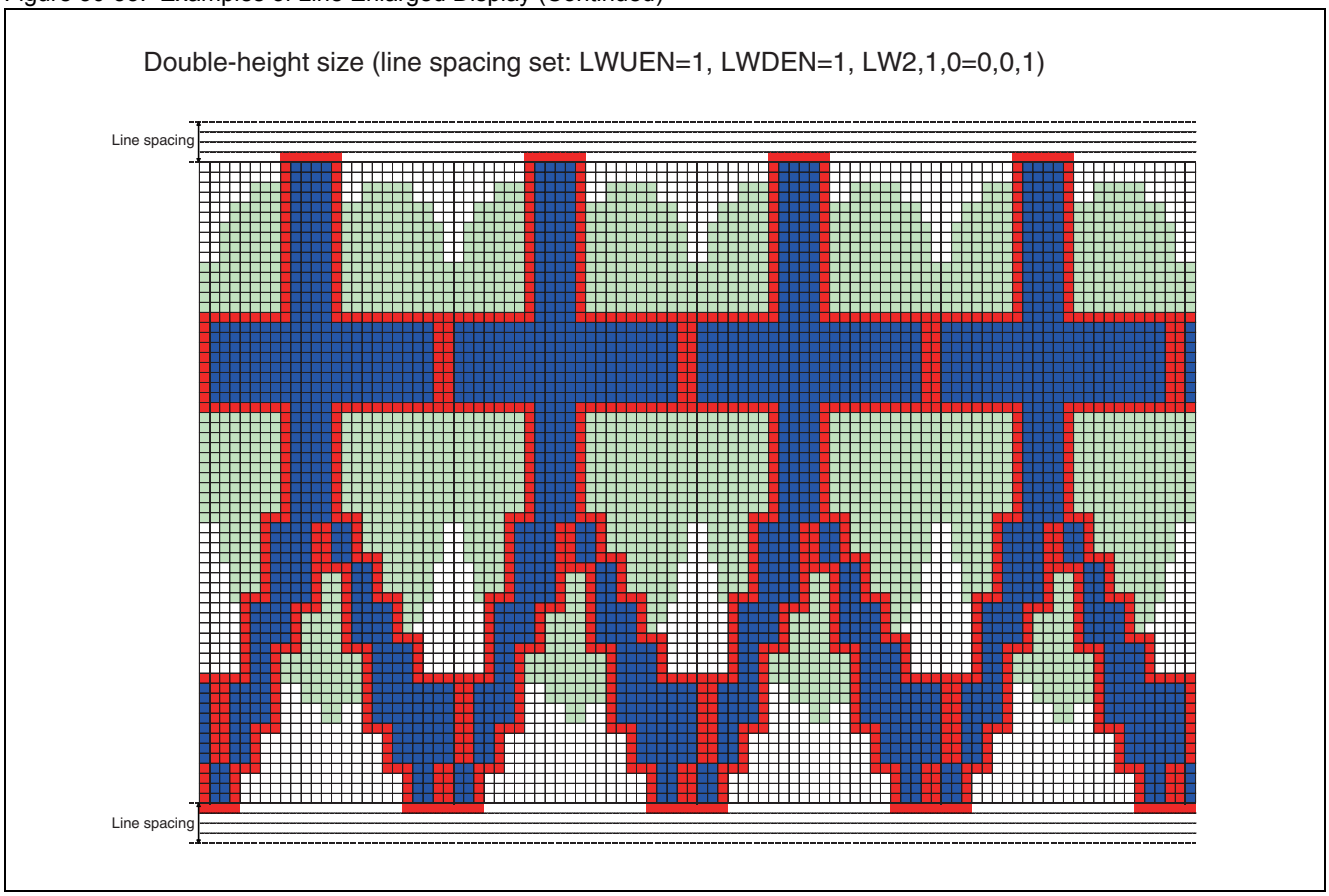


Figure 30-34. Examples of Line Enlarged Display (Continued)

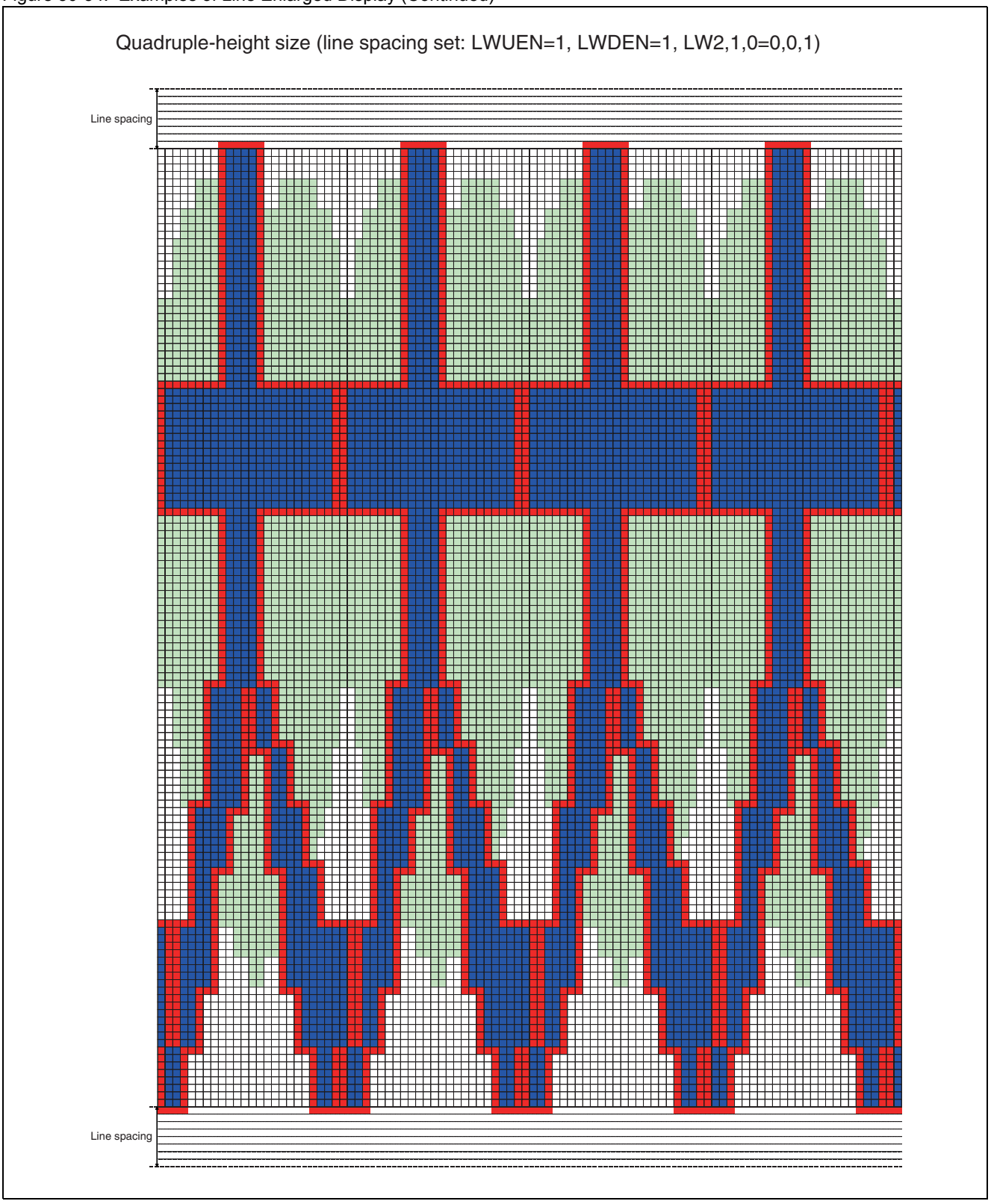
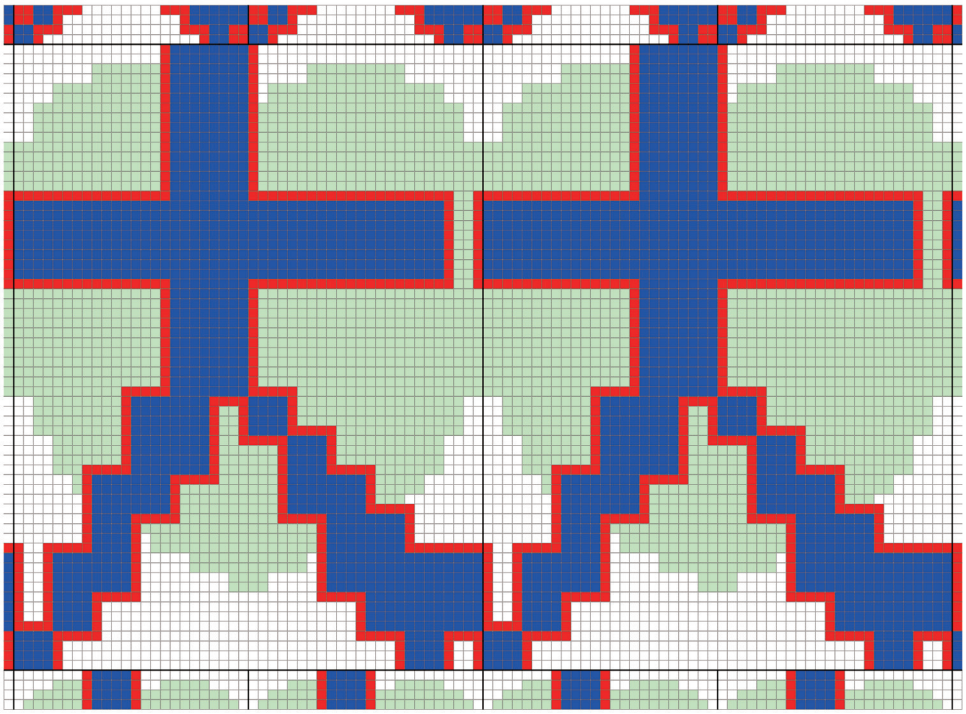
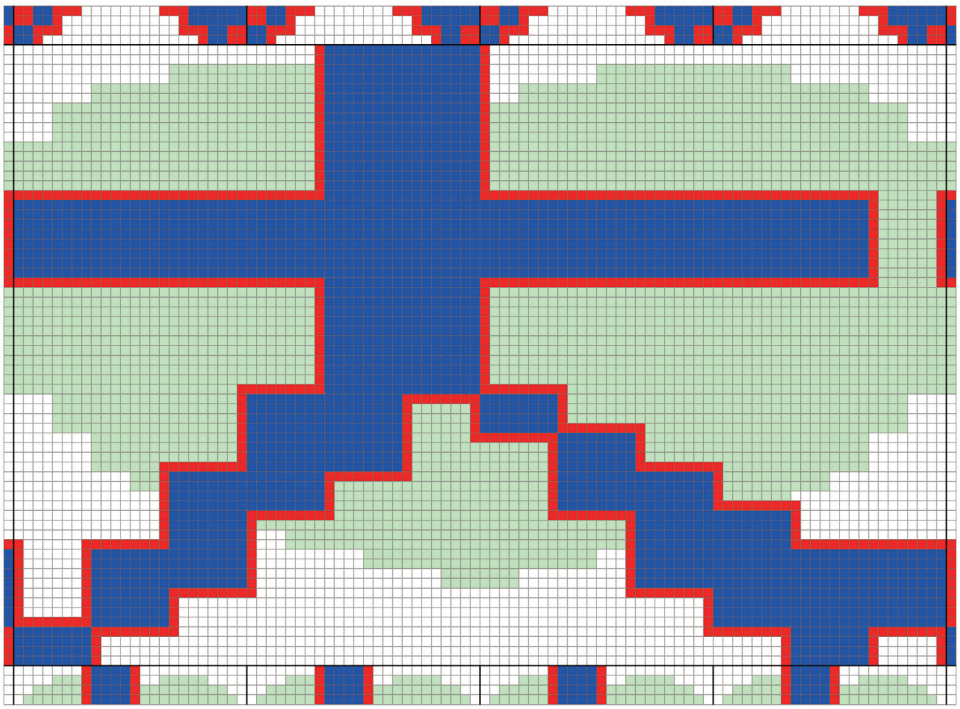


Figure 30-35. Examples of Line Enlarged Display (Continued)

Double-width/height size (line spacing not set: LWUEN=0,LWDEN=0, LW2,1,0=0,0,1)



Double-height/quadruple-width size (line spacing not set: LWUEN=1,LWDEN=1, LW2,1,0=0,0,0)



30.2.9.6 Graphics Character Control

Graphics character display is a function that displays the dot patterns of 8 characters stored in font memory as a 32-dot × 32-dot graphic. Each dot is able to display 256 colors.

Character/graphics character control (per character setting)

Graphics characters use 8 consecutive characters in font ROM. When displaying graphics character, set the lower 3 bits of the character code to 0.

Table 30-17 shows each of the character code settings.

Table 30-17. Codes for Specifying Graphics Characters

Display layer	Character code	
	Bits that can be set arbitrarily	Fixed bits Set these bits to "0".
Main screen	M13 to M3	M2 to M0
Screen background characters	PM13 to PM3	PM2 to PM0
Sprites	SM13 to SM3	SM2 to SM0

Note: Create the graphics characters by using the corresponding OSDC pattern editor (PEDWIN2).

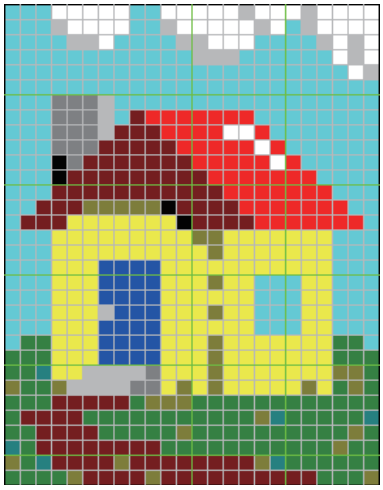
Table 30-18 to Figure 30-36 show the character/graphics character control of character data set 2 (command 2): bit MG.

Table 30-18. Character/Graphics Character Control (Per Character Setting)

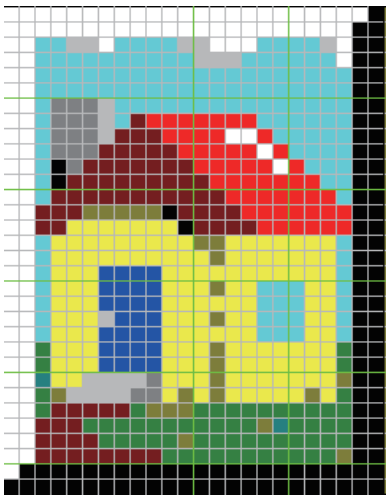
MG	Character/graphics character control
0	Normal character
1	Graphics character

Figure 30-36. Character/Graphics Character Control (Per Character Setting)

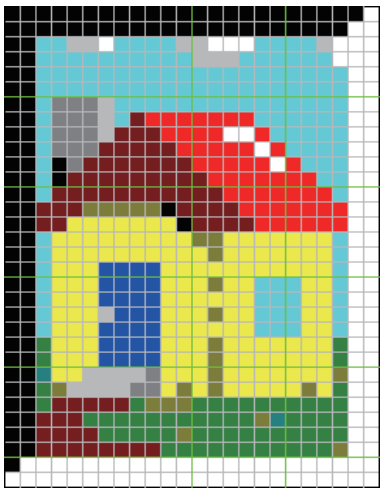
Graphic display example



(MG=1)



(MM1=1, MM0=1)



(MM1=1, MM0=0)

*: Shaded background display is enabled even for graphic display.

Graphics color/trimming color replacement control (per screen setting)

Table 30-19 shows the graphics color/trimming color replacement control of the graphics color control (command 6-1): bit GFC.

This control replaces an arbitrary color within the graphics characters (the color specified by bits GF7 to GF0) with the trimming color (LF7 to LF0) configured by line control data set 1 (command 3).

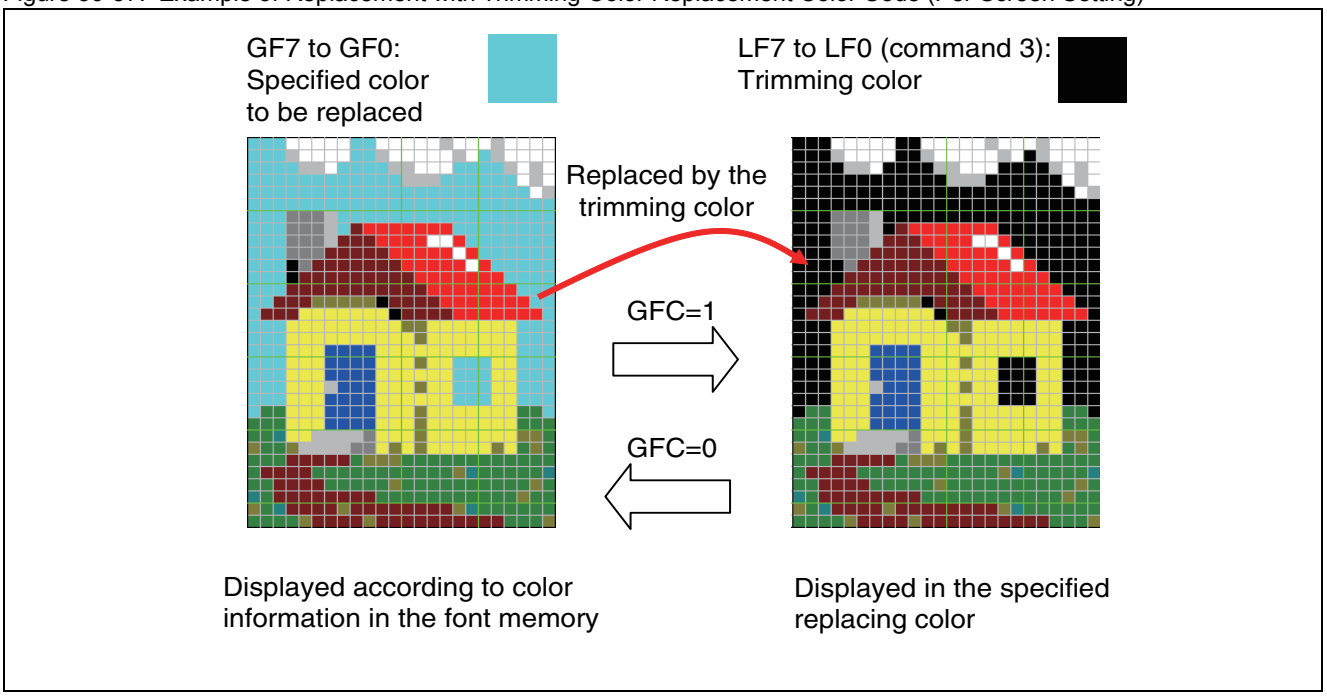
Table 30-19. Graphics Color/Trimming Color Replacement Control (Per Screen Setting)

GFC	Graphics character/trimming character replacement control
0	Do not replace the specified color.
1	Replace the specified color with the trimming color.

Trimming color replacement color code (per screen setting)

Figure 30-37 shows an example of replacement with graphics color control (command 6-1): bits GF7 to GF0.

Figure 30-37. Example of Replacement with Trimming Color Replacement Color Code (Per Screen Setting)



Notes:

- If graphics color/trimming color replacement control is ON (the GFC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the graphics color/trimming color replacement color code and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the trimming color replacement takes precedence.
- If graphics color/trimming color replacement control is ON (the GFC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the replacement trimming color code (trimming color (LF7 to LF0) of line control data set 1 (command 3)) and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the color becomes transparent and the lower layer color is displayed.
- If graphics color/character color replacement control is ON (the GCC bit is set to 1) and graphics color/trimming color replacement control is ON (the GFC bit is set to 1), set the value of the character color replacement color (bits GC7 to GC0) and the value of the trimming color replacement color code (bits GF7 to GF0) to different color codes.

Graphics color/character color replacement control (per screen setting)

Table 30-20 shows an example of graphics color/character color replacement control using graphics color control (command 6-1): bit GCC.

This control replaces an arbitrary color within the graphics characters (the color specified by bits GC7 to GC0) with the character color (MC7 to MC0) configured by character data set 1 (command 1).

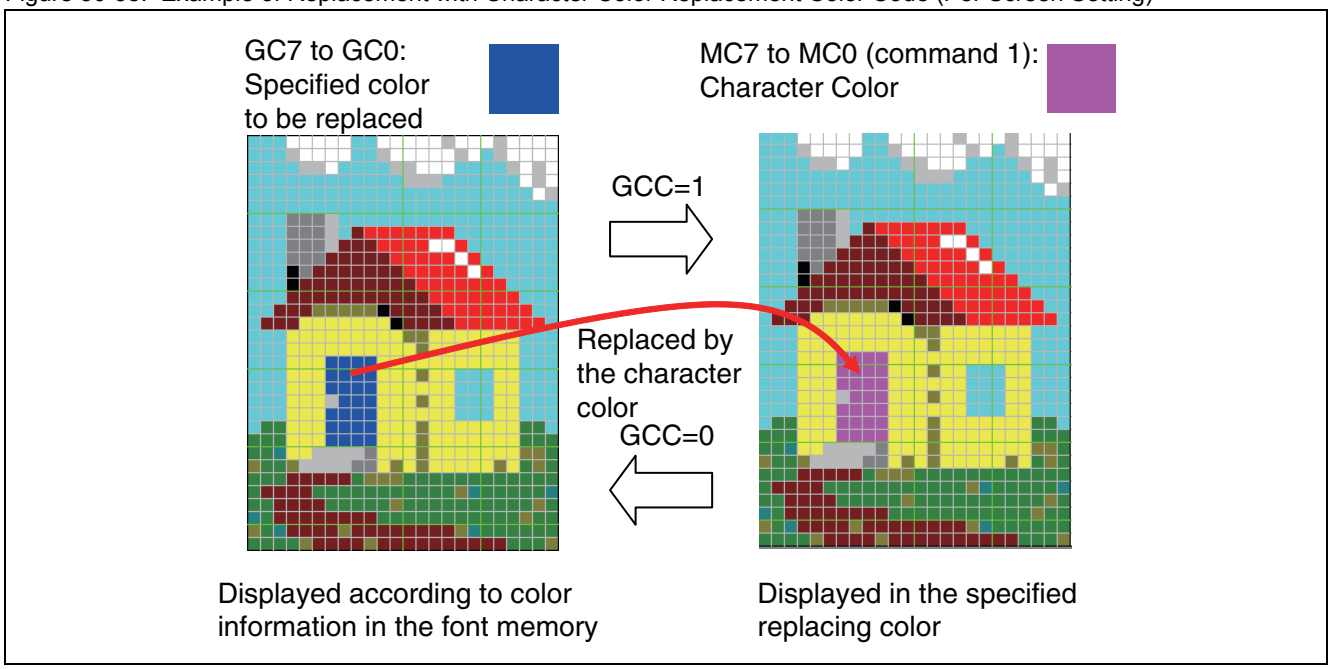
Table 30-20. Graphics Color/Character Color Replacement Control (Per Screen Setting)

GCC	Graphics color/character color replacement control
0	Do not replace the specified color.
1	Replace the specified color with the character color.

Character Color Replacement Color (Per Screen Setting)

Figure 30-38 shows an example of replacement with graphics color control (command 6-1): bits GC7 to GC0.

Figure 30-38. Example of Replacement with Character Color Replacement Color Code (Per Screen Setting)



Notes:

- If graphics color/character color replacement control is ON (the GCC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the graphics color/character color replacement color code and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the character color replacement takes precedence.
- If graphics color/character color replacement control is ON (the GCC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the replacement character color code (trimming color (LF7 to LF0) of line control data set 1 (command 1)) and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the color becomes transparent and the lower layer color is displayed.
- If graphics color/character color replacement control is ON (the GCC bit is set to 1) and graphics color/trimming color replacement control is ON (the GFC bit is set to 1), set the value of the character color replacement color code (bits GC7 to GC0) and the value of the trimming color replacement color code (bits GF7 to GF0) to different color codes.

30.2.9.7 Blink Control

The OSDC is able to configure per-character blinking (flashing) display settings. Furthermore, the period and duty ratio of the blinking display are also configurable.

Blinking control (per character setting)

Table 30-21 shows the blink control of character data set 1 (command 1): bits MBL and MBB.

Table 30-21. Blink Control (Per Character Setting)

MBL	MBB	Blink control
0	0	Blinking OFF (normal display)
1	0	Character blinking ON
0	1	Character background blinking ON
1	1	Character and character background blinking ON

Display examples

Figure 30-39 to Figure 30-47 show examples of blinking displayed on different backgrounds.

Figure 30-39. Example of Blinking Display With No Background (MM2=0, MM1=0, MM0=0)

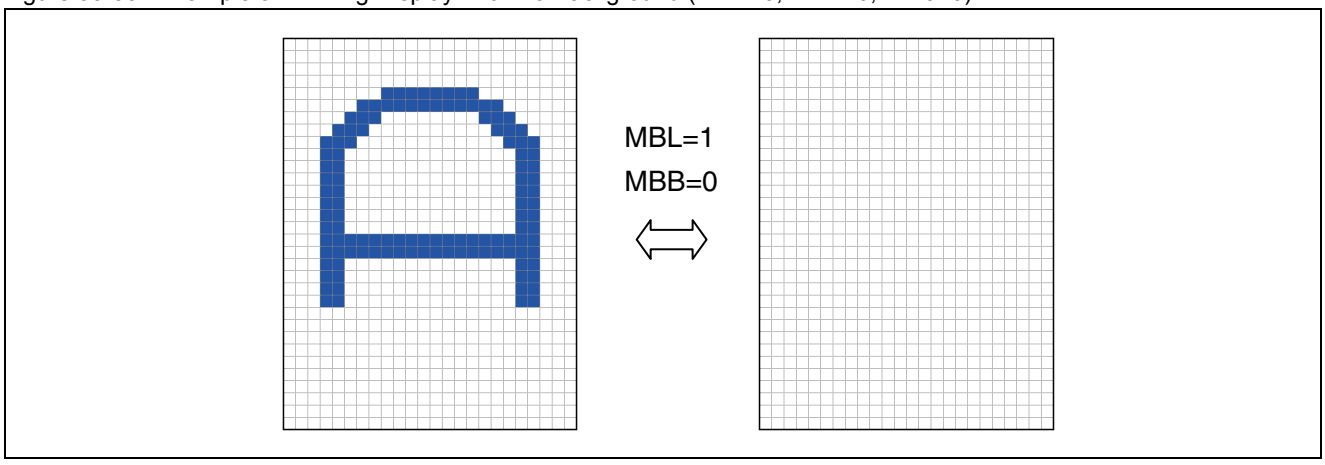


Figure 30-40. Example 1 of Blinking Display With Solid Background (MM2=0, MM1=0, MM0=1)

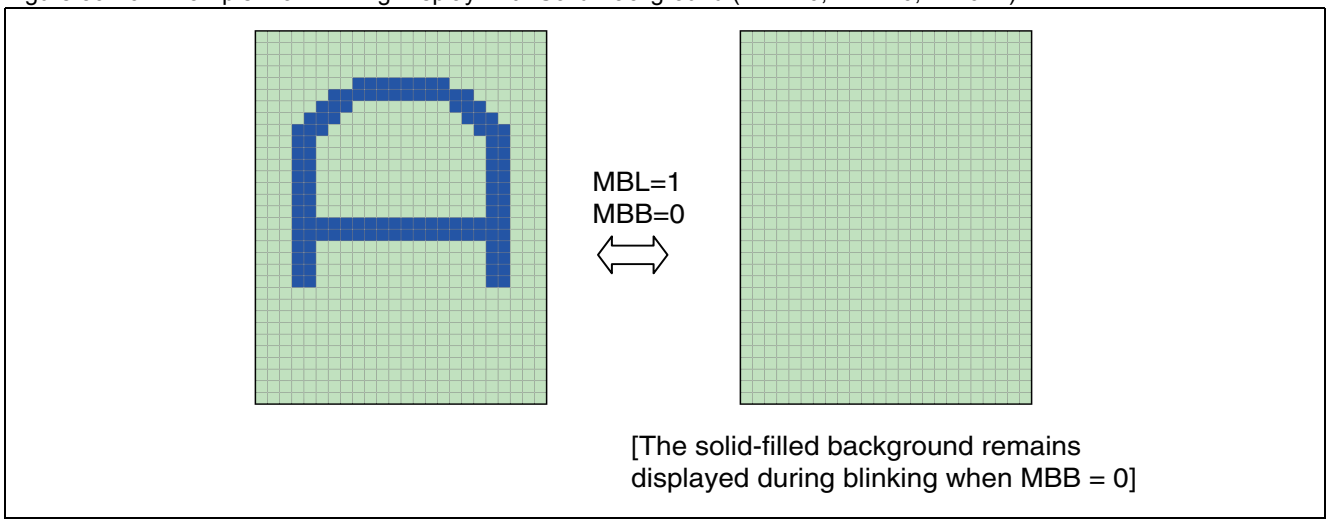


Figure 30-41. Example 2 of Blinking Display With Solid Background (MM2=0, MM1=0, MM0=1)

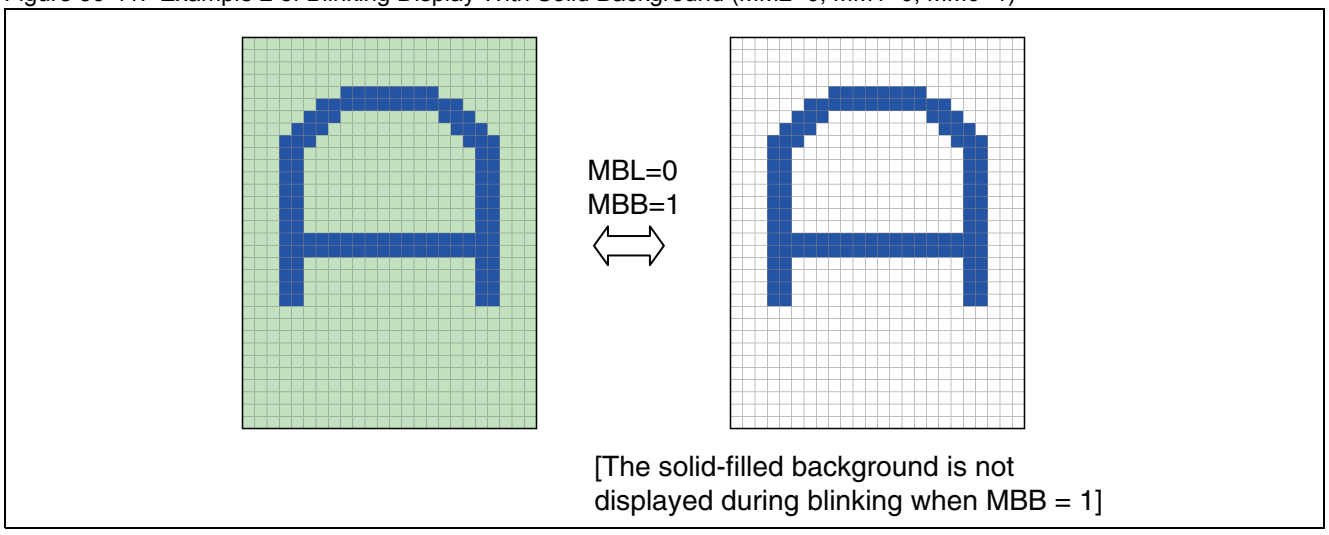


Figure 30-42. Example 3 of Blinking Display With Solid Background (MM2=0, MM1=0, MM0=1)

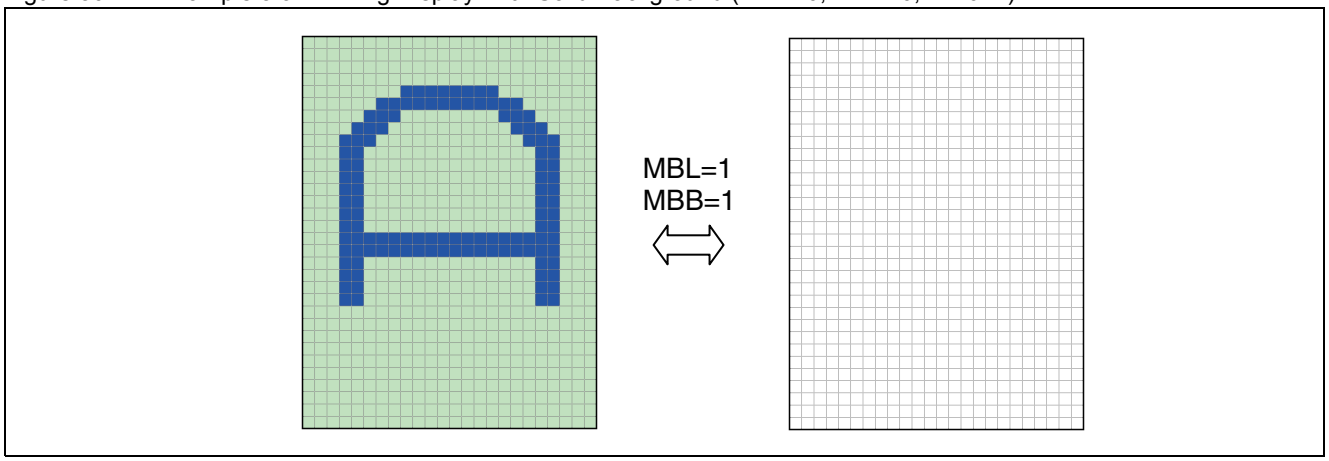


Figure 30-43. Example of Blinking Display With Background Character Display (MM2=1, MM1=0, MM0=0)

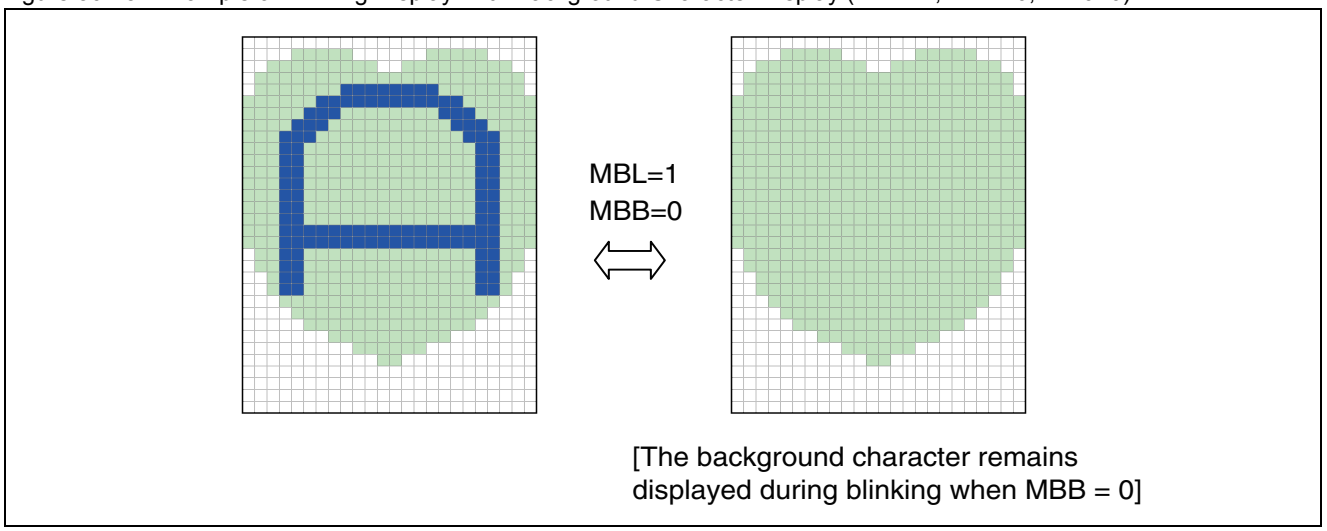


Figure 30-44. Example 1 of Blinking Display With Shaded Background Display (MM2=0, MM1=1, MM0=1)

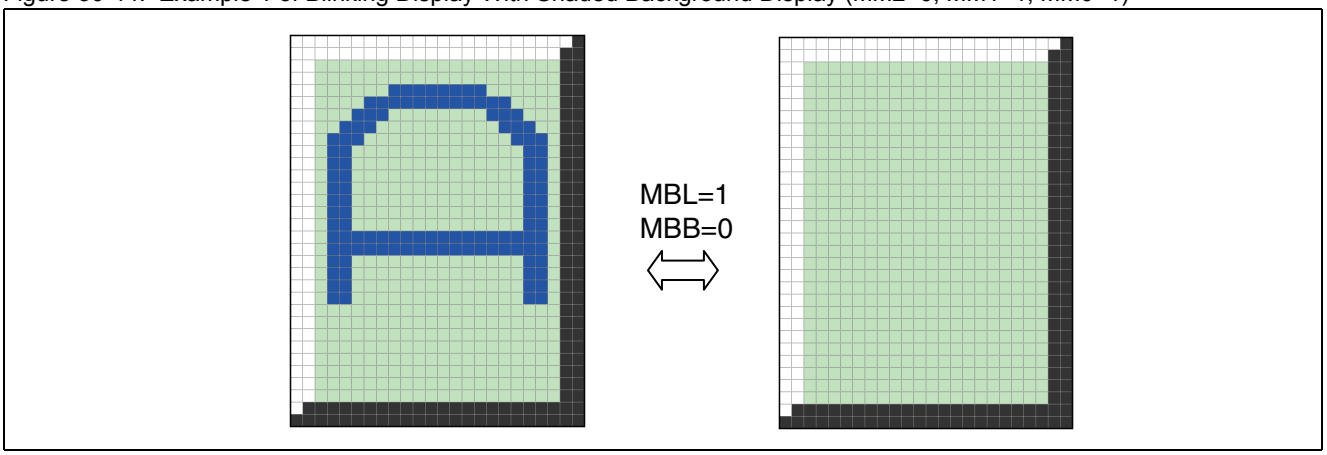


Figure 30-45. Example 2 of Blinking Display With Shaded Background Display (MM2=0, MM1=1, MM0=1)

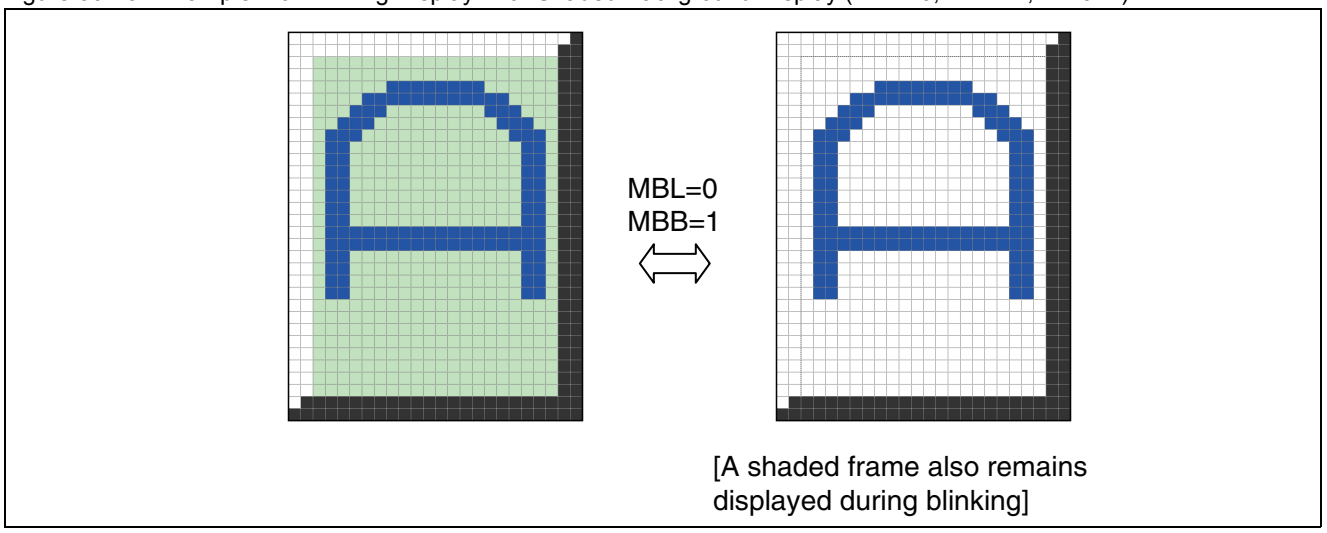


Figure 30-46. Example 3 of Blinking Display With Shaded Background Display (MM2=0, MM1=1, MM0=1)

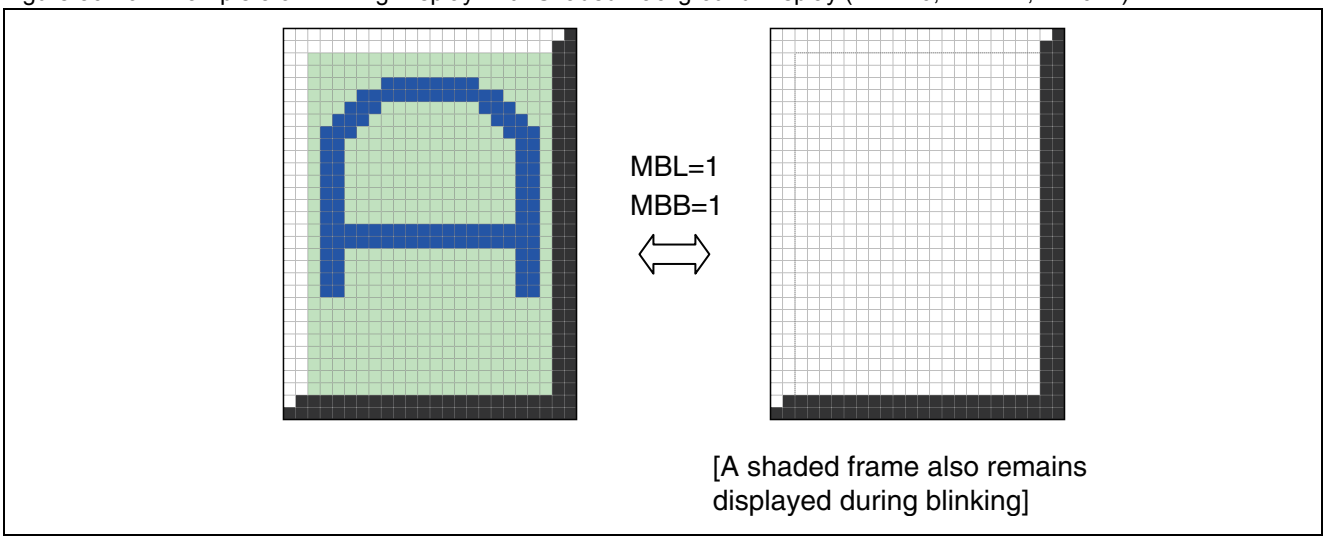
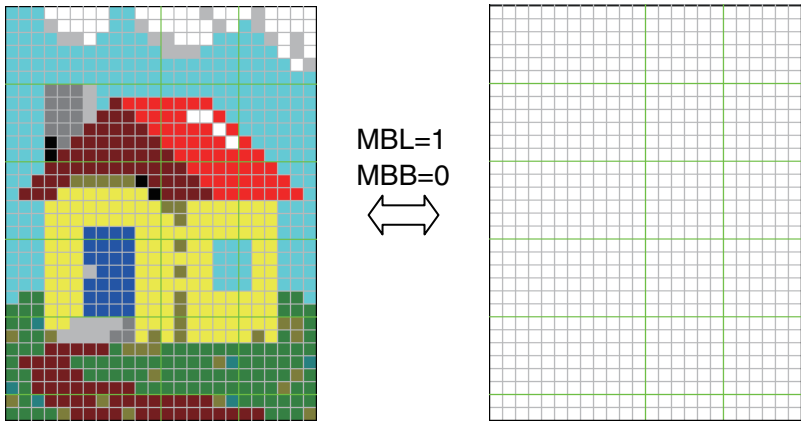


Figure 30-47. Example of Blinking Display of Graphic With No Background (MM2=0, MM1=0, MM0=0)



Note: If a background is configured when displaying a blinking graphic, then if MBB is set to 0, the background is displayed during the OFF part of the blinking, the same as when displaying a character.
The shaded frame of shaded display continues to be displayed during the OFF part of the blinking. (Shaded frames do not blink.)
Configuring blinking on the background character [(MBL=0, MBB=1) or (MBL=1, MBB=1)] of a character configured with character background character display (MM2=1, MM1=0, MM0=0) is forbidden.

Blink period

Table 30-22 shows the blinking period of the screen output control (command 5-0): bits BT1 and BT0.

Table 30-22. Blink Period Control (Per Screen Setting)

BT1	BT0	Blink period
0	0	16 × Vsync
0	1	32 × Vsync
1	0	48 × Vsync
1	1	64 × Vsync

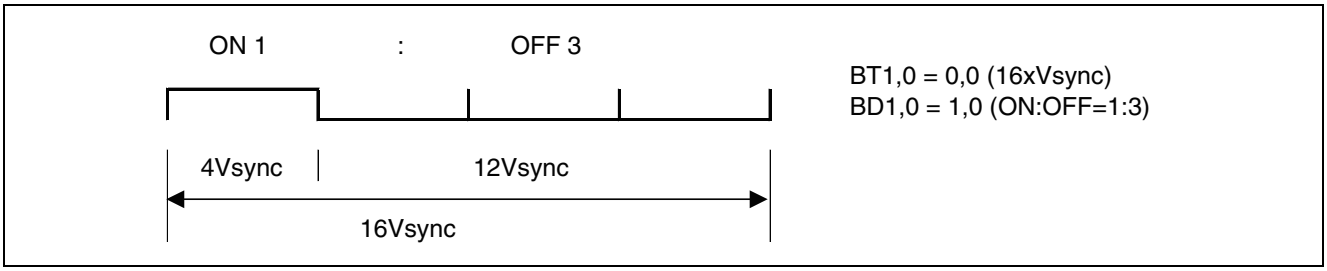
Blink duty ratio

Table 30-23 shows the blink duty control of the screen output control (command 5-0): bits BD1 and BD0.

Table 30-23. Blink Duty Control (Per Screen Setting)

BD1	BD0	Blink duty ratio (ON : OFF)
0	0	1 : 0 (Always ON)
0	1	1 : 1
1	0	1 : 3
1	1	3 : 1

Note: The relationship between the blink period and duty ratio is shown by the following example.



30.2.9.8 Transparency Control

An arbitrary display color can be used to display the color of the layer below by using transparency control.

Transparency Control (Per Screen Setting)

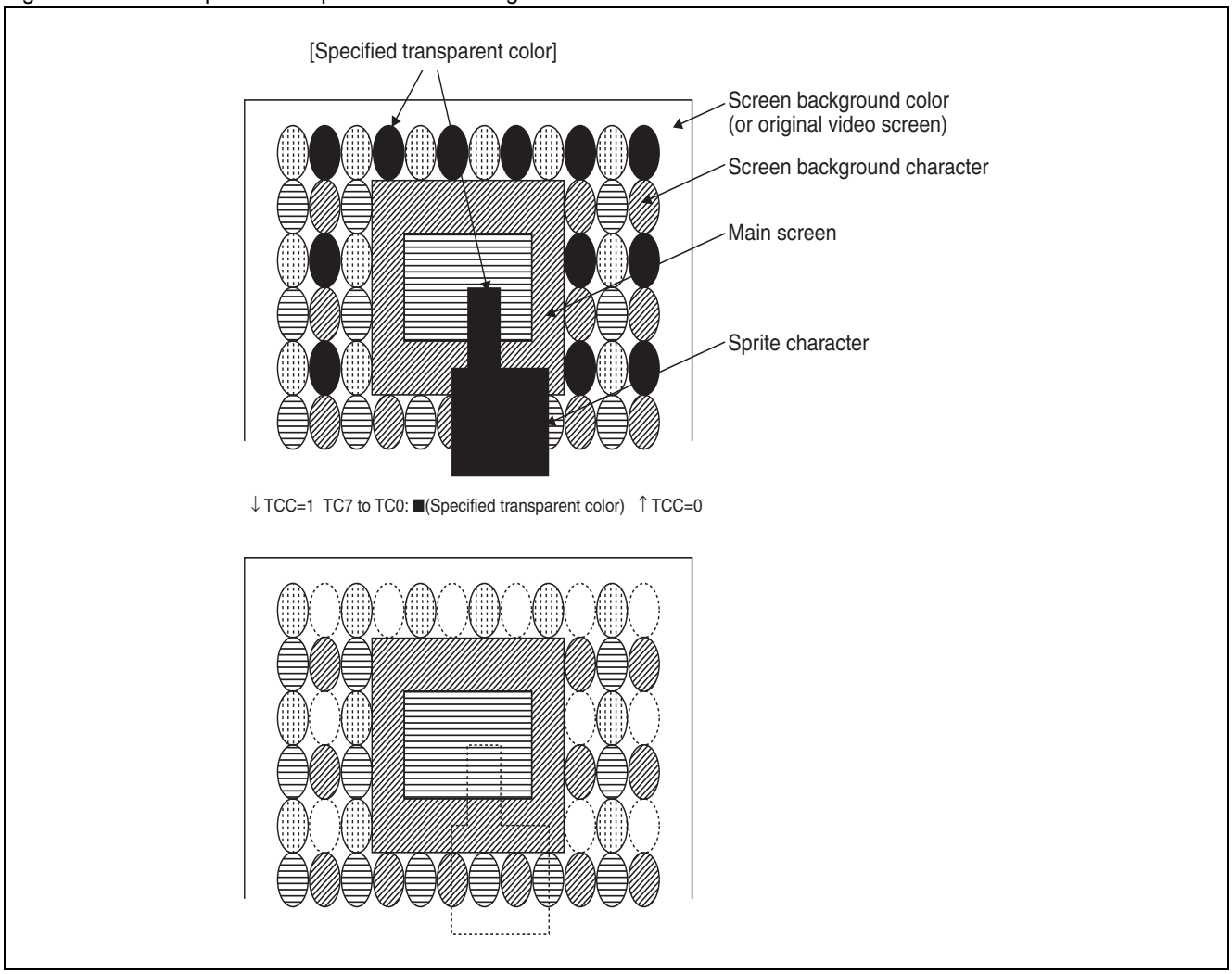
Table 30-24 shows the transparent color control of the transparent color control (command 6-0): bit TCC.

Table 30-24. Transparent Color Control (Per Screen Setting)

TCC	Transparent color control
0	Does not perform transparent color control.
1	Performs transparent color control.

Figure 30-48 shows an example of setting the transparent color code of the transparent color control (command 6-0): bits TC7 to TC0 (where the solid black parts in the diagram are set to the transparent color).

Figure 30-48. Example of Transparent Color Setting



30.2.9.9 Alpha Blend Output Control

The alpha blend output control outputs the timing and quantity of alpha blend. This makes it possible to perform alpha blending using an external circuit.

Alpha blend output control 1 (per line setting)

Table 30-25 shows the commands related to line alpha blend settings in line control data set 1 (command 3).

Table 30-25. Line Alpha Blend Settings (Per Line Settings)

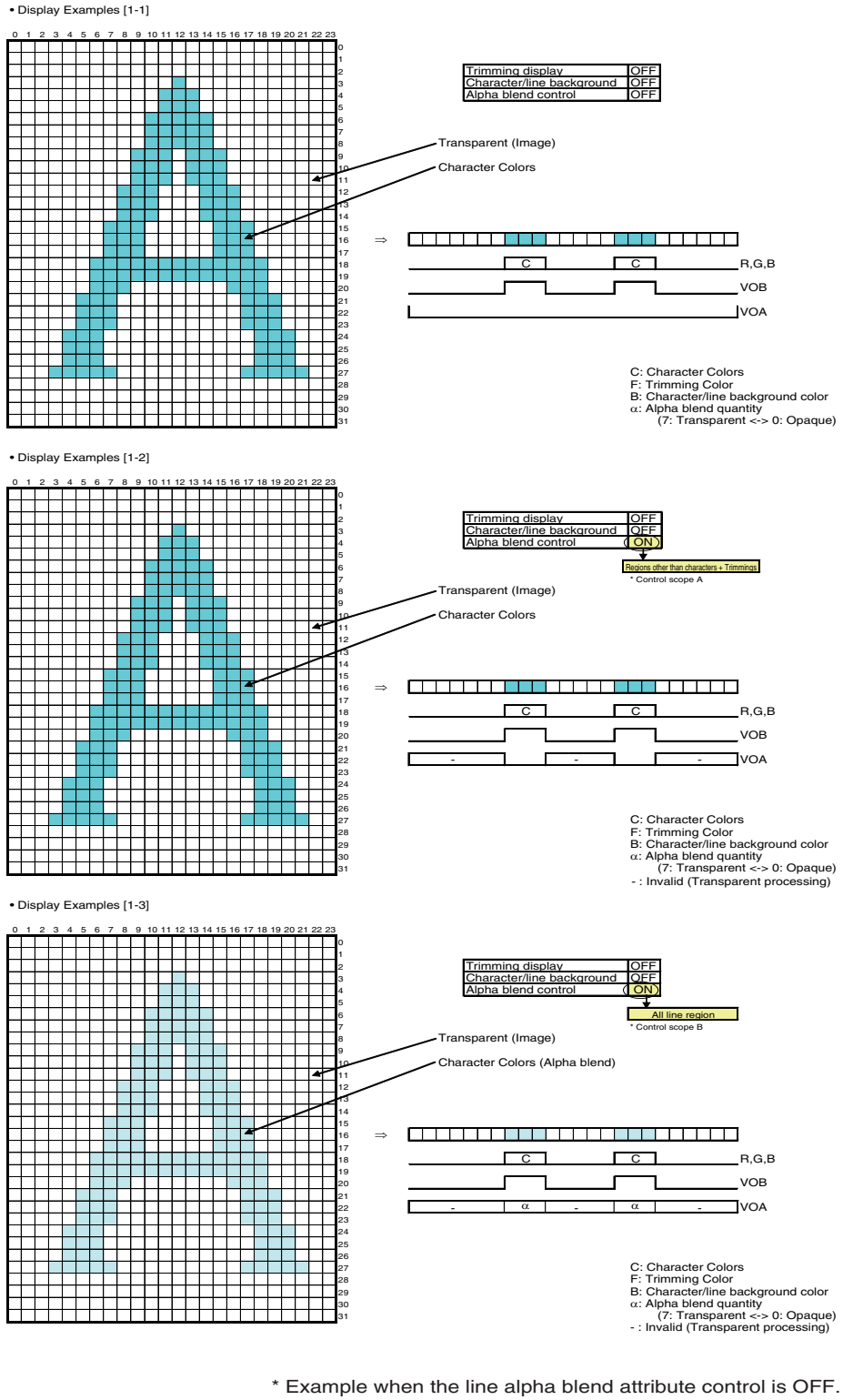
LALEN	Line alpha blend output control
0	OFF
1	ON

LALCM	Line alpha blend control scope control
0	Control scope A [Regions other than characters and trimmings]
1	Control scope B [All line region]

LAL2, LAL1, LAL0	Line alpha blend quantity
1, 1, 1	7 [Transparent]
1, 1, 0	6 ↑
1, 0, 1	5
1, 0, 0	4
0, 1, 1	3
0, 1, 0	2
0, 0, 1	1 ↓
0, 0, 0	0 [Opaque]

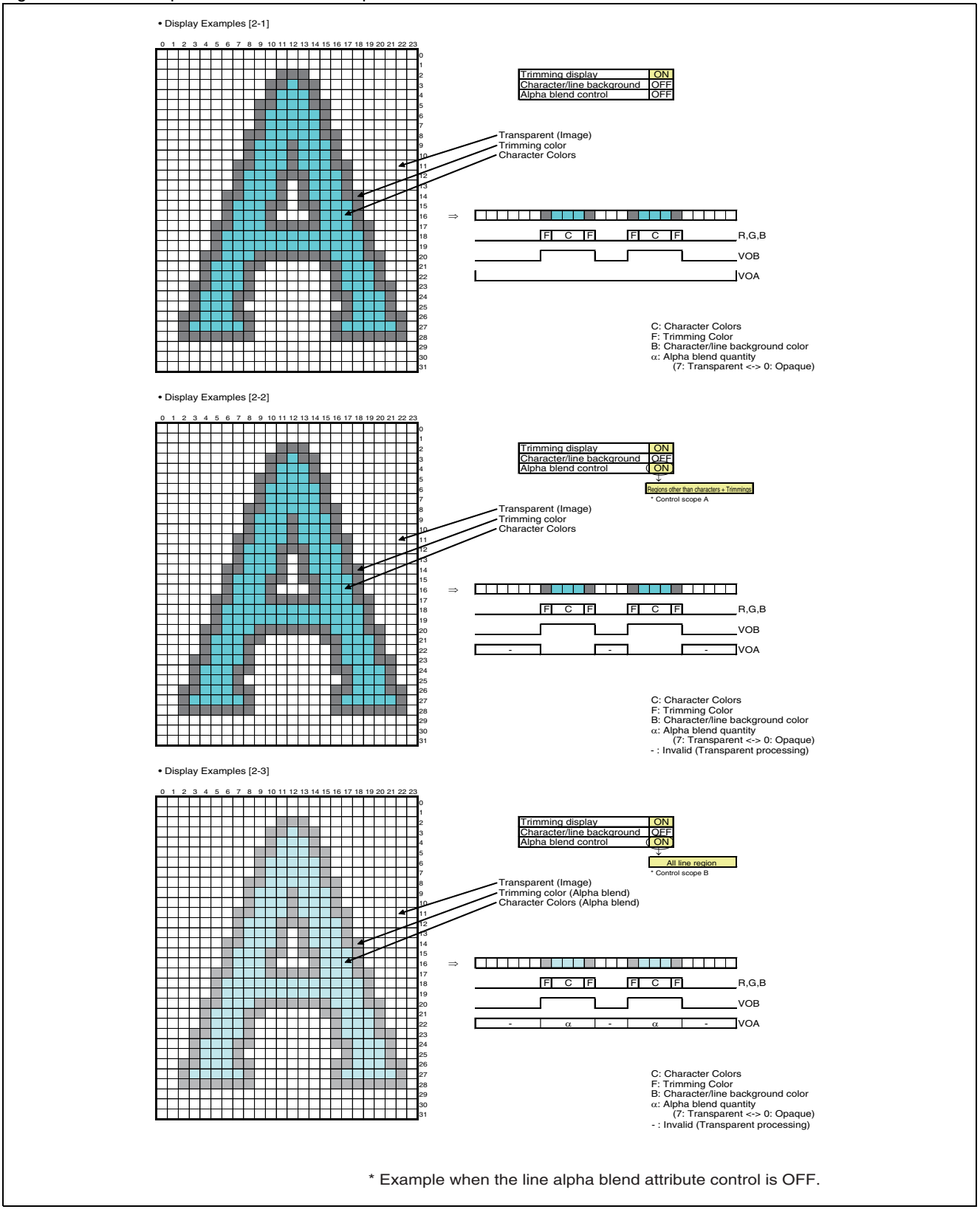
Figure 30-49 to Figure 30-52 show examples of line alpha blend control.

Figure 30-49. Line Alpha Blend Control Example 1



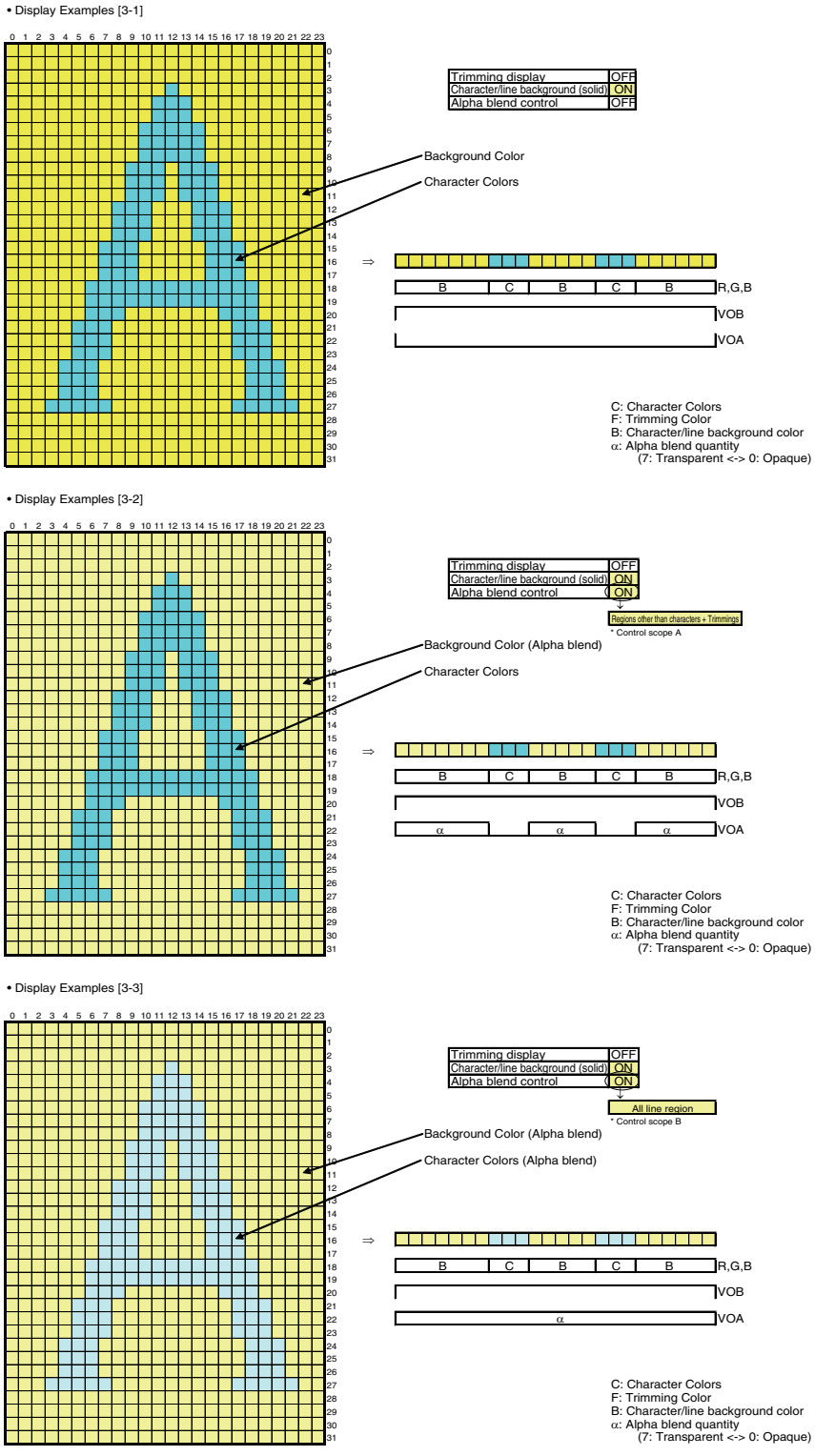
* Example when the line alpha blend attribute control is OFF.

Figure 30-50. Line Alpha Blend Control Example 2



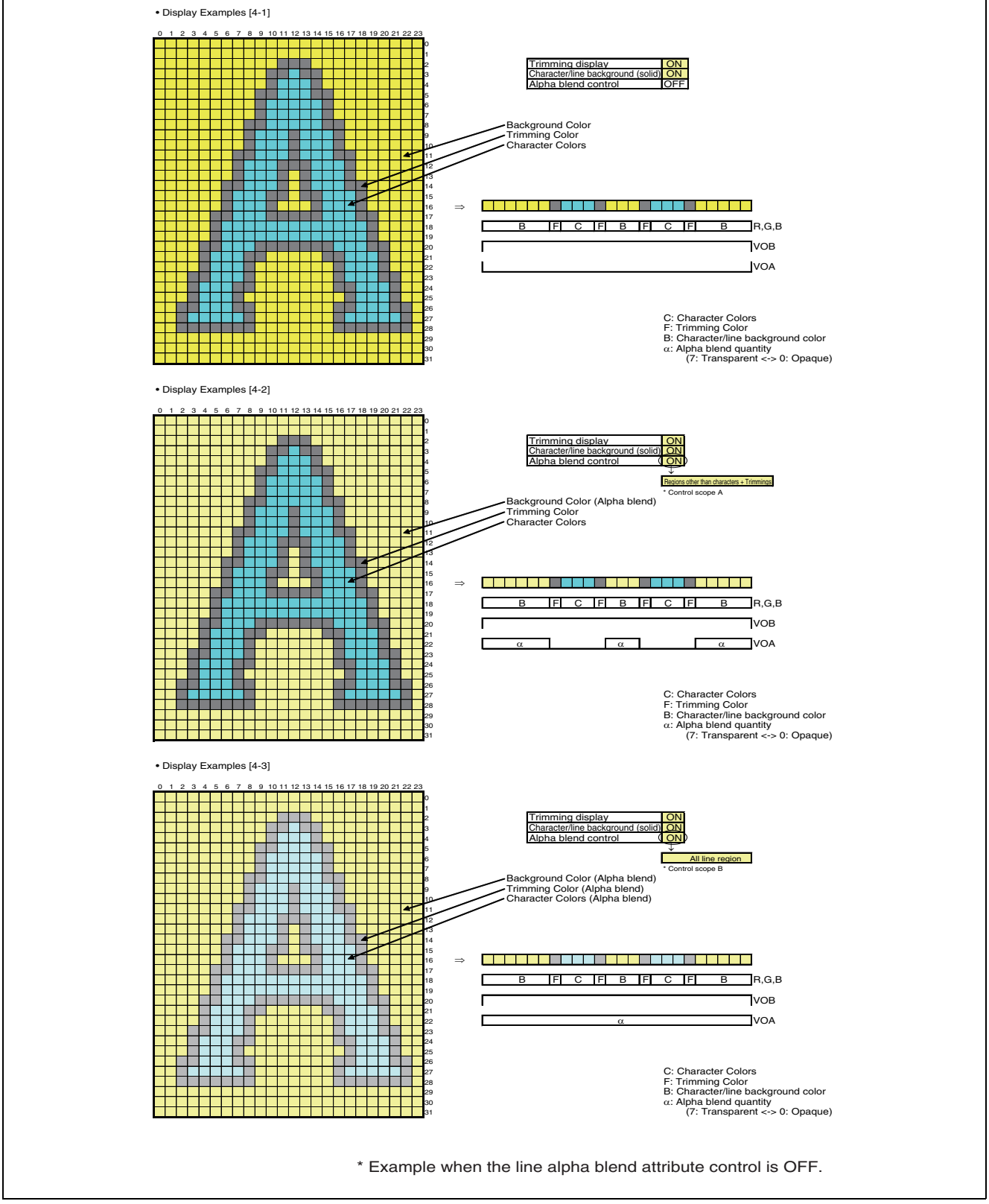
* Example when the line alpha blend attribute control is OFF.

Figure 30-51. Line Alpha Blend Control Example 3



* Example when the line alpha blend attribute control is OFF.

Figure 30-52. Line Alpha Blend Control Example 4



Alpha blend output control 2 (per line settings and per screen settings)

Table 30-26 shows the commands related to line alpha blend attribute control in line control data set 1 (command 3).

Table 30-26. Line Alpha Blend Attribute Control (Per Line Settings)

LALCA1	LALCA0	Line alpha blend attribute control
0	0	OFF
0	1	Only character alpha blend A attributes[1] are enabled
1	0	Only character alpha blend B attributes[1] are enabled
1	1	Both character alpha blend A and B attributes are enabled

[1]: The character alpha blend A attribute and B attribute are configured using the MA bit of character data set 2 (command 2).

Table 30-27 shows the commands related to the quantity setting when the alpha blend is disabled in the screen output control (command 5-0).

Table 30-27. Quantity Setting when Alpha Blend Attribute Disabled (Per Screen Setting)

SALCC	Quantity setting when alpha blend attribute disabled[1]
0	Fixed at alpha blend quantity 0 (opaque)
1	Fixed at alpha blend quantity 7 (transparent)

[1]: The SALCC setting for the quantity setting when the alpha blend attribute is disabled is only enabled when the line alpha blend attribute control (LALCA1 and LALCA0) settings is set to character alpha blend A attribute only enabled or B attribute only enabled.

Figure 30-53 to Figure 30-54 show examples of line alpha blend attribute control.

Figure 30-53. Line Alpha Blend Attribute Control Example 1

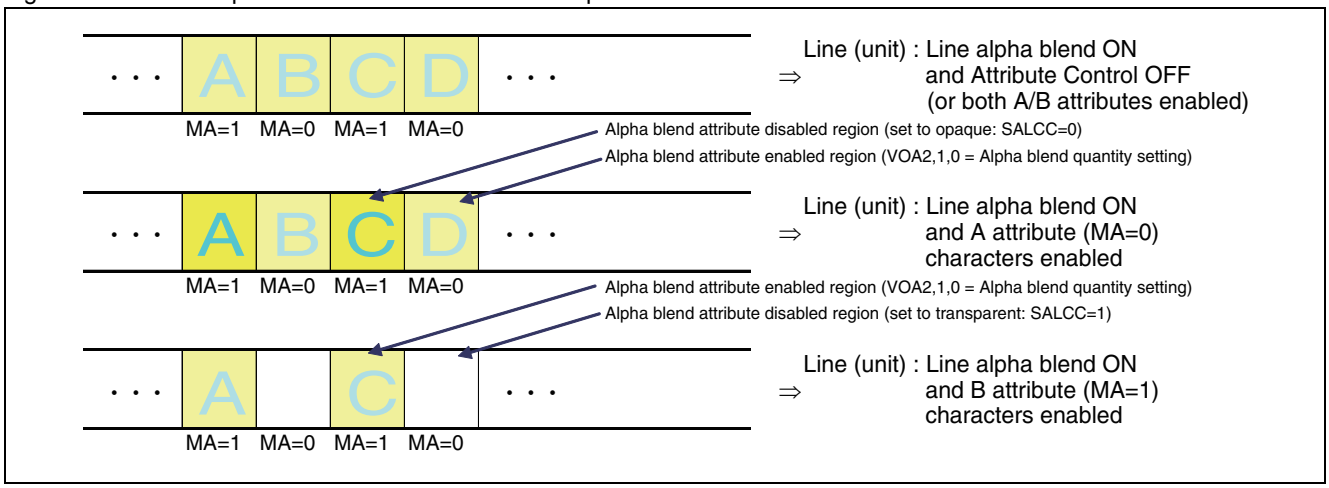
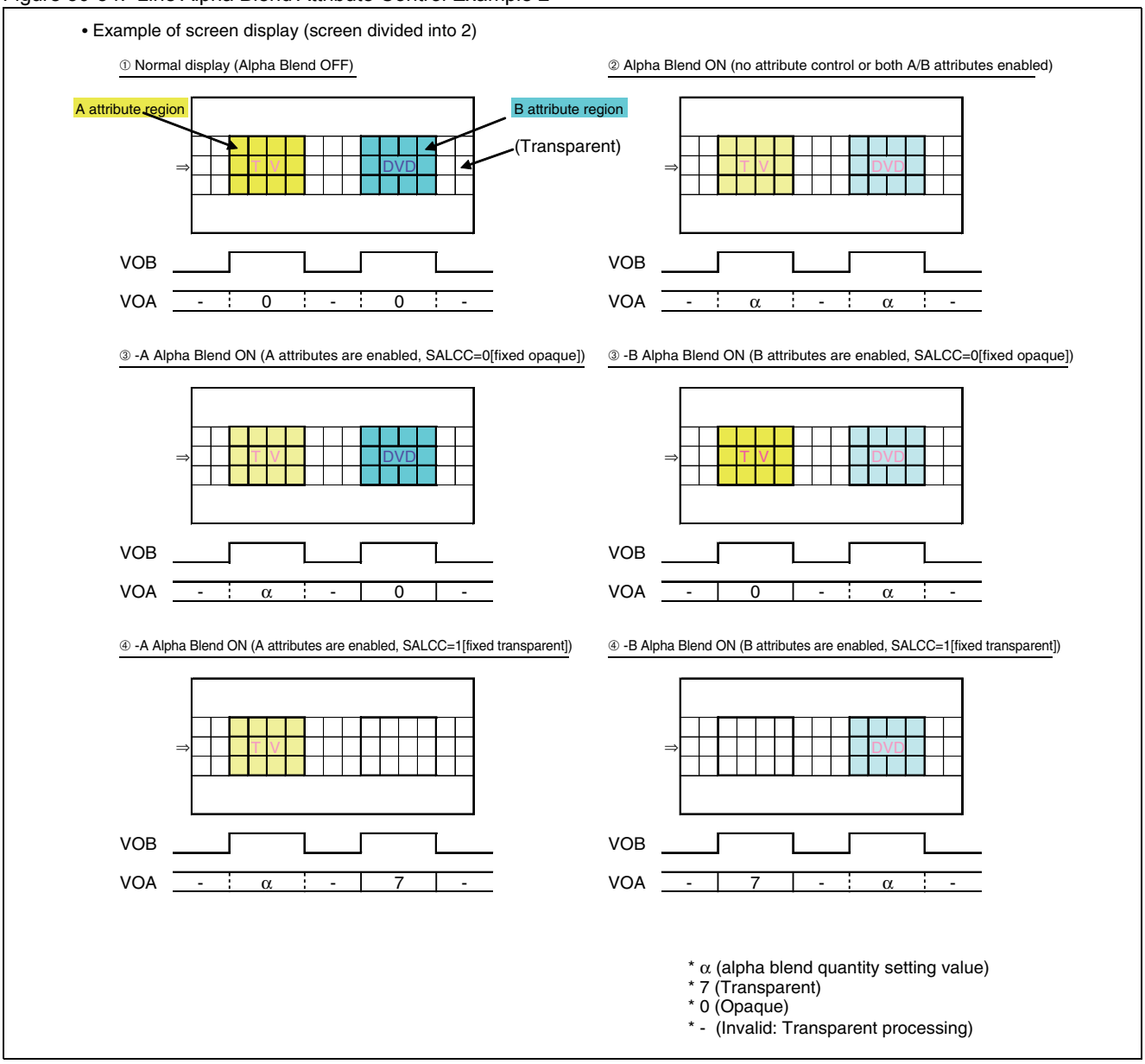


Figure 30-54. Line Alpha Blend Attribute Control Example 2



Note: Line alpha blend attribute control is only valid for dot regions of character attributes (character + trimming, character background).

Alpha blend output control 3 (per screen setting)

Table 30-28 shows the commands related to screen background alpha blend control in the screen background control (command 8-0).

Table 30-28. Screen Background Alpha Blend Control (Per Screen Setting)

UALEN	Screen background alpha blend control
0	OFF
1	ON

UAL2, UAL1, UAL0	Screen background alpha blend quantity
1, 1, 1	7 [Transparent]
1, 1, 0	6 ↑
1, 0, 1	5
1, 0, 0	4
0, 1, 1	3
0, 1, 0	2
0, 0, 1	1 ↓
0, 0, 0	0 [Opaque]

Note: Screen background alpha blend is enabled when screen output control (command 5-0) UDS=1.
It is also enabled when PDS=1 in the main screen.

Note: Screen background alpha blend control is also enabled in the alpha blend quantity output control in screen background character regions in the main screen.

Alpha blend output by priority of display dot and display period of the screen

Table 30-29 shows the alpha blend output by priority of display dot and display period of the screen.

Table 30-29. Alpha Blend Output by Priority of Display Dot and Display Period

Display period[1]	Window period[2]		MAIN, SUB screen priority control[3] (MCC)	Priority of Display Dot						VOA2,1,0 Alpha Blend Output
				Sprite Character		Normal Screen		Screen Background		
	MAIN	SUB		MAIN	SUB	MAIN	SUB	MAIN	SUB	
Internal	Ext.	Int.	—	0	0	—	0	—	1	SUB screen background α
				0	0	—	1	—	—	SUB line α
				0	1	—	—	—	—	Opaque
				1	—	—	—	—	—	Opaque
	Int.	Ext.	—	0	0	0	—	1	—	MAIN screen background α
				0	0	1	—	—	—	MAIN line α
				0	1	—	—	—	—	Opaque
				1	—	—	—	—	—	Opaque
	Int.	Int.	0	0	0	0	0	0	1	SUB screen background α
				0	0	0	0	1	—	MAIN screen background α
				0	0	0	1	—	—	SUB line α
				0	0	1	—	—	—	MAIN line α
				0	1	—	—	—	—	Opaque
				1	—	—	—	—	—	Opaque
			1	0	0	0	0	—	1	SUB screen background α
				0	0	0	0	1	0	MAIN screen background α
				0	0	—	1	—	—	SUB line α
				0	0	1	0	—	—	MAIN line α
				—	1	—	—	—	—	Opaque
				1	0	—	—	—	—	Opaque
External	—	—	—	0	0	0	0	0	0	Opaque

*1 Display period: Command 14-0 (display period control 1), command 14-1 (display period control 2)
*2 Window period: Command 8-1 (window period control 1), command 8-2 (window period control 2)
*3 MAIN, SUB screen priority control: [SUB operation] command 5-0 (screen output control) MCC bit

Note: For the alpha blend output signal (VOA2,1,0), the external circuit should be configured as the period, where the dots of OSDC are existing (VOB = High period), can be enabled. Configure the external circuit as transparent (image) when VOB is low.|
The alpha blend output outputs the data corresponding to the image data of the upper layer in the main screen/sub screen display order. In sprite character regions, the alpha blend output is opaque (VOA2,1,0=0,0,0).
When the alpha blend control is OFF, the alpha blend output signal (VOA2,1,0) outputs all output Low (for positive logic).

30.2.10 Character Background Display

There are 7 types of character backgrounds and 256 colors of character background color, and each of these is configurable per character.

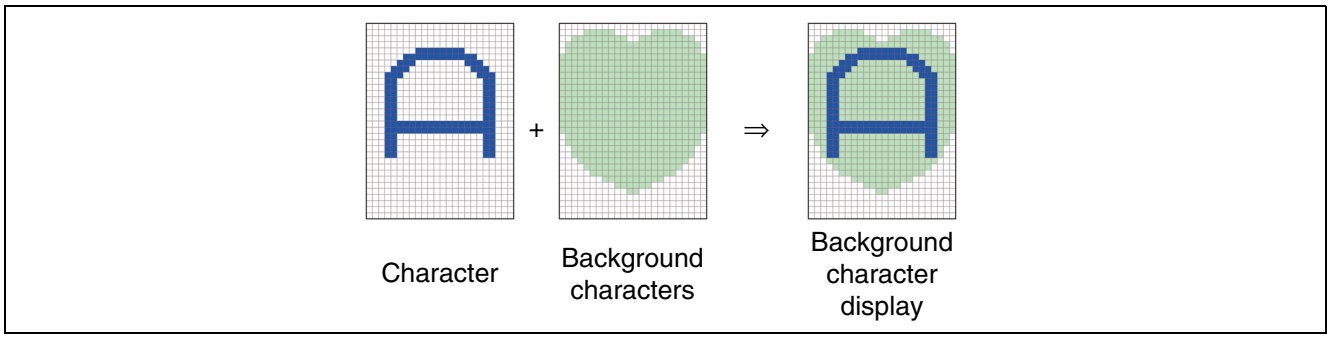
Character background control (per character setting)

Table 30-30 shows the character background control of character data set 1 (command 1): bits MM2, MM1, and MM0.

Table 30-30. Character Background Control (Per Character Setting)

MM2	MM1	MM0	Character background
0	0	0	No background (not displayed)
0	0	1	Solid background
0	1	0	Shaded background concaved (solid)
0	1	1	Shaded background convexed (solid)
1	0	0	Background character*
1	0	1	(Setting prohibited)
1	1	0	Shaded background concaved (background character[1])
1	1	1	Shaded background convexed (background character[1])

[1]: The background character displays an arbitrary character as the background.
(Do not specify a graphics character as the background character.)



Shaded background frame highlight color (configurable per character from 8 types and 256 colors)

The shaded background highlight color can be configured by setting BxH7 to BxH0 (x = 0 to 7) of the shaded background frame color control (command 15-0 to command 15-3) to the color code, and setting the character shaded background frame color type selection control (command 1) MSC2, MSC1, and MSC0 to the color type selection from among the 8 frame color types (0 to 7).

Shaded background frame shadow color (configurable per character from 8 types and 256 colors)

The shaded background shadow color can be configured by setting BxS7 to BxS0 (x = 0 to 7) of the shaded background frame color control (command 15-0 to command 15-3) to the color code, and setting the character shaded background frame color type selection control (command 1) MSC2, MSC1, and MSC0 to the color type selection from among the 8 frame color types (0 to 7).

Character background color [solid/background character] (configurable per character from 256 colors)

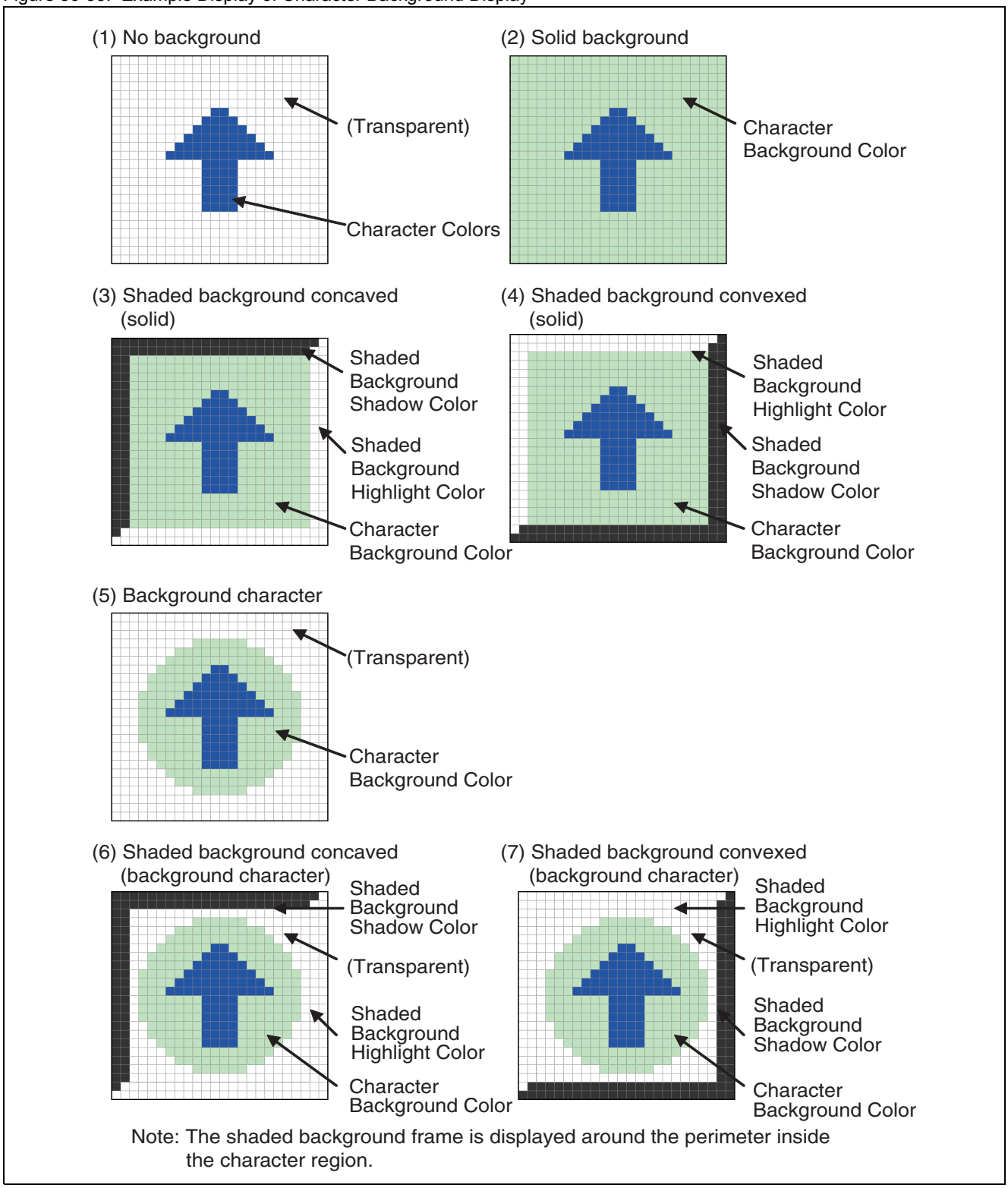
The character background color can be configured by setting bits MB7 to MB0 of character data set 1 (command 1) to the color code.

Note: The character shaded background frame color type selection is selected from 8 sets (0 to 7) of shaded background frame highlight color/shadow color.

Display examples

Figure 30-55 shows display examples for each of the background controls.

Figure 30-55. Example Display of Character Background Display



Note: The shaded background frame is displayed around the perimeter inside the character region.

Character shaded background frame width control (per line setting)

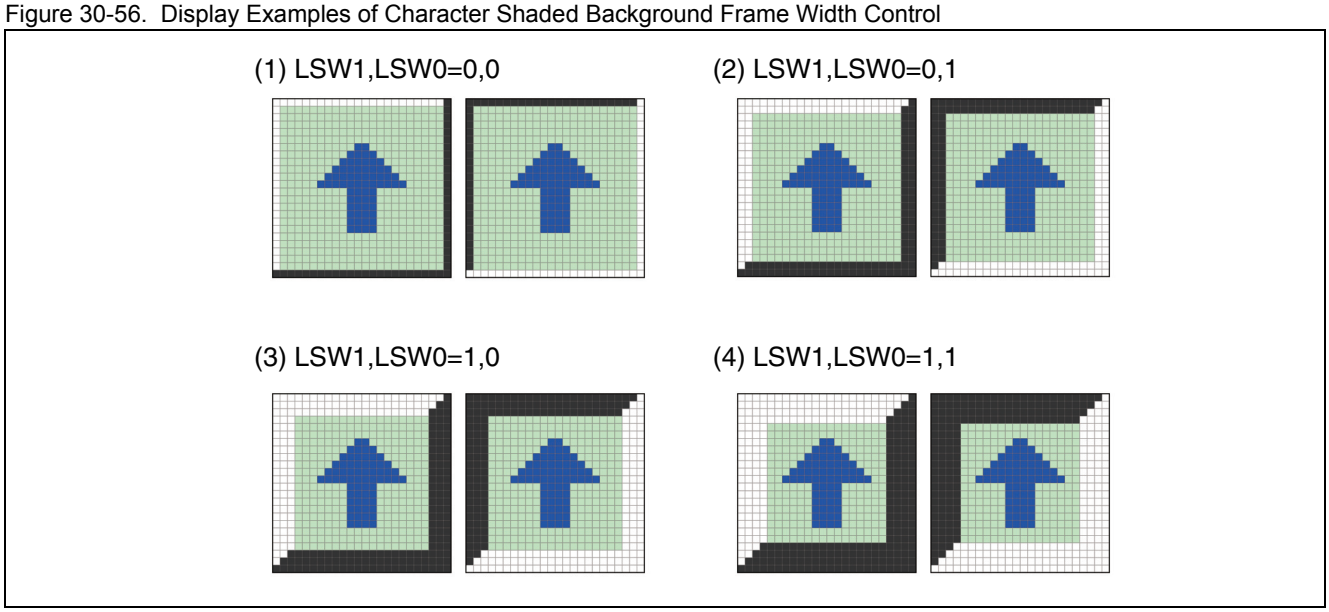
Table 30-31 shows the character shaded background frame width control of line control data set 2 (command 4): bits LSW1 and LSW0.

Table 30-31. Character Shaded Background Frame Width Control (Per Line Setting)

LSW1	LSW0	Character shaded background frame width
0	0	1 dot
0	1	2 dots
1	0	3 dots
1	1	4 dots

Figure 30-56 shows display examples when the character shaded background frame width is controlled.

Figure 30-56. Display Examples of Character Shaded Background Frame Width Control



Note: If the total character shaded frame width in the vertical direction is equal to or exceeds the line character vertical size during shaded background concaved/convexed display,
I.e., if LHS[3:0] (line character vertical size setting dot count) ≤ LSW[1:0] (character shaded background frame width setting dot count) × 2, the shaded frame cannot be displayed properly and shaded background concaved/convexed should not be configured.

30.2.10.1 Shaded Background Right Character Merge Display

The shaded background right character merge display is a function that prevents the display of the right edge of the shaded frame of a given character and the left edge of the frame of the next (right neighboring) character. This makes it possible to display a shaded background around multiple characters in the horizontal direction.

Shaded background right character merge control (per character setting)

Table 30-32 shows the shaded background right character merge control of character data set 2 (command 2): bit MR.

Table 30-32. Shaded Background Right Character Merge Control (Per Character Setting)

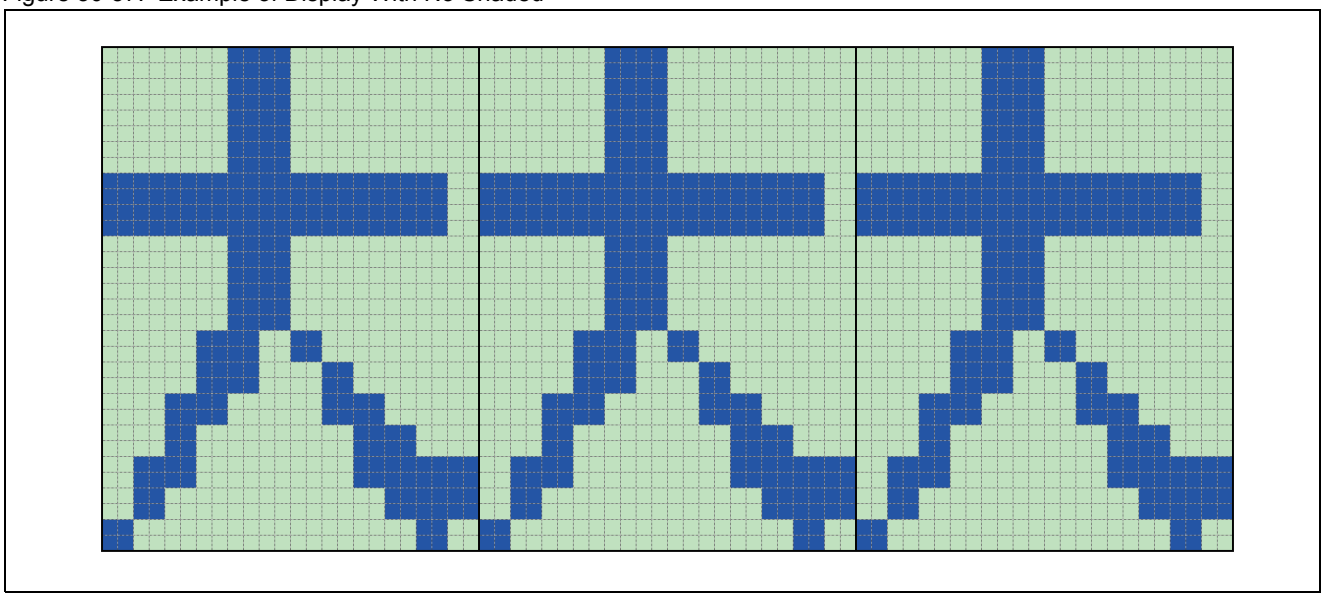
MR	Shaded background right character merge control
0	OFF
1	ON

Display examples

[No Shaded]

Figure 30-57 shows an example of the display with no shaded (solid background).

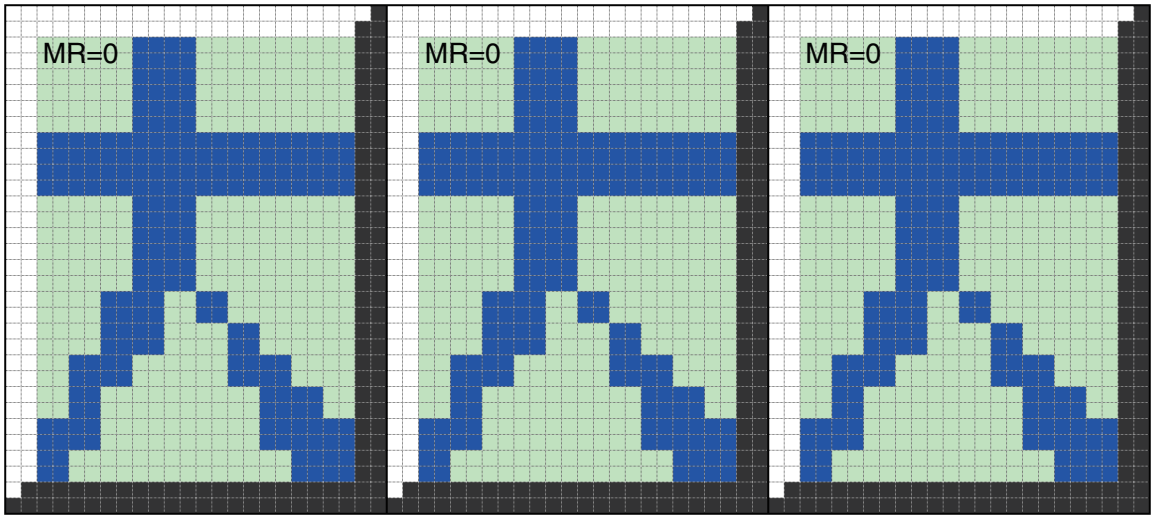
Figure 30-57. Example of Display With No Shaded



[Separate Shaded Backgrounds]

Figure 30-58 shows an example of the display when the shaded backgrounds are independent.

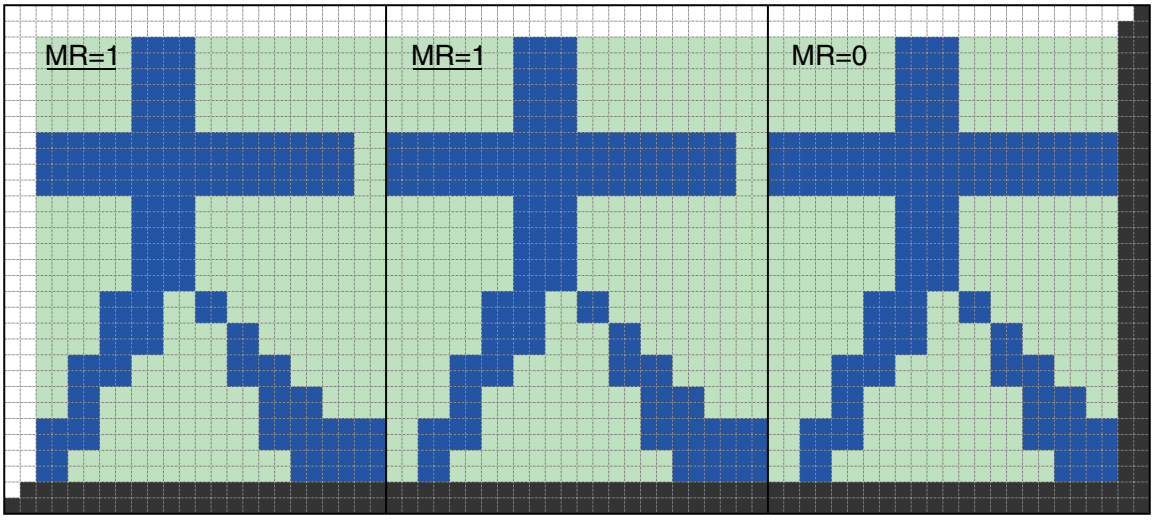
Figure 30-58. Example of Display With Separate Shaded Backgrounds



[Shaded Background Right Character Merge]

Figure 30-59 shows an example of the display with shaded background right character merge.

Figure 30-59. Example of Display With Shaded Background Right Character Merge



30.2.10.2 Character Shaded Background Frame Top/Bottom Erase Display (Character Shaded Background)

The character shaded background frame top/bottom erase control makes it possible, for example, to merge the display of multiple characters in the vertical direction by not displaying (erasing) the bottom edge of the shaded frame of a given line and the top edge of the shaded frame of the next (immediately below) line and disabling the line spacing setting between the lines.

Character shaded background frame top erase control (per character setting)

Table 30-33 shows the character shaded background frame top erase control of character data set 1 (command 1): bit MU.

Table 30-33. Character Shaded Background Frame Top Erase Control (Per Character Setting)

MU	Character shaded background frame top erase control
0	Character shaded background frame top ON
1	Character shaded background frame top OFF

Character shaded background frame bottom erase control (per character setting)

Table 30-34 shows the character shaded background frame bottom erase control of character data set 1 (command 1): bit MD.

Table 30-34. Character Shaded Background Frame Bottom Erase Control (Per Character Setting)

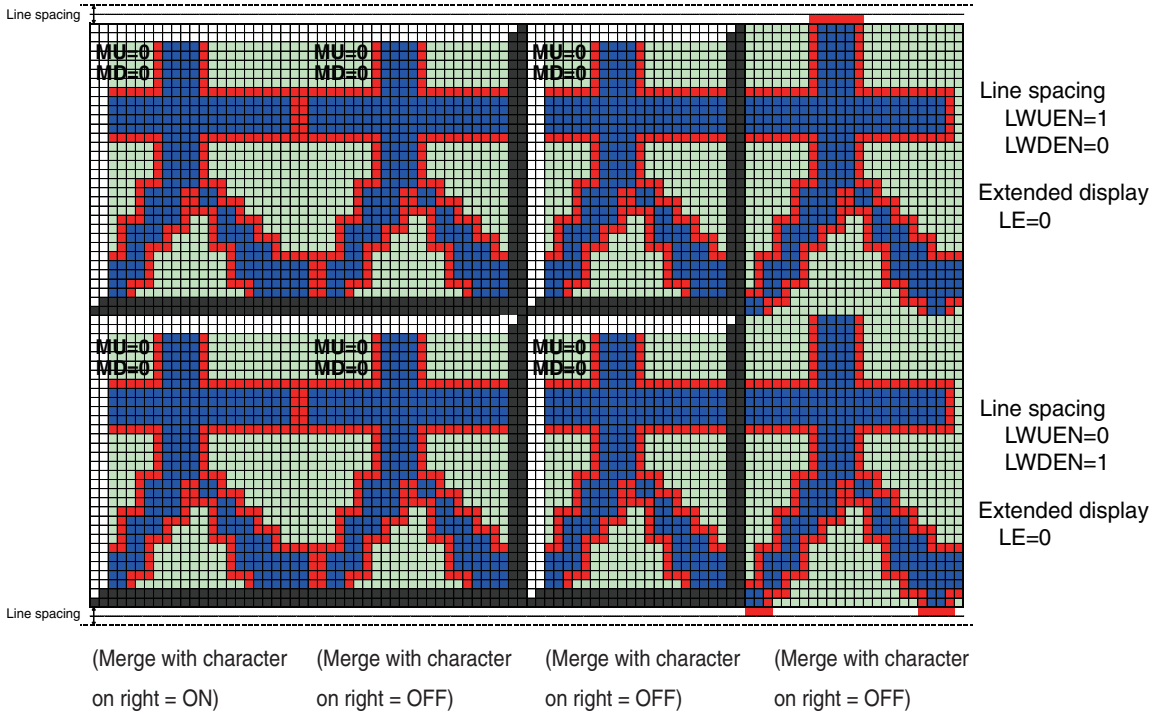
MD	Character shaded background frame bottom erase control
0	Character shaded background frame bottom ON
1	Character shaded background frame bottom OFF

Display examples (character background)

[Character Shaded Background Frame Top/Bottom Erase Control = OFF]

Figure 30-60 shows examples of the display when the character shaded background frame top/bottom erase control is OFF.

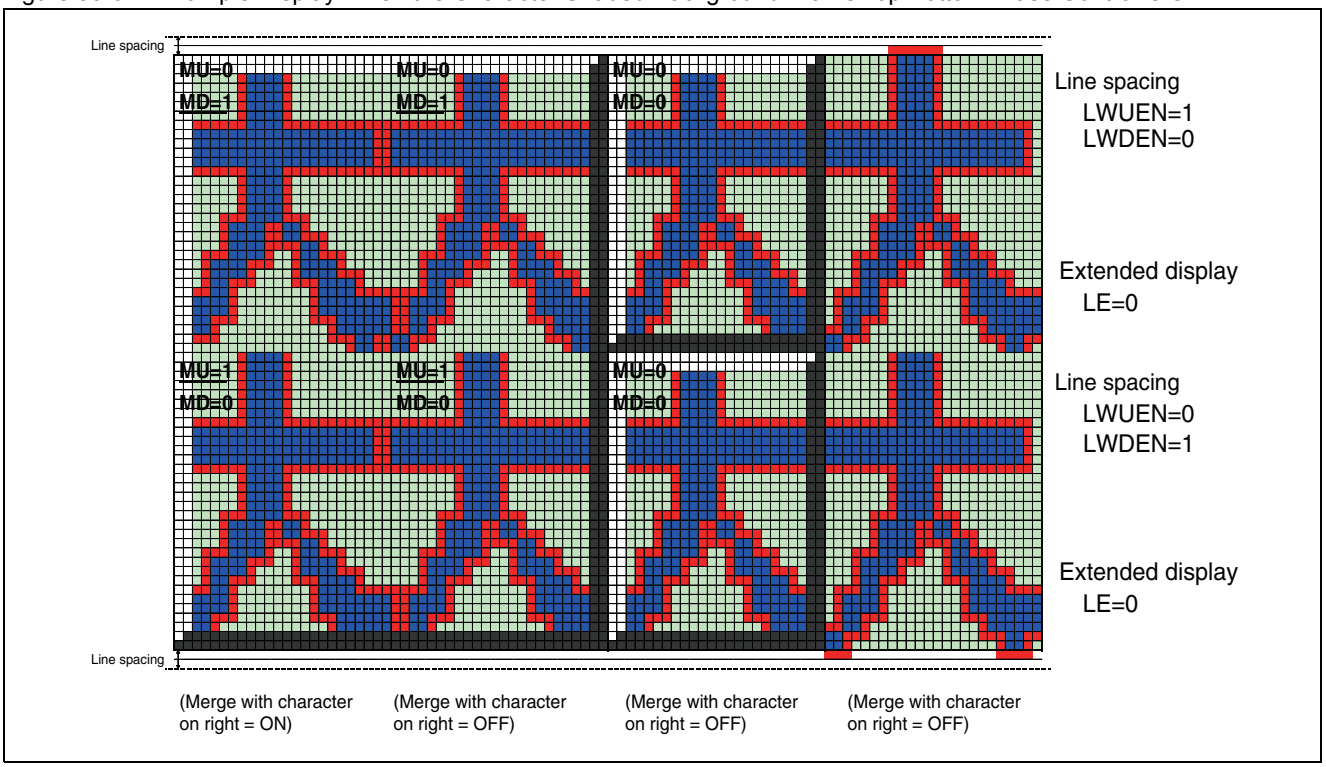
Figure 30-60. Example Display When the Character Shaded Background Frame Top/Bottom Erase Control is OFF



[Character Shaded Background Frame Top/Bottom Erase Control = ON]

Figure 30-61 shows examples of the display when the character shaded background frame top/bottom erase control is ON.

Figure 30-61. Example Display When the Character Shaded Background Frame Top/Bottom Erase Control is ON



Note: When the line spacing setting is disabled, the character trimming that extends into the character region of the lines above and below is not displayed. Furthermore, even if there is a line spacing region, the trimming that extends into the lines above and below is not displayed when there is a shaded frame.
Because the character size, line enlargement control, line spacing control, character background control, character trimming control, italic control, underline control, and each color code setting operate as attributes on each character or each line, make the same settings on the current line and the following line if there is a need to merge the lines. (Because the italic control and underline control are controlled independently for each line, these cannot be set when actually used.)

30.2.10.3 Character Background Extended Display

Character background extended display is a function that extends the display of the character background into the line spacing region.

Character background extended control (per line setting)

Table 30-35 shows the character background extended control of line control data set 2 (command 4): bit LE.

Table 30-35. Character Background Extended Control (Per Line Setting)

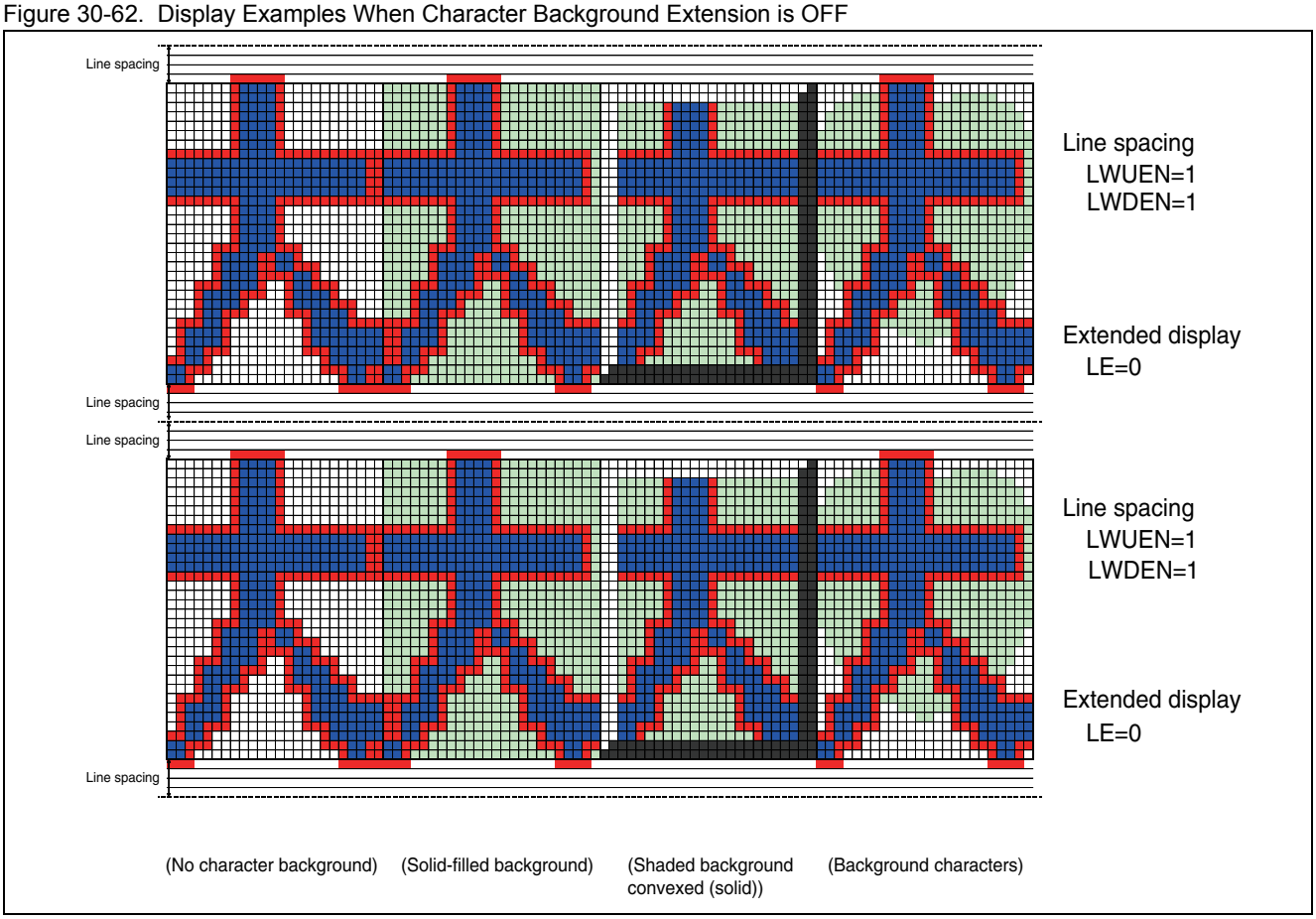
LE	Character Background Extended Control
0	OFF (Normal display)
1	ON (Extended display)

Display examples

[Character Background Extension = OFF]

Figure 30-62 shows examples of the display when character background extension is OFF.

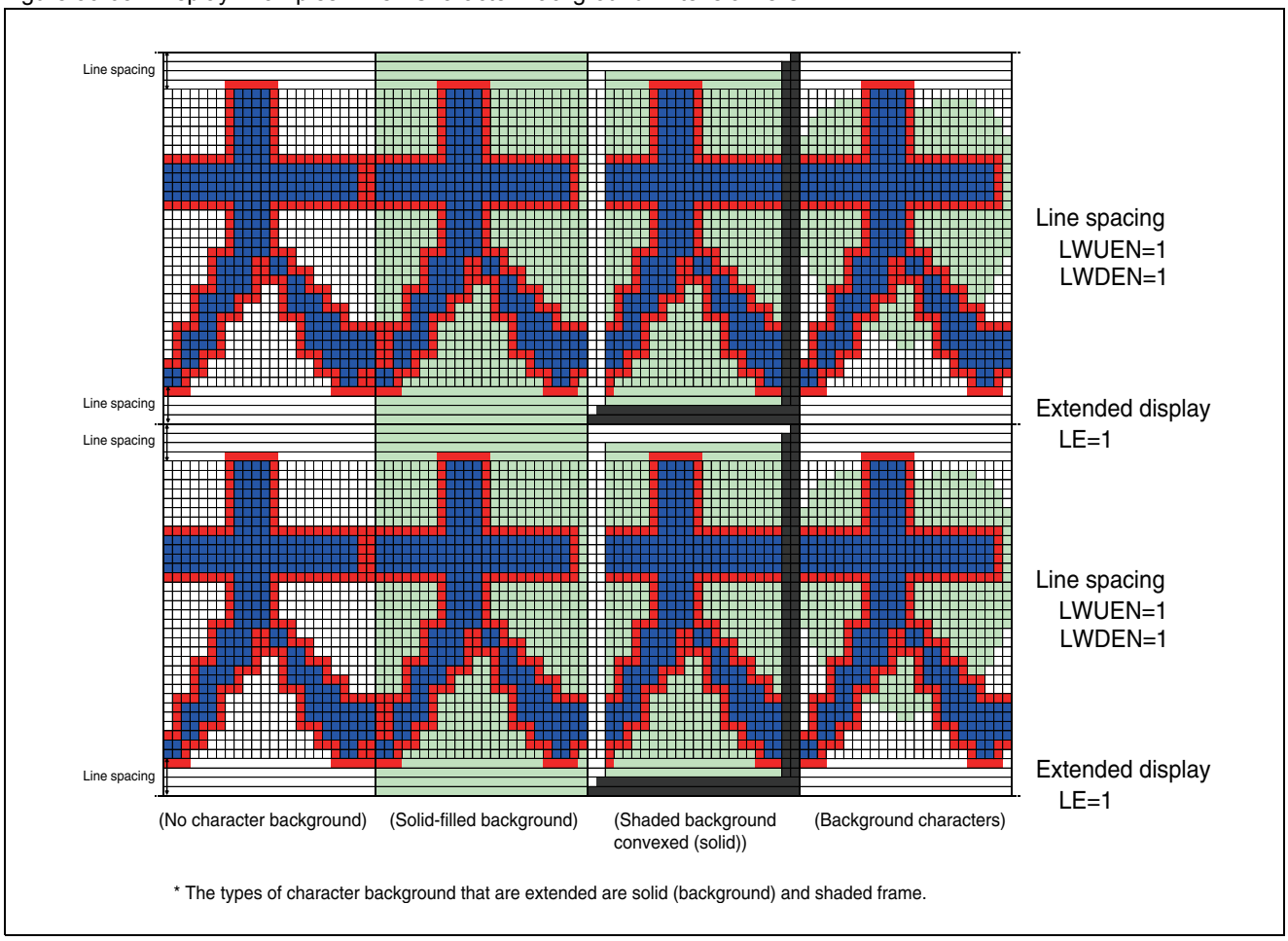
Figure 30-62. Display Examples When Character Background Extension is OFF



[Character Background Extension = ON]

Figure 30-63 shows examples of the display when character background extension is ON.

Figure 30-63. Display Examples When Character Background Extension is ON



30.2.11 Line Background Display

Line background is a function that displays the background in the character line region, the regions to the left and right, and the line spacing regions above and below.

Line background control (per line setting)

Table 30-36 shows the line background control of line control data set 2 (command 4): bits LM1 and LM0.

Table 30-36. Line Background Control (Per Line Setting)

LM1	LM0	Line background
0	0	No background (not displayed)
0	1	Solid background
1	0	Shaded background concaved (solid)[1]
1	1	Shaded background convexed (solid)[2]

[1]: Shaded background concaved display is a function that displays the shadow color in the uppermost 2 dots within the region of the line, and the highlight color in the lowermost 2 dots within the line.
[2]: Shaded background convexed display is a function that displays the highlight color in the uppermost 2 dots within the region of the line, and the shadow color in the lowermost 2 dots within the line.

Line shaded background frame top display control (per line setting)

Table 30-37 shows the line shaded background frame top display control of line control data set 2 (command 4): bit LMUEN.

Table 30-37. Line Shaded Background Frame Top Display Control (Per Line Setting)

LMUEN	Line shaded background frame top display control
0	Line shaded background frame top OFF
1	Line shaded background frame top ON

Note: If LMUEN is set to 1 when (LM1,LM0) are set to (1,0) or (1,1), the upper edge 2 dots of the shaded frame are displayed.

Line shaded background frame bottom display control (per line setting)

Table 30-38 shows the line shaded background frame bottom display control of line control data set 2 (command 4): bit LMDEN.

Table 30-38. Line Shaded Background Frame Bottom Display Control (Per Line Setting)

LMDEN	Line shaded background frame bottom display control
0	Line shaded background frame bottom OFF
1	Line shaded background frame bottom ON

Note: If LMDEN is set to 1 when (LM1,LM0) are set to (1,0) or (1,1), the lower edge 2 dots of the shaded frame are displayed.

Line background color (configurable per line from 256 colors)

The line background color can be configured per-line by setting bits LB7 to LB0 of the line control data set 2 (command 4) to the color code.

Shaded background frame highlight color (configurable per screen from 256 colors)

The line background highlight (shadow) color can be configured per-screen by setting bits B0H7 to B0H0 of the shaded background frame color control (command 15-0) to the color code.

Note: This is shared with the character shaded background frame color type selection control 0 (type) of the character background.

Shaded background frame shadow color (configurable per screen from 256 colors)

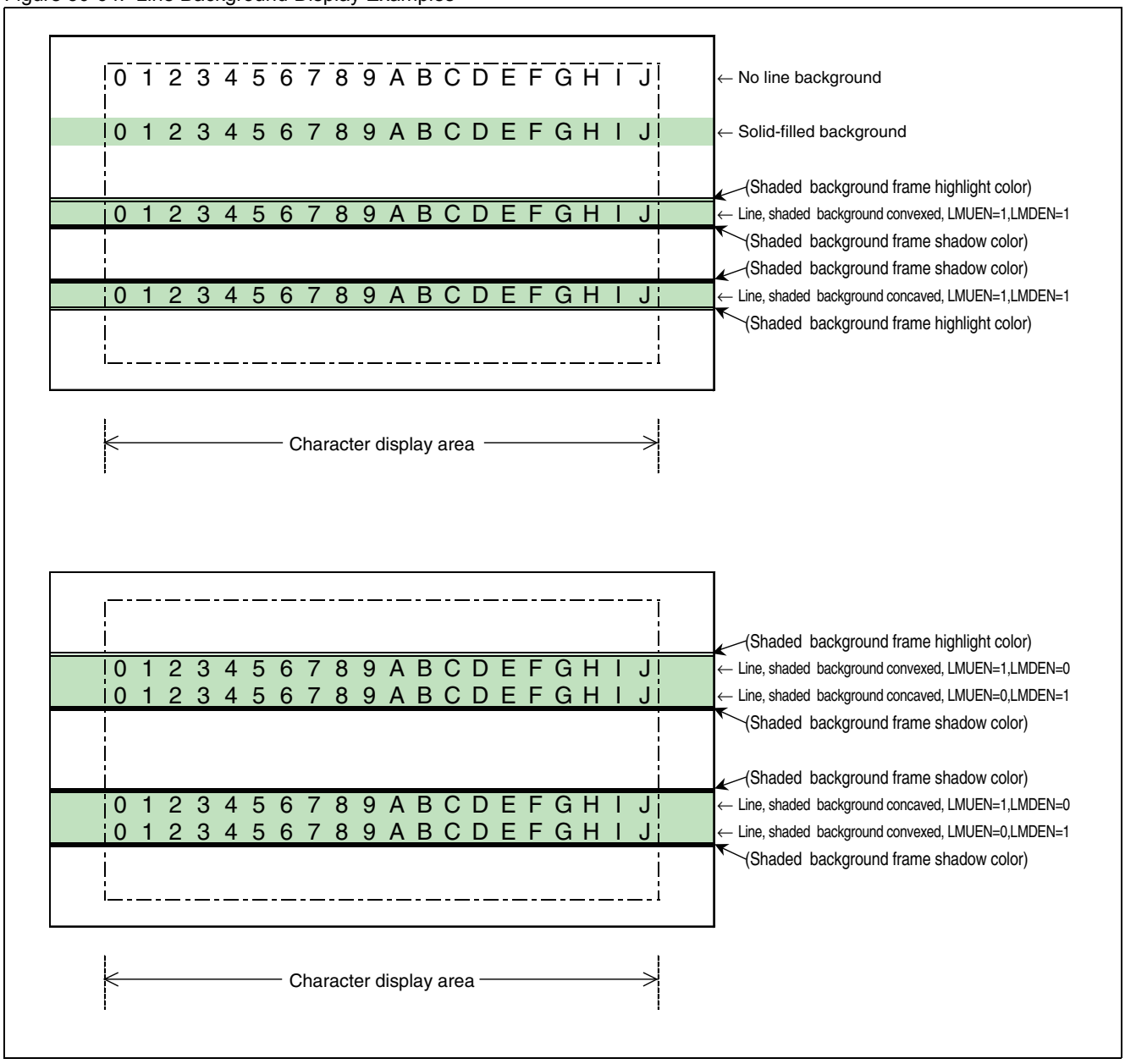
The line background highlight (shadow) color can be configured per-screen by setting bits B0S7 to B0S0 of the shaded background frame color control (command 15-0) to the color code.

Note: This is shared with the character shaded background frame color type selection control 0 (type) of the character background.

Display examples

Figure 30-64 shows examples of the line background display.

Figure 30-64. Line Background Display Examples



Note: If the line character vertical size is set to 2 dots when displaying shaded background concaved/convexed, the shaded frame cannot be displayed properly and shaded background concaved/convexed should not be configured.

30.2.11.1 Background Consecutive Line Merge Display Examples

This section shows examples of background consecutive line merge display.

Background consecutive line merge display examples

Figure 30-65 to Figure 30-69 show examples of displaying combinations of character shaded background display and line shaded background display.

Figure 30-65. Examples of Displaying Combinations of Character Shaded Background Display and Line Shaded Background Display (Part 1)

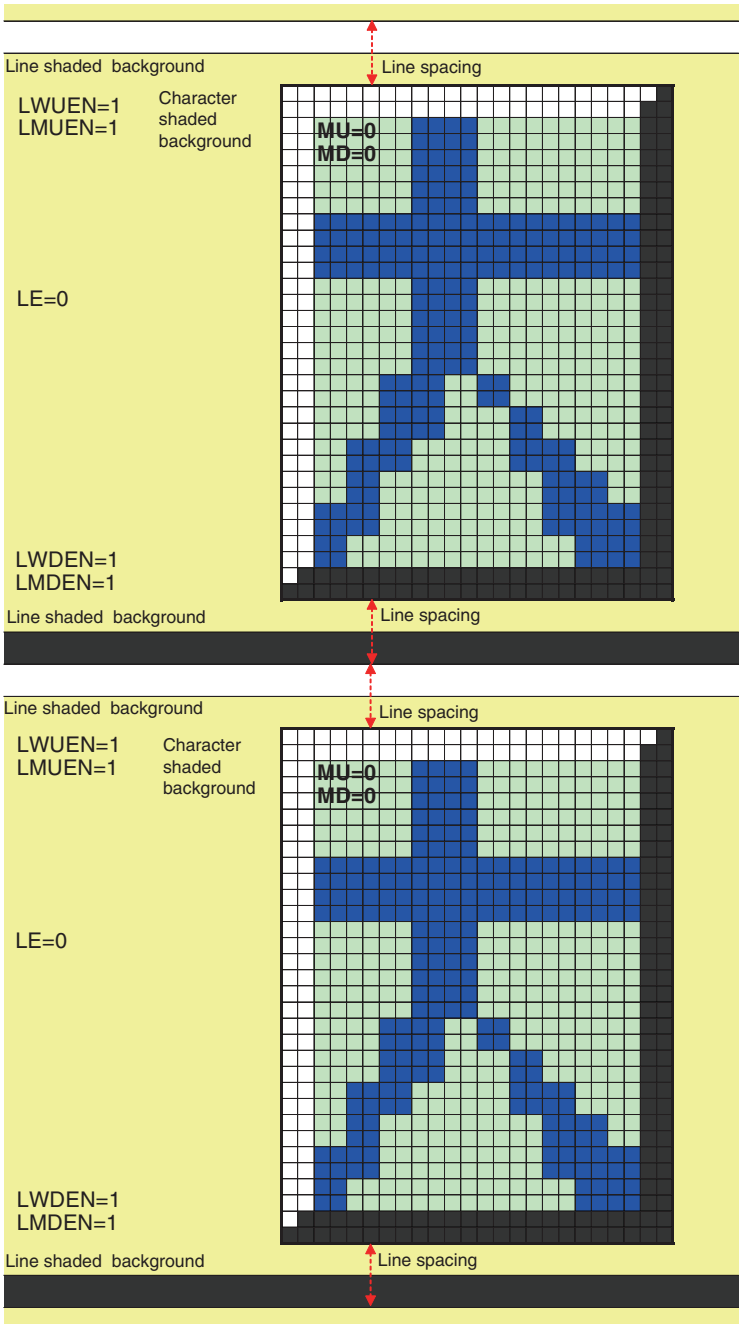


Figure 30-66. Examples of Displaying Combinations of Character Shaded Background Display and Line Shaded Background Display (Part 2)

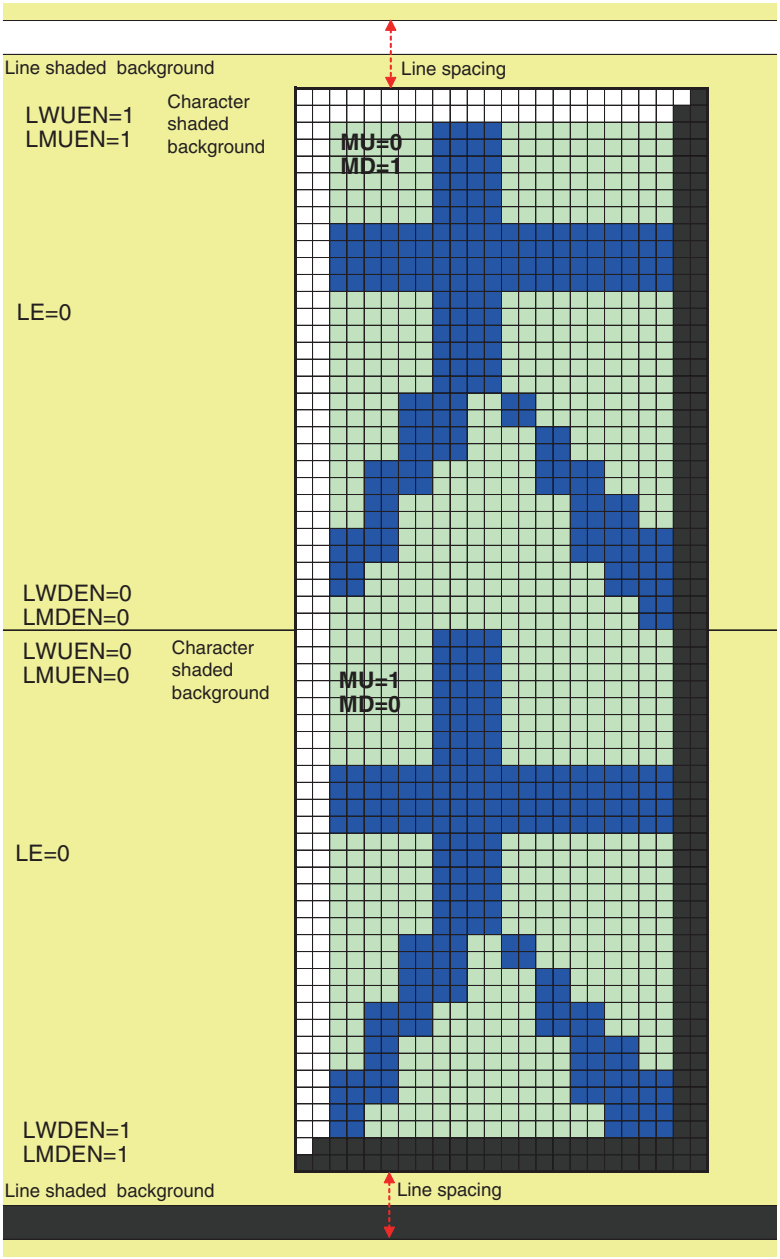


Figure 30-67. Examples of Displaying Combinations of Character Shaded Background Display and Line Shaded Background Display (Part 3)

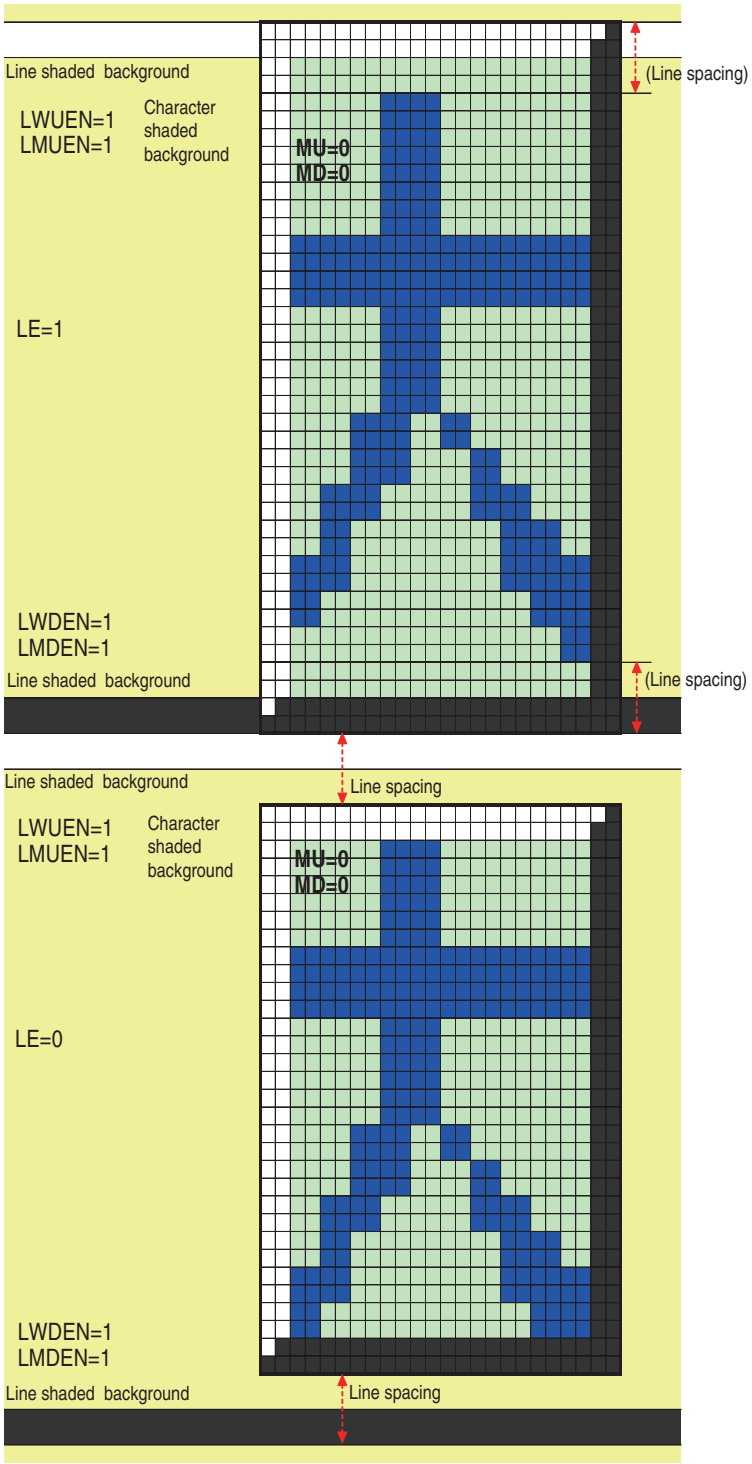


Figure 30-68. Examples of Displaying Combinations of Character Shaded Background Display and Line Shaded Background Display (Part 4)

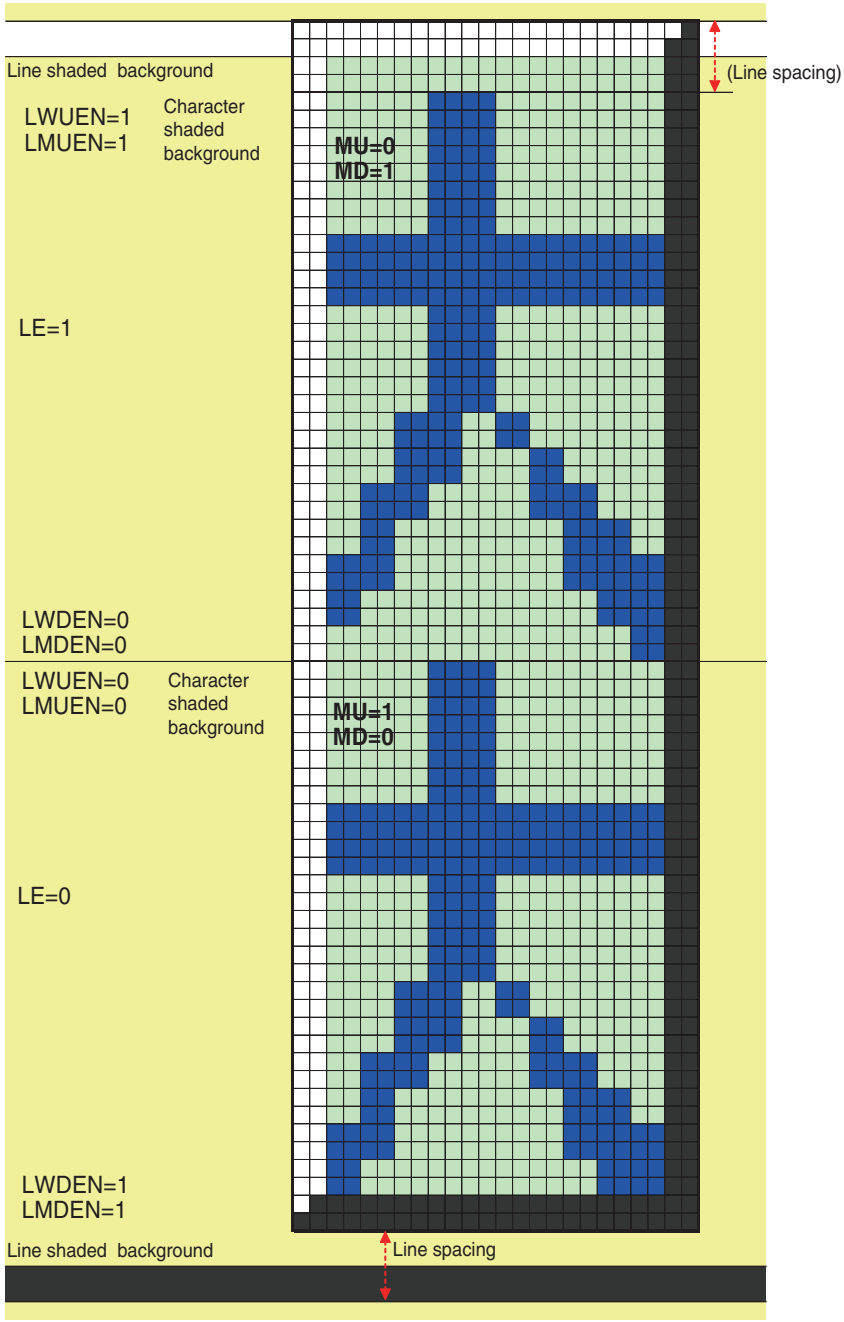
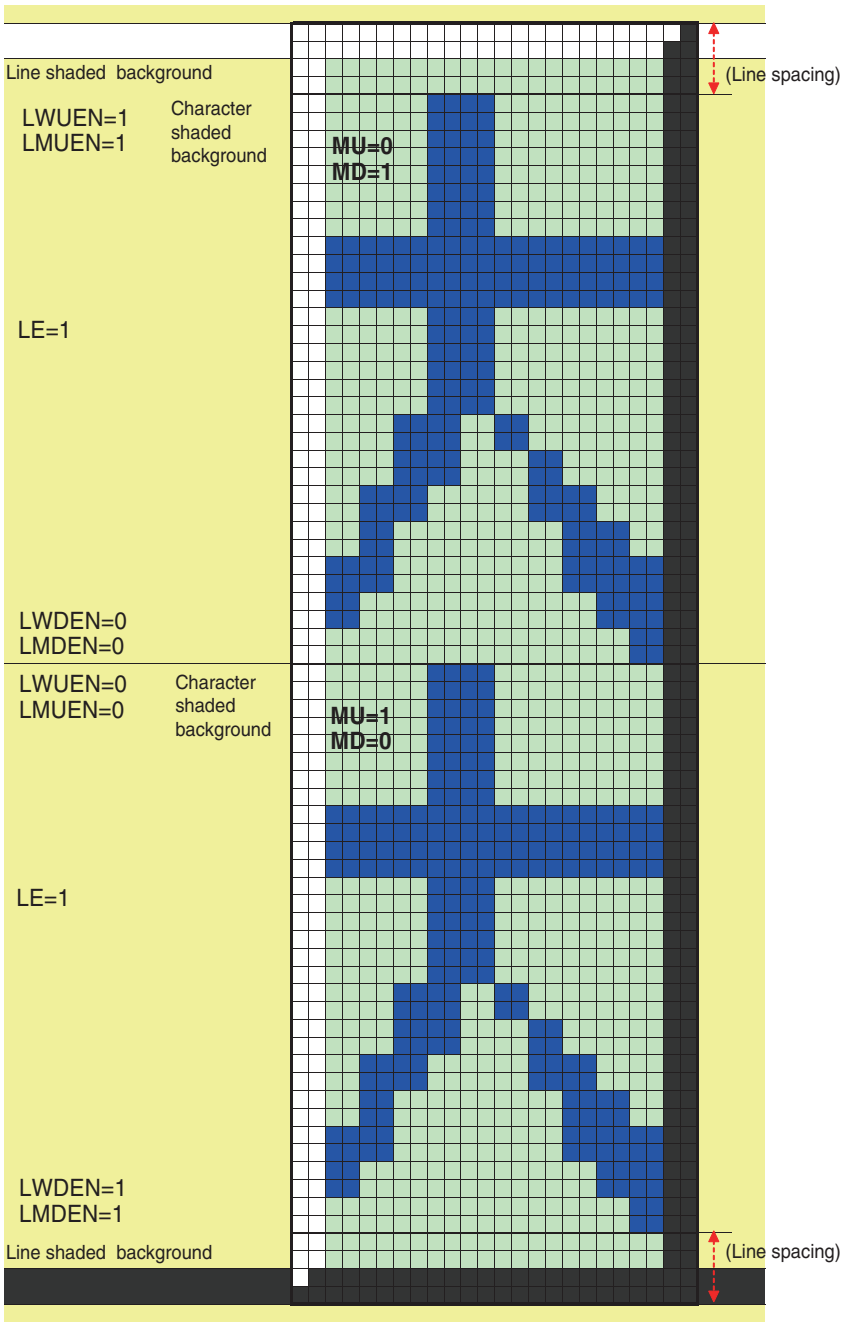


Figure 30-69. Examples of Displaying Combinations of Character Shaded Background Display and Line Shaded Background Display (Part 5)



Note: Although the shaded frame of the line shaded background is added over the character region, if it overlaps with the shaded frame of the character shaded background, the shaded frame of the character shaded background is displayed with priority.

30.2.12 Screen Background Display

The screen background display has the two functions of displaying the screen background character and displaying the screen background color.

Screen background display

- Screen background character display (main screen only)

  A graphics character can be displayed throughout the entire screen by repeatedly displaying a character block with the same structure made up of a maximum of 4 characters arranged 2 characters horizontally × 2 characters vertically.
- Screen background color display (main screen and sub screen)

  The background color is displayed throughout the entire screen as the lowest layer output.

30.2.12.1 Screen Background Character Display

Screen background character display is a function to repeatedly display a block of a maximum of 2 characters horizontally × 2 characters vertically throughout the entire screen. Screen background character display is only able to display graphics characters. Furthermore, it can only be applied to the main screen.

Screen background character display configuration

Figure 30-70 shows an example of the display of screen background characters.

Figure 30-70. Screen Background Character Display Example



Screen Background Character Display Control

- Screen background character output control

  Figure 30-39 shows the screen background character output control of the screen output control (command 5-0): bit PDS.

Table 30-39. Screen Background Character Output Control

PDS	Screen background character output
0	OFF
1	ON

- Screen background character code

  Screen background character control 1 (command 7): bits PM13 to PM0

  Note: Set PM2, PM1, and PM0 to 0.

Note: The screen background character can only use L size display of graphics characters.

- Screen background character vertical size control

  Table 30-40 shows the screen background character vertical size control of the screen background character control (command 7): bits PH2 to PH0.

Table 30-40. Screen Background Character Vertical Size Control

PH2	PH1	PH0	Screen background character vertical size
0	0	0	18 dots
0	0	1	20 dots
0	1	0	22 dots
0	1	1	24 dots
1	0	0	26 dots
1	0	1	28 dots
1	1	0	30 dots
1	1	1	32 dots

- Screen background character configuration control

  Table 30-41 shows the screen background character configuration control of the screen background character control (command 7): bits PD1 and PD0.

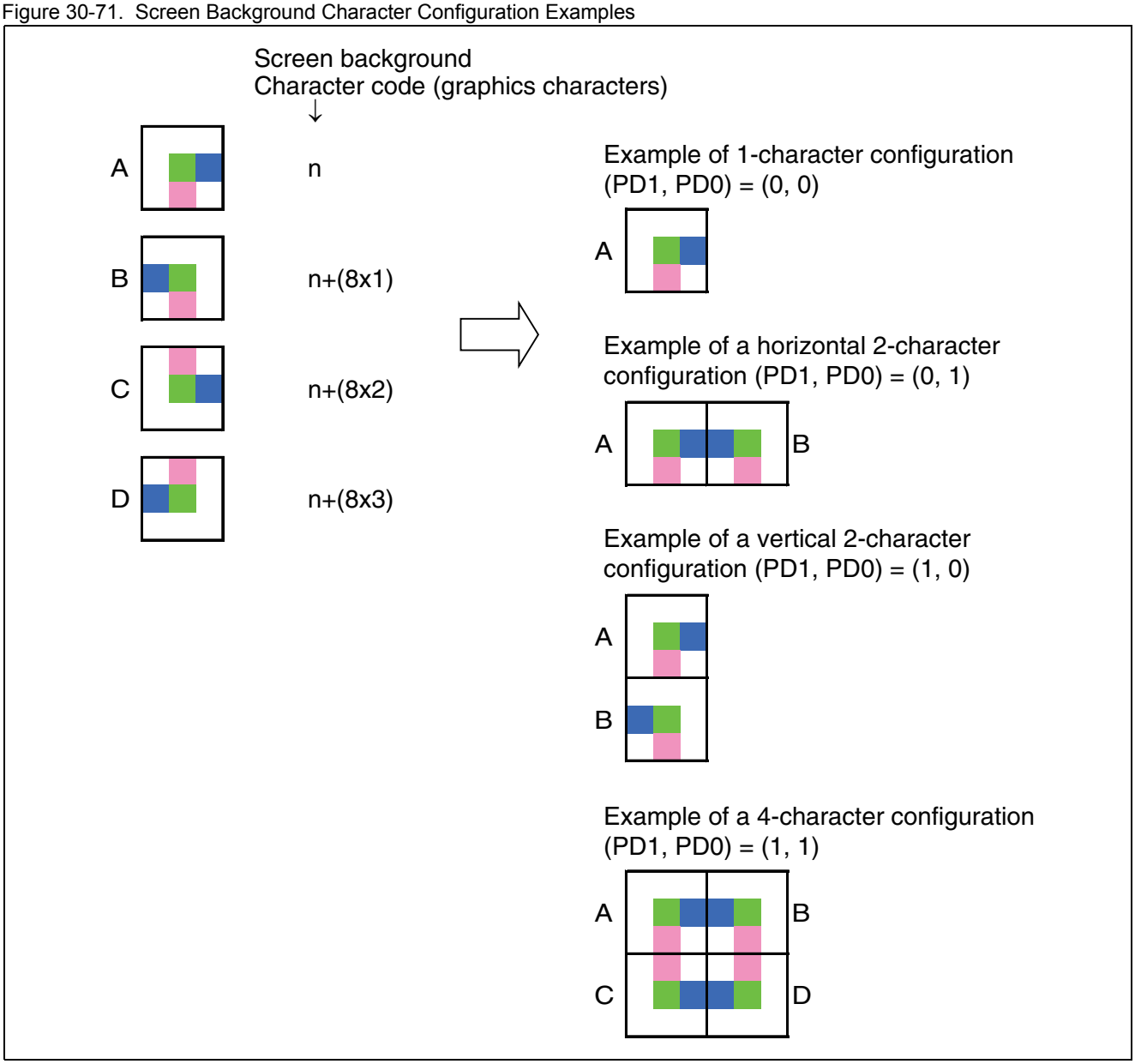
Table 30-41. Screen Background Character Configuration Control

PD1	PD0	Screen background character configuration
0	0	Composed of 1 character
0	1	Composed of 2 characters horizontally
1	0	Composed of 2 characters vertically
1	1	Composed of 2 × 2 characters (vertical × horizontal)

Note: When PD1 is set to 1 and PD0 is set to 1 (composed of 2 × 2 characters (vertical × horizontal)), do not use the character background character function on either the main screen or the sub screen. (It will not display correctly.)

Figure 30-71 shows examples of the screen background character configurations.

Figure 30-71. Screen Background Character Configuration Examples



Note: The character codes of the graphics characters used in the screen background need to be arranged sequentially when using the configurations of 2 horizontal characters, 2 vertical characters, or 4 characters. (Graphics characters are equivalent to 8 normal characters, and the character codes for graphics characters must be a multiple of 8 characters from 0000_H.)

30.2.12.2 Screen Background Color Display

Screen background color is a function that displays a background color throughout the entire screen as the bottommost layer output of the display screen.

Screen background output control

Table 30-42 shows the screen background output control of the screen output control (command 5-0): bit UDS.

Table 30-42. Screen Background Output Control

UDS	Screen background color output
0	OFF
1	ON

Screen background color control

The screen background color can be configured by setting bits U7 to U0 of screen background character control 4 (command 8-0) to the color code.

30.2.12.3 Window Background Display

Window background display controls the screen display region.

Window background display

Table 30-43 shows the commands related to window function control in the screen output control (command 5-0).

Table 30-43. Window Function Control

WE	Window function control
0	OFF
1	ON

The start and end timing of the window vertical display periods are configured as follows.

- Vertical display start timing

  Window period control 1 (command 8-1): bits WYS10 to 0

  Configurable from 0 to 2047 dots in units of 1 dot.

- Vertical display end timing

  Window period control 1 (command 8-1): bits WYE10 to 0

  Configurable from 0 to 2047 dots in units of 1 dot.

  For the start and end of window vertical display periods, the setting values of the vertical position are calculated as follows.

Window vertical display start/end positions = Vertical display start offset* +
Window period control 1 (vertical display start/end timing) [dot]

Note: The origin for the vertical start and end timing settings is the setting value of the vertical display start offset (main: controlled by command 11 bits VOF5 to 0, see "30.2.4.1 Screen Display Position Offset") after the period of 1 Hsync from the vertical sync signal (VSYNC pin input signal) synchronization pulse.

The start and end timing of the window horizontal display periods are configured as follows.

- Horizontal display start timing

  Window period control 2 (command 8-2): bits WXS11 to 0

  Configurable from 0 to 4095 dots in units of 1 dot.

- Horizontal display end timing

  Window period control 2 (command 8-2): bits WXE11 to 0

  Configurable from 0 to 4095 dots in units of 1 dot.

  For the start and end of window horizontal display periods, the setting values of the horizontal position are calculated as follows respectively for the MAIN screen and SUB screen.

  [MAIN screen]

  Window horizontal display start position = WXS + 17 [dots]

  Window horizontal display end position = WXE + 17 [dots]

  [SUB screen]

  Window horizontal display start position = WXS + 13 [dots]

  Window horizontal display end position = WXE + 13 [dots]

  Note: The origin for the horizontal start and end timing settings is from the significant edge (controlled by command 13 bit HE) of the synchronization pulse of the horizontal sync signal (HSYNC pin input signal).

Notes:

- There is relatively a 4-dot difference between the main screen and sub screen in the setting values of the start and end of the window horizontal display periods.
- If the following settings are made, the display end timing setting is invalid. Therefore, do not make these settings.
  WYS [10:0] (Vertical display start timing) + [Vertical window border width][1] ≥ WYE[10:0] (Vertical display end timing)
  WXS [11:0] (Horizontal display start timing) + [Horizontal window border width][1] ≥ WXE[11:0] (Horizontal display end timing)
  [1]: The value of the [horizontal/vertical window border width] is for the case where the window border background is configured. Otherwise the value is "0".

Table 30-44. shows the commands related to window background mode and window border width control in the screen background control (command 8-0).

Table 30-44. Window Background Mode Control and Window Border Width Control

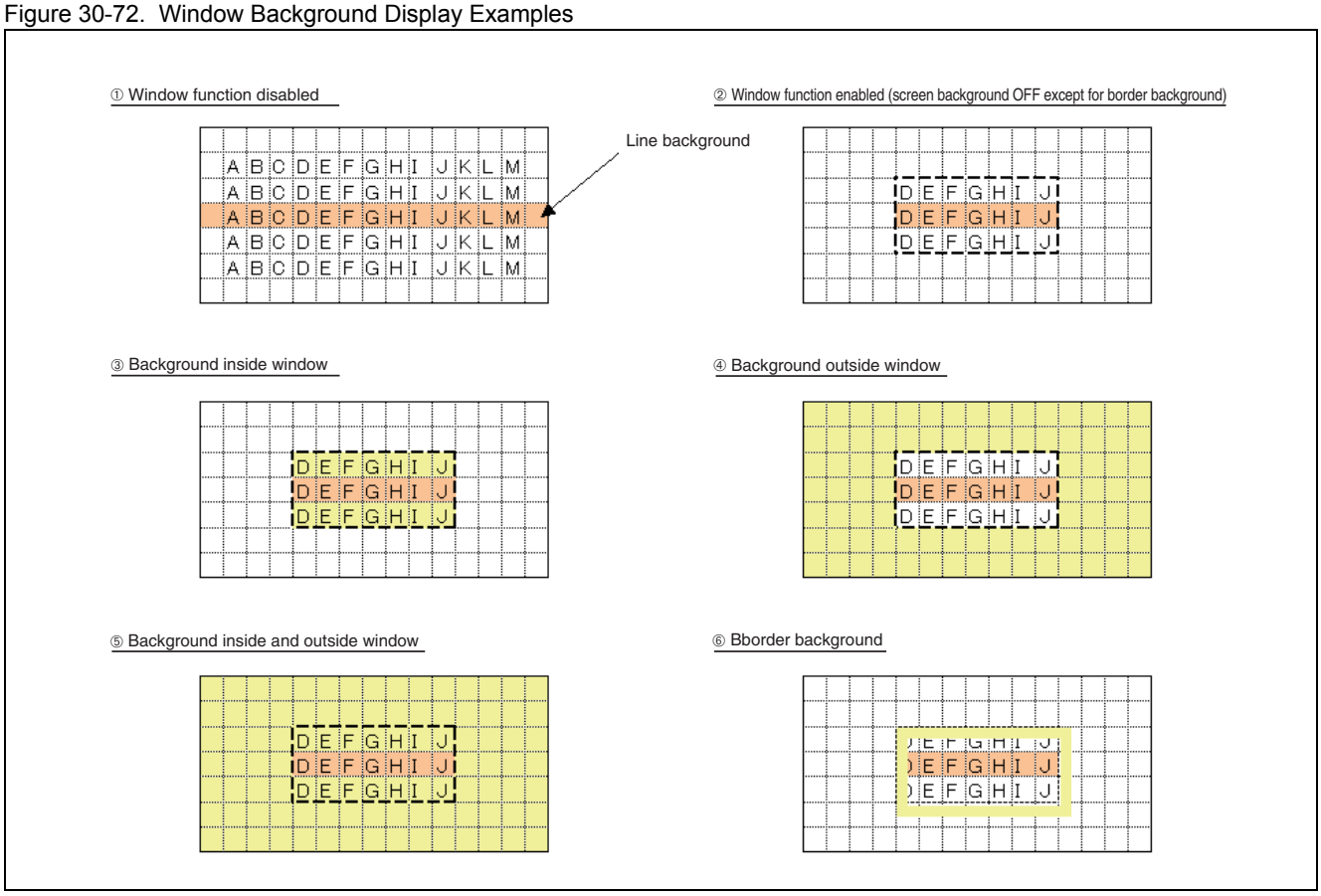
UW1	UW0	Window background mode control
0	0	Background inside window
0	1	Background outside window
1	0	Background inside and outside window
1	1	Window border background

UWBV	UWBH	Window border width control[1]
0	0	Vertical 2 dots, horizontal 2 dots
0	1	Vertical 2 dots, horizontal 4 dots
1	0	Vertical 4 dots, horizontal 2 dots
1	1	Vertical 4 dots, horizontal 4 dots

[1]: Window border width control is valid when the window border background setting (UW1=1, UW0=1) is configured in the window background mode control. Furthermore, the window border display is added in the positive direction to each of the setting values of the window period control (command 8-1, 2).

Figure 30-72 shows examples of the window background displays.

Figure 30-72. Window Background Display Examples



Note: The order of precedence of the screen configuration of the main screen and sub screen does not change during window display.
The border region of the border background corresponds to the screen background (or screen background character: main only)
Sprite characters do not have an effect on the window background display.
See "30.2 Display Functions" and "30.2.1 Screen Configuration" for the configuration within the screen order of precedence.

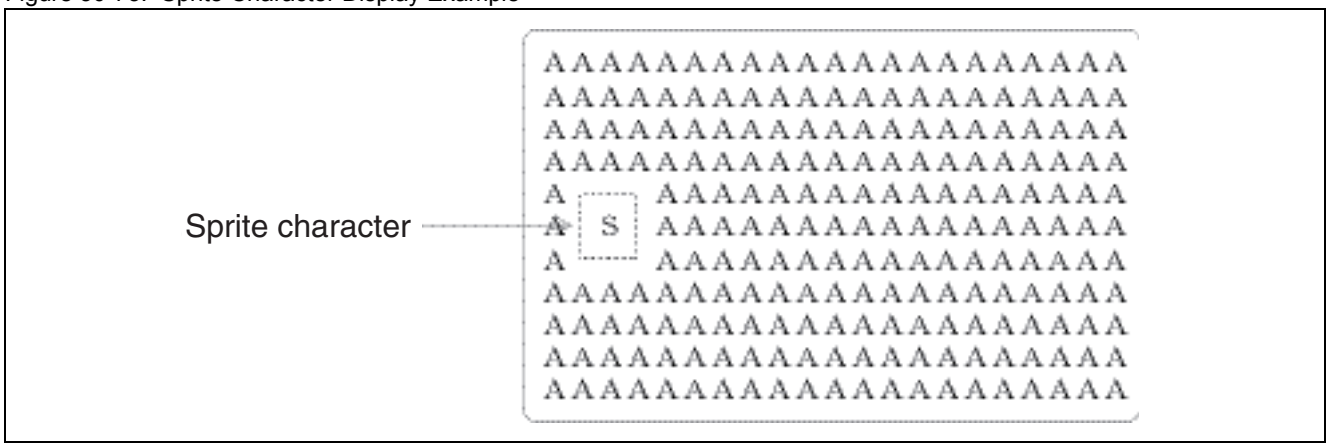
30.2.13 Sprite Character Display

Sprite characters are a function that display a character in the topmost layer of the display screen. Sprite character display can only display graphics characters.

Sprite Character Configuration

Figure 30-73 shows an example of displaying a sprite character.

Figure 30-73. Sprite Character Display Example



Sprite Character Display Control

- Sprite Character Output Control

  Table 30-45 shows the sprite character output control of the screen output control (command 5-0): bit SDS.

Table 30-45. Sprite character output control

SDS	Sprite character output
0	OFF
1	ON

- Sprite character code

  Sprite character control 1 (command 9-0): bits SM13 to SM0.

  Note: Set SM2, SM1, and SM0 to 0.

Note: Sprite character display can only display combinations of L size graphics characters.

- Sprite character vertical display position control

  Sprite character control 2 (command 9-1): bits SY10 to SY0

  Configurable from 0 to 2047 dots in units of 1 dot.

- Sprite character horizontal display position control

  Sprite character control 2 (command 9-1): bits SX11 to SX0

  Configurable from 0 to 4095 dots in units of 1 dot.

- Sprite character vertical size control

  Table 30-46 shows the sprite character vertical size control of sprite character control 1 (command 9-0): bits SH2 to SH0.

Table 30-46. Sprite Character Vertical Size Control

SH2	SH1	SH0	Sprite character vertical size
0	0	0	18 dots
0	0	1	20 dots
0	1	0	22 dots
0	1	1	24 dots
1	0	0	26 dots
1	0	1	28 dots
1	1	0	30 dots
1	1	1	32 dots

- Sprite character configuration control

  Table 30-47 shows the sprite character configuration control of sprite character control 1 (command 9-0): bits SD1 and SD0.

Table 30-47. Sprite Character Configuration Control

SD1	SD0	Sprite character configuration
0	0	Composed of 1 character
0	1	Composed of 2 characters horizontally
1	0	Composed of 2 characters vertically
1	1	Composed of 2 × 2 characters (vertical × horizontal)

- Sprite character blinking control

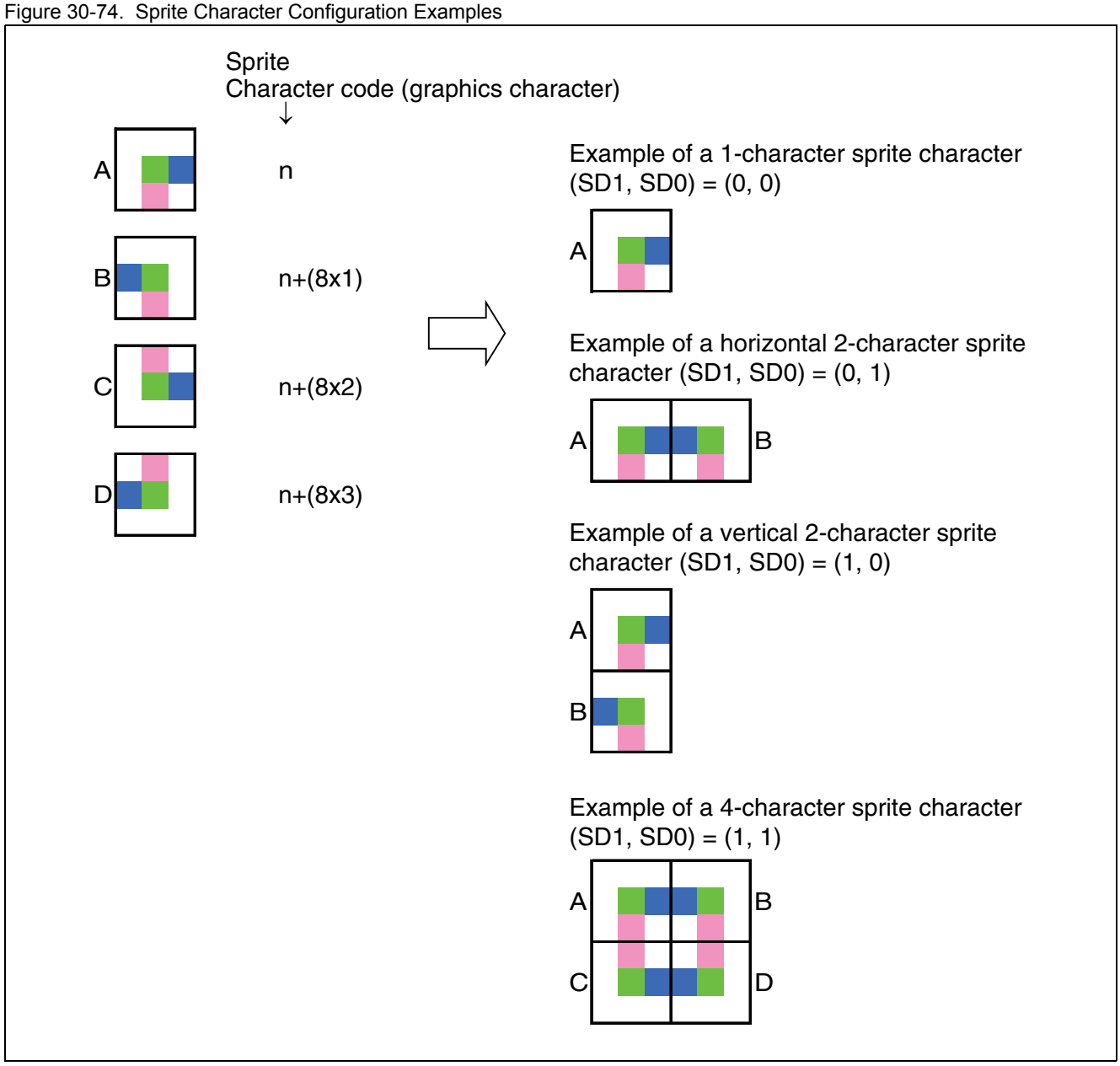
  Table 30-48 shows the sprite character blinking control of sprite character control 1 (command 9-0): bit SBL.

Table 30-48. Sprite Character Blinking Control

SBL	Sprite character blinking
0	OFF
1	ON

Figure 30-74 shows examples of sprite character configurations.

Figure 30-74. Sprite Character Configuration Examples



Note: The character codes of the graphics characters used in sprites need to be arranged sequentially when using the configurations of 2 horizontal characters, 2 vertical characters, or 4 characters. (Graphics characters are equivalent to 8 normal characters, and the character codes for graphics characters must be a multiple of 8 characters from 0000_H.)

30.3 Control Functions

This section describes the OSDC control functions.

30.3.1 Dot Clock Control

30.3.2 Sync Signal Input

30.3.3 Display Signal Output

30.3.4 Display Period Control

30.3.5 Synchronization Control

30.3.6 Interrupt Control

30.3.7 OSDC Operation Control

30.3.1 Dot Clock Control

Dot clock control allows the dot clock to be selected from an external dot clock input or an internal VCO generated dot clock input. The dot clock can be controlled by commands 17 and 18.

Input dot clock selection control

Table 30-49 shows the dot clock selection control of dot clock control 1 (command 17): bit DCK. If the internal VCO generated dot clock is used, select the input dot clock using this command after setting the VCO-related controls in command 18.

Table 30-49. Dot Clock Selection Control

DCK	Dot clock control
0	External dot clock input (default setting)
1	Internal VCO generated dot clock input

External dot clock input

The dot clock can be input from an external oscillator.

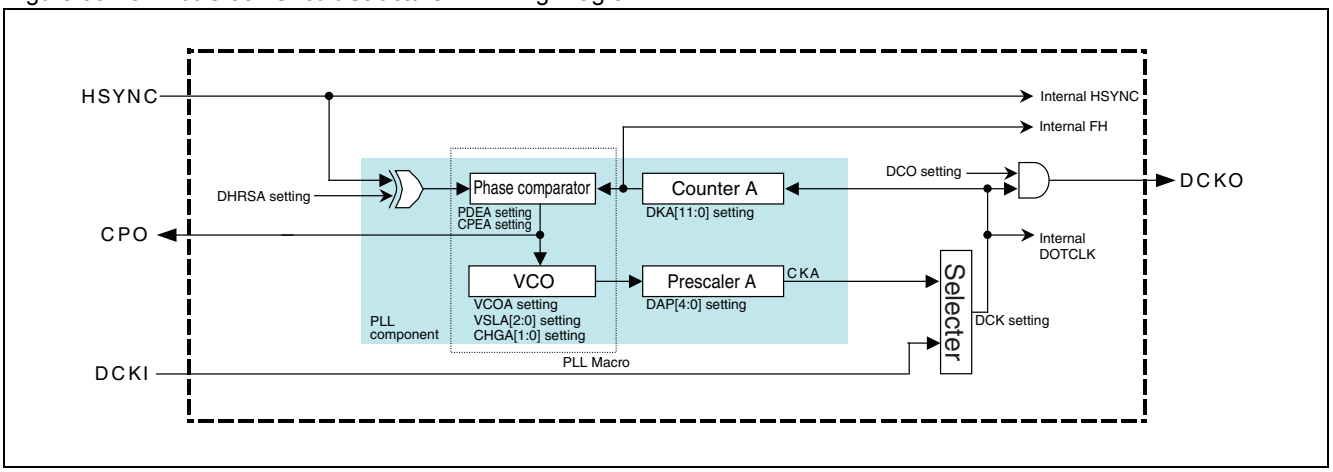
This clock is assumed to be synchronized with the input horizontal sync signals.

Internal VCO generated dot clock input

- Dot clock circuit configuration

  The VCO-generated clock for the OSDC can be used as the dot clock by controlling the internal prescalers, etc. Figure 30-75 shows a diagram outlining the dot clock circuit structure.

Figure 30-75. Dot Clock Circuit Structure Trimming Diagram



- Dot clock prescaler control

  Table 30-50 shows the dot clock prescaler control of PLL clock control (command 18): bits DAP4 to DAP0.

Table 30-50. Dot Clock Prescaler Control

DAP4 to DAP0	Dot clock prescaler configuration
00000	VCO-generated clock
00001	VCO-generated clock × 1/2
00010	VCO-generated clock × 1/4
00011	VCO-generated clock × 1/6
00100	VCO-generated clock × 1/8
• • •	• • •
11110	VCO-generated clock × 1/60
11111	VCO-generated clock × 1/62

- Phase comparator edge selection control

  Table 30-51 shows the phase comparator edge selection control of clock selection control (command 18): bit DHRSA.

Table 30-51. Phase Comparator Edge Selection Control

DHRSA	Phase comparator edge selection control
0	HSYNC rising edge
1	HSYNC falling edge

- Clock frequency division control

  The internal horizontal sync signal is generated by setting the PLL clock control (command 18): bits DKA11 to DKA0 to the dot clock frequency division required for the horizontal sync width. (This is set to the same frequency as the input external horizontal sync signal HSYNC.)

  Table 30-52 shows the clock frequency division control of PLL clock control (command 18): bits DKA11 to DKA0.

Table 30-52. Clock Frequency Division Control

DKA11 to DKA0	Clock frequency division
000_H	129 clocks
• • •	• • •
080_H	129 clocks
081_H	130 clocks
• • •	• • •
FFE_H	4095 clocks
FFF_H	4096 clocks

- VCO oscillator control

  Table 30-53 shows the VCO oscillator control of PLL clock control (command 18): bit VCOA.

Table 30-53. VCO Oscillator Control

VCOA	VCO oscillator control
0	VCO oscillator stopped
1	VCO oscillator running

- Oscillator VCO selection control

  Table 30-54 shows the VCO selection control of PLL clock control (command 18): bits VSLA2 to VSLA0.

Table 30-54. Oscillator VCO Selection Control

VSLA2	VSLA1	VSLA0	VSO selection configuration	VSO oscillator guaranteed frequency range
0	0	0	VCO0	25 to 50 MHz
0	0	1		
0	1	0	VCO1	40 to 70 MHz
0	1	1		
1	0	0	-	Setting prohibited
1	0	1	-	Setting prohibited
1	1	0	-	Setting prohibited
1	1	1	-	Setting prohibited

Note: The guaranteed range of the VCO oscillator is different from the guaranteed operating frequency of the OSDC.

- PLL charge pump control

  Table 30-55 shows the charge pump control of PLL clock control (command 18): bit CPEA.

Table 30-55. Charge Pump Control

CPEA	Charge pump control
0	Stopped
1	Normal operation

Output dot clock control

The dot clock output from the DCK0 pin can be controlled.

Table 30-56 shows the output dot clock selection control of PLL clock control (command 18): bit DCO.

Table 30-56. Output Dot Clock Control

DCO	Output dot clock control
0	Dot clock output OFF
1	Dot clock output ON

30.3.2 Sync Signal Input

This section explains the vertical sync detection and horizontal sync operation during sync signal input.

Sync signal input

- Vertical sync control

  The level of the vertical sync signal is sampled on the leading edge and trailing edge of the horizontal sync pulse, and any changes are detected.
- Horizontal sync control

  The horizontal sync operation can be selected from operating on the leading edge or operating on the trailing edge.
- Field control

  The field state during interlace display is detected by monitoring the falling edge of the vertical sync signal.

30.3.2.1 Vertical Sync Control

Vertical sync detection is a function that samples the level of the vertical sync signal on the leading edge and trailing edge of the horizontal sync pulse and detects any changes. The vertical display position of the screen depends on the position of this detection.

Vertical sync detection

Operation of the vertical sync requires the vertical sync detection HSYNC edge selection control and vertical sync signal input logic control to be configured in the I/O pin control (command 13).

Configure the vertical sync detection HSYNC edge selection control by considering the phases of the input horizontal sync signal and vertical sync signal. Furthermore, the vertical sync signal input logic control is configured to match the input vertical sync signal logic.

Table 30-57 and Table 30-58 show each of these controls.

Table 30-57. Vertical Sync Detection HSYNC Edge Selection

VHE	Vertical sync detection HSYNC edge
0	Vertical sync detected at HSYNC leading edge
1	Vertical sync detected at HSYNC trailing edge

Table 30-58. Vertical Sync Signal Input Logic Control

IVX	Vertical sync signal input logic
0	The VSYNC pin is negative logic input
1	The VSYNC pin is positive logic input

Examples of vertical sync detection operation

Figure 30-76 and Figure 30-77 show examples of vertical sync detection operation.

Figure 30-76. Detection of Vertical Sync Signal on Horizontal Sync Leading Edge (VHE=0, IVX=0, IHX=0)

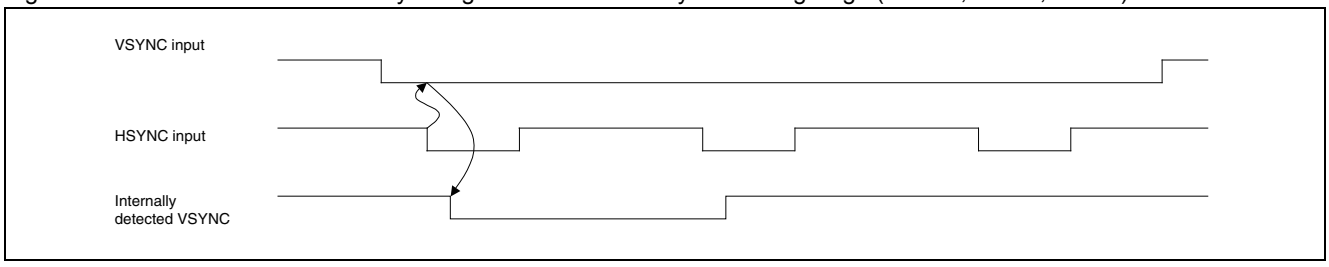
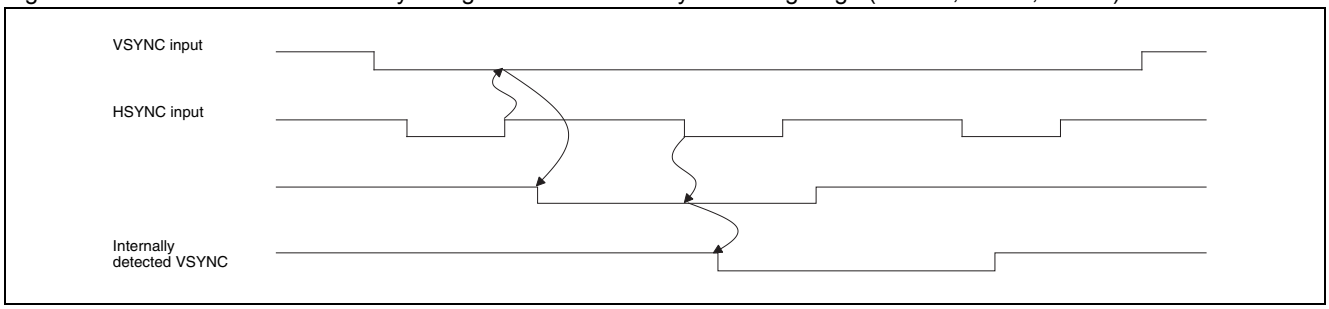


Figure 30-77. Detection of Vertical Sync Signal on Horizontal Sync Trailing Edge (VHE=1, IVX=0, IHX=0)



30.3.2.2 Horizontal Sync Control

The operation of the horizontal sync can be selected from operating on the leading edge of the horizontal sync or the trailing edge of the horizontal sync.

Horizontal sync operation

The operation of the horizontal sync requires the horizontal sync operation edge selection and horizontal sync signal input logic control to be configured in the I/O pin control (command 13).

Table 30-59 and Table 30-60 show the horizontal sync operation edge selection.

Table 30-59. Horizontal Sync Operation Edge Selection

HE	Horizontal sync operation edge
0	Trailing edge
1	Leading edge

Table 30-60. Horizontal Sync Signal Input Logic Control

IHX	Horizontal sync signal input logic
0	The HSYNC pin is negative logic input
1	The HSYNC pin is positive logic input

Examples of horizontal sync operation

Figure 30-78 and Figure 30-79 show examples of horizontal sync operation.

Figure 30-78. Horizontal Sync Trailing Edge Operation (HE=0, IHX=0)

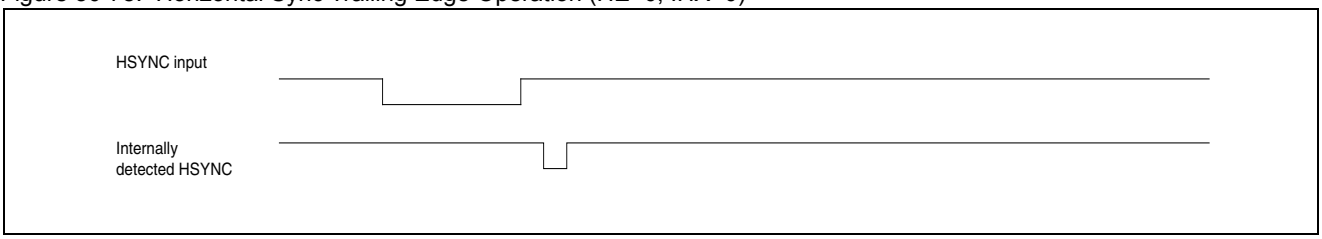
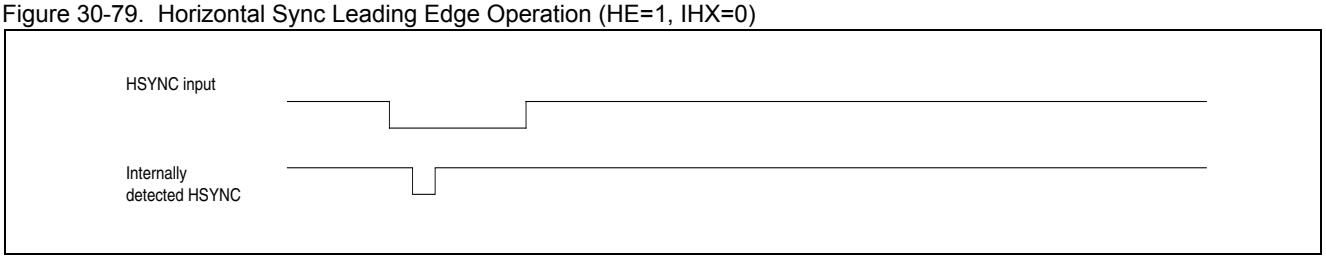


Figure 30-79. Horizontal Sync Leading Edge Operation (HE=1, IHX=0)



Note: Basically, the horizontal sync operation edge selection should be set to the leading edge operation (HE=1). In case of the trailing edge is set (HE=0), the sprite character, screen background character, and character background character cannot be used.

30.3.2.3 Field Control

When interlaced display is performed (the input sync signals display the interface timing), the field can be detected from the phase timing of the input vertical sync signal and the input horizontal sync signal. Selective output of the font display raster (even/odd) is controlled by the result of this field detection.

To perform interfaced display during dot clock external input operation, the number of clocks in the horizontal sync signal period needs to be set in command 18 (bits DKA11 to DKA0). This does not need to be controlled when performing non-interlaced display.

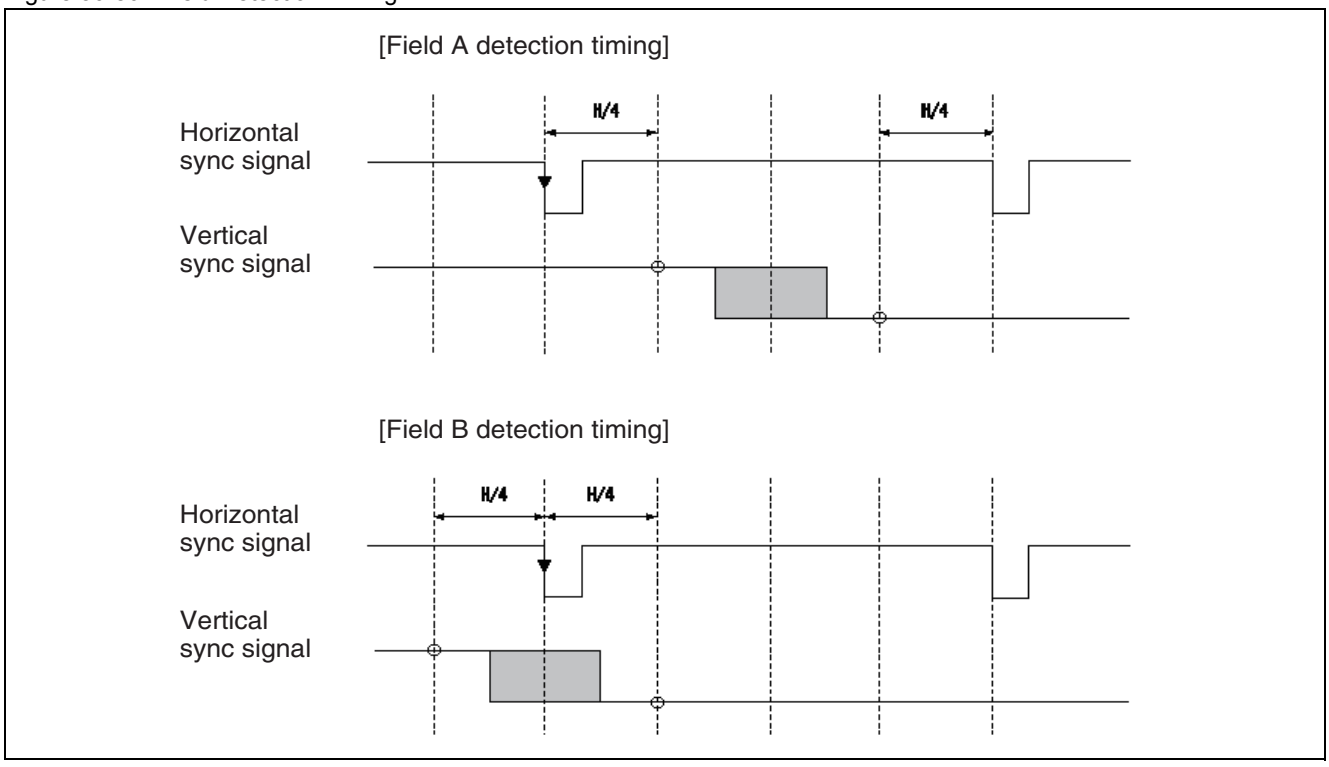
Field detection control

When interlacing is configured in the IN bit of the synchronization control (command 11), the field for performing interlaced display is detected from the phase state of the input horizontal sync signal and vertical sync signal.

- Field detection control is performed by detecting the vertical sync pulse leading edge by measuring the level of the vertical sync signal at a position of H/4 before and after the synchronization pulse leading edge of the horizontal sync signal. If there are variations in the level of the vertical sync signal in the vicinity of this detection position, field detection may not be able to be performed correctly. Furthermore, if there are variations in the period of the horizontal sync signal in the vicinity of the synchronization pulse of the vertical sync signal, the field might not be detected correctly. In this situation, stabilize the horizontal sync signal using an external circuit before the signal is input.
- The field detection timing of "H/4" is calculated by taking the number of clocks in the horizontal sync period set configured in bits DKA11 to DKA0 of the PLL clock control (command 18) as 1 Hsync.

  Note: See "30.3.5 Synchronization Control" for details on the font display rasters displayed in each field when performing interlacing.
- Figure 30-80 shows the input timing of the vertical sync signal (VSYNC pin input signal) and horizontal sync signal (HSYNC pin input signal) for performing interlaced display.

Figure 30-80. Field Detection Timing



Note: See "30.3.5 Synchronization Control" for details on the font display rasters displayed in each field when performing interlacing.

Field Correction Control

- Field correction control

  Field correction control is a function that converts the display states of the even raster display field and odd raster display field of fonts displayed in the field state generated by the field detection control. The display output is corrected in the case where the display output for each field is not performed correctly (output fields are incorrect) during interlaced display output.

  Each of the field outputs can be exchanged by the field correction control of synchronization control (command 11): bit FC.

  Table 30-61 shows the field correction control (FC) of the synchronization control (command 11).

Table 30-61. Field Correction Control

FC	Field correction control
0	No correction
1	Correction

30.3.3 Display Signal Output

This section describes the display signal output timing.

Display signal output timing

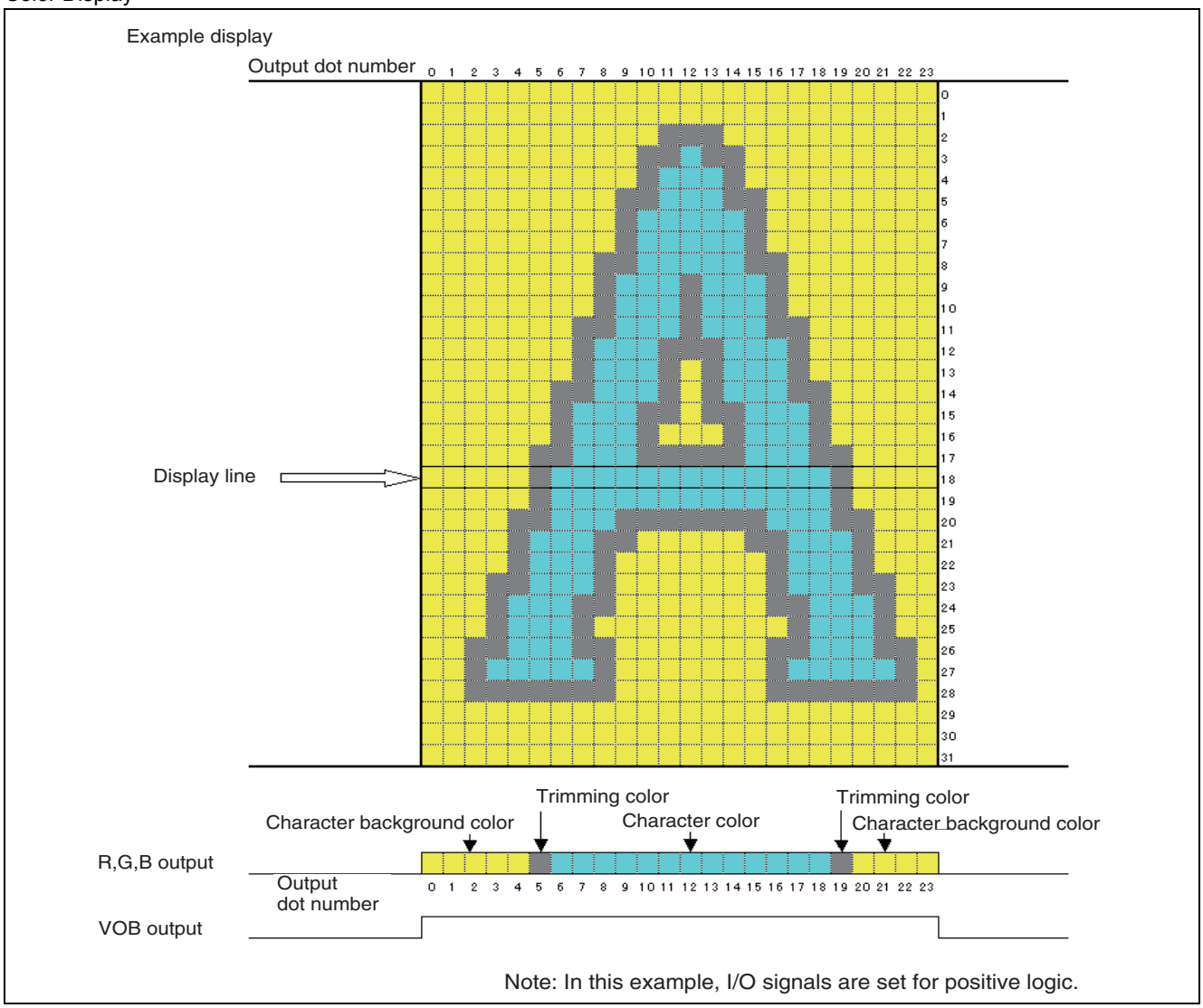
Display Signal Output

- Display period signal: VOB
- Display color code: R, G, B

Display signal output example 1

Figure 30-81 shows an example of the output when displaying character display, character background display, and trimming display.

Figure 30-81. Example of Output When Displaying Character Color Display, Character Background Display, and Trimming Color Display



Note: During the period where the VOB output is not active, the R, B, and G outputs of the OSDC macro (R, B, G) output Low when the I/O signals are set to positive logic.

Display signal output example 2

The non-displayable periods are indicated by the horizontal sync signal (HSYNC) and vertical sync signal (VSYNC) inputs.

Figure 30-82 and Figure 30-83 show examples of non-displayable periods during sync signal input.

Figure 30-82. Example of Operation of Display Output Mask by HSYNC Input Signal

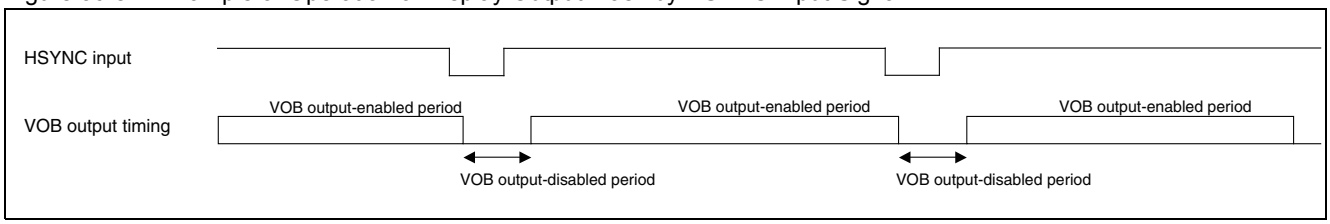
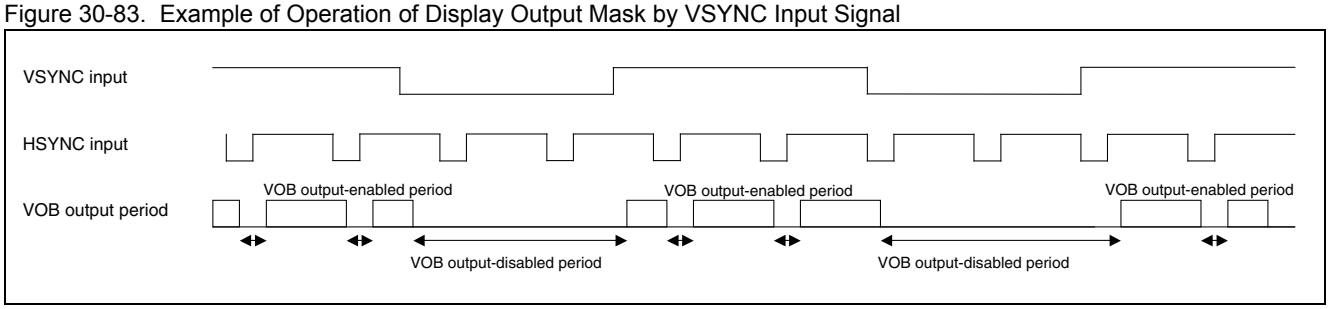


Figure 30-83. Example of Operation of Display Output Mask by VSYNC Input Signal



30.3.4 Display Period Control

There are two types of display period control as follows.

- Vertical display period control
- Horizontal display period control

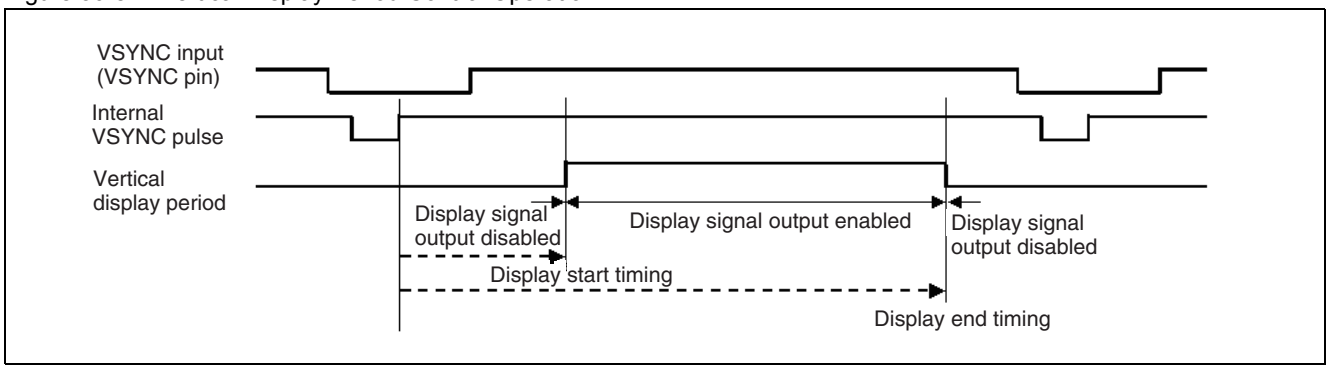
These functions enable the horizontal and vertical blanking to be controlled.

Vertical display period control

The vertical display period control internally generates the vertical display period and performs display signal output control. During the vertical display period, display output is performed in the vertical direction. The display period can be configured using the command setting.

Figure 30-84 shows the operation of the vertical display period control.

Figure 30-84. Vertical Display Period Control Operation



The start and end timing of the vertical display period can be configured as follows.

- Vertical display start timing
  Display period control 1 (command 14-0): bits DYS10 to 0
  Configurable from 0 to 2047 Hsyncs in units of 1 Hsync.
- Vertical display end timing
  Display period control 1 (command 14-0): bits DYE10 to 0
  Configurable from 0 to 2047 Hsyncs in units of 1 Hsync.

Note: If the following settings are made, the display end timing setting is invalid. Therefore, do not make these settings.

DYS [10:0] (Vertical display start timing) ≥ DYE[10:0] (Vertical display end timing)

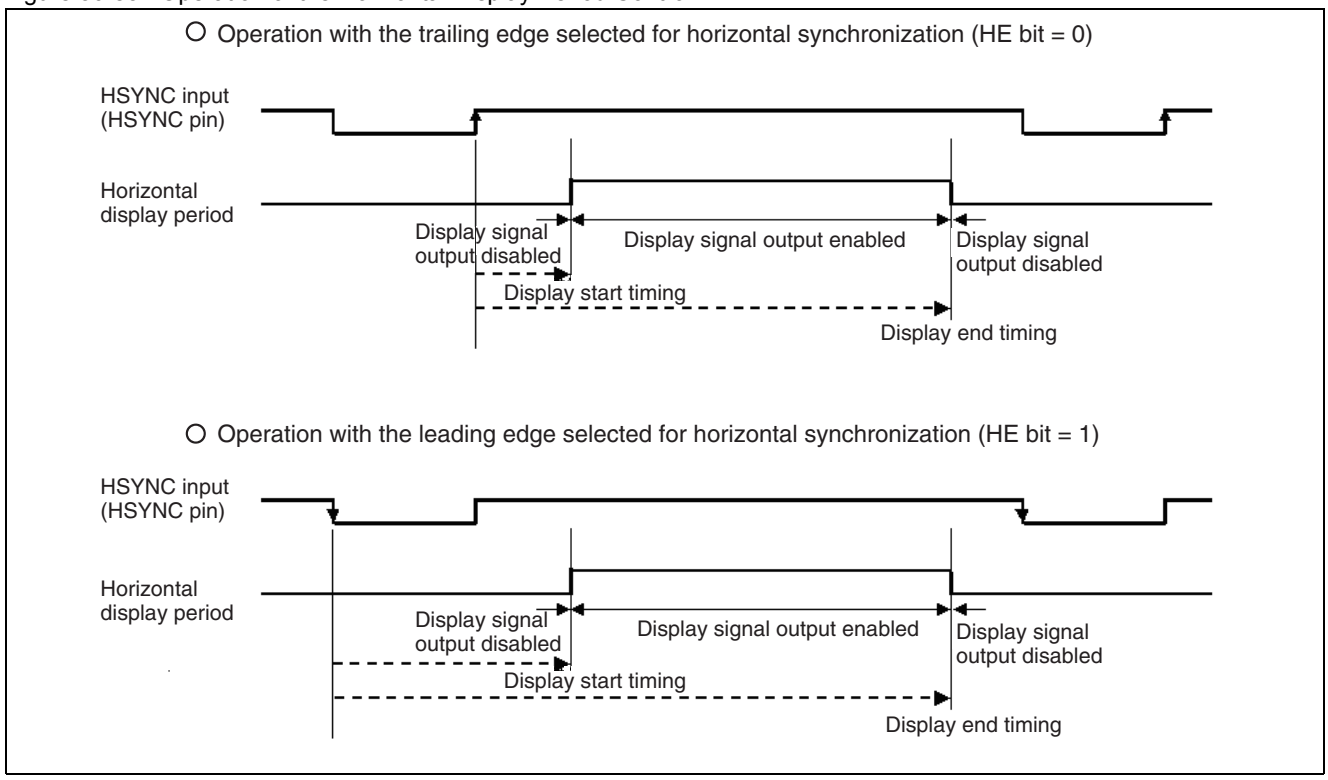
Horizontal display period control

The horizontal display period control internally generates the horizontal display period and performs display signal output control. During the horizontal display period, display output is performed in the horizontal direction. The display period can be configured using the command setting. The following two types of operation are performed depending on the horizontal sync operation edge selection (HE bit) control of the I/O pin control (command 13).

- Operation on trailing edge (HE bit = 0) of horizontal sync
- Operation on leading edge (HE bit = 1) of horizontal sync

Figure 30-85 shows the operation of the horizontal display period control.

Figure 30-85. Operation of the Horizontal Display Period Control



The start and end timing of the horizontal display period can be configured as follows.

- Horizontal display start timing
  Display period control 2 (command 14-1): bits DXS11 to 0
  Configurable from 0 to 4095 dot clocks in units of 1 dot clock.
- Horizontal display end timing
  Display period control 2 (command 14-1): bits DXE11 to 0
  Configurable from 0 to 4095 dot clocks in units of 1 dot clock.

For the start and end of window horizontal display periods, the setting values of the horizontal position are calculated as follows.

- Horizontal display start position = DXS + 7 [dots]
- Horizontal display end position = DXE + 7 [dots]

Note: If the following settings are made, the display end timing setting is invalid. Therefore, do not make these settings.

DXS [11:0] (Horizontal display start timing) ≥ DXE[11:0] (Horizontal display end timing)

30.3.5 Synchronization Control

Capable of interlaced or non-interlaced (progressive) display by setting the interlaced/non-interlaced control bit (IN bit) of the synchronization control (command 11).

Synchronization control

Table 30-62 shows the synchronization control.

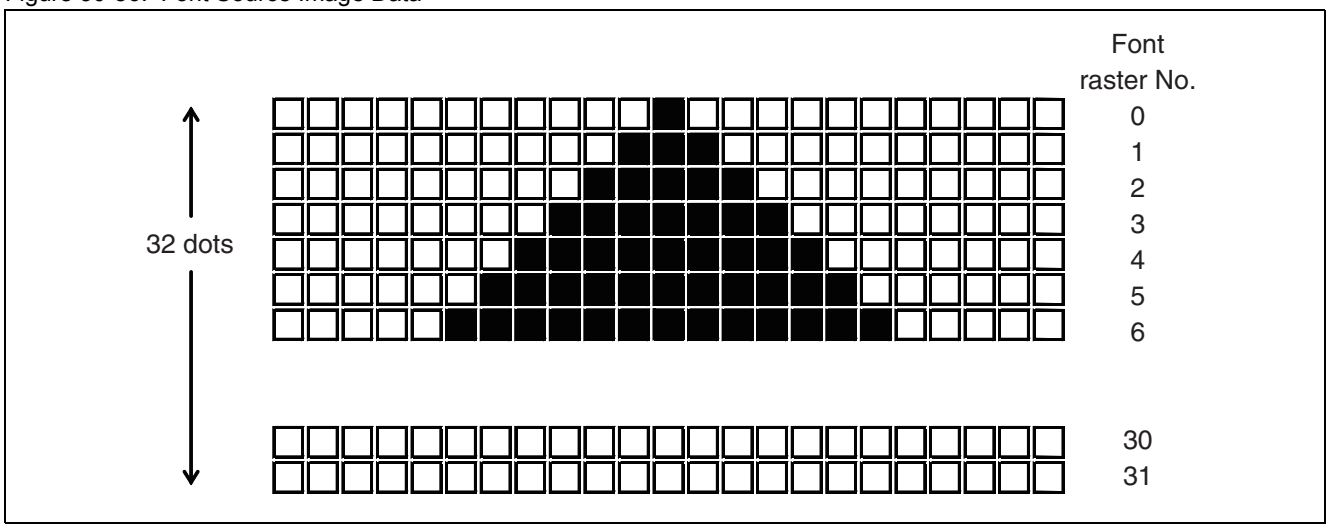
Table 30-62. Synchronization Control

IN	Synchronization control
0	Interlaced operation
1	Non-interlaced operation (progressive operation)

Display examples

Figure 30-86 shows the font source image data.

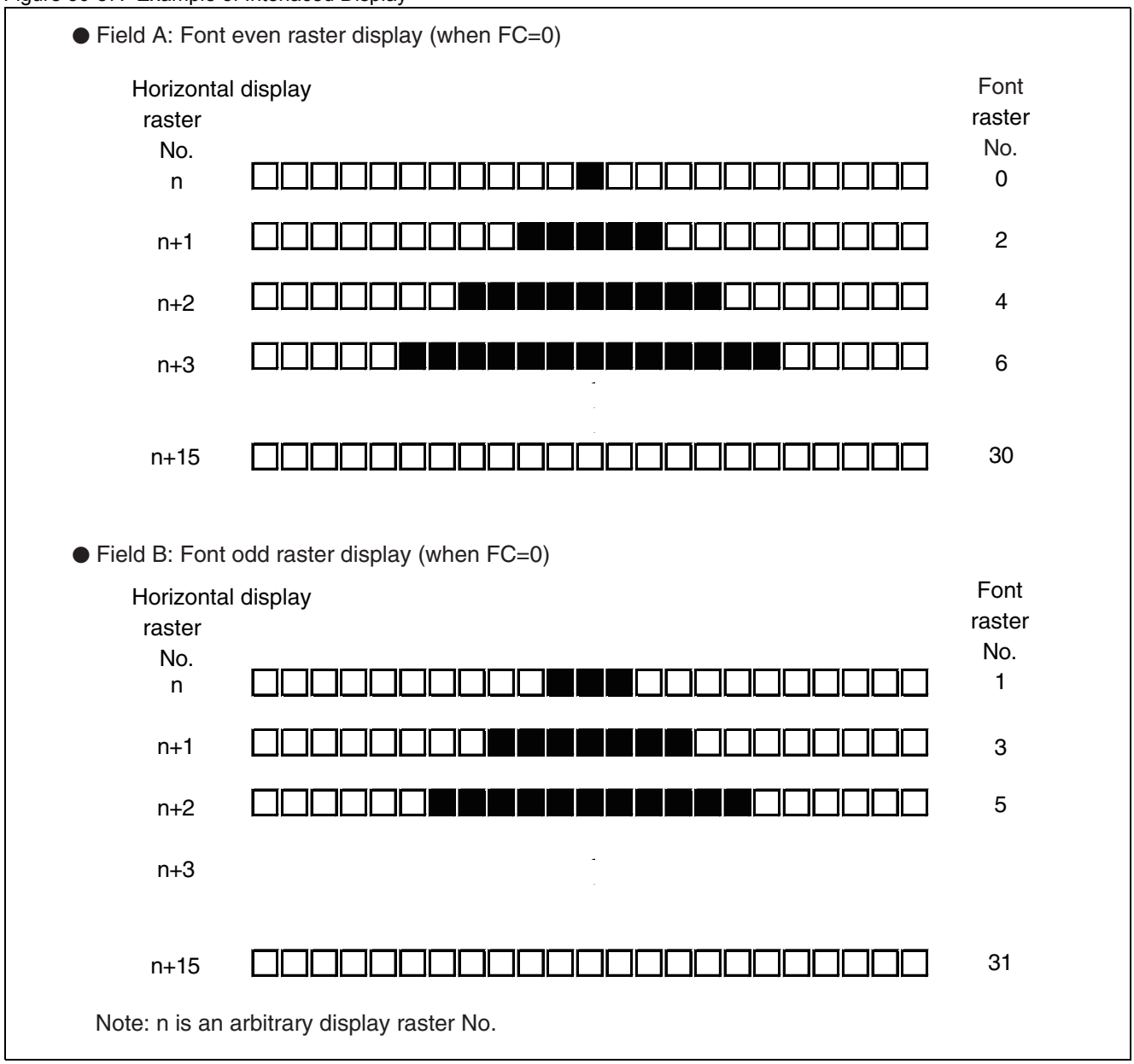
Figure 30-86. Font Source Image Data



- Figure 30-87 shows an example of interlaced display (IN set to 0).

  In an interlaced display, the display image is composed by displaying different dots from the font source image data in field A and field B detected by "30.3.2.3 Field Control".

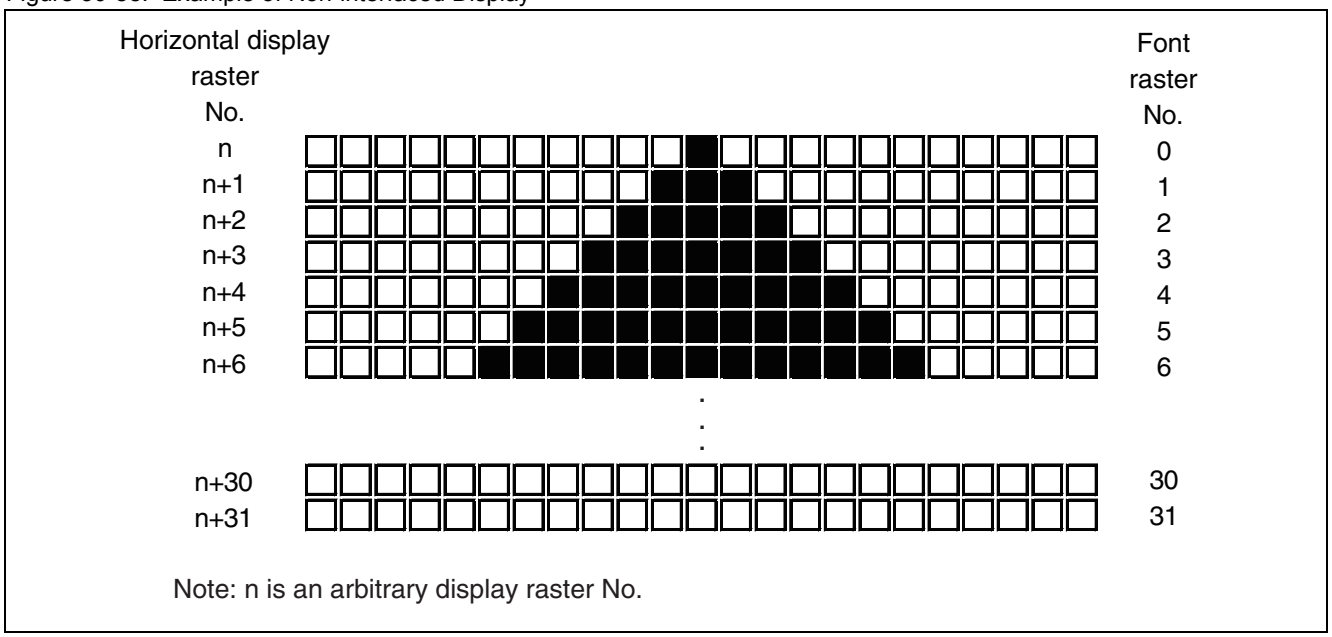
Figure 30-87. Example of Interlaced Display



Note: If the display rasters of font data displayed during interlaced display are inverted, this can be corrected by controlling the FC bit of the field correction control (command 11).

- Figure 30-88 shows an example of non-interlaced display (IN set to 1).
  In a non-interlaced display, the display image is composed by sequentially outputting the font source image data.

Figure 30-88. Example of Non-interfaced Display



30.3.6 Interrupt Control

The OSDC control interrupts have the following three sources.

- VRAM fill end detection
- Line display end detection
- Vertical sync signal detection

Interrupt requests to the CPU are the logical OR of 3 types for main and logical OR of 2 types for sub.

Interrupt control

Interrupt control is a function to control the interrupts that occur due to the internal operation status.

Interrupts are controlled by the interrupt source flags and interrupt generation control in the interrupt control (command 15).

Interrupt source flags

The interrupt source flag is set to "1" when the interrupt enabled control bit (LIE, VIE, FIE) is enabled and the interrupt source occurs. In order to clear a flag that has been set, "1" needs to be written to the source flag.

There are three interrupt source flags.

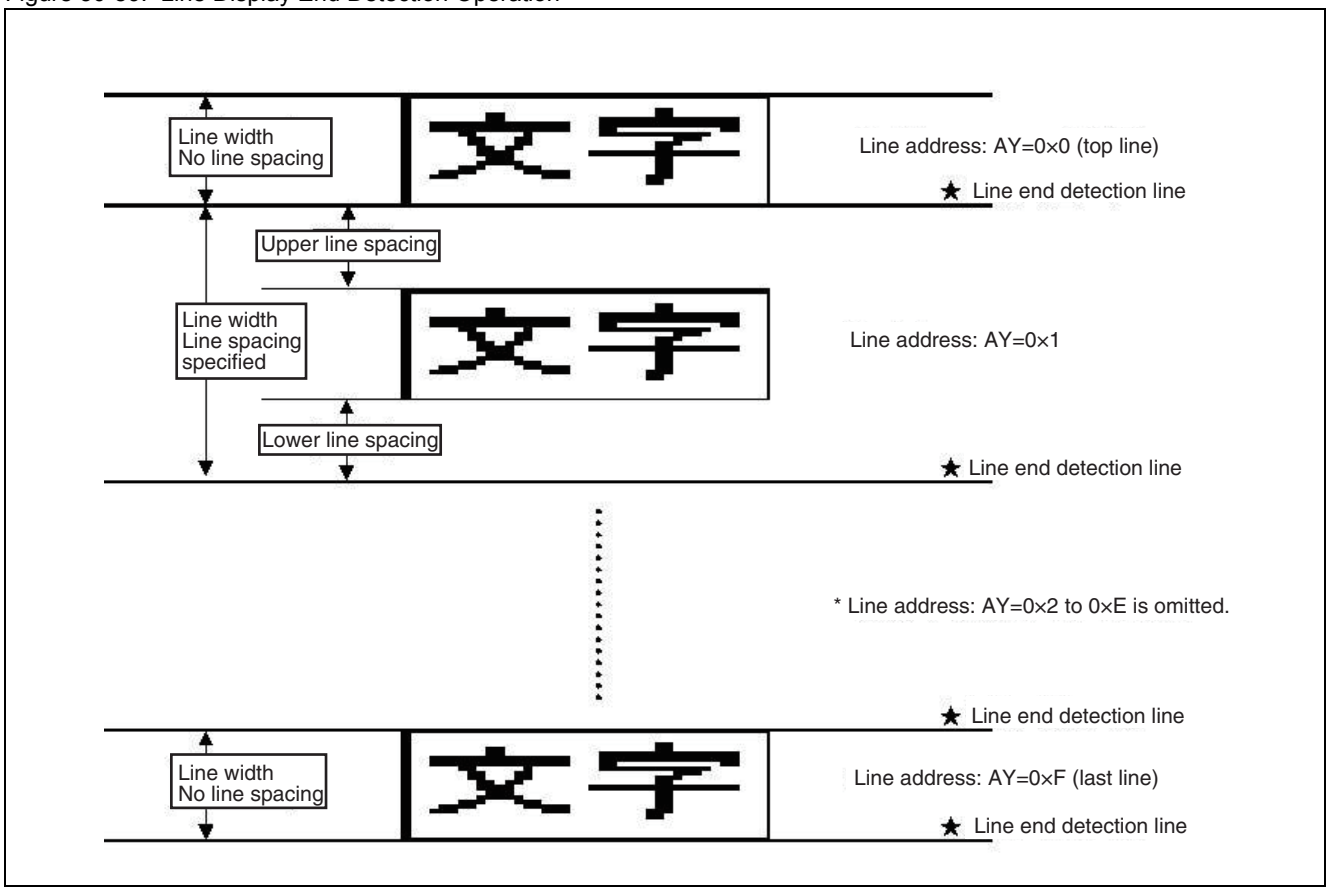
- VRAM fill end flag (command 15: bit FIF)
- Line display end detection flag (command 15: LIF bit)
- Vertical sync signal detection flag (command 15: VIF bit) *Main operation only

- Line display end detection flag

  As shown in Figure 30-89, the line display end source occurs when the character display ends in the line end detection line of the final raster (the raster section immediately before the start of the first line) of each line.

  In order to allow "line end detection line" interrupts to be generated, the line spacing bottom control "LWDEN" needs to be set to "1" in command 4 (line control data set 2) for the corresponding line. (If the current line and line below are treated as joined in other settings, the line interrupt for the "line end detection line" in the current line does not occur.)

Figure 30-89. Line Display End Detection Operation

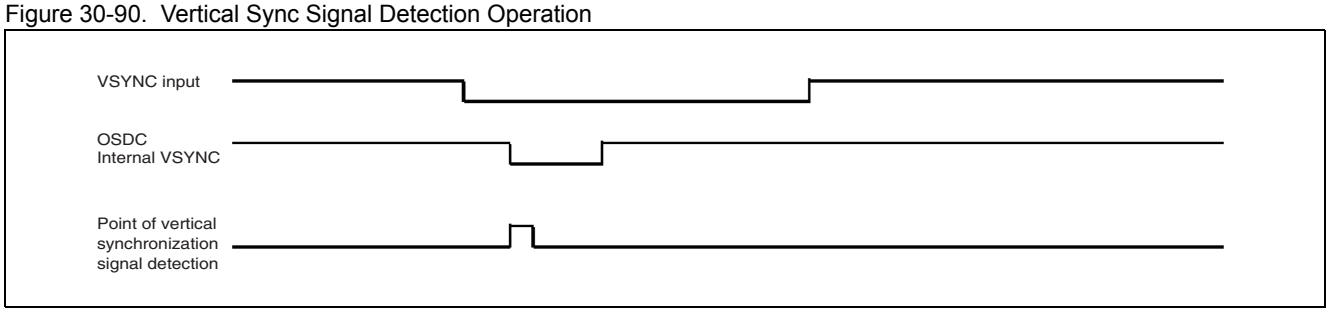


- Vertical sync signal detection flag

  This flag detects the falling edge of the internal vertical sync signal.

  Figure 30-90 shows the detection point of the vertical sync signal.

Figure 30-90. Vertical Sync Signal Detection Operation



- VRAM fill end detection flag

  The VRAM fill end source occurs when the VRAM setting started by the VRAM fill command (command 0 to 2) has finished.

Interrupt source flags and interrupt enable control

The interrupt source flags and generation of interrupts can be controlled.

There are interrupt source flags and interrupt generation control for the following three interrupts.

- VRAM fill end detection interrupt (command 15: bit FIE)
  Table 30-63 shows the VRAM fill end detection interrupt control.

Table 30-63. VRAM Fill End Detection Interrupt Control

FIE	VRAM fill end detection interrupt
0	Interrupt disabled
1	Interrupt enabled

- Line display end detection interrupt (command 15: LIE bit)
  Table 30-64 shows the line display end detection interrupt control.

Table 30-64. Line Display End Detection Interrupt Control

LIE	Line display end detection interrupt
0	Interrupt disabled
1	Interrupt enabled

- Vertical sync signal detection interrupt (command 15: VIE bit) *Main operation only
  Table 30-65 shows the vertical sync signal detection interrupt control.

Table 30-65. Vertical Sync Signal Input Interrupt Control

VIE	Vertical sync signal detection interrupt
0	Interrupt disabled
1	Interrupt enabled

30.3.7 OSDC Operation Control

The OSDC operation control function controls the operation of the OSDC.

OSDC operation control

In order to control the OSDC, the input dot clock selection setting, DAC setting, and output pin setting need to be configured as necessary and the OSDC needs to be activated.

- Input Dot Clock Selection Control

  Table 30-66 shows the input dot clock control of the OSDC operation control (command 17): bit DCK.

  To select the VCO oscillator clock, configure the VCO-related settings in clock control, clock control 2, and clock control 3 before enabling this bit.

Table 30-66. Input Dot Clock Selection Control

DCK	Dot clock control
0	External dot clock input
1	Internal VCO generated dot clock input

- DAC control

  Table 30-67 shows the DAC control of the OSDC operation control (command 17): bit DPD.

Table 30-67. DAC Control

DPD	DAC control
0	Stopped
1	Running

- Output pin control

  Table 30-68 shows the output pin control of the OSDC operation control (command 17): bits ANO and DGO.

Table 30-68. Output Pin Control

ANO	Analog RGB output pin control
0	Analog RGB output OFF
1	Analog RGB output ON

DGO	Digital RGB pin output control
0	Digital RGB output OFF
1	Digital RGB output ON

- OSDC active control

  Table 30-69 shows the OSDC active control of the OSDC operation control (command 17): bit OSDEN.

  When the OSDEN bit is "0" (OSDC is disabled), the status of the OSDC active control is as follows.

  - The access to the font memory is disabled.
  - Do not access the following commands.

    Command 0: VRAM Write Address Set

    Command 1: Character Data Set 1

    Command 2: Character Data Set 2

    Command 3: Line Control Data Set 1

    Command 4: Line Control Data Set 2

    Command 15: Interrupt Control

    Command 16-0 to 127: Palette Control
  - The external pin is not operated. However, it operates internally, as long as the dot clock is supplied.

Table 30-69. OSDC Active Control

OSDEN	OSDC active control
0	OSDC disabled
1	OSDC enabled

- Sub operation control

  Table 30-70 shows the sub operation control of the OSDC operation control (command 17): bit SUBEN.

  In order to issue sub screen display control commands, first set this SUBEN bit to 1 (sub operation enabled).

  (When the SUBEN bit is 0, sub screen display control commands are not accepted.)

Table 30-70. Sub Operation Control

SUBEN	Sub operation control
0	Sub operation disabled
1	Sub operation enabled

- OSDC soft reset control

  Table 30-71 shows the OSDC soft reset control of the OSDC operation control (command 17): bit OSDR.

  This bit resets the OSDC unit. A reset occurs when the OSDR bit is set to "1". Furthermore, the reset is cleared by setting this bit to "0".

  Note: The registers are not initialized.

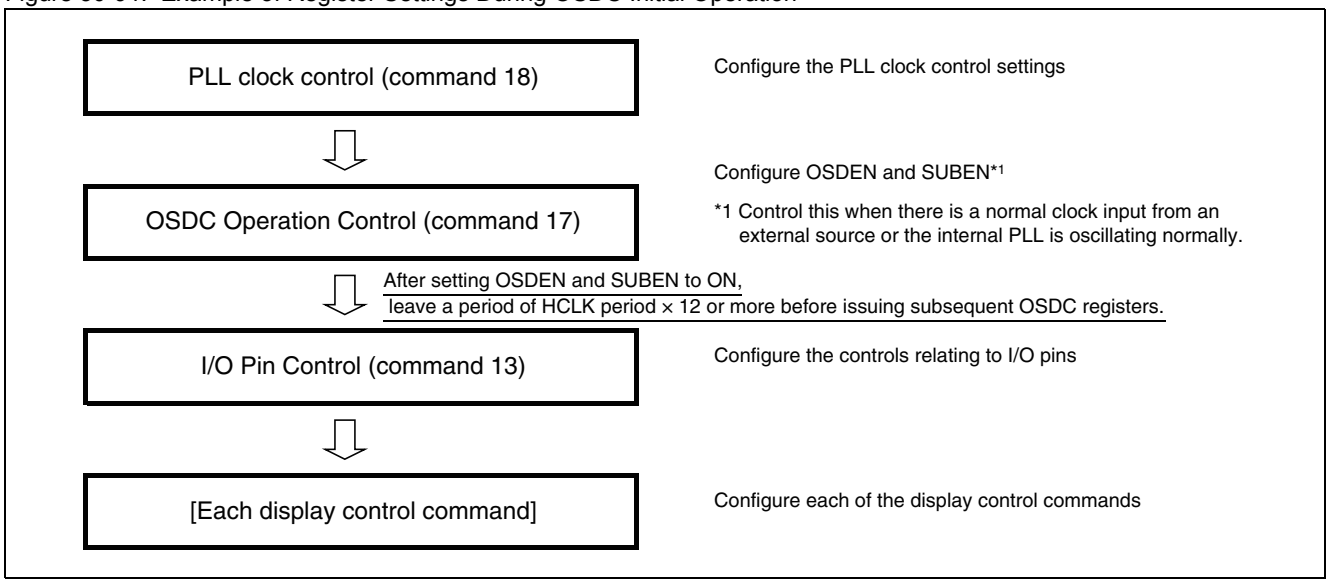
Table 30-71. OSDC Soft Reset Control

OSDR	OSDC soft reset control
0	Disabled (reset clear)
1	OSDC macro reset

Example of Register Settings During OSDC Initial Operation

Figure 30-91 shows an example of the register settings for OSDC initial operation.

Figure 30-91. Example of Register Settings During OSDC Initial Operation



OSDC operation control during transition to CPU core regulator standby mode

When not in the modes listed below, the internal regulator enters standby mode (lower power consumption) and the OSDC cannot operate.

- Main RUN/Doze mode/Main sleep/Main clock watch mode

Therefore, when in a mode other than listed above (sub clock mode, stop mode), stop the operation of the external dot clock input to the OSDC and the operation of the PLL for the dot clock.

Note: If the dot clock is stopped suddenly, register access cannot be made to commands 0, 1, 2, 3, 4, or 15. When stopping the dot clock, always set OSDEN="0" and SUBEN="0" before stopping the dot clock.
After setting OSDEN and SUBEN to ON, leave a period of HCLK period × 12 or more before issuing subsequent OSDC registers. Furthermore, when entering main operation from regulator standby mode, reconfigure all of the commands of the OSDC.

Limitations when performing sprite character display, screen background character display, or character background character display

When performing sprite character display, screen background character display, or character background character display, set the horizontal sync operation edge selection control of the I/O pin control (command 13) to leading edge (HE=1).

It is prohibited to perform sprite character display, screen background character display and character background character display when trailing edge is set (HE=0).

The OSDC transfers data from font memory into OSDC dedicated font buffer RAM on each VSYNC. Note that the displays described above are enabled after this operation finishes.

The transfer conditions and required time are as follows.

Each of the following times is required after the origin and synchronization pulse (falling edge: IVX=0) of the vertical sync signal (VSYNC pin input signal).

- When performing main screen sprite character display, +2055 [dot clocks]
- When performing sub screen sprite character display, +2055 [dot clocks]
- When performing screen background character (main screen) or character background character (main or sub screen) display, +2055 [dot clocks]

In other words, if main screen sprite character display, sub screen sprite character display, and screen background character (main screen) or character background character (main or sub screen) display are all used at the same time:

The time that needs to be reserved after the synchronization pulse (falling edge: IVX=0) of the vertical sync signal (VSYNC pin input signal) is

2055 + 2055 + 2055 -> Total 6165 [dot clocks]

Furthermore, the period of 2055 [dot clocks] for screen background character (main screen) or character background character (main or sub screen) needs to be reserved regardless of each of the settings.

If the OSDC display is started before this, the display will not be correct. Therefore, adjust the vertical blanking period using the synchronization control (command 11) or vertical display position offset control (VOF5 to VOF0), etc. to ensure that the above time is reserved.

30.4 Display Control Commands (Main/OSDC Operation)

This section explains the OSDC main display and OSDC operation control commands.

30.4.1 List of Main Screen Display/OSDC Operation and Control Commands
30.4.2 VRAM Write Address Set (Command 0)
30.4.3 Character Data Set (Command 1, Command 2)
30.4.4 Line Control Data Set (Command 3, Command 4)
30.4.5 Screen Output Control (Command 5-0)
30.4.6 Screen Display Position Control (Command 5-1)
30.4.7 Transparent Color Control (Command 6-0)
30.4.8 Graphics Color Control (Command 6-1)
30.4.9 Screen Background Character Control (Command 7)
30.4.10 Screen Background Control (Command 8-0)
30.4.11 Window Period Control (Command 8-1, Command 8-2)
30.4.12 Sprite Character Control (Command 9-0, Command 9-1)
30.4.13 Synchronization Control (Command 11)
30.4.14 Character Background Character Code Set (Command 12-0 to Command 12-7)
30.4.15 I/O Pin Control (Command 13)
30.4.16 Display Period Control (Command 14-0, Command 14-1)
30.4.17 Interrupt Control (Command 15)
30.4.18 Shaded Background Frame Color Control (Command 15-0 to Command 15-3)
30.4.19 Palette Control (Command 16-0 to Command 16-127)
30.4.20 Operation Control (Command 17)
30.4.21 PLL Clock Control (Command 18)

30.4.1 List of Main Screen Display/OSDC Operation and Control Commands

This section lists the display control commands. Table 30-72 shows a list of MAIN_OSDC operation control commands.

List of display control commands

Table 30-72. List of Display Control Commands

Address lower 16 bits (HEX)	MAIN_OSDC command no.	Data																																Function
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
40 00	0																FL				AY4	AY3	AY2	AY1	AY0			AX5	AX4	AX3	AX2	AX1	AX0	VRAM write address set [MOSD_VADR]
40 04	1		MSC2	MSC1	MSC0	MIT	MUL	MBL	MBB	MU	MD		MS1	MS0	MM2	MM1	MM0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0	Character data set 1 [MOSD_CDS1]
40 08	2									MB M3	MB M2	MB M1	MB M0		MA	MR	MG			M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0	Character data set 2 [MOSD_CDS2]
40 0C	3	LAL CA1	LAL CA0	LAL CM	LAL EN		LAL 2	LAL 1	LAL 0	LHS 3	LHS 2	LHS 1	LHS 0		LW2	LW1	LW0					LFD	LFC	LFB	LFA	LF7	LF6	LF5	LF4	LF3	LF2	LF1	LF0	Line control data set 1 [MOSD_LDS1]
40 10	4							LSW 1	LSW 0				LDS	LGY 1	LGY 0	LGX 1	LGX 0	LWU EN	LWD EN	LMU EN	LMD EN		LE	LM1	LM0	LB7	LB6	LB5	LB4	LB3	LB2	LB1	LB0	Line control data set 2 [MOSD_LDS2]
40 14	5-0							FM1	FM0	BT1	BT0	BD1	BD0								SAL CC				WE	SDS	UDS	PDS	DSP					Screen output control [MOSD_SCOC]
40 18	5-1						X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0						Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Screen display position control [MOSD_HVDP]
40 1C	6-0																								TCC	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	Transparent color control [MOSD_TSBC]
40 20	6-1								GFC	GF7	GF6	GF5	GF4	GF3	GF2	GF1	GF0								GCC	GC7	GC6	GC5	GC4	GC3	GC2	GC1	GC0	Graphics color control [MOSD_GRCC]
40 24	7						PH2	PH1	PH0							PD1	PD0			PM13	PM12	PM11	PM10	PM9	PM8	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0	Screen background character control [MOSD_SBCC]
40 28	8-0											UW BV	UW BH			UW1	UW0				UAL EN		UAL 2	UAL 1	UAL 0	U7	U6	U5	U4	U3	U2	U1	U0	Screen background control [MOSD_SCBC]
40 2C	8-1						WY E10	WY E9	WY E8	WY E7	WY E6	WY E5	WY E4	WY E3	WY E2	WY E1	WY E0						WY S10	WY S9	WY S8	WY S7	WY S6	WY S5	WY S4	WY S3	WY S2	WY S1	WY S0	Window period control 1 [MOSD_WPC1]
40 30	8-2					WX E11	WX E10	WX E9	WX E8	WX E7	WX E6	WX E5	WX E4	WX E3	WX E2	WX E1	WX E0					WX S11	WX S10	WX S9	WX S8	WX S7	WX S6	WX S5	WX S4	WX S3	WX S2	WX S1	WX S0	Window period control 2 [MOSD_WPC2]
40 34	9-0				SBL		SH2	SH1	SH0							SD1	SD0			SM13	SM12	SM11	SM10	SM9	SM8	SM7	SM6	SM5	SM4	SM3	SM2	SM1	SM0	Sprite character control 1 [MOSD_SPC1]
40 38	9-1					SX11	SX10	SX9	SX8	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0						SY10	SY9	SY8	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0	Sprite character control 2 [MOSD_SPC2]
40 3C	11											VOF 5	VOF 4	VOF 3	VOF 2	VOF 1	VOF 0										IN		FC					Synchronization control [MOSD_SYNC]
40 40	12-0			BM1 C13	BM1 C12	BM1 C11	BM1 C10	BM1 C9	BM1 C8	BM1 C7	BM1 C6	BM1 C5	BM1 C4	BM1 C3	BM1 C2	BM1 C1	BM1 C0			BM0 C13	BM0 C12	BM0 C11	BM0 C10	BM0 C9	BM0 C8	BM0 C7	BM0 C6	BM0 C5	BM0 C4	BM0 C3	BM0 C2	BM0 C1	BM0 C0	Character background character code set [MOSD_CBC0 to MOSD_CBC7]
40 44	12-1			BM3 C13	BM3 C12	BM3 C11	BM3 C10	BM3 C9	BM3 C8	BM3 C7	BM3 C6	BM3 C5	BM3 C4	BM3 C3	BM3 C2	BM3 C1	BM3 C0			BM2 C13	BM2 C12	BM2 C11	BM2 C10	BM2 C9	BM2 C8	BM2 C7	BM2 C6	BM2 C5	BM2 C4	BM2 C3	BM2 C2	BM2 C1	BM2 C0	
40 48	12-2			BM5 C13	BM5 C12	BM5 C11	BM5 C10	BM5 C9	BM5 C8	BM5 C7	BM5 C6	BM5 C5	BM5 C4	BM5 C3	BM5 C2	BM5 C1	BM5 C0			BM4 C13	BM4 C12	BM4 C11	BM4 C10	BM4 C9	BM4 C8	BM4 C7	BM4 C6	BM4 C5	BM4 C4	BM4 C3	BM4 C2	BM4 C1	BM4 C0	
40 4C	12-3			BM7 C13	BM7 C12	BM7 C11	BM7 C10	BM7 C9	BM7 C8	BM7 C7	BM7 C6	BM7 C5	BM7 C4	BM7 C3	BM7 C2	BM7 C1	BM7 C0			BM6 C13	BM6 C12	BM6 C11	BM6 C10	BM6 C9	BM6 C8	BM6 C7	BM6 C6	BM6 C5	BM6 C4	BM6 C3	BM6 C2	BM6 C1	BM6 C0	
40 50	12-4			BM9 C13	BM9 C12	BM9 C11	BM9 C10	BM9 C9	BM9 C8	BM9 C7	BM9 C6	BM9 C5	BM9 C4	BM9 C3	BM9 C2	BM9 C1	BM9 C0			BM8 C13	BM8 C12	BM8 C11	BM8 C10	BM8 C9	BM8 C8	BM8 C7	BM8 C6	BM8 C5	BM8 C4	BM8 C3	BM8 C2	BM8 C1	BM8 C0	
40 54	12-5			BMB C13	BMB C12	BMB C11	BMB C10	BMB C9	BMB C8	BMB C7	BMB C6	BMB C5	BMB C4	BMB C3	BMB C2	BMB C1	BMB C0			BMA C13	BMA C12	BMA C11	BMA C10	BMA C9	BMA C8	BMA C7	BMA C6	BMA C5	BMA C4	BMA C3	BMA C2	BMA C1	BMA C0	
40 58	12-6			BMD C13	BMD C12	BMD C11	BMD C10	BMD C9	BMD C8	BMD C7	BMD C6	BMD C5	BMD C4	BMD C3	BMD C2	BMD C1	BMD C0			BMC C13	BMC C12	BMC C11	BMC C10	BMC C9	BMC C8	BMC C7	BMC C6	BMC C5	BMC C4	BMC C3	BMC C2	BMC C1	BMC C0	
40 5C	12-7			BMF C13	BMF C12	BMF C11	BMF C10	BMF C9	BMF C8	BMF C7	BMF C6	BMF C5	BMF C4	BMF C3	BMF C2	BMF C1	BMF C0			BME C13	BME C12	BME C11	BME C10	BME C9	BME C8	BME C7	BME C6	BME C5	BME C4	BME C3	BME C2	BME C1	BME C0	
40 60	13								VHE	HE					IHX	IVX															OHX	OBX	OCX	I/O pin control [MOSD_IOTC]
40 64	14-0						DYE 10	DYE 9	DYE 8	DYE 7	DYE 6	DYE 5	DYE 4	DYE 3	DYE 2	DYE 1	DYE 0						DYS 10	DYS 9	DYS 8	DYS 7	DYS 6	DYS 5	DYS 4	DYS 3	DYS 2	DYS 1	DYS 0	Display period control 1 [MOSD_CDP1]
40 68	14-1					DXE 11	DXE 10	DXE 9	DXE 8	DXE 7	DXE 6	DXE 5	DXE 4	DXE 3	DXE 2	DXE 1	DXE 0					DXS 11	DXS 10	DXS 9	DXS 8	DXS 7	DXS 6	DXS 5	DXS 4	DXS 3	DXS 2	DXS 1	DXS 0	Display period control 2 [MOSD_CDP2]
40 6C	15																						FIF	LIF	VIF						FIE	LIE	VIE	Interrupt control [MOSD_INTC]

Table 30-72. List of Display Control Commands

Address lower 16 bits (HEX)	MAIN_OSDC command no.	Data																															Function
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
40 70	15-0	B1H 7	B1H 6	B1H 5	B1H 4	B1H 3	B1H 2	B1H 1	B1H 0	B1S 7	B1S 6	B1S 5	B1S 4	B1S 3	B1S 2	B1S 1	B1S 0	B0H 7	B0H 6	B0H 5	B0H 4	B0H 3	B0H 2	B0H 1	B0H 0	B0S 7	B0S 6	B0S 5	B0S 4	B0S 3	B0S 2	B0S 1	B0S 0	Shaded background frame color control [MOSD_SBC0 to MOSD_SBC3]
40 74	15-1	B3H 7	B3H 6	B3H 5	B3H 4	B3H 3	B3H 2	B3H 1	B3H 0	B3S 7	B3S 6	B3S 5	B3S 4	B3S 3	B3S 2	B3S 1	B3S 0	B2H 7	B2H 6	B2H 5	B2H 4	B2H 3	B2H 2	B2H 1	B2H 0	B2S 7	B2S 6	B2S 5	B2S 4	B2S 3	B2S 2	B2S 1	B2S 0	
40 78	15-2	B5H 7	B5H 6	B5H 5	B5H 4	B5H 3	B5H 2	B5H 1	B5H 0	B5S 7	B5S 6	B5S 5	B5S 4	B5S 3	B5S 2	B5S 1	B5S 0	B4H 7	B4H 6	B4H 5	B4H 4	B4H 3	B4H 2	B4H 1	B4H 0	B4S 7	B4S 6	B4S 5	B4S 4	B4S 3	B4S 2	B4S 1	B4S 0	
40 7C	15-3	B7H 7	B7H 6	B7H 5	B7H 4	B7H 3	B7H 2	B7H 1	B7H 0	B7S 7	B7S 6	B7S 5	B7S 4	B7S 3	B7S 2	B7S 1	B7S 0	B6H 7	B6H 6	B6H 5	B6H 4	B6H 3	B6H 2	B6H 1	B6H 0	B6S 7	B6S 6	B6S 5	B6S 4	B6S 3	B6S 2	B6S 1	B6S 0	
42 00	16-0	PL0 1R4	PL0 1R3	PL0 1R2	PL0 1R1	PL0 1R0	PL0 1G5	PL0 1G4	PL0 1G3	PL0 1G2	PL0 1G1	PL0 1G0	PL0 1B4	PL0 1B3	PL0 1B2	PL0 1B1	PL0 1B0	PL0 0R4	PL0 0R3	PL0 0R2	PL0 0R1	PL0 0R0	PL0 0G5	PL0 0G4	PL0 0G3	PL0 0G2	PL0 0G1	PL0 0G0	PL0 0B4	PL0 0B3	PL0 0B2	PL0 0B1	PL0 0B0	Palette control [MOSD_PL0 to MOSD_PL127]
to	to																																	
43 FC	16-127	PLF-FR4	PLF-FR3	PLF-FR2	PLF-FR1	PLF-FR0	PLF FG5	PLF FG4	PLF FG3	PLF FG2	PLF FG1	PLF FG0	PLF FB4	PLF FB3	PLF FB2	PLF FB1	PLF FB0	PLF ER4	PLF ER3	PLF ER2	PLF ER1	PLF ER0	PLF EG5	PLF EG4	PLF EG3	PLF EG2	PLF EG1	PLF EG0	PLF EB4	PLF EB3	PLF EB2	PLF EB1	PLF EB0	
44 00	17											SUB EN	OSD EN			DG O	ANO								OSD R				DCK				DPD	OSDC operation control [MOSD_OSDC]
44 04	18			DHR SA	DCO	DKA 11	DKA 10	DKA 9	DKA 8	DKA 7	DKA 6	DKA 5	DKA 4	DKA 3	DKA 2	DKA 1	DKA 0	VCO A	VSL A2	VSL A1	VSL A0			CPE A	PDE A				DAP 4	DAP 3	DAP 2	DAP 1	DAP 0	PLL clock control [MOSD_PLLC]

Notes:

- When reset is input, the SDS, UDS, PDS, and DSP bits of the screen output control, the FIF, LIF, VIF, FIE, LIE, and VIE bits of the interrupt control, the OHX, OBX, and OCX bits of the I/O pin control, and the SUBEN, OSDEN, DGO, ANO, OSDR, DCK, and DPD bits of the OSDC operation control are initialized to "0". Other register bits and the contents of VRAM are undefined. During the reset period, the R[4:0], G[5:0], B[4:0], VOB, and VOA[2:0] pin outputs are Low.
- After input of a reset signal, be sure to set all of the register bits and all of the VRAM (character data and line control data).
- Blanks indicate reserved bits. Always set these to "0" when issuing a command.
- Access from the CPU to each of the OSDC commands (registers) is only supported by word access (32-bit access). Furthermore, burst access is not supported.
- In order to issue display control commands to the sub screen, it is necessary to set the SUBEN bit (sub operation enable) of command 17 (OSDC operation control) to 1 (sub operation enabled).
  (When the SUBEN bit is 0, do not issue SUB_OSDC display control commands.)
- If the command 17 or command 18 settings are changed during OSDC normal display operation, OSDC operation and operations related to the OSDC cannot be guaranteed when the command is changed.

30.4.2 VRAM Write Address Set (Command 0)

Command 0 configures the VRAM write address and specifies the VRAM fill.

Command 0 (VRAM write address set)

Address: 4000_H

- Format (write-only, 32-bit access only)

  [MOSD_VADR]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															FL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			AY4	AY3	AY2	AY1	AY0			AX5	AX4	AX3	AX2	AX1	AX0

FL: Specify VRAM fill

(0: OFF, 1: ON)

AY4 to AY0: Line address

(00_H to $1F_H$)

AX5 to AX0: Column address

(00_H to $3B_H$)

- Function

  This command sets the VRAM write address and specifies the VRAM fill.

  This is used to set the line/column address before the character data set (issuing command 1, 2) and to set the line address before the line control data set (issuing command 3, 4).

  VRAM fill is started by executing character data set 2 (command 2).

- Additional Information
  - During normal writes (writing 1 character data or 1 line control data), set the VRAM fill specification to OFF (FL=0).
  - The VRAM write address specified by issuing this command is automatically incremented after executing character data set 2 (command 2).
    (on the last column this is incremented to the first column of the next line, and on the last column of the last line this is incremented to the first column of the first line).
  - The VRAM fill function writes the single character data specified by character data set 1, 2 (commands 1, 2) to character VRAM from the line and column address specified by command 0 to the last line (line 32) and last column (column 60). VRAM fill is started by executing character data set 2 (command 2).
    After the fill has finished executing, it is possible to generate a VRAM fill interrupt.
    Do not issue commands 1 to 4 while the VRAM fill is executing.

  Notes:

  - "OCX: Display color signal output logic control" inversely controls OSDC analog output level also.
  - Basically, the horizontal sync operation edge selection should be set to the leading edge operation (HE=1).
  - In case of the trailing edge is set (HE=0), the sprite character, screen background character, and character background character cannot be used.

Notes:

- Line control data set (issuing command 3, 4) ignores the column address (AX5 to AX0).
  Furthermore, automatic address incrementing is not performed after line control data set.
- Specifying VRAM fill is only valid for character data set (command 1, 2).

30.4.3 Character Data Set (Command 1, Command 2)

Character data is configured using command 1 and the VRAM is set and the screen updated by executing command 2.

Command 1 (character data set 1)

Address: 4004_H

- Format (write-only, 32-bit access only)

  [MOSD_CDS1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSC2	MSC1	MSC0	MIT	MUL	MBL	MBB	MU	MD		MS1	MS0	MM2	MM1	MM0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0

MSC2, MSC1, MSC0: Character shaded background frame color type selection control

(color types 0 to 7)

MIT: Italic control

(0: Italics OFF)

(1: Italics ON)

MBL, MBB: Blink control

(0, 0: Blinking OFF)

(0, 1: Character background blinking ON)

(1, 0: Character + trimming blinking ON)

(1, 1: Character + trimming + character background blinking ON)

MUL: Underline control

(0: Underline OFF)

(1: Underline ON)

MU: Character shaded background frame top erase control

(0: Character shaded background frame top ON)

(1: Character shaded background frame top OFF)

MD: Character shaded background frame bottom erase control

(0: Character shaded background frame bottom ON)

(1: Character shaded background frame bottom OFF)

MS1, MS0: Character horizontal size control

(0, 0: 16 bits)

(0, 1: 24 bits)

(1, 0: 32 bits)

(1, 1: Setting prohibited)

MB7 to MB0: Character background color

(256 colors)

MM2, MM1, MM0: Character background control

(0, 0, 0: OFF)

(0, 0, 1: Solid display)

(0, 1, 0: Shaded concaved display (solid))

(0, 1, 1: Shaded convexed display (solid))

(1, 0, 0: Background character display)

(1, 0, 1: Setting prohibited)

(1, 1, 0: Shaded concaved display (background character))

(1, 1, 1: Shaded convexed display (background character))

MC7 to MC0: Character color

(256 colors)

- Function

  This command configures the character data. The VRAM settings are updated and the changes reflected on the screen when character data set 2 (command 2) is executed.

- Additional Information
  - The character color, character background color, character background display, character horizontal size, italic display, underline display, and blink display can be freely configured in any combination separately for each character.
  - The shaded background display can be configured to merge with the upper, lower, left or right characters by the combination of the MU and MD bits with the MR bit in character data set 2 (command 2).
  - The shaded background display frame color settings can be configured by setting bits BxH7 to BxH0 and BxS7 to BxS0 (for x = 0 to 7) in the shaded background frame color control (command 15-0 to command 15-3) and then selecting the background frame color type (0 to 7) by setting the character shaded background frame color type selection control bits MSC2, MSC1, and MSC0.
  - When blink control is set to on, the blinking (flashing) display is performed according to the setting of bits BT1, BT0, BD1, and BD0 of the screen output control (command 5-0).
  - The background character for the background character display is selected from the character background character types 0 to F by setting bits MBM3 to MBM0 of character data set 2 (command 2). Furthermore, background character types 0 to F can be set to an arbitrary character (background character) by setting character background character codes 1, 0 to F, E (command 12-0 to command 12-7) to the character code.

Command 2 (character data set 2)

Address: 4008_H

- Format (write-only, 32-bit access only)

  [MOSD_CDS2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								MBM3	MBM2	MBM1	MBM0		MA	MR	MG

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

MBM3 to MBM0: Character background character type control

(0_H to F_H: 16 character types)

MR: Shaded background right character merge control

(0: Do not merge right)

(1: Merge right)

M13 to M0: Character code

(0000_H to $3FFF_H$: 16384 character types)

MA: Character alpha blend attribute control

(0: A attribute, 1: B attribute)

MG: Character/graphics character control

(0: Character, 1: Graphics character)

- Function

  This setting data is written together with the character data configured by character data set 1 (command 1) to the VRAM specified by the VRAM write address set (command 0).

  The VRAM write address is automatically incremented after this command is executed.

- Additional Information
  - The shaded background right character merge control bit (MR) only applies to character where shaded background is specified (MM1 is set to 1) in character data set 1 (command 1).
  - Alpha blend can be enabled or disabled for each individual character by the combination of the character alpha blend attribute control bit (MA) and the line character alpha blend attribute control (LALCA1 and LALCA0) of the line control data set 1 (command 3).
  - The character background character type control bits (MBM3 to MBM0) are enabled when the (MM2, MM1, MM0) bits of character data set 1 (command 1) = (1, 0, 0), (1, 1, 0), or (1, 1, 1). The character codes that correspond to the character background character types of MBM3 to MBM0 are configured by setting the character codes in the character background character code set (command 12-0 to command 12-7) registers.

Note: Because the contents of VRAM are undefined when the power is turned ON, be sure to set all of the VRAM data before starting the display.

30.4.4 Line Control Data Set (Command 3, Command 4)

Line control data is configured using command 3 and the line VRAM is set and the screen updated by executing command 4.

Command 3 (line control data set 1)

Address: $400C_H$

- Format (write-only, 32-bit access only)

  [MOSD_LDS1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LALCA1	LALCA0	LALCM	LALEN		LAL2	LAL1	LAL0	LHS3	LHS2	LHS1	LHS0		LW2	LW1	LW0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				LFD	LFC	LFB	LFA	LF7	LF6	LF5	LF4	LF3	LF2	LF1	LF0

LALCA1, LALCA0: Line alpha blend attribute control

(0, 0: Alpha blend attribute control OFF)

(0, 1:Only character alpha blend A attributes are enabled)

(1, 0: Only character alpha blend B attributes are enabled)

(1, 1: Character alpha blend A and B attributes both enabled)

LALCM: Line alpha blend control scope control

(0: Control scope A [Regions other than character and trimming])

(1: Control scope B [Entire line region])

LALEN: Line alpha blend output control

(0: OFF, 1: ON)

LAL2 to LAL0: Line alpha blend quantity

(7 to 0 [7: Transparent <-> 0: Opaque)

LHS3 to LHS0: Line character vertical size control

(2 to 32 dots in units of 2 dots)

LW2 to LW0: Line spacing control

(0 to 14 dots in units of 2 dots)

LFD, LFC: Trimming output control

(0, 0: All OFF)

(0, 1: Trimming only on characters with no character background)

(1, 0: Trimming only on characters with no character background, solid background, or background character)

(1, 1: Trimming output ON)

LFB, LFA: Trimming control

(0, 0: Trimming OFF)

(0, 1: Full perimeter trimming)

(1, 0: Right trimming)

(1, 1: Shadow trimming)

LF7 to LF0: Trimming color

(256 colors)

- Function

  This command sets the line control data. The line VRAM settings are updated and the changes reflected on the screen when line control data set 2 (command 4) is executed.

- Additional Information
  - The shadow trimming style setting is configured using shadow trimming style control (bits FM1 and FM0) of the screen output control (command 5-0).
  - For lines where the character vertical size is set to 2 dots, line spacing control is prohibited.

    Either set (LW2, LW1, LW0) to (0, 0, 0), or set line spacing upper control LWUEN to 0 and line spacing lower control LWDEN to 0 in line control data set 2 (command 4).

Command 4 (line control data set 2)

Address: 4010_H

- Format (write-only, 32-bit access only)

  [MOSD_LDS2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						LSW1	LSW0				LDS	LGY1	LGY0	LGX1	LGX0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LWUEN	LWDEN	LMUEN	LMDEN		LE	LM1	LM0	LB7	LB6	LB5	LB4	LB3	LB2	LB1	LB0

LSW1, LSW0: Character shaded background frame width control

(0, 0: 1 dot)

(0, 1: 2 dots)

(1, 0: 3 dots)

(1, 1: 4 dots)

LDS: Line character output control

(0: OFF, 1: ON)

LGY1, LGY0: Line vertical enlargement control

(0, 0: Normal)

(0, 1: Double-height)

(1, 0: Setting prohibited)

(1, 1: Quadruple-height)

LGX1, LGX0: Line horizontal enlargement control

(0, 0: Normal)

(0, 1: Double-width)

(1, 0: Setting prohibited)

(1, 1: Quadruple-width)

LWUEN: Line spacing upper control

(0: Line spacing upper OFF)

(1: Line spacing upper ON)

LWDEN: Line spacing lower control

(0: Line spacing lower OFF)

(1: Line spacing lower ON)

LMUEN: Line shaded background frame top display control

(0: Line shaded background frame top OFF)

(1: Line shaded background frame top ON)

LMDEN: Line shaded background frame bottom display control

(0: Line shaded background frame bottom OFF)

(1: Line shaded background frame bottom ON)

LE: Character background extension control

(0: Normal, 1: Extended)

LM1, LM0: Line background control

(0, 0: OFF)

(0, 1: Solid display)

(1, 0: shaded concaved (solid) display)

(1, 1: shaded convexed (solid) display)

LB7 to LB0: Line background color

(256 colors)

- Function

  This command writes the data specified in this command together with the line control data configured by line control data set 1 (command 3) to the line VRAM at the line address specified by the VRAM write address set (command 0).

Notes:

- Because the contents of VRAM are undefined when the power is turned ON, be sure to set all of the VRAM data before starting the display.
- Issuing this command does not automatically increment the VRAM write address. The VRAM write address set (command 0) needs to be set for each line when making line control settings.
- The frame color setting of the shaded concaved and convexed display in the line background control is the setting value of bits B0H7 to B0H0 and B0S7 to B0S0 in the shaded background frame color control (command 15-0). Furthermore, the line shaded frame width can only be 2 dots regardless of the setting of the character shaded background frame width control (LSW1 and LSW0).

30.4.5 Screen Output Control (Command 5-0)

Command 5-0 controls the screen display output.

Command 5-0 (screen output control)

Address: 4014_H

- Format (write-only, 32-bit access only)

  [MOSD_SCOC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						FM1	FM0	BT1	BT0	BD1	BD0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SAL CC				WE	SDS	UDS	PDS	DSP				

FM1: Shadow trimming style control

(0: Lower-right)

(1: Lower-right + right)

FM0: Trimming dot count control

(0: 1 dot)

(1: 2 dots)

BT1, BT0: Blink period control

(0, 0: 16 Vsync)

(0, 1: 32 Vsync)

(1, 0: 48 Vsync)

(1, 1: 64 Vsync)

BD1, BD0: Blink duty ratio control

(0, 0: ON : OFF = 1 : 0 - Always displayed)

(0, 1: ON : OFF = 1 : 1)

(1, 0: ON : OFF = 1 : 3)

(1, 1: ON : OFF = 3 : 1)

SALCC: Alpha blend attribute disabled quantity control

(0: Alpha blend quantity fixed at 0 [Opaque])

(1: Alpha blend quantity fixed at 7 [Transparent])

WE: Window function control

(0: OFF, 1: ON)

SDS: Sprite character output control

(0: OFF, 1: ON)

PDS: Screen background character output control

(0: OFF, 1: ON)

UDS: Screen background output control

(0: OFF, 1: ON)

DSP: Display output control

(Controls characters + character background + line background)

(0: OFF, 1: ON)

- Function

  This command controls the screen display output.

- Additional Information
  - The initial values of SDS, UDS, PDS, and DSP after reset are "0".
  - The blink period control and blink duty ratio control perform control over characters and character backgrounds that have been specified as blinking ((MBL, MBB) set to (0, 1), (1, 0) or (1, 1)) in character data set 1 (command 1). This command also controls sprite characters if sprite character blinking has been specified (SBL set to 1) in sprite character control 1 (command 9-0).
  - Window function control has no effect on sprite characters.

30.4.6 Screen Display Position Control (Command 5-1)

Command 5-1 controls the horizontal and vertical display position of the screen.

Command 5-1 (screen display position control)

Address: 4018_H

- Format (write-only, 32-bit access only)
  [MOSD_HVDP]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

X10 to X0: Horizontal display position control
(0 to 2047 in units of 1 dot)

Y10 to Y0: Vertical display position control
(0 to 2047 in units of 1 dot)

- Function
  This command controls the horizontal display position and vertical display position of the main screen.

30.4.7 Transparent Color Control (Command 6-0)

Command 6-0 controls the transparent color.

Command 6-0 (transparent color control)

Address: $401C_H$

- Format (write-only, 32-bit access only)

  [MOSD_TSBC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							TCC	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0

TCC: Transparent color control
(0: OFF, 1: ON)

TC7 to TC0: Transparent color code
(256 colors)

- Function

  This command controls the transparent color.

- Additional Information
  - Arbitrary color display areas can be not displayed by setting the transparent color code (TC7 to TC0) to an arbitrary color code and setting the transparent color control to ON (TCC=1). The arbitrary color display areas perform display output of the lower layer.

Note: The alpha blend quantity signal from pins VOA2 to VOA0 and the non-displayed areas due to the transparency setting output the alpha blend quantity signal to the lower layer display output.

30.4.8 Graphics Color Control (Command 6-1)

Command 6-1 is a command to display an arbitrarily specified color in graphics characters replaced by the character color or trimming color.

Command 6-1 (graphics color control)

Address: 4020_H

- Format (write-only, 32-bit access only)

  [MOSD_GRCC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							GFC	GF7	GF6	GF5	GF4	GF3	GF2	GF1	GF0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							GCC	GC7	GC6	GC5	GC4	GC3	GC2	GC1	GC0

GFC: Graphics color/trimming color replacement control
(0: OFF, 1: ON)

GCC: Graphics color/character color replacement control
(0: OFF, 1: ON)

GF7 to GF0: Color code replaced by trimming color
(256 colors)

GC7 to GC0: Color code replaced by character color
(256 colors)

- Function

  Displays an arbitrary color within the graphics characters replaced by the character color or trimming color.

- Additional Information
  - An arbitrary color within the graphics characters (the color specified by bits GF7 to GF0) is displayed replaced by the trimming color (LF7 to LF0) configured by line control data set 1 (command 3) by setting GFC to 1.
  - An arbitrary color within the graphics characters (the color specified by bits GC7 to GC0) is displayed replaced by the character color (MC7 to MC0) configured by character data set 1 (command 1) by setting GCC to 1.
  - If graphics color/trimming color replacement control is ON (the GFC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the graphics color/trimming color replacement color code and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the trimming color replacement takes precedence.
  - If graphics color/trimming color replacement control is ON (the GFC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the replacement trimming color code (trimming color (LF7 to LF0) of line control data set 1 (command 3)) and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the color becomes transparent and the lower layer color is displayed.
  - If graphics color/character color replacement control is ON (the GCC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the graphics color/character color replacement color code and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the character color replacement takes precedence.
  - If graphics color/character color replacement control is ON (the GCC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the replacement character color code (character color (MC7 to MC0) of character data set 1 (command 1)) and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the color becomes transparent and the lower layer color is displayed.
  - If graphics color/character color replacement control is ON (the GCC bit is set to 1) and graphics color/trimming color replacement control is ON (the GFC bit is set to 1), set the value of the character color replacement color code (bits GC7 to GC0) and the value of the trimming color replacement color code (bits GF7 to GF0) to different color codes.

Note: The settings of this command only apply to colors displayed in graphics characters on the main screen. This has no effect on sprite characters or screen background character dot colors.

30.4.9 Screen Background Character Control (Command 7)

Command 7 controls the screen background character.

Command 7 (screen background character control)

Address: 4024_H

- Format (write-only, 32-bit access only)

  [MOSD_SBCC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					PH2	PH1	PH0							PD1	PD0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		PM13	PM12	PM11	PM10	PM9	PM8	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0

PH2 to PH0: Screen background character vertical size control

(18 to 32 dots in units of 2 dots)

PM13 to PM0: Screen background character code

(0000_H to $3FF8_H$: 2048 character types)

PD1, PD0: Screen background character configuration control

(0, 0: 1 character)

(0, 1: 2 characters horizontally)

(1, 0: 2 characters vertically)

(1, 1: 2 × 2 characters (vertical × horizontal))

- Function

  This command controls the screen background character.

- Additional Information

  - The screen background character is displayed by setting the screen background character output control to ON (PDS=1) in the screen output control (command 5-0).

Notes:

- Screen background characters can only display graphics characters.
- The graphics color of screen background characters cannot be replaced by the graphics color/trimming color replacement control or graphics color/character color replacement control in the graphics color control (command 6-1).
- When stopping the display control by the screen background character operation control, set the screen background character output control to OFF (PDS=0) in the screen output control (command 5-0).

30.4.10 Screen Background Control (Command 8-0)

Command 8-0 controls the screen background color and window screen.

Command 8-0 (screen background control)

Address: 4028_H

- Format (write-only, 32-bit access only)

  [MOSD_SCBC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										UWBV	UWBH			UW1	UW0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			UALEN		UAL2	UAL1	UAL0	U7	U6	U5	U4	U3	U2	U1	U0

UWBV, UWBH: Window border width control

(0, 0: 2 dots vertically, 2 dots horizontally)

(0, 1: 2 dots vertically, 4 dots horizontally)

(1, 0: 4 dots vertically, 2 dots horizontally)

(1, 1: 4 dots vertically, 4 dots horizontally)

UALEN: Screen background alpha blend control

(0: OFF, 1: ON)

U7 to U0: Screen background color

(256 colors)

UW1, UW0: Window background mode control

(0, 0: Background inside window)

(0, 1: Background outside window)

(1, 0: Background inside and outside window)

(1, 1: Window border background)

UAL2 to UAL0: Screen background alpha blend quantity

(7 to 0 [7: Transparent <-> 0: Opaque])

- Function

  This command controls the screen background.

- Additional Information
  - The screen background color is displayed by setting the screen background output control to ON (UDS=1) in the screen output control (command 5-0).
  - Functions related to the window (UWBV, UWBH, UW1, UW0) are enabled by setting the window function control to ON (WE=1) in the screen output control (command 5-0).

30.4.11 Window Period Control (Command 8-1, Command 8-2)

Command 8-1 and command 8-2 control the window period.

Command 8-1 (window period control 1)

Address: $402C_H$

- Format (write-only, 32-bit access only)

  [MOSD_WPC1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					WYE 10	WYE 9	WYE 8	WYE 7	WYE 6	WYE 5	WYE 4	WYE 3	WYE 2	WYE 1	WYE 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					WYS 10	WYS 9	WYS 8	WYS 7	WYS 6	WYS 5	WYS 4	WYS 3	WYS 2	WYS 1	WYS 0

WYE10 to WYE0: Window vertical period end control

(0 to 2047 in units of 1 dot)

WYS10 to WYS0: Window vertical period start control

(0 to 2047 in units of 1 dot)

- Function

  This command controls the display start and end timing in the vertical direction in the window period.

- Additional Information
  - This command is enabled by setting the window function control to ON (WE=1) in the screen output control (command 5-0).

Command 8-2 (window period control 2)

Address: 4030_H

- Format (write-only, 32-bit access only)

  [MOSD_WPC2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				WXE 11	WXE 10	WXE 9	WXE 8	WXE 7	WXE 6	WXE 5	WXE 4	WXE 3	WXE 2	WXE 1	WXE 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				WXS 11	WXS 10	WXS 9	WXS 8	WXS 7	WXS 6	WXS 5	WXS 4	WXS 3	WXS 2	WXS 1	WXS 0

WXE11 to WXE0: Window horizontal period end control

(0 to 4095 in units of 1 dot)

WXS11 to WXS0: Window horizontal period start control

(0 to 4095 in units of 1 dot)

- Function

  This command controls the display start and end timing in the horizontal direction in the window period.

- Additional Information

  This command is enabled by setting the window function control to ON (WE=1) in the screen output control (command 5-0).

30.4.12 Sprite Character Control (Command 9-0, Command 9-1)

Command 9-0 and command 9-1 control the sprite character.

Command 9-0 (sprite character control 1)

Address: 4034_H

- Format (write-only, 32-bit access only)

  [MOSD_SPC1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			SBL		SH2	SH1	SH0							SD1	SD0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		SM13	SM12	SM11	SM10	SM9	SM8	SM7	SM6	SM5	SM4	SM3	SM2	SM1	SM0

SBL: Sprite character blink control

(0: OFF, 1: ON)

SD1, SD0: Sprite character configuration control

(0, 0: 1 character)

(0, 1: 2 characters horizontally)

(1, 0: 2 characters vertically)

(1, 1: 2 × 2 characters (vertical × horizontal))

SH2 to SH0: Sprite character vertical size control

(18 to 32 dots in units of 2 dots)

SM13 to SM0: Sprite character code

(0000_H to $3FF8_H$: 2048 character types)

- Function

  This command controls the sprite character display.

- Additional Information
  - The sprite character is displayed by setting the sprite character output control to ON (SDS=1) in the screen output control (command 5-0).
  - The blink period and blink duty ratio of the sprite character are controlled by the settings of bits BT1, BT0, BD1, and BD0 of the screen output control (command 5-0).

Notes:

- Sprite characters can only display graphics characters.
- The graphics color of sprite characters cannot be replaced by the graphics color/trimming color replacement control or graphics color/character color replacement control in the graphics color control (command 6-1).
- When stopping the display control by the sprite operation control, set the sprite character output control to OFF (SDS=0) in the screen output control (command 5-0).

Command 9-1 (sprite character control 2)

Address: 4038_H

- Format (write-only, 32-bit access only)

  [MOSD_SPC2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				SX11	SX10	SX9	SX8	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SY10	SY9	SY8	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

SX11 to SX0: Sprite character horizontal display position control

(0 to 4095 in units of 1 dot)

SY10 to SY0: Sprite character vertical display position control

(0 to 2047 in units of 1 dot)

- Function

  This command controls the horizontal and vertical display position of the sprite character.

- Additional Information

  See the additional information in sprite character control 1 (command 9-0).

30.4.13 Synchronization Control (Command 11)

Command 11 controls the synchronization display.

Command 11 (synchronization control)

Address: $403C_H$

- Format (write-only, 32-bit access only)

  [MOSD_SYNC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										VOF5	VOF4	VOF3	VOF2	VOF1	VOF0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									IN		FC				

VOF5 to VOF0: Vertical display start offset

(0 to 63 Hsync)

IN: Synchronization control

(0: Interlaced)

(1: Non-interlaced (progressive))

FC: Field correction control

(0: Not corrected, 1: Corrected)

- Function

  This command configures the vertical display start offset position and controls interlaced/non-interlaced display.

- Additional Information
  - Field correction control (FC) only has an effect during interlaced display (IN=0).

30.4.14 Character Background Character Code Set (Command 12-0 to Command 12-7)

Commands 12-0 to 12-7 configure the character background character codes.

Command 12-0 to command 12-7 (character background character code set)

Address: See the following table

- Format (write-only, 32-bit access only)

  [MOSD_CBC0 to MOSD_CBC7]

Address	Command No.	Data			
		31 to 30	29 to 16	15 to 14	13 to 0
4040	12-0	0, 0	BM1C13 to BM1C0	0, 0	BM0C13 to BM0C0
4044	12-1	0, 0	BM3C13 to BM3C0	0, 0	BM2C13 to BM2C0
4048	12-2	0, 0	BM5C13 to BM5C0	0, 0	BM4C13 to BM4C0
404C	12-3	0, 0	BM7C13 to BM7C0	0, 0	BM6C13 to BM6C0
4050	12-4	0, 0	BM9C13 to BM9C0	0, 0	BM8C13 to BM8C0
4054	12-5	0, 0	BMBC13 to BMBC0	0, 0	BMAC13 to BMAC0
4058	12-6	0, 0	BMDC13 to BMDC0	0, 0	BMCC13 to BMCC0
405C	12-7	0, 0	BMFC13 to BMFC0	0, 0	BMEC13 to BMEC0

BMxC13 to BMxC0: Character background character code 14 bits (0000_H to $3FFF_H$: 16384 characters)

(x => Character background character type: 0 to F [16 types])

- Function

  This command configures the character codes for the character background characters.

- Additional Information
  - The character background character codes (BMxC13 to BMxC0) configure the actual character background character codes that correspond to the 16 type settings of MBM3 to MBM0 (character background character type control) in command 2 (character data set 2).

30.4.15 I/O Pin Control (Command 13)

Command 13 controls the I/O pins.

Command 13 (I/O pin control)

Address: 4060_H

- Format (write-only, 32-bit access only)

  [MOSD_IOTC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							VHE	HE					IHX	IVX	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													OHX	OBX	OCX

VHE: Vertical sync detection HSYNC edge selection

(0: Leading, 1: Trailing)

IHX: Horizontal sync signal input logic control

(0: Negative logic, 1: Positive logic)

OHX: Alpha blend quantity signal output logic control

(0: Positive logic, 1: Negative logic)

OBX: Display output period signal output logic control

(0: Positive logic, 1: Negative logic)

OCX: Display color signal output logic control

(0: Positive logic, 1: Negative logic)

HE: Horizontal sync operation edge selection

(0: Trailing, 1: Leading)

IVX: Vertical sync signal input logic control

(0: Negative logic, 1: Positive logic)

- Function

  This command controls the I/O pins.

- Additional Information
  - The initial values of OHX, OBX, and OCX after reset are "0".
  - The problem of shaking in the vertical direction can be prevented by the input phase timing of the vertical sync signal and horizontal sync signal by configuring the vertical sync detection HSYNC edge selection (VHE). See "30.3.2.1 Vertical Sync Control" for details.

Notes:

- "OCX: Display color signal output logic control" inversely controls OSDC analog output level also.
- Basically, the horizontal sync operation edge selection should be set to the leading edge operation (HE=1).

  In case of the trailing edge is set (HE=0), the sprite character, screen background character, and character background character cannot be used.

30.4.16 Display Period Control (Command 14-0, Command 14-1)

Command 14-0 and command 14-1 control the display period.

Command 14-0 (display period control 1)

Address: 4064_H

- Format (write-only, 32-bit access only)
  [MOSD_CDP1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					DYE10	DYE9	DYE 8	DYE7	DYE6	DYE5	DYE4	DYE3	DYE2	DYE1	DYE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					DYS10	DYS9	DYS8	DYS7	DYS6	DYS5	DYS4	DYS3	DYS2	DYS1	DYS0

DYE10 to DYE0: Vertical direction display end control
(0 to 2047 in units of 1 Hsync)
DYS10 to DYS0: Vertical direction display start control
(0 to 2047 in units of 1 Hsync)

- Function
  This command controls the display start and end timing in the vertical direction.
  Note: Always set this command as it is required for the blanking processing.

Command 14-1 (display period control 2)

Address: 4068_H

- Format (write-only, 32-bit access only)
  [MOSD_CDP2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				DXE11	DXE10	DXE9	DXE8	DXE7	DXE6	DXE5	DXE4	DXE3	DXE2	DXE1	DXE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DXS11	DXS10	DXS9	DXS8	DXS7	DXS6	DXS5	DXS4	DXS3	DXS2	DXS1	DXS0

DXE11 to DXE0: Horizontal direction display end control
(0 to 4095 in units of 1 dot)
DXS11 to DXS0: Horizontal direction display start control
(0 to 4095 in units of 1 dot)

- Function
  This command controls the display start and end timing in the horizontal direction.
  Note: Always set this command as it is required for the blanking processing.

30.4.17 Interrupt Control (Command 15)

Command 15 controls the OSDC interrupts.

Command 15 (interrupt control)

Address: $406C_H$

- Format (write/read, 32-bit access only)

  [MOSD_INTC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					FIF	LIF	VIF						FIE	LIE	VIE

FIF: VRAM fill end detection flag

(0: VRAM fill end not detected, 1: VRAM fill end detected)

LIF: Line display end detection flag

(0: Line display end not detected, 1: Line display end detected)

VIF: VSYNC detection flag

(0: VSYNC not detected, 1: VSYNC detected)

FIE: VRAM fill end detection interrupt control

(0: Interrupt disabled, 1: Interrupt enabled)

LIE: Line display end detection interrupt control

(0: Interrupt disabled, 1: Interrupt enabled)

VIE: VSYNC detection interrupt control

(0: Interrupt disabled, 1: Interrupt enabled)

- Function

  This command controls the interrupts.

  In order to enable each of the interrupts and interrupt detection flags, each interrupt control (FIE, LIE, VIE) needs to be set to "1".

  FIF, LIF, and VIF are then set to "1" when the interrupt source occurs. (Writing "1" is ignored.)

  Once a flag has been set, the flag can be cleared by writing "0".

- Additional Information
  - The initial values of FIF, LIF, VIF, FIE, LIE, and VIE after reset are "0".
  - The read values of empty bits are always "0".
  - In order to allow LIF (line end detection flag) interrupts to be generated, the line spacing bottom control "LWDEN" needs to be set to "1" in command 4 (line control data set 2) for the corresponding line.

30.4.18 Shaded Background Frame Color Control (Command 15-0 to Command 15-3)

Commands 15-0 to 15-3 control the frame color of shaded backgrounds.

Command 15-0 to command 15-3 (shaded background frame color control)

Address: See the following table

- Format (write-only, 32-bit access only)

  [MOSD_SBC0 to MOSD_SBC3]

Address	Command No.	Data			
		31 to 24	23 to 16	15 to 8	7 to 0
4070	15-0	B1H7 to B1H0	B1S7 to B1S0	B0H7 to B0H0	B0S7 to B0S0
4074	15-1	B3H7 to B3H0	B3S7 to B3S0	B2H7 to B2H0	B2S7 to B2S0
4078	15-2	B5H7 to B5H0	B5S7 to B5S0	B4H7 to B4H0	B4S7 to B4S0
407C	15-3	B7H7 to B7H0	B7S7 to B7S0	B6H7 to B6H0	B6S7 to B6S0

BxH7 to BxH0: Shaded background frame highlight color

(256 colors)

BxS7 to BxS0: Shaded background frame shadow color

(256 colors)

(x => Character shaded background frame color type [MSC2, MSC1, MSC0] setting value: Color type 0 to 7 [8 combinations of highlight color and shadow color])

- Function

  This command controls the shaded background frame color.

- Additional Information
  - This command sets the shaded background frame color of the character shaded background frame color type selection (MSC2, MS1, MSC0 => 0 to 7 types) for characters where shaded character background is specified (MM1 is set to 1) in character data set 1 (command 1).
  - On lines where shaded line background is specified (LM1 is set to 1) in line control data set 2 (command 4), the shaded background frame color specified in B0H7 to B0H0 and B0S7 to B0S0 is used.
  - Table 30-73 shows the display parts of the shaded background frame highlight color and shadow color.

Table 30-73. Shaded Background Frame Highlight Color and Shadow Color Display Parts

	Character background		Line background	
	Shaded background concaved display	Shaded background convexed display	Shaded background concaved display	Shaded background convexed display
Shaded background frame highlight color	Bottom edge, right edge	Top edge, left edge	Bottom edge	Top edge
Shaded background frame shadow color	Top edge, left edge	Bottom edge, right edge	Top edge	Bottom edge

Note: The shaded background frame color type of line backgrounds uses the B0H7 to B0H0 and B0S7 to B0S0 setting values, which are shared with the character shaded background frame color type for MSC2=0, MSC1=0, and MSC0=0 (type 0) for character shaded backgrounds.

30.4.19 Palette Control (Command 16-0 to Command 16-127)

Commands 16-0 to 16-127 control the palette values.

Command 16-0 to command 16-127 (palette control)

Address: See the following table

- Format (write-only, 32-bit access only)

  [MOSD_PL0 to MOSD_PL127]

Address	Command No.	Data					
		31 to 27	26 to 21	20 to 16	15 to 11	10 to 5	4 to 0
4200	16-0	PL01R4 to PL01R0	PL01G5 to PL01G0	PL01B4 to PL01B0	PL00R4 to PL00R0	PL00G5 to PL00G0	PL00B4 to PL00B0
4204	16-1	PL03R4 to PL03R0	PL03G5 to PL03G0	PL03B4 to PL03B0	PL02R4 to PL02R0	PL02G5 to PL02G0	PL02B4 to PL02B0
4208	16-2	PL05R4 to PL05R0	PL05G5 to PL05G0	PL05B4 to PL05B0	PL04R4 to PL04R0	PL04G5 to PL04G0	PL04B4 to PL04B0
420C	16-3	PL07R4 to PL07R0	PL07G5 to PL07G0	PL07B4 to PL07B0	PL06R4 to PL06R0	PL06G5 to PL06G0	PL06B4 to PL06B0
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
42FC	16-63	PL7FR4 to PL7FR0	PL7FG5 to PL7FG0	PL7FB4 to PL7FB0	PL7ER4 to PL7ER0	PL7EG5 to PL7EG0	PL7EB4 to PL7EB0
4300	16-64	PL81R4 to PL81R0	PL81G5 to PL81G0	PL81B4 to PL81B0	PL80R4 to PL80R0	PL80G5 to PL80G0	PL80B4 to PL80B0
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
43F0	16-124	PLF9R4 to PLF9R0	PLF9G5 to PLF9G0	PLF9B4 to PLF9B0	PLF8R4 to PLF8R0	PLF8G5 to PLF8G0	PLF8B4 to PLF8B0
43F4	16-125	PLFBR4 to PLFBR0	PLFBG5 to PLF-BG0	PLFBB4 to PLFBB0	PLFAR4 to PLFAR0	PLFAG5 to PLFAG0	PLFAB4 to PLFAB0
43F8	16-126	PLFDR4 to PLF-DR0	PLFDG5 to PLF-DG0	PLFDB4 to PLFDB0	PLFCR4 to PLF-CR0	PLFCG5 to PLF-CG0	PLFCB4 to PLFCB0
43FC	16-127	PLFFR4 to PLFFR0	PLFFG5 to PLFFG0	PLFFB4 to PLFFB0	PLFER4 to PLFER0	PLFEG5 to PLFEG0	PLFEB4 to PLFEB0

PLxxR4 to PLxxR0: Red color signal palette value 5 bits (xx => 8-bit color code: 00 to FF)

PLxxG5 to PLxxG0: Green color signal palette value 6 bits (xx => 8-bit color code: 00 to FF)

PLxxB4 to PLxxB0: Blue color signal palette value 5 bits (xx => 8-bit color code: 00 to FF)

- Function

  The command controls the palette values.

- Additional Information
  - The RGB color settings of the OSDC 8-bit color codes "00" to "FF" are configured by setting the palette values in "Command 16-0" to "Command 16-127".
  - The following types of color codes are used.
    Characters (MC7 to MC0)
    Character background color (MB7 to MB0)
    Trimming color (LF7 to LF0)
    Line background color (LB7 to LB0)
    Shaded background frame color (BH7 to BH0, BS7 to BS0)
    Graphics color control (GF7 to GF0, GC7 to GC0)
    Screen background color (U7 to U0)
    Graphics colors

30.4.20 Operation Control (Command 17)

Command 17 controls the initial operation of the OSDC.

Command 17 (OSDC operation control)

Address: 4400_H

- Format (write-only, 32-bit access only)
  [MOSD_OSDC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										SUBEN	OSDEN			DGO	ANO

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							OSDR				DCK				DPD

SUBEN: Sub screen operation enable
(0: Sub operation disabled, 1: Sub operation enabled)
OSDEN: OSDC enable
(0: OSDC disabled, 1: OSDC enabled)
DGO: Digital system output control
(0: Output OFF, 1: Output ON)
ANO: Analog system output control
(0: Output OFF, 1: Output ON)
OSDR: OSDC reset
(0: Inactive [reset cleared], 1: Active [reset])
DCK: Input dot clock selection control
(0: External dot clock input, 1: Internal PLL generated dot clock input)
DPD: DAC enable
(0: OSDC DAC access OFF, 1: OSDC DAC access ON)

- Function

  This command controls access to and operation of OSDC related resources.

- Additional Information
  - The initial values of SUBEN, OSDEN, DGO, ANO, OSDDR, DCK, and DPD after reset are "0".
  - Commands for controlling the main screen display should be issued after setting OSDC Enable (OSDEN) to ON. Furthermore, if the OSDC Reset (OSDR) is active, it needs to be set to inactive before setting OSDC Enable (OSDEN) to ON.
  - In order to issue sub screen display control commands, it is necessary to set the SUBEN (sub operation enable) bit to 1 (sub operation enabled). (Do not issue SUB_OSDC display control commands while the SUBEN bit is 0.)
  - OSDC enable and sub operation enable need to be controlled while a normal clock is input externally or while the internal PLL is oscillating normally.
  - When reset using the OSDR bit, the value of this register is not initialized.
  - When either the analog system output or digital system output is in the output on state, the VOB and VOA signals which are common to both output system are output.
  - In order to turn both the VOB and VOA outputs OFF (low output when using positive logic), both the analog system output and digital system output need to be set to the output OFF state.
  - When the OSDEN bit is "0" (OSDC is disabled), the status of the OSDC active control is as follows:
    1. The access to the font memory is disabled.
    2. Do not access to the following commands.

       Command 0: VRAM Write Address Set

       Command 1: Character Data Set 1

       Command 2: Character Data Set 2

       Command 3: Line Control Data Set 1

       Command 4: Line Control Data Set 2

       Command 15: Interrupt Control

       Command 16-0 to 127: Palette Control
    3. The external pin is not operated. However, it operates internally, as long as the dot clock is supplied.

30.4.21 PLL Clock Control (Command 18)

Command 18 controls the PLL clock.

Command 18 (PLL clock control)

Address: 4404_H

- Format (write-only, 32-bit access only)

  [MOSD_PLLC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DHRSA	DCO	DKA11	DKA10	DKA9	DKA8	DKA7	DKA6	DKA5	DKA4	DKA3	DKA2	DKA1	DKA0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCOA	VSLA2	VSLA1	VSLA0			CPEA	PDEA				DAP4	DAP3	DAP2	DAP1	DAP0

DHRSA: PLL phase comparison edge selection

(0: Horizontal sync signal rising edge)

(1: Horizontal sync signal falling edge)

DCO: Dot clock output pin control

(0: OFF, 1: ON)

DKA11 to DKA0: Clock frequency division

(1 clock unit: Min = 129 clocks, Max = 4096 clocks)

VCOA: PLL VCO oscillator control

(0: OFF, 1: ON)

VSLA2 to VSLA0: PLL VCO selection control

(0, 0, 0: VCO0)

(0, 0, 1: VCO0)

(0, 1, 0: VCO1)

(0, 1, 1: VCO1)

(Other settings are prohibited)

CPEA: PLL charge pump control

(0: OFF, 1: ON)

PDEA: PLL phase comparator control

(0: OFF, 1: ON)

DAP4 to DAP0: PLL clock frequency division

(Frequency division units: Min = Divide by 1, Max = Divide by 62)

- Function

  This command controls the PLL clock.

30.5 Display Control Commands (Sub Operation)

This section explains the OSDC sub screen display control commands.

Display control commands (sub operation)

30.5.1 List of Sub Screen Display and Control Commands
30.5.2 VRAM Write Address Set (Command 0)
30.5.3 Character Data Set (Command 1, Command 2)
30.5.4 Line Control Data Set (Command 3, Command 4)
30.5.5 Screen Output Control (Command 5-0)
30.5.6 Screen Display Position Control (Command 5-1)
30.5.7 Transparent Color Control (Command 6-0)
30.5.8 Graphics Color Control (Command 6-1)
30.5.9 Screen Background Control (Command 8-0)
30.5.10 Window Period Control (Command 8-1, Command 8-2)
30.5.11 Sprite Character Control (Command 9-0, Command 9-1)
30.5.12 Interrupt Control (Command 15)
30.5.13 Shaded Background Frame Color Control (Command 15-0 to Command 15-3)

30.5.1 List of Sub Screen Display and Control Commands

This section lists the sub screen display control commands. Table 30-74. shows a list of display control commands.

List of display control commands

Table 30-74. List of Display Control Commands

Address lower 16 bits (HEX)	SUB_OSDC command No.	Data																														Function
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
41 00	0																FL				AY4	AY3	AY2	AY1	AY0			AX5	AX4	AX3	AX2	AX1	AX0	VRAM write address set [SOSD_VADR]
41 04	1		MSC2	MSC1	MSC0	MIT	MUL	MBL	MBB	MU	MD		MS1	MS0	MM2	MM1	MM0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0	Character data set 1 [SOSD_CDS1]
41 08	2									BM3	BM2	BM1	BM0		MA	MR	MG			M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0	Character data set 2 [SOSD_CDS2]
41 0C	3	LALCA1	LALCA0	LALCM	LALEN		LAL2	LAL1	LAL0	LHS3	LHS2	LHS1	LHS0		LW2	LW1	LW0					LFD	LFC	LFB	LFA	LF7	LF6	LF5	LF4	LF3	LF2	LF1	LF0	Line control data set 1 [SOSD_LDS1]
41 10	4							LSW1	LSW0				LDS	LGY1	LGY0	LGX1	LGX0	LWUEN	LWDEN	LMUEN	LMDEN		LE	LM1	LM0	LB7	LB6	LB5	LB4	LB3	LB2	LB1	LB0	Line control data set 2 [SOSD_LDS2]
41 14	5-0							FM1	FM0	BT1	BT0	BD1	BD0								SALCC				WE	SDS	UDS		DSP				MCC	Screen output control [SOSD_SCOC]
41 18	5-1						X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0						Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Screen display position control [SOSD_HVDP]
41 1C	6-0																								TCC	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	Transparent color control [SOSD_TSBC]
41 20	6-1								GFC	GF7	GF6	GF5	GF4	GF3	GF2	GF1	GF0								GCC	GC7	GC6	GC5	GC4	GC3	GC2	GC1	GC0	Graphics color control [SOSD_GRCC]
-																																		-
41 28	8-0											UWBV	UWBH			UW1	UW0				UALEN		UAL2	UAL1	UAL0	U7	U6	U5	U4	U3	U2	U1	U0	Screen background control [SOSD_SCBC]
41 2C	8-1						WYE10	WYE9	WYE8	WYE7	WYE6	WYE5	WYE4	WYE3	WYE2	WYE1	WYE0						WYS10	WYS9	WYS8	WYS7	WYS6	WYS5	WYS4	WYS3	WYS2	WYS1	WYS0	Window period control 1 [SOSD_WPC1]
41 30	8-2					WXE11	WXE10	WXE9	WXE8	WXE7	WXE6	WXE5	WXE4	WXE3	WXE2	WXE1	WXE0					WXS11	WXS10	WXS9	WXS8	WXS7	WXS6	WXS5	WXS4	WXS3	WXS2	WXS1	WXS0	Window period control 2 [SOSD_WPC2]
41 34	9-0				SBL		SH2	SH1	SH0							SD1	SD0			SM13	SM12	SM11	SM10	SM9	SM8	SM7	SM6	SM5	SM4	SM3	SM2	SM1	SM0	Sprite character control 1 [SOSD_SPC1]
41 38	9-1					SX11	SX10	SX9	SX8	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0						SY10	SY9	SY8	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0	Sprite character control 2 [SOSD_SPC2]
-																																		-
41 6C	15																						FIF	LIF	(VIF)						FIE	LIE	(VIE)	Interrupt control [SOSD_INTC]
41 70	15-0	B1H7	B1H6	B1H5	B1H4	B1H3	B1H2	B1H1	B1H0	B1S7	B1S6	B1S5	B1S4	B1S3	B1S2	B1S1	B1S0	B0H7	B0H6	B0H5	B0H4	B0H3	B0H2	B0H1	B0H0	B0S7	B0S6	B0S5	B0S4	B0S3	B0S2	B0S1	B0S0	Shaded background frame color control [SOSD_SBC0 to SOSD_SBC3]
41 74	15-1	B3H7	B3H6	B3H5	B3H4	B3H3	B3H2	B3H1	B3H0	B3S7	B3S6	B3S5	B3S4	B3S3	B3S2	B3S1	B3S0	B2H7	B2H6	B2H5	B2H4	B2H3	B2H2	B2H1	B2H0	B2S7	B2S6	B2S5	B2S4	B2S3	B2S2	B2S1	B2S0	
41 78	15-2	B5H7	B5H6	B5H5	B5H4	B5H3	B5H2	B5H1	B5H0	B5S7	B5S6	B5S5	B5S4	B5S3	B5S2	B5S1	B5S0	B4H7	B4H6	B4H5	B4H4	B4H3	B4H2	B4H1	B4H0	B4S7	B4S6	B4S5	B4S4	B4S3	B4S2	B4S1	B4S0	
41 7C	15-3	B7H7	B7H6	B7H5	B7H4	B7H3	B7H2	B7H1	B7H0	B7S7	B7S6	B7S5	B7S4	B7S3	B7S2	B7S1	B7S0	B6H7	B6H6	B6H5	B6H4	B6H3	B6H2	B6H1	B6H0	B6S7	B6S6	B6S5	B6S4	B6S3	B6S2	B6S1	B6S0	

Notes:

- Each of the SDS, UDS, DPS, and MCC bits in the screen output control and the FIF, LIF, (VIF), FIE, LIE, and (VIE) bits in the interrupt control are initialized to "0" when a reset is input. Other register bits and the contents of VRAM are undefined.
- During the reset period, the R[4:0], G[5:0], B[4:0], VOB, and VOA[2:0] pin outputs are Low. After input of a reset signal, be sure to set all of the register bits and all of the VRAM (character data and line control data).
- Blanks indicate reserved bits. Always set these to "0" when issuing a command.
- In order to issue sub screen display control commands, it is necessary to set the SUBEN bit (sub operation enable) of main/OSDC operation command 17 (OSDC operation control) to 1 (sub operation enabled). (When the SUBEN bit is 0, sub screen display control commands are not accepted.)
- The main/OSDC operation command 11 (synchronization control), commands 12-0 to 12-7 (character background character code set), command 13 (I/O pin control), command 14-0 and 14-1 (display period control), commands 16-0 to 16-127 (palette control), command 17 (OSDC operation control), and command 18 (PLL clock control) are shared with the register settings under main/OSDC operation.
- The sub screen does not have a screen background character display function.

30.5.2 VRAM Write Address Set (Command 0)

Command 0 configures the VRAM write address and specifies the VRAM fill.

Command 0 (VRAM write address set)

Address: 4100_H

- Format (write-only, 32-bit access only)

  [SOSD_VADR]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															FL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			AY4	AY3	AY2	AY1	AY0			AX5	AX4	AX3	AX2	AX1	AX0

FL: Specify VRAM fill

(0: OFF, 1: ON)

AY4 to AY0: Line address

(00_H to $1F_H$)

AX5 to AX0: Column address

(00_H to $3B_H$)

- Function

  This command sets the VRAM write address and specifies the VRAM fill.

  This is used to set the line/column address before the character data set (issuing command 1, 2) and to set the line address before the line control data set (issuing command 3, 4).

  VRAM fill is started by executing character data set 2 (command 2).

- Additional Information
  - During normal writes (writing 1 character data or 1 line control data), set the VRAM fill specification to OFF (FL=0).
  - The VRAM write address specified by issuing this command is automatically incremented after executing character data set 2 (command 2)
    (on the last column this is incremented to the first column of the next line, and on the last column of the last line this is incremented to the first column of the first line).
  - The VRAM fill function writes the single character data specified by character data set 1, 2 (commands 1, 2) to character VRAM from the line and column address specified by command 0 to the last line (line 32) and last column (column 60). VRAM fill is started by executing character data set 2 (command 2).
    After the fill has finished executing, it is possible to generate a VRAM fill interrupt.
    Do not issue commands 1 to 4 while the VRAM fill is executing.

Notes:

- Line control data set (issuing command 3, 4) ignores the column address (AX5 to AX0).

  Furthermore, automatic address incrementing is not performed after line control data set.

- Specifying VRAM fill is only valid for character data set (command 1, 2).

30.5.3 Character Data Set (Command 1, Command 2)

Character data is configured using command 1 and the VRAM is set and the screen updated by executing command 2.

Command 1 (character data set 1)

Address: 4104_H

- Format (write-only, 32-bit access only)
  [SOSD_CDS1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSC2	MSC1	MSC0	MIT	MUL	MBL	MBB	MU	MD		MS1	MS0	MM2	MM1	MM0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0

MSC2, MSC1, MSC0: Character shaded background frame color type selection control
(Type 0 to 7)

MIT: Italic control
(0: Italics OFF)
(1: Italics ON)

MUL: Underline control
(0: Underline OFF)
(1: Underline ON)

MBL, MBB: Blink control
(0, 0: Blinking OFF)
(0, 1: Character background blinking ON)
(1, 0: Character + trimming blinking ON)
(1, 1: Character + trimming + character background blinking ON)

MU: Character shaded background frame top erase control
(0: Character shaded background frame top ON)
(1: Character shaded background frame top OFF)

MD: Character shaded background frame bottom erase control
(0: Character shaded background frame bottom ON)
(1: Character shaded background frame bottom OFF)

MS1, MS0: Character horizontal size control
(0, 0: 16 dots)
(0, 1: 24 dots)
(1, 0: 32 dots)
(1, 1: Setting prohibited)

MM2, MM1, MM0: Character background control
(0, 0, 0: OFF)
(0, 0, 1: Solid display)
(0, 1, 0: Shaded concaved display (solid))
(0, 1, 1: Shaded convexed display (solid))
(1, 0, 0: Background character display)
(1, 0, 1: Setting prohibited)
(1, 1, 0: Shaded concaved display (background character))
(1, 1, 1: Shaded convexed display (background character))

MB7 to MB0: Character background color
(256 colors)

MC7 to MC0: Character color
(256 colors)

- Function

  This command configures the character data. The VRAM settings are updated and the changes reflected on the screen when character data set 2 (command 2) is executed.

- Additional Information
  - The character color, character background color, character background display, character horizontal size, italic display, underline display, and blink display can be freely configured in any combination separately for each character.
  - The shaded background display can be configured to merge with the upper, lower, left or right characters by the combination of the MU and MD bits with the MR bit in character data set 2 (command 2).
  - The shaded background display frame color settings can be configured by setting bits BxH7 to BxH0 and BxS7 to BxS0 (for x = 0 to 7) in the shaded background frame color control (command 15-0 to command 15-3) and then selecting the background frame color type (0 to 7) by setting the character shaded background frame color type selection control bits MSC2, MSC1, and MSC0.
  - When blink control is set to ON, the blinking (flashing) display is performed according to the setting of bits BT1, BT0, BD1, and BD0 of the screen output control (command 5-0).
  - The background character for the background character display is selected from the character background character types 0 to F by setting bits MBM3 to MBM0 of character data set 2 (command 2). Furthermore, background character types 0 to F can be set to an arbitrary character (background character) by setting character background character codes 1, 0 to F, E (command 12-0 to command 12-7) in the main/OSDC operation to the character code.

Command 2 (character data set 2)

Address: 4108_H

- Format (write-only, 32-bit access only)

  [SOSD_CDS2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								MBM3	MBM2	MBM1	MBM0		MA	MR	MG

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

MBM3 to MBM0: Character background character type control

(0_H to F_H: 16 character types)

MR: Shaded background right character merge control

(0: Do not merge right)

(1: Merge right)

M13 to M0: Character code

(0000_H to $3FFF_H$: 16384 character types)

MA: Character alpha blend attribute control

(0: A attribute, 1: B attribute)

MG: Character/graphics character control

(0: Character, 1: Graphics character)

- Function

  This setting data is written together with the character data configured by character data set 1 (command 1) to the VRAM specified by the VRAM write address set (command 0).

  The VRAM write address is automatically incremented after this command is executed.

- Additional Information
  - The shaded background right character merge control bit (MR) only applies to character where shaded background is specified (MM1 is set to 1) in character data set 1 (command 1).
  - Alpha blend can be enabled or disabled for each individual character by the combination of the character alpha blend attribute control bit (MA) and the line character alpha blend attribute control (LALCA1 and LALCA0) of the line control data set 1 (command 3).
  - The character background character type control bits (MBM3 to MBM0) are enabled when the (MM2, MM1, MM0) bits of character data set 1 (command 1) = (1, 0, 0), (1, 1, 0), or (1, 1, 1). The character codes that correspond to the character background character types of MBM3 to MBM0 are configured by setting the character codes in the character background character code set (command 12-0 to command 12-7) registers.

Note: Because the contents of VRAM are undefined when the power is turned ON, be sure to set all of the VRAM data before starting the display.

30.5.4 Line Control Data Set (Command 3, Command 4)

Line control data is configured using command 3 and the line VRAM is set and the screen updated by executing command 4.

Command 3 (line control data set 1)

Address: $410C_H$

- Format (write-only, 32-bit access only)

  [SOSD_LDS1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LALCA1	LALCA0	LALCM	LALEN		LAL2	LAL1	LAL0	LHS3	LHS2	LHS1	LHS0		LW2	LW1	LW0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				LFD	LFC	LFB	LFA	LF7	LF6	LF5	LF4	LF3	LF2	LF1	LF0

LALCA1, LALCA0: Line alpha blend attribute control

(0, 0: Alpha blend attribute control OFF)

(0, 1: Only character alpha blend A attributes are enabled)

(1, 0: Only character alpha blend B attributes are enabled)

(1, 1: Character alpha blend A and B attributes both enabled)

LAL2 to LAL0: Line alpha blend quantity

(7 to 0 [7: Transparent <-> 0: Opaque)

LHS3 to LHS0: Line character vertical size control

(2 to 32 dots in units of 2 dots)

LFD, LFC: Trimming output control

(0, 0: All OFF)

(0, 1: Trimming only on characters with no character background)

(1, 0: Trimming only on characters with no character background, solid background, or background character)

(1, 1: Trimming output ON)

LF7 to LF0: Trimming color

(256 colors)

LALCM: Line alpha blend control scope control

(0: Control scope A [Regions other than character and trimming])

(1: Control scope B [Entire line region])

LALEN: Line alpha blend output control

(0: OFF, 1: ON)

LW2 to LW0: Line spacing control

(0 to 14 dots in units of 2 dots)

LFB, LFA: Trimming control

(0, 0: Trimming OFF)

(0, 1: Full perimeter trimming)

(1, 0: Right trimming)

(1, 1: Shadow trimming)

- Function

  This command sets the line control data. The line VRAM settings are updated and the changes reflected on the screen when line control data set 2 (command 4) is executed.

- Additional Information
  - The shadow trimming style setting is configured using shadow trimming style control (bits FM1 and FM0) of the screen output control (command 5-0).
  - For lines where the character vertical size is set to 2 dots, line spacing control is prohibited.
    Either set (LW2, LW1, LW0) to (0, 0, 0), or set line spacing upper control LWUEN to 0 and line spacing lower control LWDEN to 0 in line control data set 2 (command 4).

Command 4 (line control data set 2)

Address: 4110_H

- Format (write-only, 32-bit access only)

  [SOSD_LDS2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						LSW1	LSW0				LDS	LGY1	LGY0	LGX1	LGX0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LWUEN	LWDEN	LMUEN	LMDEN		LE	LM1	LM0	LB7	LB6	LB5	LB4	LB3	LB2	LB1	LB0

LSW1, LSW0: Character shaded background frame width control

(0, 0: 1 dot)

(0, 1: 2 dots)

(1, 0: 3 dots)

(1, 1: 4 dots)

LGY1, LGY0: Line vertical enlargement control

(0, 0: Normal)

(0, 1: Double-height)

(1, 0: Setting prohibited)

(1, 1: Quadruple-height)

LWUEN: Line spacing upper control

(0: Line spacing upper OFF)

(1: Line spacing upper ON)

LMUEN: Line shaded background frame top display control

(0: Line shaded background frame top OFF)

(1: Line shaded background frame top ON)

LE: Character background extension control

(0: Normal, 1: Extended)

LM1, LM0: Line background control

(0, 0: OFF)

(0, 1: Solid display)

(1, 0: Shaded concaved (solid) display)

(1, 1: Shaded convexed (solid) display)

LDS: Line character output control

(0: OFF, 1: ON)

LGX1, LGX0: Line horizontal enlargement control

(0, 0: Normal)

(0, 1: Double-width)

(1, 0: Setting prohibited)

(1, 1: Quadruple-width)

LWDEN: Line spacing lower control

(0: Line spacing lower OFF)

(1: Line spacing lower ON)

LMDEN: Line shaded background frame bottom display control

(0: Line shaded background frame bottom OFF)

(1: Line shaded background frame bottom ON)

LB7 to LB0: Line background color

(256 colors)

- Function

  This command writes the data specified in this command together with the line control data configured by line control data set 1 (command 3) to the line VRAM at the line address specified by the VRAM write address set (command 0).

Notes:

- Because the contents of VRAM are undefined when the power is turned ON, be sure to set all of the VRAM data before starting the display.
- Issuing this command does not automatically increment the VRAM write address. The VRAM write address set (command 0) needs to be set for each line when making line control settings.
- The frame color setting of the shaded concaved and convexed display in the line background control is the setting value of bits B0H7 to B0H0 and B0S7 to B0S0 in the shaded background frame color control (command 15-0). Furthermore, the line shaded frame width can only be 2 dots regardless of the setting of the character shaded background frame width control (LSW1 and LSW0).

30.5.5 Screen Output Control (Command 5-0)

Command 5-0 controls the screen display output.

Command 5-0 (screen output control)

Address: 4114_H

- Format (write-only, 32-bit access only)

  [SOSD_SCOC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						FM1	FM0	BT1	BT0	BD1	BD0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SALCC				WE	SDS	UDS		DSP				MCC

FM1: Shadow trimming style control

(0: Lower-right)

(1: Lower-right + right)

FM0: Trimming dot count control

(0: 1 dot)

(1: 2 dots)

BT1, BT0: Blink period control

(0, 0: 16 Vsync)

(0, 1: 32 Vsync)

(1, 0: 48 Vsync)

(1, 1: 64 Vsync)

BD1, BD0: Blink duty ratio control

(0, 0: ON : OFF = 1 : 0 - Always displayed)

(0, 1: ON : OFF = 1 : 1)

(1, 0: ON : OFF = 1 : 3)

(1, 1: ON : OFF = 3 : 1)

SALCC: Alpha blend attribute disabled quantity control

(0: Alpha blend quantity fixed at 0 [Opaque])

(1: Alpha blend quantity fixed at 7 [Transparent])

WE: Window function control

(0: OFF, 1: ON)

SDS: Sprite character output control

(0: OFF, 1: ON)

DSP: Display output control (character + character background + line background control)

(0: OFF, 1: ON)

UDS: Screen background output control

(0: OFF, 1: ON)

MCC: Main, sub screen priority control

(0: Main screen priority display, 1: Sub screen priority display)

- Function

  This command controls the screen display output.

- Additional Information
  - The initial values of SDS, UDS, DSP, and MCC after reset are "0".
  - The blink period control and blink duty ratio control perform control over characters and character backgrounds that have been specified as blinking ((MBL, MBB) set to (0, 1), (1, 0) or (1, 1)) in character data set 1 (command 1). This command also controls sprite characters if sprite character blinking has been specified (SBL set to 1) in sprite character control 1 (command 9-0).
  - Window function control has no effect on sprite characters.

30.5.6 Screen Display Position Control (Command 5-1)

Command 5-1 controls the horizontal and vertical display position of the screen.

Command 5-1 (screen display position control)

Address: 4118_H

- Format (write-only, 32-bit access only)

  [SOSD_HVDP]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

X10 to X0: Horizontal display position control
(0 to 2047 in units of 1 dot)

Y10 to Y0: Vertical display position control
(0 to 2047 in units of 1 dot)

- Function

  This command controls the horizontal display position and vertical display position of the sub screen.

30.5.7 Transparent Color Control (Command 6-0)

Command 6-0 controls the transparent color.

Command 6-0 (transparent color control)

Address: $411C_H$

- Format (write-only, 32-bit access only)

  [SOSD_TSBC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							TCC	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0

TCC: Transparent color control
(0: OFF, 1: ON)

TC7 to TC0: Transparent color code
(256 colors)

- Function

  This command controls the transparent color.

- Additional Information
  - Arbitrary color display areas can be not displayed by setting the transparent color code (TC7 to TC0) to an arbitrary color code and setting the transparent color control to ON (TCC=1). The arbitrary color display areas perform display output of the lower layer.

Note: The alpha blend quantity signal from pins VOA2 to VOA0 and the non-displayed areas due to the transparency setting output the alpha blend quantity signal to the lower layer display output.

30.5.8 Graphics Color Control (Command 6-1)

Command 6-1 is a command to display an arbitrarily specified color in graphics characters replaced by the character color or trimming color.

Command 6-1 (graphics color control)

Address: 4120_H

- Format (write-only, 32-bit access only)

  [SOSD_GRCC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							GFC	GF7	GF6	GF5	GF4	GF3	GF2	GF1	GF0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							GCC	GC7	GC6	GC5	GC4	GC3	GC2	GC1	GC0

GFC: Graphics color/trimming color replacement control
(0: OFF, 1: ON)

GCC: Graphics color/character color replacement control
(0: OFF, 1: ON)

GF7 to GF0: Color code replaced by trimming color
(256 colors)

GC7 to GC0: Color code replaced by character color
(256 colors)

- Function

  Displays an arbitrary color within the graphics characters replaced by the character color or trimming color.

- Additional Information
  - An arbitrary color within the graphics characters (the color specified by bits GF7 to GF0) is displayed replaced by the trimming color (LF7 to LF0) configured by line control data set 1 (command 3) by setting GFC to 1.
  - An arbitrary color within the graphics characters (the color specified by bits GC7 to GC0) is displayed replaced by the character color (MC7 to MC0) configured by character data set 1 (command 1) by setting GCC to 1.
  - If graphics color/trimming color replacement control is ON (the GFC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the graphics color/trimming color replacement color code and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the trimming color replacement takes precedence.
  - If graphics color/trimming color replacement control is ON (the GFC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the replacement trimming color code (trimming color (LF7 to LF0) of line control data set 1 (command 3)) and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the color becomes transparent and the lower layer color is displayed.
  - If graphics color/character color replacement control is ON (the GCC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the graphics color/character color replacement color code and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the character color replacement takes precedence.
  - If graphics color/character color replacement control is ON (the GCC bit is set to 1) and transparent color control is ON (the TCC bit of the transparent color control (command 6-0) is set to 1), and the replacement character color code (character color (MC7 to MC0) of character data set 1 (command 1)) and the transparent color code (TC7 to TC0) of the transparent color control (command 6-0) are set to the same value, the color becomes transparent and the lower layer color is displayed.
  - If graphics color/character color replacement control is ON (the GCC bit is set to 1) and graphics color/trimming color replacement control is ON (the GFC bit is set to 1), set the value of the character color replacement color code (bits GC7 to GC0) and the value of the trimming color replacement color code (bits GF7 to GF0) to different color codes.

Note: The settings of this command only apply to colors displayed in graphics characters on the sub screen. This has no effect on sprite character dot colors.

30.5.9 Screen Background Control (Command 8-0)

Command 8-0 controls the screen background color and window screen.

Command 8-0 (screen background control)

Address: 4128_H

- Format (write-only, 32-bit access only)

  [SOSD_SCBC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										UWBV	UWBH			UW1	UW0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			UAL EN		UAL2	UAL1	UAL0	U7	U6	U5	U4	U3	U2	U1	U0

UWBV, UWBH: Window border width control

(0, 0: 2 dots vertically, 2 dots horizontally)

(0, 1: 2 dots vertically, 4 dots horizontally)

(1, 0: 4 dots vertically, 2 dots horizontally)

(1, 1: 4 dots vertically, 4 dots horizontally)

UALEN: Screen background alpha blend control

(0: OFF, 1: ON)

U7 to U0: Screen background color

(256 colors)

UW1, UW0: Window background mode control

(0, 0: Background inside window)

(0, 1: Background outside window)

(1, 0: Background inside and outside window)

(1, 1: Window border background)

UAL2 to UAL0: Screen background alpha blend quantity

(7 to 0 [7: Transparent <-> 0: Opaque])

- Function

  This command controls the screen background.

- Additional Information
  - The screen background color is displayed by setting the screen background output control to ON (UDS=1) in the screen output control (command 5-0).
  - Functions related to the window (UWBV, UWBH, UW1, UW0) are enabled by setting the window function control to ON (WE=1) in the screen output control (command 5-0).

30.5.10 Window Period Control (Command 8-1, Command 8-2)

Command 8-1 and command 8-2 control the window period.

Command 8-1 (window period control 1)

Address: $412C_H$

- Format (write-only, 32-bit access only)

  [SOSD_WPC1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					WYE10	WYE9	WYE8	WYE7	WYE6	WYE5	WYE4	WYE3	WYE2	WYE1	WYE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					WYS10	WYS9	WYS8	WYS7	WYS6	WYS5	WYS4	WYS3	WYS2	WYS1	WYS0

WYE10 to WYE0: Window vertical period end control

(0 to 2047 in units of 1 dot)

WYS10 to WYS0: Window vertical period start control

(0 to 2047 in units of 1 dot)

- Function

  This command controls the display start and end timing in the vertical direction in the window period.
- Additional Information
  - This command is enabled by setting the window function control to ON (WE=1) in the screen output control (command 5-0).

Command 8-2 (window period control 2)

Address: 4130_H

- Format (write-only, 32-bit access only)

  [SOSD_WPC2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				WXE11	WXE10	WXE9	WXE8	WXE7	WXE6	WXE5	WXE4	WXE3	WXE2	WXE1	WXE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				WXS11	WXS10	WXS9	WXS8	WXS7	WXS6	WXS5	WXS4	WXS3	WXS2	WXS1	WXS0

WXE11 to WXE0: Window horizontal period end control

(0 to 4095 in units of 1 dot)

WXS11 to WXS0: Window horizontal period start control

(0 to 4095 in units of 1 dot)

- Function

  This command controls the display start and end timing in the horizontal direction in the window period.
- Additional Information
  - This command is enabled by setting the window function control to ON (WE=1) in the screen output control (command 5-0).

30.5.11 Sprite Character Control (Command 9-0, Command 9-1)

Command 9-0 and command 9-1 control the sprite character.

Command 9-0 (sprite character control 1)

Address: 4134_H

- Format (write-only, 32-bit access only)

  [SOSD_SPC1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			SBL		SH2	SH1	SH0							SD1	SD0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		SM13	SM12	SM11	SM10	SM9	SM8	SM7	SM6	SM5	SM4	SM3	SM2	SM1	SM0

SBL: Sprite character blink control

(0: OFF, 1: ON)

SD1, SD0: Sprite character configuration control

(0, 0: 1 character)

(0, 1: 2 characters horizontally)

(1, 0: 2 characters vertically)

(1, 1: 2 × 2 characters (vertical × horizontal))

SH2 to SH0: Sprite character vertical size control

(18 to 32 dots in units of 2 dots)

SM 13 to SM0: Sprite character code

(0000_H to $3FF8_H$: 2048 character types)

- Function

  This command controls the sprite character display.

- Additional Information
  - The sprite character is displayed by setting the sprite character output control to ON (SDS=1) in the screen output control 1 (command 5-0).
  - The blink period and blink duty ratio of the sprite character are controlled by the settings of bits BT1, BT0, BD1, and BD0 of the screen output control (command 5-0).

Notes:

- Sprite characters can only display graphics characters.
- The graphics color of sprite characters cannot be replaced by the graphics color/trimming color replacement control or graphics color/character color replacement control in the graphics color control (command 6-1).
- When stopping the display control by the sprite operation control, set the sprite character output control to OFF (SDS=0) in the screen output control (command 5-0).

Command 9-1 (sprite character control 2)

Address: 4138_H

- Format (write-only, 32-bit access only)

  [SOSD_SPC2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				SX11	SX10	SX9	SX8	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SY10	SY9	SY8	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

SX11 to SX0: Sprite character horizontal display position control

(0 to 4095 in units of 1 dot)

SY10 to SY0: Sprite character vertical display position control

(0 to 2047 in units of 1 dot)

- Function

  This command controls the horizontal and vertical display position of the sprite character.
- Additional Information
  - See the additional information in sprite character control 1 (command 9-0).

30.5.12 Interrupt Control (Command 15)

Command 15 controls the OSDC interrupts.

Command 15 (interrupt control)

Address: $416C_H$

- Format (write/read, 32-bit access only)

  [SOSD_INTC]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					FIF	LIF	(VIF)						FIE	LIE	(VIE)

FIF: VRAM fill end detection flag

(0: VRAM fill end not detected, 1: VRAM fill end detected)

LIF: Line display end detection flag

(0: Line display end not detected, 1: Line display end detected)

(VIF): VSYNC detection flag

Note: Always set this bit to 0 because the control is enabled on the main side.

FIE: VRAM fill end detection interrupt control

(0: Interrupt disabled, 1: Interrupt enabled)

LIE: Line display end detection interrupt control

(0: Interrupt disabled, 1: Interrupt enabled)

(VIE): VSYNC detection interrupt control

Note: Always set this bit to 0 because the control is enabled on the main side.

- Function

This command controls the interrupts.

In order to enable each of the interrupts and interrupt detection flags, each interrupt control (FIE, LIE) needs to be set to "1".

FIF and LIF are then set to "1" when the interrupt source occurs. (Writing "1" has no effect.)

Once a flag has been set, the flag can be cleared by writing "0".

- Additional Information
  - The initial values of FIF, LIF, (VIF), FIE, LIE, and (VIE) after reset are "0".
  - The read values of empty bits are always "0".
  - In order to allow LIF (line end detection flag) interrupts to be generated, the line spacing bottom control "LWDEN" needs to be set to "1" in command 4 (line control data set 2) for the corresponding line.

30.5.13 Shaded Background Frame Color Control (Command 15-0 to Command 15-3)

Commands 15-0 to 15-3 control the frame color of shaded backgrounds.

Command 15-0 to command 15-3 (shaded background frame color control)

Address: See the following table

- Format (write-only, 32-bit access only)

  [SOSD_SBC0 to SOSD_SBC3]

Address	Command No.	Data			
		31 to 24	23 to 16	15 to 8	7 to 0
4170	15-0	B1H7 to B1H0	B1S7 to B1S0	B0H7 to B0H0	B0S7 to B0S0
4174	15-1	B3H7 to B3H0	B3S7 to B3S0	B2H7 to B2H0	B2S7 to B2S0
4178	15-2	B5H7 to B5H0	B5S7 to B5S0	B4H7 to B4H0	B4S7 to B4S0
417C	15-3	B7H7 to B7H0	B7S7 to B7S0	B6H7 to B6H0	B6S7 to B6S0

BxH7 to BxH0: Shaded background frame highlight color

(256 colors)

BxS7 to BxS0: Shaded background frame shadow color

(256 colors)

(x => Character shaded background frame color type [MSC2, MSC1, MSC0] setting value: Color type 0 to 7 [8 combinations of highlight color and shadow color])

- Function

  This command controls the shaded background frame color.

- Additional Information
  - This command sets the shaded background frame color of the character shaded background frame color type selection (MSC2, MS1, MSC0 => 0 to 7 types) for characters where shaded character background is specified (MM1 is set to 1) in character data set 1 (command 1).
  - On lines where shaded line background is specified (LM1 is set to 1) in line control data set 2 (command 4), the shaded background frame color specified in B0H7 to B0H0 and B0S7 to B0S0 is used.
  - Table 30-75 shows the display parts of the shaded background frame highlight color and shadow color

.

Table 30-75. Shaded Background Frame Highlight Color and Shadow Color Display Parts

	Character background		Line background	
	Shaded background concaved display	Shaded background convexed display	Shaded background concaved display	Shaded background convexed display
Shaded background frame highlight color	Bottom edge, right edge	Top edge, left edge	Bottom edge	Top edge
Shaded background frame shadow color	Top edge, left edge	Bottom edge, right edge	Top edge	Bottom edge

Note: The shaded background frame color type of line backgrounds uses the B0H7 to B0H0 and B0S7 to B0S0 setting values, which are shared with the character shaded background frame color type for MSC2=0, MSC1=0, and MSC0=0 (type 0) for character shaded backgrounds.

30.6 Display Control Command Write Conditions and Update Timing

This section describes the conditions for writing the OSDC display control commands, and the timing with which the display control commands are applied.

30.6.1 List of Display Control Command Write Conditions and Update Timing

30.6.1 List of Display Control Command Write Conditions and Update Timing

Table 30-76 shows a list of write conditions and update timings for the display control commands.

List of display control command write conditions and command update timing

Table 30-76. List of Display Control Command Write Conditions and Update Timings

Address lower 16 bits (HEX)	Command no.	Function	Bit name	Command acceptance timing	Image output timing	Command write when dot clock is stopped	Remarks
40 00	0	VRAM write address set [MOSD_VADR]	FL	Dot clock	Dot clock	Disabled	
			AY4-AY0				
			AX5-AX0				
40 04	1	Character data set 1 [MOSD_CDS1]	MSC2-MSC0	Dot clock	Dot clock	Disabled	
			MIT				
			MUL				
			MBL		HSYNC[1]		
			MBB				
			MU		Dot clock		
			MD				
			MS1,MS0				
			MM2-MM0				
			MB7-MB0				
			MC7-MC0				
40 08	2	Character data set 2 [MOSD_CDS2]	MBM3-MBM0	Dot clock	Dot clock	Disabled	
			MA				
			MR				
			MG				
			M13-M0				
40 0C	3	Line control data set 1 [MOSD_LDS1]	LALCA1, LALCA0	Dot clock	HSYNC[2]	Disabled	
			LALCM				
			LALEN				
			LAL2-LAL0				
			LHS3-LHS0				
			LW2-LW0				
			LFD				
			LFC				
			LFB				
			LFA				
			LF7-LF0				

Table 30-76. List of Display Control Command Write Conditions and Update Timings

Address lower 16 bits (HEX)	Command no.	Function	Bit name	Command acceptance timing	Image output timing	Command write when dot clock is stopped	Remarks
40 10	4	Line control data set 2 [MOSD_LDS2]	LSW1,LSW0	Dot clock	HSYNC[2]	Disabled	
			LDS				
			LGY1,LGY0				
			LGX1,LGX0				
			LWUEN				
			LWDEN				
			LMUEN				
			LMDEN				
			LE				
			LM1,LM0				
			LB7-LB0				
40 14	5-0	Screen output control [MOSD_SCOC]	FM1,FM0	Dot clock	Dot clock	Disabled	
			BT1,BT0		HSYNC[2]		
			BD1,BD0				
			SALCC		Dot clock		
			WE				
			SDS		VSYNC[3]		
			UDS				
			PDS				
			DSP				
40 18	5-1	Screen display position control [MOSD_HVDP]	X10-X0	Dot clock	Dot clock	Disabled	
			Y10-Y0		HSYNC[1]		When interlacing is configured, the sprite display data is controlled by Y0 bit; therefore, the display data changes by dot clock.
40 1C	6-0	Transparent color control [MOSD_TSBC]	TCC	Dot clock	Dot clock	Disabled	
			TC7-TC0				
40 20	6-1	Graphics color control [MOSD_GRCC]	GFC	Dot clock	Dot clock	Disabled	
			GF7-GF0				
			GCC				
			GC7-GC0				
40 24	7	Screen background character control [MOSD_SBCC]	PH2-PH0	Dot clock	HSYNC[1]	Disabled	
			PD1,PD0		Dot clock		
			PM13-PM0		VSYNC[3]		The screen output timing is VSYNC; however, the screen will not be displayed properly when it is command acceptance during OSDC dedicated font buffer RAM data is transferred, which is mentioned in "Limitations When Performing Sprite Character Display, Screen Background Character Display, or Character Background Character Display" of "30.3.7 OSDC Operation Control".

Table 30-76. List of Display Control Command Write Conditions and Update Timings

Address lower 16 bits (HEX)	Command no.	Function	Bit name	Command acceptance timing	Image output timing	Command write when dot clock is stopped	Remarks
40 28	8-0	Screen background control [MOSD_SCBC]	UWBV	Dot clock	Dot clock	Disabled	
			UWBH				
			UW1,UW0				
			UALEN				
			UAL2-UAL0				
			U7-U0				
40 2C	8-1	Window period control 1 [MOSD_WPC1]	WYE10-WYE0	Dot clock	HSYNC[1]	Disabled	The screen output timing is HSYNC; however, when the command acceptance timing and the screen output timing are the same, the timing of dot clock is applied.
			WYS10-WYS0				
40 30	8-2	Window period control 2 [MOSD_WPC2]	WXE11-WXE0	Dot clock	Dot clock	Disabled	
			WXS11-WXS0				
40 34	9-0	Sprite character control 1 [MOSD_SPC1]	SBL	Dot clock	HSYNC[1]	Disabled	
			SH2-SH0				
			SD1,SD0		HSYNC (vertical direction)[1] Dot clock (horizontal direction)		
			SM13-SM0		VSYNC[3]		The screen output timing is VSYNC; however, the screen will not be displayed properly when it is command acceptance during OSDC dedicated font buffer RAM data is transferred, which is mentioned in "Limitations When Performing Sprite Character Display, Screen Background Character Display, or Character Background Character Display" of "30.3.7 OSDC Operation Control".
40 38	9-1	Sprite character control 2 [MOSD_SPC2]	SX11-SX0	Dot clock	Dot clock	Disabled	
			SY10-SY0		HSYNC[1]	Disabled	When interlacing is configured, the sprite display data is controlled by SY0 bit; therefore, the display data changes by dot clock unit.
40 3C	11	Synchronization control [MOSD_SYNC]	VOF5-VOF0	Dot clock	HSYNC[1]	Disabled	
			IN		Dot clock		
			FC				
40 40 to 40 5C	12-0 to 12-7	Character background character code set [MOSD_CBC0 to MOSD_CBC7]	BMxC13-BMxC0	Dot clock	VSYNC[3]	Disabled	BMxC13-BMxC0:x => 0 to F
40 60	13	I/O pin control [MOSD_IOTC]	VHE	Dot clock	HSYNC[1]	Disabled	
			HE		Dot clock		
			IHX				
			IVX		HSYNC[1]		
			OHX	IP clock (IPCLK) *4	IP clock (IPCLK)[4]		
			OBX				
			OCX				

Table 30-76. List of Display Control Command Write Conditions and Update Timings

Address lower 16 bits (HEX)	Command no.	Function	Bit name	Command acceptance timing	Image output timing	Command write when dot clock is stopped	Remarks
40 64	14-0	Display period control 1 [MOSD_CDP1]	DYE10-DYE0	Dot clock	HSYNC[1]	Disabled	The screen output timing is HSYNC; however, when the command changing timing and the display timing are the same, the timing of dot clock is applied.
			DYS10-DYS0				
40 68	14-1	Display period control 2 [MOSD_CDP2]	DXE11-DXE0	Dot clock	Dot clock	Disabled	
			DXS11-DXS0				
40 6C	15	Interrupt control [MOSD_INTC, SOSD_INTC]	FIF	Dot clock	Dot clock	Disabled	
			LIF				
			VIF				
			FIE				
			LIE				
			VIE				
40 70 to 40 7C	15-0 to 15-3	Shaded background frame color control [MOSD_SBC3 to MOSD_SBC0]	BxH7-BxH0	Dot clock	Dot clock	Disabled	BxH7-BxH0:x => 0 to 7
			BxS7-BxS0				BxS7-BxS0:x => 0 to 7
-	-	-	-	-	-	-	
41 00	0	VRAM write address set [SOSD_VADR]	FL	Dot clock	Dot clock	Disabled	
			AY4-AY0				
			AX5-AX0				
41 04	1	Character data set 1 [SOSD_CDS1]	MSC2-MSC0	Dot clock	Dot clock	Disabled	
			MIT				
			MUL				
			MBL		HSYNC[1]		
			MBB				
			MU		Dot clock		
			MD				
			MS1,MS0				
			MM2-MM0				
			MB7-MB0				
			MC7-MC0				
41 08	2	Character data set 2 [SOSD_CDS2]	MBM3-MBM0	Dot clock	Dot clock	Disabled	
			MA				
			MR				
			MG				
			M13-M0				

Table 30-76. List of Display Control Command Write Conditions and Update Timings

Address lower 16 bits (HEX)	Command no.	Function	Bit name	Command acceptance timing	Image output timing	Command write when dot clock is stopped	Remarks
41 0C	3	Line control data set 1 [SOSD_LDS1]	LALCA1, LALCA0	Dot clock	HSYNC[2]	Disabled	
			LALCM				
			LALEN				
			LAL2-LAL0				
			LHS3-LHS0				
			LW2-LW0				
			LFD				
			LFC				
			LFB				
			LFA				
			LF7-LF0				
41 10	4	Line control data set 2 [SOSD_LDS2]	LSW1,LSW0	Dot clock	HSYNC[2]	Disabled	
			LDS				
			LGY1,LGY0				
			LGX1,LGX0				
			LWUEN				
			LWDEN				
			LMUEN				
			LMDEN				
			LE				
			LM1,LM0				
			LB7-LB0				
41 14	5-0	Screen output control [SOSD_SCOC]	FM1,FM0	Dot clock	Dot clock	Disabled	
			BT1,BT0		HSYNC[2]		
			BD1,BD0				
			SALCC		Dot clock		
			WE				
			SDS		VSYNC[3]		
			UDS				
			DSP				
			MCC		Dot clock		
41 18	5-1	Screen display position control [SOSD_HVDP]	X10-X0	Dot clock	Dot clock	Disabled	
			Y10-Y0		HSYNC[1]		When interlacing is configured, the sprite display data is controlled by Y0 bit, so that the display data is changed in units of 1 dot clock.
41 1C	6-0	Transparent color control [SOSD_TSBC]	TCC	Dot clock	Dot clock	Disabled	
			TC7-TC0				
41 20	6-1	Graphics color control [SOSD_GRCC]	GFC	Dot clock	Dot clock	Disabled	
			GF7-GF0				
			GCC				
			GC7-GC0				
-	-	-	-	-	-	-	

Table 30-76. List of Display Control Command Write Conditions and Update Timings

Address lower 16 bits (HEX)	Command no.	Function	Bit name	Command acceptance timing	Image output timing	Command write when dot clock is stopped	Remarks
41 28	8-0	Screen background control [SOSD_SCBC]	UWBV	Dot clock	Dot clock	Disabled	
			UWBH				
			UW1,UW0				
			UALEN				
			UAL2-UAL0				
			U7-U0				
41 2C	8-1	Window period control 1 [SOSD_WPC1]	WYE10-WYE0	Dot clock	HSYNC[1]	Disabled	The screen output timing is HSYNC; however, when the command acceptance timing and the screen output timing are the same, the timing of dot clock is applied.
			WYS10-WYS0				
41 30	8-2	Window period control 2 [SOSD_WPC2]	WXE11-WXE0	Dot clock	Dot clock	Disabled	
			WXS11-WXS0				
41 34	9-0	Sprite character control 1 [SOSD_SPC1]	SBL	Dot clock	HSYNC[1]	Disabled	
			SH2-SH0				
			SD1,SD0		HSYNC (vertical direction)[1] Dot clock (horizontal direction)		
			SM13-SM0		VSYNC[3]		The screen output timing is VSYNC; however, the screen will not be displayed properly when it is command acceptance during OSDC dedicated font buffer RAM data is transferred, which is mentioned in "Limitations When Performing Sprite Character Display, Screen Background Character Display, or Character Background Character Display" of "30.3.7 OSDC Operation Control".
41 38	9-1	Sprite character control 2 [SOSD_SPC2]	SX11-SX0	Dot clock	Dot clock	Disabled	
			SY10-SY0		HSYNC[1]		* When interlacing is configured, the sprite display data is controlled by SY0 bit, so that the display data is changed in units of 1 dot clock.
-	-	-	-	-	-	-	
41 6C	15	Interrupt control [SOSD_INTC]	FIF	Dot clock	Dot clock	Disabled	
			LIF				
			FIE				
			LIE				
41 70 to 41 7C	15-0 to 15-3	Shaded background frame color control [SOSD_SBC3 to SOSD_SBC0]	BxH7-BxH0	Dot clock	Dot clock	Disabled	BxH7-BxH0:x => 0 to 7
			BxS7-BxS0				BxS7-BxS0:x => 0 to 7

Table 30-76. List of Display Control Command Write Conditions and Update Timings

Address lower 16 bits (HEX)	Command no.	Function	Bit name	Command acceptance timing	Image output timing	Command write when dot clock is stopped	Remarks
42 00 to 42 FC	16-0 to 16-127	Palette control	PLxxR4-PLxxR0	Dot clock	Dot clock	Disabled	PLxxR4 - PLxxR0: Red color signal palette value 5 bits (xx => 8-bit color code: 00 to FF)
			PLxxG5-PLxxG0				PLxxG5 - PLxxG0: Green color signal palette value 6 bits (xx => 8-bit color code: 00 to FF)
			PLxxB4-PLxxB0				PLxxB4 - PLxxB0: Blue color signal palette value 5 bits (xx => 8-bit color code: 00 to FF)
44 00	17	OSDC operation control [MOSD_OSDC]	SUBEN	IP clock (IPCLK)[4]	Dot clock	Disabled	
			OSDEN				
			DGO	Dot clock	Dot clock		
			ANO				
			OSDR	IP clock (IPCLK)[4]	IP clock (IPCLK)[4]		
			DCK				
			DPD				
44 04	18	PLL clock control [MOSD_PLLC]	DHRSA	IP clock (IPCLK)[4]	IP clock (IPCLK)[4]	❍	
			DCO		Dot clock		
			DKA11-DKA0	Dot clock	Dot clock		
			VCOA	IP clock (IPCLK)[4]	IP clock (IPCLK)[4]		
			VSLA2-VSLA0				
			CPEA				
			PDEA				
			DAP4-DAP0				

[1]: The HSYNC timing is a rising of the internally detected HSYNC of Examples of Horizontal Sync Operation (Figure 30-78and Figure 30-79) in "30.3.2.2 Horizontal Sync Control".
[2]: The HSYNC timing is 13-dot clock after from rising of the internally detected HSYNC of Examples of Horizontal Sync Operation (Figure 30-78 and Figure 30-79) in "30.3.2.2 Horizontal Sync Control".
[3]: The VSYNC timing is the internally detected VSYNC period of Examples of Vertical Sync Detection Operation (Figure 30-76 and Figure 30-77) in "30.3.2.1 Vertical Sync Control".
[4]: The IP clock (IPCLK) timing is the register timing from CPU as is.

30.7 Font Memory • CPU Read Access

This section describes the access (read) to the font memory from the CPU.

The font memory • CPU read access settings

When read access to the font memory of OSDC from the CPU, the register settings must be done to the addresses (reserved area) shown in Table 30-77.

Table 30-77. The Font Memory • CPU Read Access Settings

Address	Register setting value
$0000\ 0600_H$	$0020\ 0001_H$
$0000\ 0608_H$	$0040\ 0065_H$
$0000\ 0648_H$	$0000\ 0080_H$
$0000\ 0688_H$	$0412\ 0000_H$

The font memory • CPU read access area

When read access to the font memory of OSDC from the CPU, the responded area (addresses) is as follows:

- MB91F610A (Flash memory product): $0040\ 0000_H$ to $005F\ FFFF_H$ (2MB)
- MB91613 (MASK ROM product): $0040\ 0000_H$ to $004D\ FFFF_H$ (896KB)

Note: The read access to the font memory from the CPU is enabled, only when OSDC operation is disabled [OSDC operation control (command 17) OSDEN=0,SUBEN=0].

31. DMA Controller (DMAC)

This chapter explains the functions and operations of the DMA controller (DMAC).

31.1 Overview

31.2 Configuration

31.3 Registers

31.4 Interrupts

31.5 An Explanation of Operations and Setting Procedure Examples

31.1 Overview

The DMA controller (DMAC) is used for the DMA (Direct Memory Access) transfer. This controller enhances system performance by enabling high performance of data transfers without going through the CPU.

This series has 8 built-in channels for the DMA controller (DMAC).

Overview

This section explains the features of the DMA controller (DMAC).

- Address space: 32-bit (4 GB)
- Transfer mode: One of the following 3 modes can be selected.
  - Block transfer

    This mode is used to transfer 1 block of data when a transfer request is generated. When a transfer request is detected once again after 1 block of data is transferred, the following data is transferred by 1 block. In this mode, the transfer is repeated for the number of times specified.
  - Burst transfer

    In this mode, once a transfer request is generated, data is sequentially transferred 1 block at a time until the entire data transfer is completed.
  - Demand transfer

    In this mode, once a transfer request is generated, data is transferred sequentially either until the transfer request is canceled or the transfer is completed. If the transfer count is reloaded when the data transfer is completed, the data transfer continues until the transfer request is canceled.
- Data size: One of the following 3 sizes (widths) can be selected for the data to be transferred.
  - 8 bit
  - 16 bit
  - 32 bit
- Block size: It is possible to select between 1 and 16.
- Transfer count: It is possible to select a figure between 1 and 65535.
- Address update: The transfer source and destination addresses can be updated when data of a specified size (8-bit/16-bit/32-bit) is transferred. One of the following 3 update methods is available.
  - Address increment
  - Address decrement
  - No update (transfer source/destination addresses are fixed)
- Reload function: Whether to reload the following information can be specified upon the completion of data transfer for the specified number of times.
  - Address of transfer source
  - Address of transfer destination
  - Transfer count
- Transfer request: One of the following 3 methods can be selected to generate a transfer request.
  - Generate a transfer request by using software
  - Generate a transfer request by detecting whether there is an interrupt request from a peripheral function.
  - Generate a transfer request by detecting whether there is an interrupt request from on-chip bus IP (USB HOST, USB function)

  The source that generates a transfer request (transfer request source) varies depending on the transfer mode.

Table 31-1 shows the relationship between the transfer mode and transfer request source.

Table 31-1. Relationship between the transfer mode and transfer request source.

Transfer Request Source		Block Transfer	Burst Transfer	Demand Transfer
Software		O	O	X
Interrupt request from a peripheral function		O	O	X
On-chip bus IP	USB HOST	O	X	O
	USB function	O	X	O

- Priority: One of the following 2 modes can be selected as having priority when multiple transfer requests are generated.
  - Fixed

    The lowest channel number has priority.

    Order of ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7
  - Round robin

    The following example shows that the channel which started a transfer became the one with the lowest priority and the channels placed lower than it are moved up in terms of priority.

    Example) Data is transferred from ch.0 to ch.1

    Initial state: ch.0 > ch.1> ch.2> ch.3 > ch.4 > ch.5 > ch.6 > ch.7

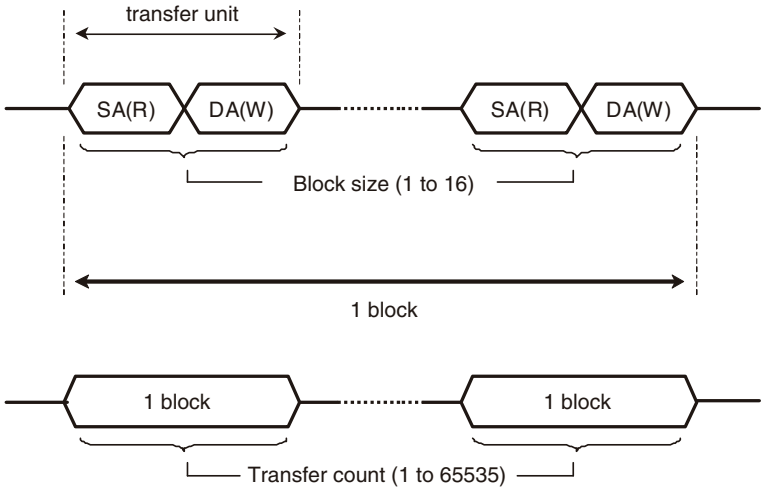
    After ch.0 is transferred: ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0

    After ch.1 is transferred: ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1
- Interrupt request: Can be generated in the following cases:
  - Normal end of DMA transfer
  - Abnormal end of DMA transfer
  - Generation of transfer stop requests

Definition of terms

Each term used for the DMA controller (DMAC) is shown in Figure 31-1.

Figure 31-1. Each Term for the DMA Controller (DMAC)



[transfer unit]

Indicating one minimum transfer of Read (R) from the Source Address (SA) and Write (W) to the Destination Address (DA).

[Block size]

Indicating the transfer count of “1 transfer unit” set with block size bits (bit3 to 0: BLK3 to BLK0) of DMA channel control registers (DCCR0 to DCCR7).

[1 block]

Indicating “1 transfer unit” multiplied by “block size”.

[Transfer count]

Indicating the transfer count of “1 block” set with DMA transfer count registers (DTCR0 to DTCR7).

31.2 Configuration

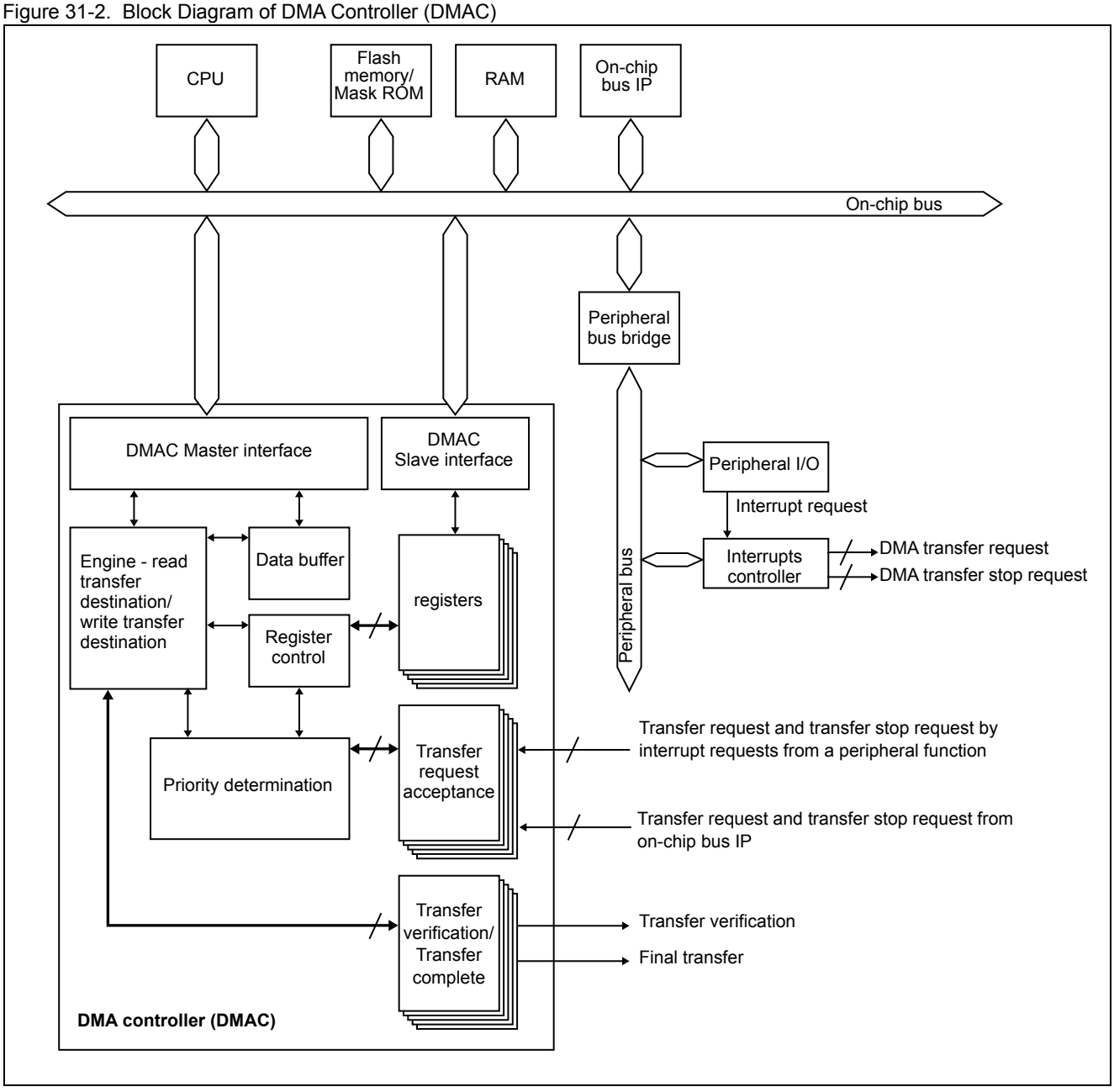
This section explains the DMA controller (DMAC) configuration.

Block diagram of DMA controller (DMAC)

Figure 31-2 is a block diagram of the DMA controller (DMAC).

The DMA controller (DMAC) is shown in where it appears inside the frame of DMA controller (DMAC).

Figure 31-2. Block Diagram of DMA Controller (DMAC)



- Engine - read transfer destination/write transfer destination
  They are used to read data from the transfer destination of DMA transfer or to write data to the transfer destination.
- Priority determination circuit
  This circuit is used for determining the level of priority of channels for DMA transfers.
- Transfer request acceptance
  Accepts DMA transfer requests.
- Transfer acceptance/transfer complete
  Outputs transfer acceptance or transfer complete.

Clocks

Table 31-2 shows the clocks used in the DMA controller (DMAC).

Table 31-2. Clocks Used in the DMA Controller (DMAC)

Clock Name	Description
Operation clock	On-chip bus clock (HCLK)

31.3 Registers

This section explains the configurations and functions of the registers used by the DMA controller (DMAC).

List of registers

Table 31-3 shows the registers of the DMA controller (DMAC).

Table 31-3. Registers of the DMA Controller (DMAC) (Sheet 1 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
Common	DMACR	DMA Control register	31.3.1
	DILVR	DMA-halt by interrupt level register	31.3.7
0	DCCR0	DMA channel control register 0	31.3.5
	DCSR0	DMA channel status register 0	31.3.6
	DTCR0	DMA transfer count register 0	31.3.4
	DSAR0	DMA source address register 0	31.3.2
	DDAR0	DMA destination address register 0	31.3.3
1	DCCR1	DMA channel control register 1	31.3.5
	DCSR1	DMA channel status register 1	31.3.6
	DTCR1	DMA transfer count register 1	31.3.4
	DSAR1	DMA source address register 1	31.3.2
	DDAR1	DMA destination address register 1	31.3.3
2	DCCR2	DMA channel control register 2	31.3.5
	DCSR2	DMA channel status register 2	31.3.6
	DTCR2	DMA transfer count register 2	31.3.4
	DSAR2	DMA source address register 2	31.3.2
	DDAR2	DMA destination address register 2	31.3.3
3	DCCR3	DMA channel control register 3	31.3.5
	DCSR3	DMA channel status register 3	31.3.6
	DTCR3	DMA transfer count register 3	31.3.4
	DSAR3	DMA source address register 3	31.3.2
	DDAR3	DMA destination address register 3	31.3.3
4	DCCR4	DMA channel control register 4	31.3.5
	DCSR4	DMA channel status register 4	31.3.6
	DTCR4	DMA transfer count register 4	31.3.4
	DSAR4	DMA source address register 4	31.3.2
	DDAR4	DMA destination address register 4	31.3.3
5	DCCR5	DMA channel control register 5	31.3.5
	DCSR5	DMA channel status register 5	31.3.6
	DTCR5	DMA transfer count register 5	31.3.4
	DSAR5	DMA source address register 5	31.3.2
	DDAR5	DMA destination address register 5	31.3.3
6	DCCR6	DMA channel control register 6	31.3.5
	DCSR6	DMA channel status register 6	31.3.6
	DTCR6	DMA transfer count register 6	31.3.4
	DSAR6	DMA source address register 6	31.3.2
	DDAR6	DMA destination address register 6	31.3.3

Table 31-3. Registers of the DMA Controller (DMAC) (Sheet 2 of 2)

Channel	Abbreviated Register Name	Register Name	Reference
7	DCCR7	DMA channel control register 7	31.3.5
	DCSR7	DMA channel status register 7	31.3.6
	DTCR7	DMA transfer count register 7	31.3.4
	DSAR7	DMA source address register 7	31.3.2
	DDAR7	DMA destination address register 7	31.3.3

31.3.1 DMA Cont2rol Register (DMACR)

This register controls the entire

DMA controller (DMAC).

Figure 31-3 shows the bit configuration of the DMA control register (DMACR).

Figure 31-3. Bit Configuration of DMA Control Register (DMACR)

bit	31	30	29	28	27	26	25	24
	DME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
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	23	22	21	20	19	18	17	16
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
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	15	14	13	12	11	10	9	8
	AT	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
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
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
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

R/W: Read/Write

Note: Be sure to access this register in units of words.

[bit31]: DME (Operation enable bit of DMA)

This bit enables/disables the entire operation of the DMA controller (DMAC).

Written Value	Explanation
0	Disables the entire operation of the DMA controller (DMAC).
1	Enables the entire operation of the DMA controller (DMAC).

Notes:

- When "0" is written to this bit to disable the entire operation of the DMA controller (DMAC), DMA transfer cannot be executed even if channel operations are enabled by the CE bit (CE = 1) of DMA channel control registers (DCCR0 to DCCR7).
- If "0" is written to this bit during DMA transfer, the transfer stops when 1 block of the data for transfer has been transferred.

[bit30 to bit16]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit15]: AT (Priority setting bit)

This bit selects one of the following for setting priority if multiple transfer requests are generated.

- Fixed: The lowest channel number is selected.
- Round robin: The priority is determined every time 1 block of data is transferred. The channel which started a transfer became the one with the lowest priority and the channels placed lower than it are moved up in terms of priority.

  Example) Transferring data from ch.0 to ch.1

  Initial state: ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7

  After ch.0 is transferred: ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0

  After ch.1 is transferred: ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1

Written Value	Explanation
0	Fixed
1	Round robin

Note: The priority set by this bit is determined per transfer of the block specified by the BLK3 to BLK0 bits of the DMA channel control registers (DCCR0 to DCCR7).
The priority is not determined during transferring by the demand transfer.

[bit14 to bit0]: Reserved bits

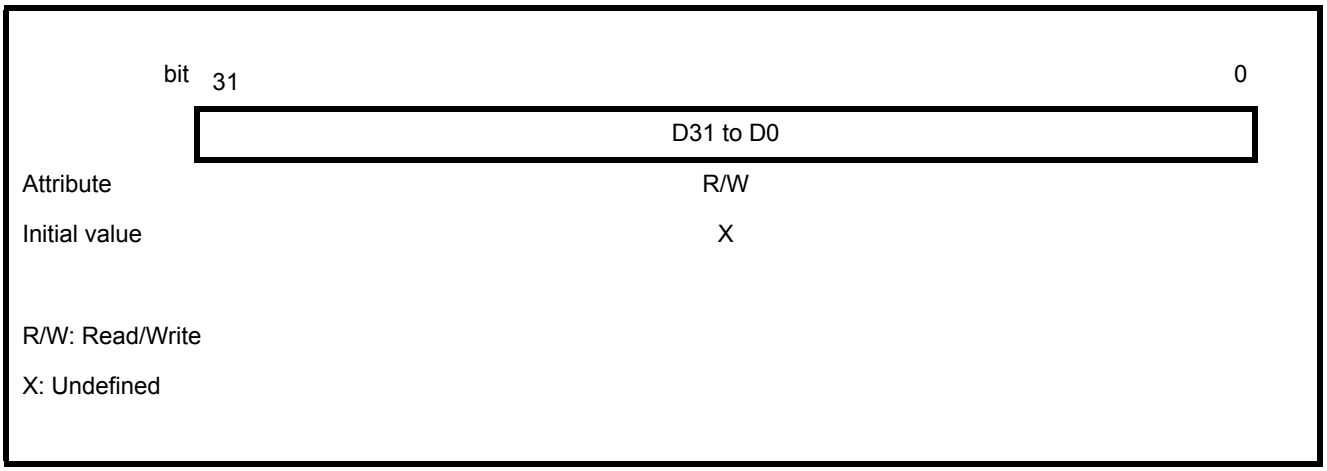
In case of writing	Always write "0" to this bit (these bits).
In case of reading	"0" is read.

31.3.2 DMA Source Address Registers (DSAR0 to DSAR7)

These registers are used to set the transfer source address. These registers are provided for each channel.

Figure 31-4 shows the bit configuration of the DMA source address registers (DSAR0 to DSAR7).

Figure 31-4. Bit Configuration of the DMA Source Address Registers (DSAR0 to DSAR7)



If the option of updating the transfer source address is selected by the SAC1 and SAC0 bits (SAC1, SAC0 = 00 or 01) of the DMA channel control registers (DCCR0 to DCCR7), the value of this register (address) is updated per completion of a DMA transfer of which size is specified by the TS1 and TS0 bits.

If the transfer of data by the number of blocks specified at DMA transfer count registers (DTCR0 to DTCR7) has completed, the value of this register becomes as follows depending on the SAR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

- SAR = 0: The value of this register becomes the next address subsequent to the last-accessed address after transfer completion.
- SAR = 1: After transfer completion, the value of this register returns to the value that is written prior to the transfer.

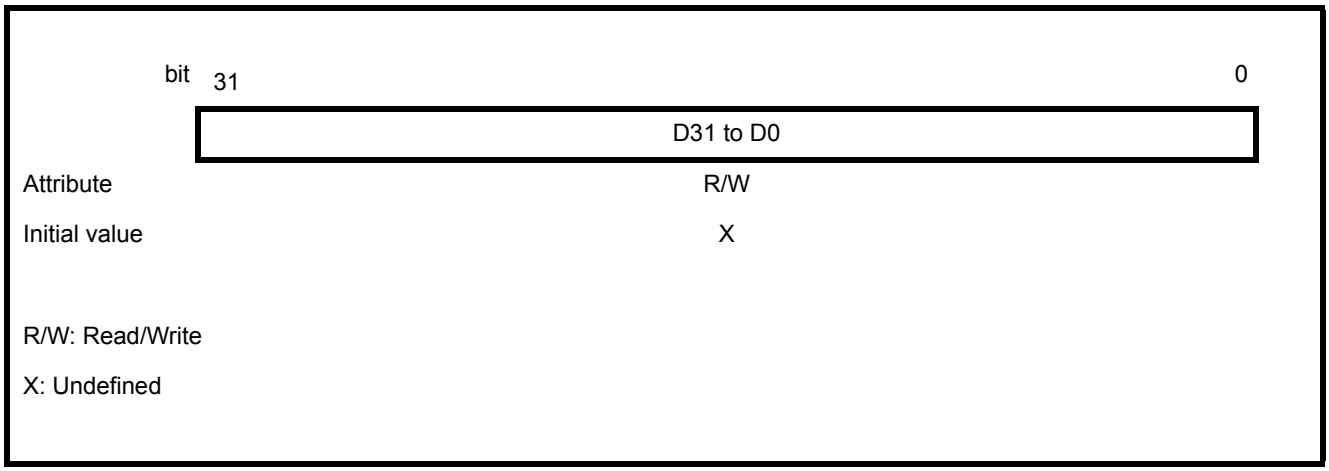
Note: Be sure to access this register in units of words.

31.3.3 DMA Destination Address Registers (DDAR0 to DDAR7)

These registers are used to set the address of transfer destination. These registers are provided for each channel.

Figure 31-5 shows the bit configuration of the DMA destination address registers (DDAR0 to DDAR7).

Figure 31-5. Bit Configuration of DMA Destination Address Registers (DDAR0 to DDAR7)



If the option of updating transfer destination address is selected by the DAC1 and DAC0 bits (DAC1, DAC0 = 00 or 01) of DMA channel control registers (DCCR0 to DCCR7), the value of this register (address) is updated per completion of a DMA transfer of which size is specified by the TS1 and TS0 bits.

If a transfer of data by the number of blocks specified at DMA transfer count registers (DTCR0 to DTCR7) has completed, the value of this register becomes as follows depending on the DAR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

- DAR = 0: The value of this register becomes the next address subsequent to the last-accessed address after transfer completion.
- DAR = 1: After transfer completion, the value of this register returns the value that is written prior to the transfer.

Note: Be sure to access this register in units of words.

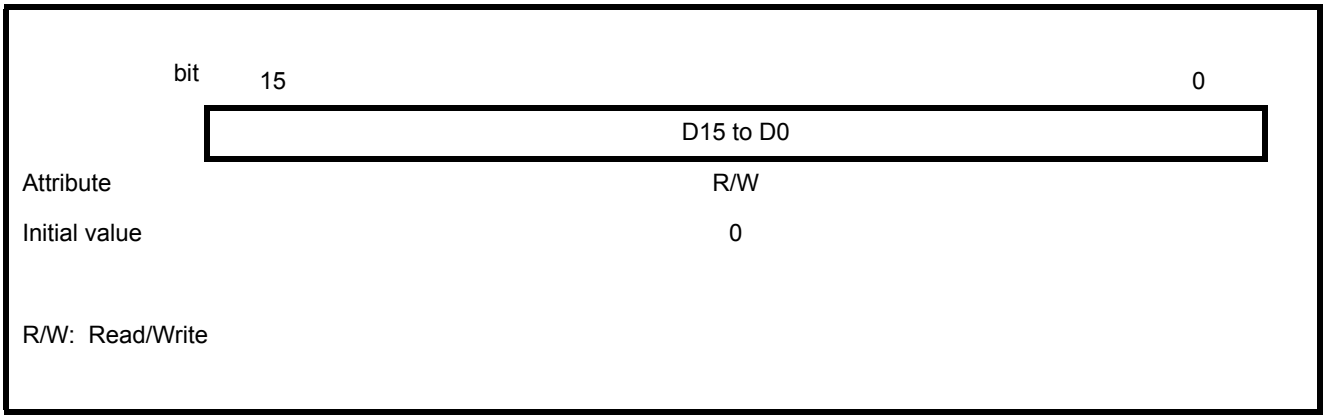
31.3.4 DMA Transfer Count Registers (DTCR0 to DTCR7)

The number of data blocks to transfer can be selected from within the range of 1 to 65535. Also, reading this value shows the number of remaining data blocks to be transferred. These registers are provided for each channel.

The value of these registers decrements by 1 per transfer of 1 block of data. When this register value becomes "0", the transfer ends.

Figure 31-6 shows the bit configuration of the DMA transfer count registers (DTCR0 to DTCR7).

Figure 31-6. Bit Configuration of DMA Transfer Count Registers (DTCR0 to DTCR7)



If a transfer of data by the number of blocks specified at these registers has completed, the value of this register becomes as follows depending on the TCR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

- TCR = 0: The value of this register becomes "0" after transfer completion.
- TCR = 1: After transfer completion, the value of this register returns to the value that is written prior to the transfer.

Notes:

- Transfer is not executed if "0" is set to this register.
- Be sure to access this register in units of halfwords.
- If the DMA transfer is suspended or ends abnormally, this register shows the number of remaining transfers.

31.3.5 DMA Channel Control Registers (DCCR0 to DCCR7)

These registers control the channels of the DMA controller (DMAC). These registers are provided for each channel.

Figure 31-7 shows the bit configuration of the DMA channel control registers (DCCR0 to DCCR7).

Figure 31-7. Bit Configuration of the DMA Channel Control Registers (DCCR0 to DCCR7)

bit	31	30	29	28	27	26	25	24
	CE	Reserved	Reserved	Reserved	Reserved	AIE	SIE	NIE
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	23	22	21	20	19	18	17	16
	Reserved	Reserved	RS1	RS0	Reserved	Reserved	TM1	TM0
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	15	14	13	12	11	10	9	8
	ST	SAR	SAC1	SAC0	DT	DAR	DAC1	DAC0
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
	TCR	Reserved	TS1	TS0	BLK3	BLK2	BLK1	BLK0
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

R/W: Read/Write

Note: Be sure to access this register in units of words.

[bit31]: CE (Channel operation enable bit)

This bit enables/disables the operation of channels.

Written Value	Explanation
0	Disables channel operation.
1	Enables channel operation.

If "1" is written to this bit when a request source of DMA transfer is set to software by using the RS1 and RS0 bits (RS1, RS0 = 00), DMA transfer starts. After transfer completion, this bit is automatically cleared to "0".

If RS1 and RS0 bits are set to other than "00", only the channel operation is enabled by writing "1" to this bit.

In this case, if a transfer request specified by the RS1 and RS0 bits is detected, transfer starts. The value of this bit becomes as follows by the TCR bit settings.

- TCR = 0: The bit is cleared to "0" after transfer completion.
- TCR = 1: The bit is not cleared to "0" even after transfer completion.

Note: If "0" is written to this bit during the DMA transfer, the transfer stops when 1 block of the data for transfer has been transferred.
In such case, transfer is not re-executed until "1" is written to this bit again and a transfer request is detected.

[bit30 to bit27]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit26]: AIE (Abnormal end interrupt enable bit)

During enabling an abnormal end interrupt of a channel (AIE=1), an abnormal end interrupt request is output when a prohibited value is set to this register.

However, a flag bit (AC bit of DMA channel status registers (DCSR0 to DCSR7)) indicating the abnormal end will be changed to "1" regardless of the settings.

If a value set to this register is any of the following, it is regarded as having an abnormal end of DMA transfer.

- TM1, TM0 bits = 10 (setting prohibited)
- SAC1, SAC0 bits = 10 (setting prohibited)
- DAC1, DAC0 bits = 10 (setting prohibited)
- TS1, TS0 bits = 11 (setting prohibited)
- RS1, RS0 bits = 00 and TM1, TM0 bits = 11 (transfer request source: software, transfer mode: demand transfer)

Written Value	Explanation
0	Disables generation of abnormal end interrupt requests.
1	Enables generation of abnormal end interrupt requests.

Notes:

- When AIE=0, writing AIE=1 and setting the prohibited value to the register, it becomes AC=1 and AIE=1 of DMA channel status register (DCSR0 to DCSR7); however, an abnormal end interrupt does not occur.
- When the end interrupt request is generated, the interrupt request will not be cleared, even if AIE is set to "0". Please write "0" to AC to clear the interrupt request.

[bit25]: SIE (Transfer suspension interrupt enable bit)

During enabling a transfer stop interrupt of a channel (SIE=1), an interrupt request is output when a transfer is stopped by the transfer stop request.

However, when the transfer stop request is generated, the flag bit indicating transfer stop by the transfer stop request (SP bit of DMA channel status register (DCSR0 to DCSR7)) changes to "1" regardless of the settings.

Written Value	Explanation
0	Disables generation of transfer suspension interrupt requests.
1	Enables generation of transfer suspension interrupt requests.

Note: When the transfer suspension interrupt request is generated, the interrupt request will not be cleared, even if SIE is set to "0". Please write "0" to SP to clear the interrupt request.

[bit24]: NIE (Normal end interrupt enable bit)

During enabling a normal end interrupt of a channel (NIE=1), an interrupt request is output when DMA transfer normally ends.

However, the flag bit indicating normal end (NC bit (DCSR0 to DCSR7) of DMA channel status register) changes to "1" regardless of the settings when DMA transfer normally ends.

Under any of the following conditions, DMA transfer is regarded as having a normal end.

- Transfer completes for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7).
- If "0" is set for the value of DMA transfer count registers (DTCR0 to DTCR7), channel operations are enabled by the CE bit (CE = 1).

Written Value	Explanation
0	Disables generation of normal end interrupt requests.
1	Enables generation of normal end interrupt requests.

Note: When the normal end interrupt request is generated, the interrupt request will not be cleared, even if NIE is set to "0". Please write "0" to NC to clear the interrupt request.

[bit23, bit22]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit21, bit20]: RS1, RS0 (Transfer request source bits)

These bits set the source that generates a transfer request (transfer request source) from any of the following 3 options.

- Generate a transfer request by using software.
- Generate a transfer request by detecting an interrupt request from a peripheral function.
- Generate a transfer request by detecting whether there is an interrupt request from on-chip bus IP (USB HOST, and USB function)

RS1	RS0	Explanation
0	0	Software
0	1	Interrupt request from a peripheral function
1	0	Setting prohibited
1	1	On-chip bus IP

Notes:

- If transfer mode is set to demand transfer by the TM1 and TM0 bits (TM1, TM0 = 11), only the on-chip bus IP can be set for the transfer request source.
- If interruption requests from a peripheral function are set as the transfer request source, the following register settings are required.
  - IO-data request registers (IORR0 to IORR7)
  - Select registers for DMA transfer request clear by a peripheral (ICSEL0 to ICSEL11)

  See "32. Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function" for details of each register.
- If the on-chip bus IP is set as the transfer request source, the DMA transfer setting for it is required. See "28. USB Host" for USB HOST, and "27. USB Function" for the USB function.

[bit19, bit18]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit17, bit16]: TM1, TM0 (Transfer mode bits)

These bits set the transfer mode from any of the following 3 modes.

- Block transfer

  This mode is used to transfer 1 block of data when a transfer request is generated. When a transfer request is detected again after 1 block of data is transferred, the following data is transferred by 1 block. In this mode, the transfer is repeated for the number of times specified.
- Burst transfer

  In this mode, once a transfer request is generated, data is sequentially transferred 1 block at a time until the entire data transfer is completed.
- Demand transfer

  In this mode, once a transfer request is generated, data is transferred sequentially either until the transfer request is canceled or the transfer is completed. If the reload option of transfer count is enabled at the completion of the data transfer, the data transfer continues until the transfer request is canceled.

TM1	TM0	Explanation
0	0	Block transfer
0	1	Burst transfer
1	0	Setting prohibited
1	1	Demand transfer

Note: When demand transfer is selected, these bits need to be set to "1" by using the ST bit or DT bit.

[bit15]: ST (Transfer source type bits)

This bit sets whether to output the transfer request acceptance signal and the transfer complete signal in the reading cycle of the transfer source.

Written Value	Explanation
0	Does not output.
1	Outputs.

In the case where, interruption requests from a peripheral function are given to the transfer request source and the peripheral is specified as the transfer destination, by setting this bit to "1", the transfer request reception signal is output and the transfer request can be cleared.

Notes:

- If transfer mode is set to demand transfer by the TM1 and TM0 bits (TM1, TM0 = 11), set "1" to both or either of this bit/DT bit.
- In case of performing the DMA transfer with the USB function or USB HOST, always set the ST bit to "0" regardless of the transfer source type.

[bit14]: SAR (Transfer source address reload bit)

This bit specifies whether to return the value of DMA source address registers (DSAR0 to DSAR7) to the value before transfer upon the completion of data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) (enable/disable reload of transfer source address).

Written Value	Explanation
0	Disables reload. The value of DMA source address registers (DSAR0 to DSAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	Enables reload. After transfer completion, the value of DMA source address registers (DSAR0 to DSAR7) returns to the value that is written prior to the transfer.

[bit13, bit12]: SAC1, SAC0 (Transfer source address count bits)

These bits specify whether to update the value of DMA source address registers (DSAR0 to DSAR7) from one of the following 3 options per completion of data transfer of the size specified with TS1 and TS0 bits.

SAC1	SAC0	Explanation
0	0	Address increment
0	1	Address decrement
1	0	Setting prohibited
1	1	Address fixed

If the setting of address increment/address decrement is selected, the increment/decrement value varies depending on the transfer size specified with TS1 and TS0 bits.

Table 31-4 shows the relationship between the transfer size and the address increment/decrement value.

Table 31-4. Relationship between the transfer size and the value of address increment / decrement

Transfer Size	Increment/Decrement Value
8 bit	1
16 bit	2
32 bit	4

[bit11]: DT (Transfer destination type bits)

This bit sets whether to output the transfer request acceptance signal and the transfer complete signal in the writing cycle to the transfer destination.

Written Value	Explanation
0	Does not output.
1	Outputs.

In the case where, interruption requests from a peripheral function are given to the transfer request source and the peripheral is specified as the transfer destination, by setting this bit to "1", the transfer request reception signal is output and the transfer request can be cleared.

Notes:

- If transfer mode is set to demand transfer by the TM1 and TM0 bits (TM1, TM0 = 11), set "1" to both or either of this bit/ST bit.
- In case of performing the DMA transfer with the USB function or USB HOST, always set the DT bit to "1" regardless of the transfer destination type.

[bit10]: DAR (Transfer destination address reload bits)

This bit specifies whether to return the value of DMA destination address registers (DDAR0 to DDAR7) to the value before transfer upon the completion of data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) (enable/disable reload of transfer source address).

Written Value	Explanation
0	Disables reload. The value of DMA destination address registers (DDAR0 to DDAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	Enables reload. After transfer completion, the value of DMA destination address registers (DDAR0 to DDAR7) returns to the value that is written prior to the transfer.

[bit9, bit8]: DAC1, DAC0 (Transfer destination address count bits)

These bits specify whether to update the value of DMA destination address registers (DDAR0 to DDAR7) from one of the following 3 options per completion of data transfer of the size specified with TS1 and TS0 bits.

DAC1	DAC0	Explanation
0	0	Address increment
0	1	Address decrement
1	0	Setting prohibited
1	1	Address fixed

If the setting of address increment/address decrement is selected, the increment/decrement value varies depending on the transfer size specified with TS1 and TS0 bits.

Table 31-5 shows the relationship between the transfer size and the address increment/decrement value.

Table 31-5. Relationship between the Transfer Size and the Value of Address Increment/Decrement

Transfer Size	Increment/Decrement Value
8 bit	1
16 bit	2
32 bit	4

[bit7]: TCR (Transfer count reload bit)

This bit specifies whether to have the specified transfer count reloaded to DMA transfer count registers (DTCR0 to DTCR7) upon the completion of data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) (enable/disable reload of transfer count).

Written Value	Explanation
0	Disables reload. After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) is cleared to "0".
1	Enables reload. After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) returns to the value that is written prior to the transfer.

Notes:

- If "1" is set to this bit and the transfer request source is set to other than software by RS1 and RS0 bits, the CE bit is not cleared to "0" and enters the transfer request wait state even after transfer completion.
- If "0" is written to this bit, the CE bit is automatically cleared to "0" after transfer completion regardless of the transfer request source.
- If the reload is enabled by writing "1" to this bit, transfer continues sequentially in demand transfer mode while transfer requests are being output irrespective of whether transfer has completed for the number of times specified.

[bit6]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit5, bit4]: TS1, TS0 (Transfer size bits)

These bits set the size (width) of data to be transferred at one time from any of the following 3 modes.

TS1	TS0	Explanation
0	0	8 bit
0	1	16 bit
1	0	32 bit
1	1	Setting prohibited

[bit3 to bit0]: BLK3 to BLK0 (Block size bits)

These bits are used to specify the number of times for 1 transfer unit (size) within 1 block.

BLK3	BLK2	BLK1	BLK0	Explanation
0	0	0	0	1 time
0	0	0	1	2 times
0	0	1	0	3 times
0	0	1	1	4 times
0	1	0	0	5 times
0	1	0	1	6 times
0	1	1	0	7 times
0	1	1	1	8 times
1	0	0	0	9 times
1	0	0	1	10 times
1	0	1	0	11 times
1	0	1	1	12 times
1	1	0	0	13 times
1	1	0	1	14 times
1	1	1	0	15 times
1	1	1	1	16 times

31.3.6 DMA Channel Status Registers (DCSR0 to DCSR7)

These registers show the state of DMA controller (DMAC). These registers are provided for each channel.

Figure 31-8 shows the bit configuration of the DMA channel status registers (DCSR0 to DCSR7).

Figure 31-8. Bit Configuration of DMA Channel Status Registers (DCSR0 to DCSR7)

bit	15	14	13	12	11	10	9	8
	CA	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	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
	Reserved	Reserved	Reserved	Reserved	Reserved	AC	SP	NC
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

R/W: Read/Write

R: Read only

Note: Be sure to access this register in units of half words.

[bit15]: CA (Channel active bit)

This bit indicates the operating state of channels.

Read Value	Explanation
0	The channel is inactive.
1	The channel is active.

Notes:

- When "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7), this bit is changed to "1".
- This bit is changed to "0" in any of the following cases:
  - Transfer ends.
  - "0" is written to the CE bit of the DMA channel control registers (DCCR0 to DCCR7).

[bit14 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2]: AC (Abnormal end state flag bit)

This bit indicates an abnormal end of DMA transfer.

If the value of DMA channel control registers (DCCR0 to DCCR7) is any of the following, DMA transfer is regarded as having an abnormal end.

- TM1, TM0 bits = 10 (setting prohibited)
- SAC1, SAC0 bits = 10 (setting prohibited)
- DAC1, DAC0 bits = 10 (setting prohibited)
- TS1, TS0 bits = 11 (setting prohibited)
- RS1, RS0 bits = 00 and TM1, TM0 bits = 11 (transfer request source: software, transfer mode: demand transfer)

If "1" is set to the AIE bit of the DMA channel control registers (DCCR0 to DCCR7) when this bit is set to "1", an abnormal end interrupt request is generated.

AC	In Case of Reading	In Case of Writing
0	No abnormal end is detected.	This bit is cleared to "0".
1	An abnormal end is detected.	Ignored

Note: This bit is not cleared automatically. To clear the abnormal end interrupt request flag, write "0" to this bit before enabling the DMA transfer operation.
To clear this bit during the DMA transfer, make sure that this bit is "1" before writing "0".

[bit1]: SP (Transfer suspension state flag bit)

This bit indicates a suspension of transfer due to a transfer stop request from the transfer request source.

If "1" is set to the SIE bit of the DMA channel control registers (DCCR0 to DCCR7) when this bit is set to "1", a transfer suspension interrupt request is generated.

SP	In Case of Reading	In Case of Writing
0	Transfer is not suspended.	This bit is cleared to "0".
1	Transfer is suspended.	Ignored

Note: This bit is not cleared automatically. To clear the transfer suspension interrupt request flag, write "0" to this bit before enabling the DMA transfer operation.
To clear this bit during the DMA transfer, make sure that this bit is "1" before writing "0".

[bit0]: NC (Normal end state flag bit)

This bit indicates a normal end of DMA transfer.

Under any of the following conditions, the DMA transfer is regarded as ending normally.

- Transfer completes for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7).
- If "0" is set for the value of DMA transfer count registers (DTCR0 to DTCR7), channel operations are enabled by the CE bit (CE = 1) of DMA channel control registers (DCCR0 to DCCR7).

If "1" is set to the NIE bit of the DMA channel control registers (DCCR0 to DCCR7) when this bit is set to "1", a normal end interrupt request is generated.

NC	In Case of Reading	In Case of Writing
0	No normal end of transfer is detected.	This bit is cleared to "0".
1	A normal end of transfer is detected.	Ignored

Note: This bit is not cleared automatically. To clear the normal end interrupt request flag, write "0" to this bit before enabling the DMA transfer operation.
To clear this bit during the DMA transfer, make sure that this bit is "1" before writing "0".

31.3.7 DMA-Halt by Interrupt Level Register (DILVR)

This register specifies whether to halt DMA transfer if an interrupt request is generated from a peripheral function.

Figure 31-9 shows the bit configuration of DMA-halt by interrupt level register (DILVR).

Figure 31-9. Bit Configuration of the DMA-Halt by Interrupt Level Register (DILVR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	LVL4	LVL3	LVL2	LVL1	LVL0
Attribute	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	1	1	1	1

R/W: Read/Write

R: Read only

Note: Be sure to access this register in units of bytes.

[bit7 to bit5]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit4 to bit0]: LVL4 to LVL0 (DMA-halt by interrupt level bits)

These bits specify the interrupt level where DMA transfer is halted. If an interrupt request of a level that is higher than the interrupt level specified with this bit is generated from the peripheral functions, DMA transfer is halted.

LVL4	LVL3	LVL2	LVL1	LVL0	Interrupt Request Level Where DMA Transfers Are Halted
1	0	0	0	0	DMA transfers are not halted.
1	0	0	0	1	Higher level of interrupt request than "11_H"
1	0	0	1	0	Higher level of interrupt request than "12_H"
1	0	0	1	1	Higher level of interrupt request than "13_H"
1	0	1	0	0	Higher level of interrupt request than "14_H"
1	0	1	0	1	Higher level of interrupt request than "15_H"
1	0	1	1	0	Higher level of interrupt request than "16_H"
1	0	1	1	1	Higher level of interrupt request than "17_H"
1	1	0	0	0	Higher level of interrupt request than "18_H"
1	1	0	0	1	Higher level of interrupt request than "19_H"
1	1	0	1	0	Higher level of interrupt request than "$1A_H$"
1	1	0	1	1	Higher level of interrupt request than "$1B_H$"
1	1	1	0	0	Higher level of interrupt request than "$1C_H$"
1	1	1	0	1	Higher level of interrupt request than "$1D_H$"
1	1	1	1	0	Higher level of interrupt request than "$1E_H$"
1	1	1	1	1	All interrupt requests

Note: LVL4 bit is fixed to "1", and only LVL3 to LVL0 can be set.

31.4 Interrupts

An interrupt request is generated under the following conditions:

- Normal end of DMA transfer (normal end interrupt request)
- Abnormal end of DMA transfer (abnormal end interrupt request)
- Suspension of DMA transfer due to the generation of a transfer stop request (transfer suspension interrupt request)

Table 31-6 outlines the interrupts that can be used with the DMA controller (DMAC).

Table 31-6. Interrupts of the DMA Controller (DMAC)

Interrupt Request	Interrupt Request Flag	Interrupt Request Enabled	Interrupt Clear
Normal end interrupt request	NC of DCSR = 1	NIE of DCCR = 1	Write "0" to the NC bit of the DCSR.
Abnormal end interrupt request	AC of DCSR = 1	AIE of DCCR = 1	Write "0" to the AC bit of the DCSR.
Transfer suspension interrupt request	SP of DCSR = 1	SIE of DCCR = 1	Write "0" to the SP bit of the DCSR.

DCSR: DMA channel status registers (DCSR0 to DCSR7)
DCCR: DMA channel control registers (DCCR0 to DCCR7)

Notes:

- Clear the interrupt requests after disabling generation of interrupt requests, or clear the interrupt requests in the interrupt processing routine.
- For details of the interrupt vector number of each interrupt request, see "A.3 Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For information on the interrupt level settings, see "10. Interrupt Controller".
- When the interrupt request of DMA controller is generated, the interrupt request will not be dropped even if the interrupt enable bits (AIE, SIE, and NIE) are set to "0".
  Please write "0" to the interrupt request flags (AC, SP, and NC) to clear the interrupt request.

31.5 An Explanation of Operations and Setting Procedure Examples

This section explains the operations of the DMA controller (DMAC). This section also includes examples of procedures for setting the transfer mode.

31.5.1 Transfer Settings

This section explains the settings required for the use of the DMA controller (DMAC).

Overview

Settings of the entire DMA controller (DMAC) and the channels to be used for the transfer are required to use the DMA transfer.

To select an interrupt request of a peripheral function as a transfer request source of a DMA transfer, the settings of the interrupt vector number and each peripheral function are also required. For details, see "32. Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function".

The setup procedure is as follows:

1. Settings of the entire DMA controller (DMAC)
   See "Settings of the entire DMA Controller (DMAC)".
2. Settings of channels to be used for the DMA controller
   See "Channel Settings".

Settings of the entire DMA Controller (DMAC)

Settings of the entire DMA controller (DMAC) and the channels to be used for the transfer are required for the use of the DMA controller (DMAC).

This paragraph explains the required settings of the entire DMA controller (DMAC).

- Enabling operation of the DMA controller (DMAC) by the DME bit of DMA control register (DMACR)
  Operation disabled: DME = 0
  Operation enabled: DME = 1
- Settings of priority in AT bit of DMA control register (DMACR)
  Fixed: AT = 0
  Round robin: AT = 1
- Settings of interrupt levels of DMA transfer halts by the LVL4 to LVL0 bits of DMA-halt by interrupt level register (DILVR)

For details, see the explanation for each register.

Note: To select the interrupt request generated from a peripheral function as the source that generates DMA transfer requests, select an interrupt vector number before configuring DMA controller (DMAC) settings.
For details of selecting interrupt vector numbers, see "32.3.1 IO-Data Request Registers (IORR0 to IORR7)".

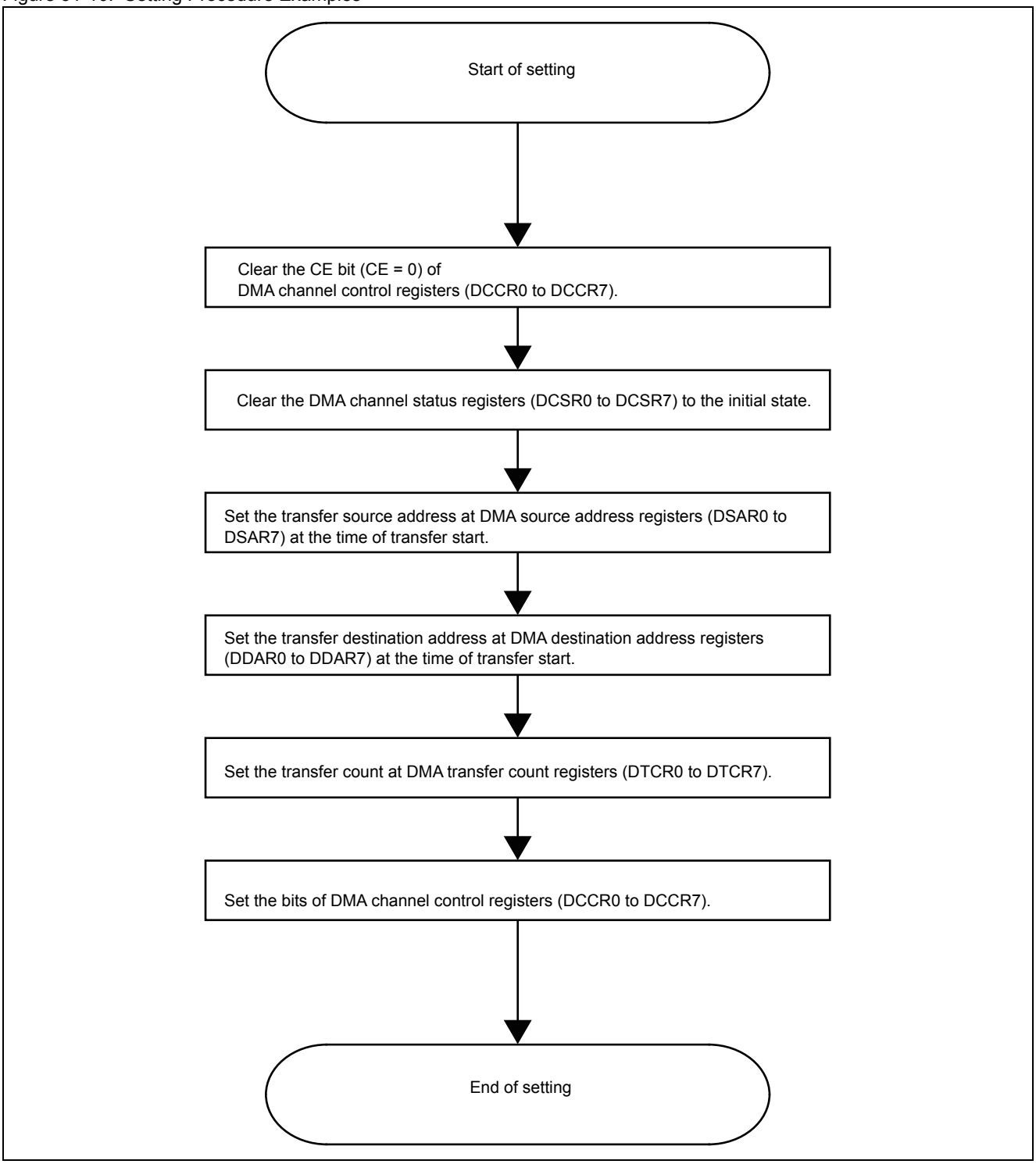
Channel Settings

This paragraph explains the settings required for the channels to be used.

Configure channel settings upon the completion of the entire DMA controller (DMAC) settings.

Figure 31-10 shows the example of the procedure for configuring channel settings.

Figure 31-10. Setting Procedure Examples



1. Disable channel operation.
   The CE bit of DMA channel control registers (DCCR0 to DCCR7) is 0.
2. Initialize the flag that indicates the channel state.
   The AC bit / SP bit / NC bit of DMA channel status registers (DCSR0 to DCSR7) is 0.
3. Set the transfer source address.
   D31 to D0 bits of DMA source address registers (DSAR0 to DSAR7)
4. Set the transfer destination address.
   D31 to D0 bits of DMA destination address registers (DDAR0 to DDAR7)
5. Set the transfer count within the range of 1 to 65535 times.
   D15 to D0 bits of DMA transfer count registers (DTCR0 to DTCR7) is 1 or more.
6. Other settings
   Set the bits of DMA channel control registers (DCCR0 to DCCR7).
7. Write "1" to the CE bit of the DMA channel control registers (DCCR0 to DCCR7).
   Channel operation is enabled.
   The transfer is started at the same time that channel operations are enabled if the transfer request source is set to software.

Note: For information on procedures for register settings, see the explanation for each register.

31.5.2 Transfer Operations

This section explains the transfer operations of the DMA controller (DMAC).

Transfer mode

The DMA controller (DMAC) has the following 3 transfer modes.

- Block transfer

  This mode is used to transfer 1 block of data when a transfer request is generated. This mode repeats the transfer of the following data by 1 block by the specified number of times when a transfer request is detected once again after 1 block of data is transferred.

  The data of which size is specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7) are transferred once at a time for the number of times specified by the BLK3 to BLK0 bits.

- Burst transfer

  In this mode, once a transfer request is generated, data is sequentially transferred 1 block at a time until the entire data transfer is completed.

  The data of which size is specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7) are transferred for the count specified at the BLK3 to BLK0 bits at a time sequentially for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7).

- Demand transfer

  In this mode, once a transfer request is generated, data is transferred sequentially either until the transfer request is canceled or the transfer is completed. If the reload option of the transfer count is on when the data transfer is completed, the data transfer continues until the transfer request is canceled.

  In the demand transfer, either or both of ST/DT needs to be set to "1".

The source that generates a transfer request varies depending on the transfer mode.

Table 31-7 shows the relationship between the transfer mode and transfer request source.

Table 31-7. Relationship between the transfer mode and transfer request source

Transfer Request Source		Block Transfer	Burst Transfer	Demand Transfer
Software		O	O	X
Interrupt request from a peripheral function		O	O	X
On-chip bus IP	USB HOST	O	X	O
	USB function	O	X	O

Detection of transfer requests

The transfer operation starts by detecting DMA transfer requests.

Detections of transfer requests vary depending on the transfer request source specified by the RS1 and RS0 bits of DMA channel control registers (DCCR0 to DCCR7).

- The transfer request source is software:

  If "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7), it determines the priority levels of channels and starts the transfer.
- Transfer source is other than software:

  When the CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "1", channel operations are enabled.

  If a transfer request is detected during that state, it determines the priority levels of channels and starts the transfer.

Notes:

- If interrupt requests of a peripheral function are set as the transfer request source, it is necessary to select an interrupt vector. See "32.3.1 IO-Data Request Registers (IORR0 to IORR7)".
- If interrupt requests of a peripheral function are set as transfer request source, set the value of interrupt level mask register (ILM) and interrupt control registers (ICR00 to ICR47) as follows when interrupt requests are generated from a peripheral function.
  ILM less than or equal to ICR
- If the on-chip bus IP is set as the transfer request source, the DMA transfer setting for it is required. See "28. USB Host" for USB HOST, and "27. USB Function" for the USB function.

Table 31-8 shows the detect condition of transfer requests and transfer request source.

Table 31-8. Detect condition of transfer requests and transfer request source

Transfer Request Source		Block Transfer	Burst Transfer	Demand Transfer
Software		Write "1" to the CE bit of the DCCR.		-
Interrupt request		Edge detection		-
On-chip bus IP	USB HOST	Edge detection	-	First time: Edge detection Second time or after: Level detection
	USB function			

DCCR: DMA channel control registers (DCCR0 to DCCR7)

Note: The interrupt request of the peripheral function does not start transfer, even if CE is changed from "0" to "1" during the interrupt request is generated because it is for edge detection.
Please execute an interrupt enable etc. of the peripheral function after setting CE to "1".

Operation

If a transfer request is detected, transfer is executed as follows:

1. Read data from the address specified at DMA source address registers (DSAR0 to DSAR7).

   Data of the bit width specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7) are read.
2. Write data to the address specified by DMA destination address registers (DDAR0 to DDAR7).

Priority

If multiple DMA transfer requests are generated, transfer is executed from the channel with the highest priority specified by the AT bit settings of DMA control registers (DMACR).

Priority is determined every time 1 block of data is transferred. Also, priority is determined at the end of transfer.

Note: Priority is not determined during the transfer in demand transfer mode.

Procedure of priority determination is any of the following:

- Fixed: The lowest channel number is selected.

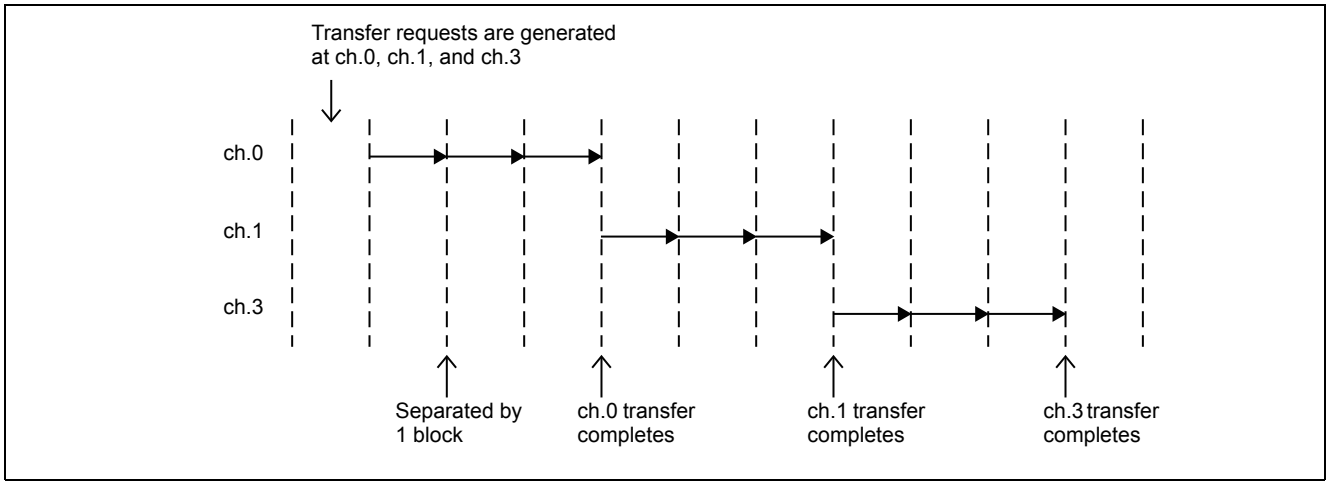
  Figure 31-11 shows an example of transfer as transfer example 1 under the following conditions.

  Transfer request: Occurs simultaneously at ch.0, ch.1, and ch.3

  Transfer mode: Burst transfer mode in all channels

  Transfer count: 3 for all channels

Figure 31-11. Transfer Example 1



1. Transfer requests are generated simultaneously at ch.0, ch.1, and ch.3.
2. Transfer at ch.0 starts.
3. Once data at ch.0 is transferred for 3 blocks, transfer at ch.1 starts.
4. Once data at ch.1 is transferred for 3 blocks, transfer at ch.3 starts.

Figure 31-12 shows an example of transfer as transfer example 2 under the following conditions.

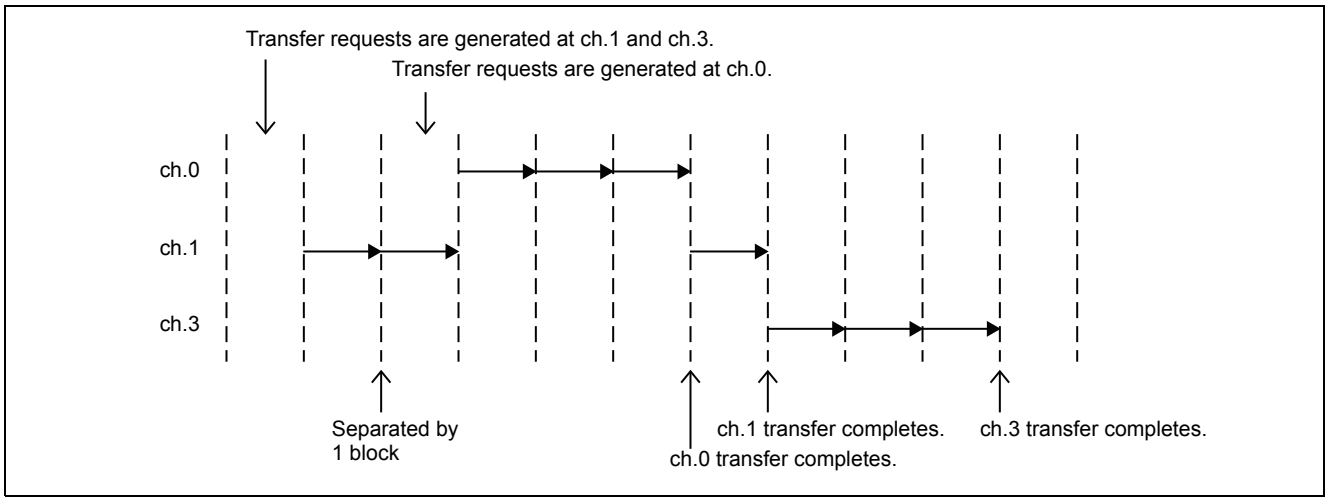
Transfer request:

- Occurs simultaneously at ch.1 and ch.3.
- Occurs at ch.0 during the transfer at ch.1.

Transfer mode: All channels are in burst transfer mode.

Transfer count: 3 for all channels.

Figure 31-12. Transfer Example 2



1. Transfer requests are generated simultaneously at ch.1 and ch.3.
2. Transfer at ch.1 starts.
3. Transfer requests are generated at ch.0 during the transfer at ch.1.
4. Transfer at ch.1 is suspended and transfer at ch.0 starts.
5. Once data at ch.0 is transferred for 3 blocks, transfer at ch.1 starts.
6. Once data at ch.1 is transferred for 3 blocks, transfer at ch.3 starts.

- Round robin: The channel which started a transfer became the one with the lowest priority and the channels placed lower than it are moved up in terms of priority one priority level at a time.

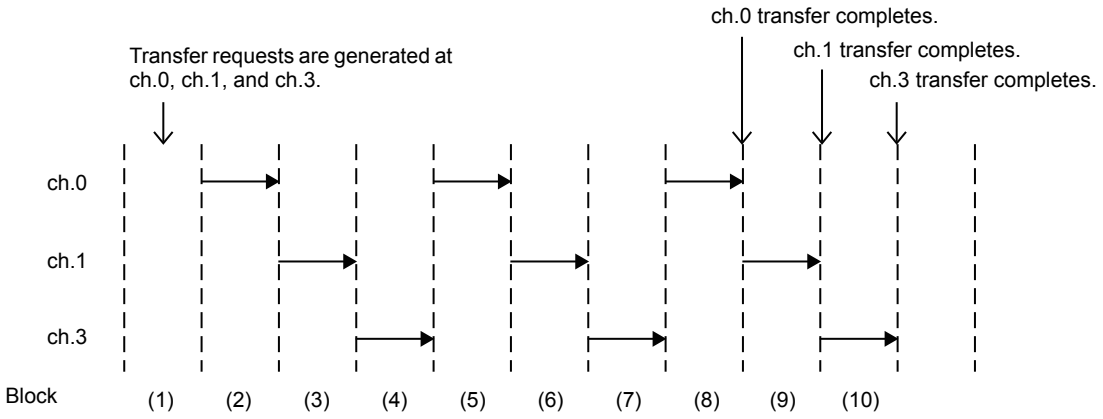
  Figure 31-13 shows an example of transfer under the following conditions.

  Transfer request: Occurs simultaneously at ch.0, ch.1, and ch.3.

  Transfer mode: Burst transfer mode in all channels

  Transfer count: 3 for all channels.

Figure 31-13. Transfer Example



Channel priority levels for each block

(1) ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7
(2) ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0
(3) ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1
(4) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1 > ch.3
(5) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.1 > ch.3 > ch.0
(6) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.3 > ch.0 > ch.1
(7) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1 > ch.3
(8) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.1 > ch.3 > ch.0
(9) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.3 > ch.0 > ch.1
(10) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1 > ch.3

1. Transfer requests are generated simultaneously at ch.0, ch.1, and ch.3.
2. 1 block of data at ch.0 is transferred.
3. After 1 block of data at ch.0 has been transferred, 1 block of data at ch.1 is transferred.
4. After 1 block of data at ch.1 has been transferred, 1 block of data at ch.3 is transferred.
5. After 1 block of data at ch.3 has been transferred, the second block of data at ch.0 is transferred.
6. After the second block of data at ch.0 has been transferred, the second block of data at ch.1 is transferred.
7. After the second block of data at ch.1 has been transferred, the second block of data at ch.3 is transferred.
8. After the second block of data at ch.3 has been transferred, the third block of data at ch.0 is transferred. Transfer at ch.0 ends.
9. After the third block of data at ch.0 has been transferred, the third block of data at ch.1 is transferred. Transfer at ch.1 ends.
10. After the third block of data at ch.1 has been transferred, the third block of data at ch.3 is transferred. Transfer at ch.3 ends.

Transfer address update operation

Upon every completion of data transfer of the data size specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7), the transfer source address and transfer destination address can be incremented/decremented.

Address updates can be set with the following registers.

- Transfer source address: SAC1 and SAC0 bits of DMA channel control registers (DCCR0 to DCCR7)
- Transfer destination address: DAC1 and DAC0 bits of DMA channel control registers (DCCR0 to DCCR7)

The increment/decrement width varies depending on the size specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7).

Table 31-9 shows the relationship between the bit setting value and the address increment/decrement width.

Table 31-9. Bit Setting Value and Increment/Decrement Width

Transfer Source Address (SAC1 And SAC0)	Transfer Destination Address (DAC1 And DAC0)	Transfer Size (TS1 And TS0)	Transfer Source Address Increment/ Decrement Width	Transfer Destination Address Increment/Decrement Width
00 (increment)	00 (increment)	00 (8 bits)	Incremented by 1	Incremented by 1
		01 (16 bits)	Incremented by 2	Incremented by 2
		10 (32 bits)	Incremented by 4	Incremented by 4
	01 (decrement)	00 (8 bits)	Incremented by 1	Decremented by 1
		01 (16 bits)	Incremented by 2	Decremented by 2
		10 (32 bits)	Incremented by 4	Decremented by 4
	11 (fixed)	00 (8 bits)	Incremented by 1	No increase or decrease
		01 (16 bits)	Incremented by 2	No increase or decrease
		10 (32 bits)	Incremented by 4	No increase or decrease
01 (decrement)	00 (increment)	00 (8 bits)	Decremented by 1	Incremented by 1
		01 (16 bits)	Decremented by 2	Incremented by 2
		10 (32 bits)	Decremented by 4	Incremented by 4
	01 (decrement)	00 (8 bits)	Decremented by 1	Decremented by 1
		01 (16 bits)	Decremented by 2	Decremented by 2
		10 (32 bits)	Decremented by 4	Decremented by 4
	11 (fixed)	00 (8 bits)	Decremented by 1	No increase or decrease
		01 (16 bits)	Decremented by 2	No increase or decrease
		10 (32 bits)	Decremented by 4	No increase or decrease
11 (fixed)	00 (increment)	00 (8 bits)	No increase or decrease	Incremented by 1
		01 (16 bits)	No increase or decrease	Incremented by 2
		10 (32 bits)	No increase or decrease	Incremented by 4
	01 (decrement)	00 (8 bits)	No increase or decrease	Decremented by 1
		01 (16 bits)	No increase or decrease	Decremented by 2
		10 (32 bits)	No increase or decrease	Decremented by 4
	11 (fixed)	00 (8 bits)	No increase or decrease	No increase or decrease
		01 (16 bits)	No increase or decrease	No increase or decrease
		10 (32 bits)	No increase or decrease	No increase or decrease

31.5.3 Transfer Suspension

DMA controller (DMAC) suspends the DMA transfer under the following cases.

This section explains the operation when DMA transfer is suspended.

Overview

DMA transfer is suspended when:

- "0" is written to the DME bit of DMA control registers (DMACR).
- "0" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7).
- Transfer stop request is output from the transfer request source.

Transfer suspension/restart

Transfer is suspended in units of blocks. For this reason, transfer is suspended after the completion of the transfer of 1 block of data if any suspension sources are generated during the transfer.

Once a transfer is suspended, DMA controller (DMAC) enters the stop state without executing new transfer.

- "0" is written to the DME bit of DMA control registers (DMACR)

  All channels enter the stop state.

  When the DME bit is cleared to "0", the channel where the transfer is being executed suspends the transfer on the completion of the transfer of 1 block of the data being transferred. Also, transfer requests that have been detected are not cleared.

  Restart the DMA transfer by using the following procedure.

  1. Write "1" to the DME bit of DMA control registers (DMACR).

- "0" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7)

  The corresponding channels enter the stop state.

  If data is being transferred at the corresponding channel, the transfer is suspended upon the completion of the transfer of 1 block of the data currently being transferred. Also, transfer requests that have been detected are cleared.

  Restart the DMA transfer by using the following procedure.

  1. Write "1" to the CE bit of the DMA channel control registers (DCCR0 to DCCR7) of which channels are in the stop state.
  2. Generate a new transfer request.

- Suspension due to a transfer stop request from transfer request source

  If a reception error occurs and transfer stop request is generated when the DMA controller (DMAC) is activated in multi-function serial interface, the transfer is suspended upon the completion of the transfer of 1 block of the data currently being transferred.

  Once the transfer is suspended, the following states are generated.

  - The SP bit of DMA channel status registers (DCSR0 to DCSR7) is changed to "1".
  - The CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "0".
  - Transfer requests that have been detected are cleared.

  While a transfer stop request is being output, no new transfer request is accepted.

  Restart the DMA transfer by using the following procedure.

  1. Disable the transfer stop request.
  2. Write "0" to the SP bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel.
  3. Write "1" to the CE bit of the DMA channel control registers (DCCR0 to DCCR7).
  4. Generate a new transfer request.

  Note: The SP bit of DMA channel status registers (DCSR0 to DCSR7) is not cleared to "0" automatically. To clear this bit, write "0" to the SP bit.

Restart operation

Follow the restart procedures to resume DMA transfer. The restart operations vary when "1" is written to the DME bit of DMA control register (DMACR) and when "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7).

Also, they vary depending on the transfer mode specified.

Table 31-10 shows the operation when the transfer is restarted.

Table 31-10. Operation of Transfer Restart

Transfer Mode	"1" Is Written to the DME Bit.	"1" Is Written to the CE Bit.
Block transfer	If a new transfer request is detected, the transfer restarts in accordance with its priority.	If a new transfer request is detected, the transfer restarts in accordance with its priority. (To generate a new transfer request in demand transfer mode, a DMA transfer request must be input once again from the on-chip bus IP)
Burst transfer	Transfer restarts immediately in accordance with the priority.	
Demand transfer	If transfer requests are generated continuously when "1" is written to the DME bit, transfer restarts immediately without determining the priority.	

31.5.4 Operation at the End of Transfer

This section explains the end operation of DMA transfer.

The end of transfer can be classified as normal end or abnormal end.

- Normal end

  DMA transfer ends normally when the data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) have completed.

  When DMA transfer ends normally, the following conditions occur.

  1. The NC bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel is changed to "1".
  2. The CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "0".

     DMA controller (DMAC) enters the stop state.

  However, if the transfer request source is set to other than software and reload of transfer count is set, the CE bit of DMA channel control registers (DCCR0 to DCCR7) is not cleared.

  Also, if "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7) for the corresponding channel when the value set for DMA transfer count registers (DTCR0 to DTCR7) is "0", the NC bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel is changed to "1" as with the normal end.

  Be sure to set "1" or greater as the value of DMA transfer count registers (DTCR0 to DTCR7) before writing "1" to the CE bit of DMA channel control registers (DCCR0 to DCCR7).

  Notes:

  - If interrupt requests are selected for the transfer request source, the interrupt request flags for peripheral functions are cleared by the DMA controller (DMAC) once DMA transfer has completed.
  - The NC bit of DMA channel status registers (DCSR0 to DCSR7) is not cleared to "0" automatically. To clear this bit, write "0" to the NC bit.

- Abnormal end

  An abnormal end interrupt request is output when the prohibited value is set to DMA channel control registers (DCCR0 to DCCR7).

  If the value of DMA channel control registers (DCCR0 to DCCR7) is any of the following, the DMA transfer is regarded as having an abnormal end.

  - TM1, TM0 bits = 10 (setting prohibited)
  - SAC1, SAC0 bits = 10 (setting prohibited)
  - DAC1, DAC0 bits = 10 (setting prohibited)
  - TS1, TS0 bits = 11 (setting prohibited)
  - RS1, RS0 bits = 00 and TM1, TM0 bits = 11 (transfer request source: software, transfer mode: demand transfer)

  When DMA transfer ends abnormally, the following conditions occur.

  1. The AC bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel is changed to "1".
  2. The CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "0".

     DMA controller (DMAC) enters the stop state.

  Note: The AC bit of DMA channel status registers (DCSR0 to DCSR7) is not cleared to "0" automatically. To clear this bit, write "0" to the AC bit.

31.5.5 Post-transfer Operation

This section explains the post-DMA-transfer operation of the blocks with the specified transfer count.

Reload operation

DMA controller (DMAC) is equipped with the reload register that maintains the transfer source address, transfer destination address, and transfer count, which have been written prior to the transfer. With this function, the transfer source address, transfer destination address, and transfer count that have been specified prior to the transfer can be reloaded after the transfer.

The following registers provide reload registers.

- DMA source address registers (DSAR0 to DSAR7)
- DMA destination address registers (DDAR0 to DDAR7)
- DMA transfer count registers (DTCR0 to DTCR7)

Note: The reload register is used as a register that memorizes the value written to each corresponding register. The value of the reload register cannot be read.

- Reload operation of transfer source address

  To specify whether to reload the transfer source address to DMA source address registers (DSAR0 to DSAR7), use the SAR bit of DMA channel control registers (DCCR0 to DCCR7).

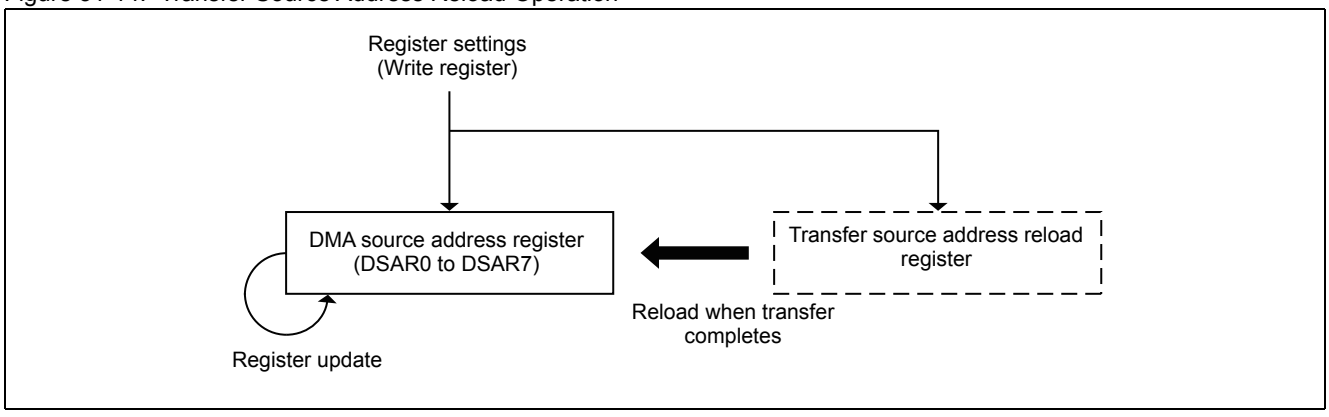
  Table 31-11 shows the setting method and post-transfer operation.

Table 31-11. Setting Method and Post-transfer Operation

SAR	Post-transfer Operation
0	The value of the DMA source address registers (DSAR0 to DSAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	After transfer completion, the value of DMA source address registers (DSAR0 to DSAR7) returns to the value that is written prior to the transfer.

Figure 31-14 shows the operation when reload is enabled by the SAR bit (SAR = 1) of DMA channel control registers (DCCR0 to DCCR7).

Figure 31-14. Transfer Source Address Reload Operation



Note: If a transfer is suspended or abnormal end occurs before completing the data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7), the transfer source address is not reloaded even when the reload is enabled by the SAR bit (SAR = 1) of DMA channel control registers (DCCR0 to DCCR7).
The value of DMA source address registers (DSAR0 to DSAR7) becomes the next address subsequent to the last-accessed address.

- Reload operation of transfer destination address

  To specify whether to reload the transfer destination address to DMA destination address registers (DDAR0 to DDAR7), use the DAR bit of DMA channel control registers (DCCR0 to DCCR7).

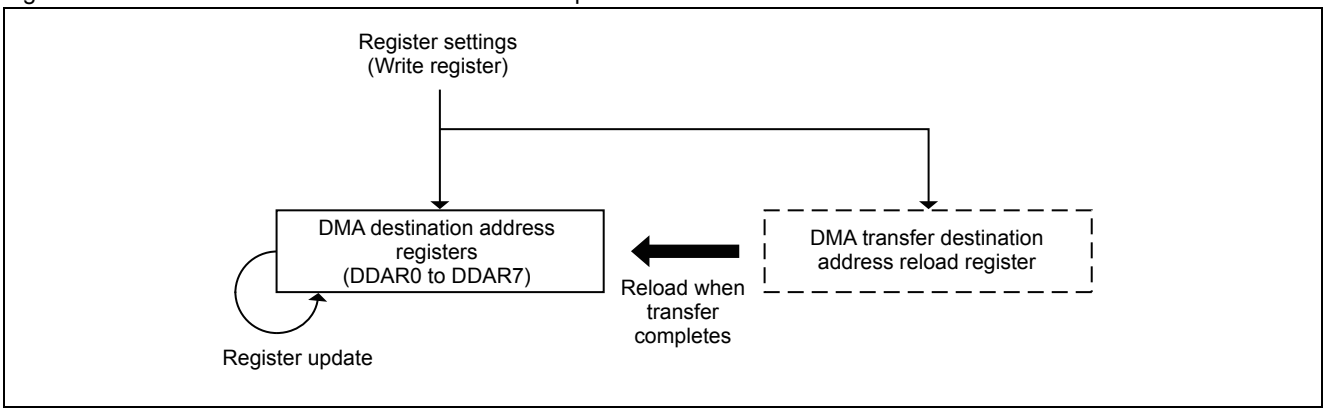
  Table 31-12 shows the setting method and post-transfer operation.

Table 31-12. Setting Method and Post-transfer Operation

DAR	Post-transfer Operation
0	The value of DMA destination address registers (DDAR0 to DDAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	After transfer completion, the value of DMA destination address registers (DDAR0 to DDAR7) returns to the value that is written prior to the transfer.

Figure 31-15 shows the operation when reload is enabled by the DAR bit (DAR = 1) of DMA channel control registers (DCCR0 to DCCR7).

Figure 31-15. Transfer Destination Address Reload Operation



Note: If a transfer is suspended or abnormal end occurs before completing the data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7), the transfer destination address is not reloaded even when the reload is enabled by the DAR bit (DAR = 1) of DMA channel control registers (DCCR0 to DCCR7).
The value of DMA destination address registers (DDAR0 to DDAR7) becomes the next address subsequent to the last-accessed address.

- Reload operation of transfer count

  To specify whether to reload the transfer count to DMA transfer count registers (DTCR0 to DTCR7), use the TCR bit of DMA channel control registers (DCCR0 to DCCR7).

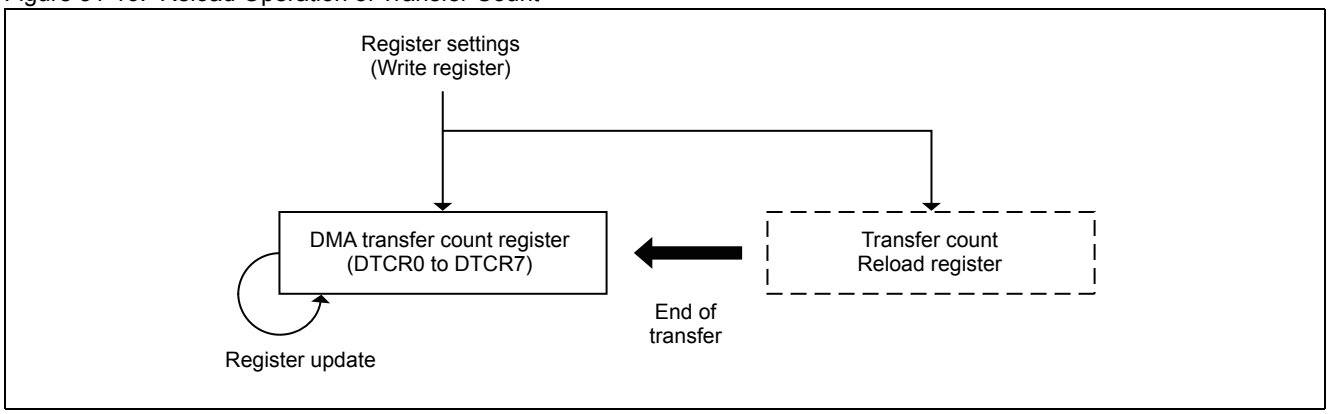
  Table 31-13 shows the setting method and post-transfer operation.

Table 31-13. Setting Method and Post-transfer Operation

TCR	Post-transfer Operation
0	After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) is cleared to "0".
1	After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) returns to the value that is written prior to the transfer.

Figure 31-16 shows the operation when reload is enabled by the TCR bit (TCR = 1) of DMA channel control registers (DCCR0 to DCCR7).

Figure 31-16. Reload Operation of Transfer Count



Also, whether to clear the CE bit of DMA channel control registers (DCCR0 to DCCR7) upon the completion of the transfer depends on the TCR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

Table 31-14 shows the relationship between the TCR bit and CE bit after transfer completion.

Table 31-14. Relationship between the TCR Bit and the Post-transfer CE Bit

TCR	Transfer Request Source	
	Software	Other Than Software
0	The CE bit is cleared to "0".	The CE bit is cleared to "0".
1	The CE bit is cleared to "0".	The CE bit is not cleared.

Notes:

- If the reload of the transfer count is enabled by the TCR bit (TCR = 1) of DMA channel control registers (DCCR0 to DCCR7), transfer continues sequentially in demand transfer mode while transfer requests are being output.
- If the DMA transfer is suspended or abnormal end occurs, DMA transfer count registers (DTCR0 to DTCR7) show the number of remaining transfers.

31.5.6 DMA Transfer Halt

If an interrupt request is generated from the a peripheral function, DMA transfer is halted.

DMA transfer is halted in units of blocks. Therefore, if an interrupt request of a level that is higher than the interrupt level specified at DMA-halt by interrupt level register (DILVR) is generated, the transfer is halted after the completion of the transfer of 1 block of the data currently being transferred.

If the DMA transfer is halted, DMA controller (DMAC) enters the halt state without executing new transfer.

The DMA transfer restarts when the interrupt request is cleared, and the interrupt level reaches or falls below the level of LVL4 to LVL0 bits of DMA-halt by interrupt level register (DILVR).

Table 31-15 shows the relationship between the LVL4 to LVL0 bit settings of DMA-halt by interrupt level register (DILVR) and the interrupt request level that halts the DMA transfer.

Table 31-15. Interrupt Request Level where DMA Transfers are Halted.

LVL4	LVL3	LVL2	LVL1	LVL0	Interrupt Request Level Where DMA Transfers Are Halted.
1	0	0	0	0	DMA transfers are not halted.
1	0	0	0	1	Higher level of interrupt request than "11_H"
1	0	0	1	0	Higher level of interrupt request than "12_H"
1	0	0	1	1	Higher level of interrupt request than "13_H"
1	0	1	0	0	Higher level of interrupt request than "14_H"
1	0	1	0	1	Higher level of interrupt request than "15_H"
1	0	1	1	0	Higher level of interrupt request than "16_H"
1	0	1	1	1	Higher level of interrupt request than "17_H"
1	1	0	0	0	Higher level of interrupt request than "18_H"
1	1	0	0	1	Higher level of interrupt request than "19_H"
1	1	0	1	0	Higher level of interrupt request than "$1A_H$"
1	1	0	1	1	Higher level of interrupt request than "$1B_H$"
1	1	1	0	0	Higher level of interrupt request than "$1C_H$"
1	1	1	0	1	Higher level of interrupt request than "$1D_H$"
1	1	1	1	0	Higher level of interrupt request than "$1E_H$"
1	1	1	1	1	All interrupt requests

32. Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function

This section explains the method of generating a DMA transfer request by using an interrupt request of peripheral functions and the method of clearing an interrupt request flag of peripheral functions from the DMA controller (DMAC).

32.1 Overview

32.2 Configuration

32.3 Registers

32.4 An Explanation of Operations and Setting Procedure Examples

32.1 Overview

This series enables the activation of a DMA transfer by using an interrupt request of peripheral functions.

Registers for selecting an interrupt request which activates a DMA transfer are provided for each channel of the DMA controller (DMAC).

If multiple interrupt requests are allocated to one interrupt vector number, you must also set which interrupt request flag is to be cleared using the DMA controller (DMAC).

Overview of generation of a DMA transfer request by using a peripheral function

The registers of the DMA controller (DMAC) can be used to set the source that triggers a DMA transfer request generation (transfer request source) for an interrupt request of a peripheral function.

Values corresponding to the interrupt vector number are specified to select the interrupt request to be used.

Overview of the select function for DMA transfer request clear by using a peripheral function

- Selection of an interrupt request

  If the source that triggers a DMA transfer request generation (transfer request source) is designated as an interrupt request of a peripheral function, the interrupt request flag is cleared by the DMA controller (DMAC) after the DMA transfer.

  For this reason, if multiple interrupt requests are allocated to the interrupt vector number to be selected as the source that triggers a DMA transfer request generation (transfer request source), you must select the interrupt request flag to be cleared by the DMA controller (DMAC) after the DMA transfer.

- Selection of a transfer stop request

  In the UART/CSIO/I^2C ch.8 to ch.11, if an interrupt request is generated upon reception, a transfer stop request is output to the DMA controller (DMAC) to interrupt the DMA transfer.

  Use this function to select the channel of which reception interrupt request is to be used as the DMA transfer stop request.

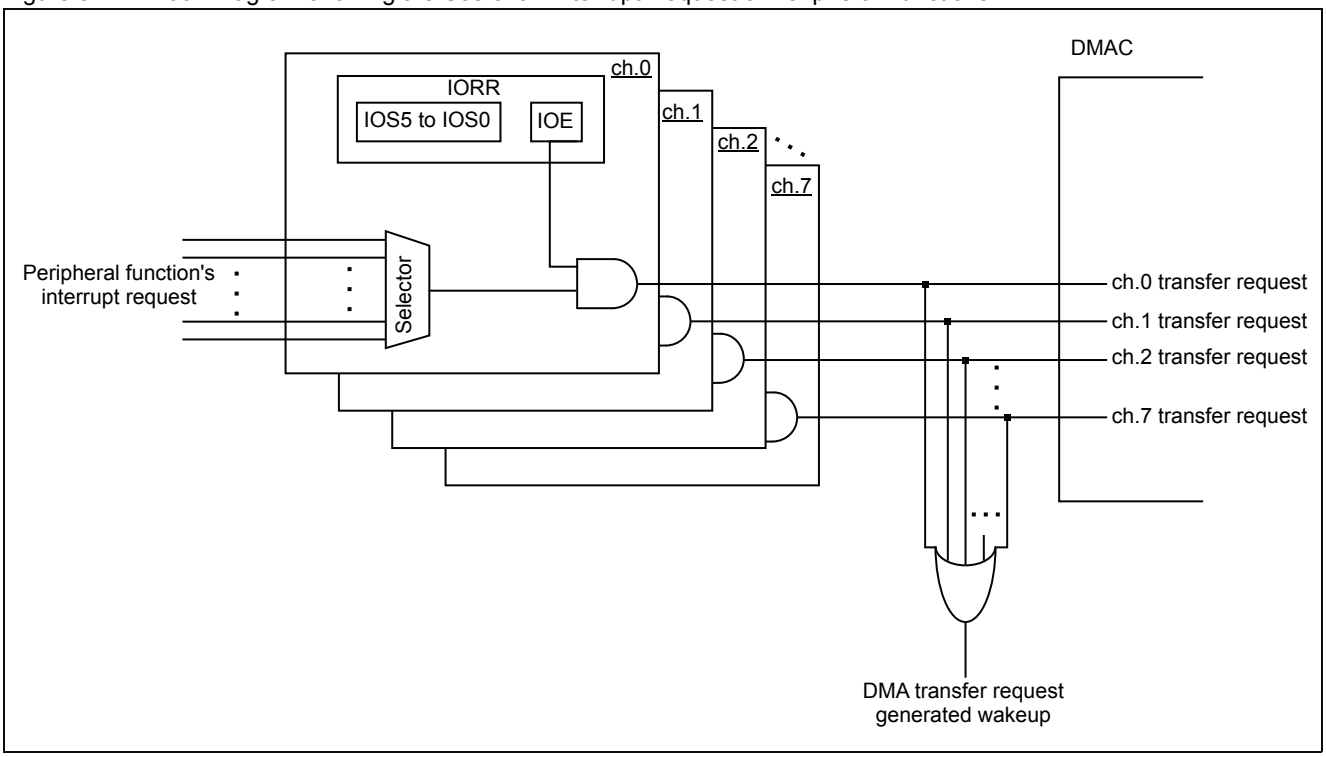
32.2 Configuration

This section explains the part at which a DMA transfer request is generated by a peripheral function and the configuration of the select function for DMA transfer request clear.

Block diagram of the part at which a DMA transfer request is generated by a peripheral function

Figure 32-1 is a block diagram of the part which uses an interrupt request of a peripheral function as a transfer request source for the DMA transfer.

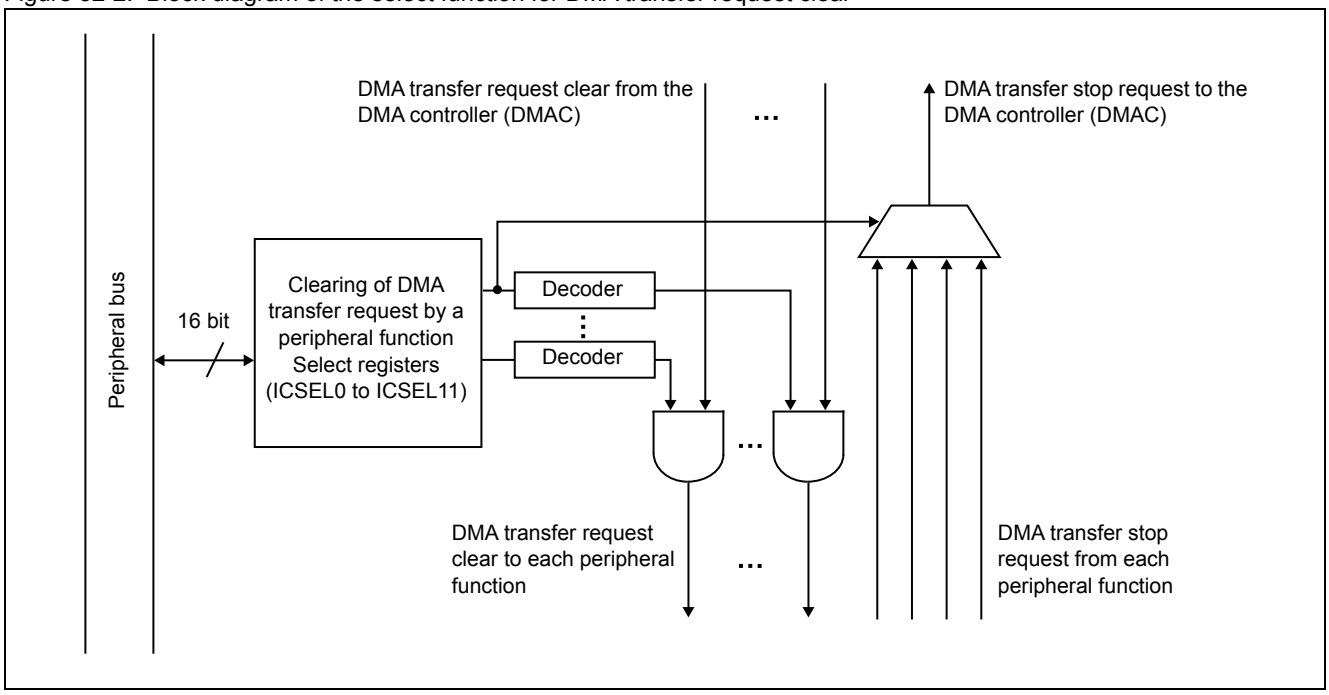
Figure 32-1. Block Diagram showing the Use of an Interrupt Request of Peripheral Functions



Block diagram of the select function for DMA transfer request clear

Figure 32-2 is a block diagram of the select function for DMA transfer request clear.

Figure 32-2. Block diagram of the select function for DMA transfer request clear



Select Register for DMA transfer request clear by a peripheral function (ICSEL0 to ICSEL11)

This register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

32.3 Registers

This section explains the configurations and functions of registers of the select function for DMA transfer request generation/clear.

List of registers of the part at which a DMA transfer request is generated by a peripheral function

Table 32-1 is a list of the registers of the part at which a DMA transfer request is generated.

Table 32-1. Registers of the part at which a DMA transfer request is generated by a peripheral function

Channel of DMAC	Abbreviated Register Name	Register Name	Reference
0	IORR0	IO-data request register 0	32.3.1
1	IORR1	IO-data request register 1	32.3.1
2	IORR2	IO-data request register 2	32.3.1
3	IORR3	IO-data request register 3	32.3.1
4	IORR4	IO-data request register 4	32.3.1
5	IORR5	IO-data request register 5	32.3.1
6	IORR6	IO-data request register 6	32.3.1
7	IORR7	IO-data request register 7	32.3.1

List of registers of the select function for DMA transfer request clear

Table 32-2 shows a list of the registers of the select function for DMA transfer request clear.

Table 32-2. List of registers of the select function for DMA transfer request clear

Channel	Abbreviated Register Name	Register Name	Reference
Common	ICSEL0	Select register 0 for DMA transfer request clear by a peripheral function	32.3.2
	ICSEL1	Select register 1 for DMA transfer request clear by a peripheral function	32.3.3
	ICSEL4	Select register 4 for DMA transfer request clear by a peripheral function	32.3.4
	ICSEL6	Select register 6 for DMA transfer request clear by a peripheral function	32.3.5
	ICSEL7	Select register 7 for DMA transfer request clear by a peripheral function	32.3.6
	ICSEL8	Select register 8 for DMA transfer request clear by a peripheral function	32.3.7
	ICSEL10	Select register 10 for DMA transfer request clear by a peripheral function	32.3.8
	ICSEL11	Select register 11 for DMA transfer request clear by a peripheral function	32.3.9

32.3.1 IO-Data Request Registers (IORR0 to IORR7)

This register sets which interrupt request of peripheral functions is to be the source that triggers a DMA transfer request generation when the source is set as an interrupt request of a peripheral function.

This register is provided for each channel of the DMA controller (DMAC).

Figure 32-3 shows the bit configuration of the I/O-data request registers (IORR0 to IORR7).

Figure 32-3. Bit Configuration of the I/O-data Request Registers (IORR0 to IORR7)

bit	7	6	5	4	3	2	1	0
	Reserved	IOE	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
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

R/W: Read/Write

Note: This register becomes enabled when the source that triggers a DMA transfer request generation is set as an interrupt request of a peripheral function (RS1, RS0 = 01) in the RS1 and RS0 bit of the DMA channel control registers (DCCR0 to DCCR7).

[bit7]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit6]: IOE (Transfer request enable bit)

This bit sets whether a DMA transfer request is output to the DMA controller (DMAC) of the corresponding channel when an interrupt request specified in bits ranging from IOS5 to IOS0 is generated.

Written Value	Explanation
0	A DMA transfer request is not output. (An interrupt request from a peripheral function is not used as a DMA transfer request.)
1	A DMA transfer request is output. (An interrupt request from a peripheral function is used as a DMA transfer request.)

[bit5 to bit0]: IOS5 to IOS0 (Transfer request selection bit)

These bits set which interrupt request generated from a peripheral function is to be used as a transfer request source by the channel of the DMA controller (DMAC) corresponding to this register.

IOS5 to IOS0	Interrupt Vector Number		Peripheral Function
	Decimal	Hexadecimal	
000000	16	10	External interrupt request ch.0 to ch.7
000001	17	11	External interrupt request ch.8 to ch.15
000010	18	12	-
000011	19	13	-
000100	20	14	16-bit reload timer ch.0 to ch.2
000101	21	15	Reception interrupt request from UART/CSIO ch.0
000110	22	16	Transmission interrupt request from UART/CSIO ch.0 Transmission bus idle interrupt request from UART/CSIO ch.0
000111	23	17	Reception interrupt request from UART/CSIO/I^2C ch.1
001000	24	18	Transmission interrupt request from UART/CSIO/I^2C ch.1 Transmission bus idle interrupt request from UART/CSIO ch.1
001001	25	19	-
001010	26	1A	Reception interrupt request from UART/CSIO/I^2C ch.2
001011	27	1B	Transmission interrupt request from UART/CSIO/I^2C ch.2 Transmission bus idle interrupt request from UART/CSIO ch.2
001100	28	1C	-
001101	29	1D	Reception interrupt request from UART/CSIO/I^2C ch.3
001110	30	1E	Transmission interrupt request from UART/CSIO/I^2C ch.3 Transmission bus idle interrupt request from UART/CSIO ch.3
001111	31	1F	-
010000	32	20	-
010001	33	21	-
010010	34	22	-
010011	35	23	-
010100	36	24	-
010101	37	25	-
010110	38	26	-
010111	39	27	Reception interrupt request from UART/CSIO/I^2C ch.8 to ch.11 Transmission interrupt request from UART/CSIO/I^2C ch.8 to ch.11 Transmission bus idle interrupt request from UART/CSIO ch.8 to ch.11 Transmission FIFO interrupt request from UART/CSIO/I^2C ch.8 to ch.11
011000	40	28	-
011001	41	29	Main timer interrupt request Sub timer interrupt request Underflow interrupt request from the watch counter
011010	42	2A	10-bit A/D converter ■ A/D scan conversion interrupt request ■ A/D priority conversion interrupt request
011011	43	2B	-
011100	44	2C	Edge detection interrupt request from 32-bit input capture ch.0 to ch.3
011101	45	2D	-
011110	46	2E	Base timer ch.0 ■ Underflow interrupt request ■ Overflow interrupt request ■ Duty match interrupt request ■ Trigger interrupt request ■ Measurement end interrupt request

IOS5 to IOS0	Interrupt Vector Number		Peripheral Function
	Decimal	Hexadecimal	
011111	47	2F	Base timer ch.1 ■ Underflow interrupt request ■ Overflow interrupt request ■ Duty match interrupt request ■ Trigger interrupt request ■ Measurement end interrupt request
100000	48	30	Base timer ch.2 ■ Underflow interrupt request ■ Overflow interrupt request ■ Duty match interrupt request ■ Trigger interrupt request ■ Measurement end interrupt request
100001	49	31	Base timer ch.3 ■ Underflow interrupt request ■ Overflow interrupt request ■ Duty match interrupt request ■ Trigger interrupt request ■ Measurement end interrupt request
100010	50	32	Base timer ch.4, ch.5 ■ Underflow interrupt request ■ Overflow interrupt request ■ Duty match interrupt request ■ Trigger interrupt request ■ Measurement end interrupt request
100011	51	33	Base timer ch.6, ch.7 ■ Underflow interrupt request ■ Overflow interrupt request ■ Duty match interrupt request ■ Trigger interrupt request ■ Measurement end interrupt request
100100	52	34	-
100101	53	35	-
100110	54	36	-
100111	55	37	-
101000	56	38	-
101001	57	39	-
101010	58	3A	-
101011	59	3B	-
101100	60	3C	-
101101	61	3D	-
101110	62	3E	-
101111	63	3F	-

Notes:

- If one interrupt vector number is used by multiple interrupt requests, only one interrupt request can be used as a DMA transfer request source.

  Disable the generation of an interrupt request which is not designated as a DMA transfer request source.
- If one interrupt vector number is used by multiple interrupt requests, set an interrupt request to clear the flag bit in the select register for DMA transfer request clear by a peripheral function (ICSEL0 to ICSEL11).
- Set an interrupt level for the interrupt request selected in this register so that values in the interrupt level mask register (ILM) and interrupt control registers (ICR00 to ICR47) indicate the following values.

  $ILM \leq ICR$

32.3.2 Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0)

The external interrupt request ch.0 to ch.7 is assigned to interrupt vector number 16 (decimal).

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-4 shows the bit configuration of select register 0 for DMA transfer request clear by a peripheral function (ICSEL0).

Figure 32-4. Bit Configuration of Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	EISEL02	EISEL01	EISEL00
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2 to bit0]: EISEL02 to EISEL00 (interrupt request select bit)

These bits select the flag bit to be cleared in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 16 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 16 (decimal), the bit selected here will be cleared.

EISEL02	EISEL01	EISEL00	Explanation	
			Interrupt Request Name	Flag Bit To Be Cleared
0	0	0	External interrupt request ch.0	EIRR0: ER0
0	0	1	External interrupt request ch.1	EIRR0: ER1
0	1	0	External interrupt request ch.2	EIRR0: ER2
0	1	1	External interrupt request ch.3	EIRR0: ER3
1	0	0	External interrupt request ch.4	EIRR0: ER4
1	0	1	External interrupt request ch.5	EIRR0: ER5
1	1	0	External interrupt request ch.6	EIRR0: ER6
1	1	1	External interrupt request ch.7	EIRR0: ER7

32.3.3 Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1)

The external interrupt request ch.8 to ch.15 is assigned to interrupt vector number 17 (decimal).

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-5 shows the bit configuration of select register 1 for DMA transfer request clear by a peripheral function (ICSEL1).

Figure 32-5. Bit Configuration of Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	EISEL12	EISEL11	EISEL10
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2 to bit0]: EISEL12 to EISEL10 (Interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 17 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 17 (decimal), the bit selected here will be cleared.

EISEL12	EISEL11	EISEL10	Explanation	
			Interrupt Request Name	Flag Bit To Be Cleared
0	0	0	External interrupt request ch.8	EIRR1: ER8
0	0	1	External interrupt request ch.9	EIRR1: ER9
0	1	0	External interrupt request ch.10	EIRR1: ER10
0	1	1	External interrupt request ch.11	EIRR1: ER11
1	0	0	External interrupt request ch.12	EIRR1: ER12
1	0	1	External interrupt request ch.13	EIRR1: ER13
1	1	0	External interrupt request ch.14	EIRR1: ER14
1	1	1	External interrupt request ch.15	EIRR1: ER15

32.3.4 Select Register 4 for DMA Transfer Request Clear by a Peripheral Function (ICSEL4)

The interrupt request of the 16-bit reload timer ch.0 to ch.2 is assigned to interrupt vector number 20 (decimal).

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-6 shows the bit configuration of select register 4 for DMA transfer request clear by a peripheral function (ICSEL4).

Figure 32-6. Bit Configuration of Select Register 4 for DMA Transfer Request Clear by a Peripheral Function (ICSEL4)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RTSEL1	RTSEL0
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: RTSEL1 to RTSEL0 (Interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 20 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 20 (decimal), the bit selected here will be cleared.

RTSEL1	RTSEL0	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Underflow interrupt request of the 16-bit reload timer ch.0	TMCSR0: UF
0	1	Underflow interrupt request of the 16-bit reload timer ch.1	TMCSR1: UF
1	0	Underflow interrupt request of the 16-bit reload timer ch.2	TMCSR2: UF
1	1		

32.3.5 Select Register 6 for DMA Transfer Request Clear by a Peripheral Function (ICSEL6)

The following interrupt requests are assigned to interrupt vector number 41 (decimal).

- Main timer interrupt request
- Sub timer interrupt request
- Underflow interrupt request from the watch counter

This register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-7 shows the bit configuration of select register 6 for DMA transfer request clear by a peripheral function (ICSEL6).

Figure 32-7. Bit Configuration of Select Register 6 for DMA Transfer Request Clear by a Peripheral Function (ICSEL6)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MWSEL1	MWSEL0
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: MWSEL1, MWSEL0 (Interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 41 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 41 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

MWSEL1	MWSEL0	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Main timer interrupt request	MTMCR: MTIF
0	1	Sub timer interrupt request	STMCR: STIF
1	0	Underflow interrupt request from the watch counter	WCCR: WCIF
1	1		

32.3.6 Select Register 7 for DMA Transfer Request Clear by a Peripheral Function (ICSEL7)

A priority conversion interrupt request and scan conversion interrupt request of the 10-bit A/D converter are assigned to interrupt vector number 42 (decimal).

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-8 shows the bit configuration of select register 7 for DMA transfer request clear by a peripheral function (ICSEL7).

Figure 32-8. Bit Configuration of the Select Register 7 for DMA Transfer Request Clear by a Peripheral Function (ICSEL7)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ADCSEL0
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit1]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit0]: ADCSEL0 (Interrupt request select bit)

This bit selects the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 42 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 42 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Priority conversion interrupt request from unit 0 of the 10-bit A/D converter	ADCR0: PCIF
1	Scan conversion interrupt request from unit 0 of the 10-bit A/D converter	ADCR0: SCIF

32.3.7 Select Register 8 for DMA Transfer Request Clear by a Peripheral Function (ICSEL8)

The interrupt request of the 32-bit input capture ch.0 to ch.3 is assigned to interrupt vector number 44 (decimal).

This register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-9 shows the bit configuration of select register 8 for DMA transfer request clear by a peripheral function (ICSEL8).

Figure 32-9. Bit Configuration of Select Register 8 for DMA Transfer Request Clear by a Peripheral Function (ICSEL8)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ICUSEL1	ICUSEL0
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: ICUSEL1, ICUSEL0 (Interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 44 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 44 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

ICUSEL1	ICUSEL0	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Edge detection interrupt request from 32-bit input capture ch.0	ICS01: ICP0
0	1	Edge detection interrupt request from 32-bit input capture ch.1	ICS01: ICP1
1	0	Edge detection interrupt request from 32-bit input capture ch.2	ICS23: ICP2
1	1	Edge detection interrupt request from 32-bit input capture ch.3	ICS23: ICP3

32.3.8 Select Register 10 for DMA Transfer Request Clear by a Peripheral Function (ICSEL10)

The interrupt request of the base timer ch.0 to ch.3 is assigned to interrupt vector number 46 to 49 (decimal).

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-10 shows the bit configuration of select register 10 for DMA transfer request clear by a peripheral function (ICSEL10).

Figure 32-10. Bit Configuration of Select Register 10 for DMA Transfer Request Clear by a Peripheral Function (ICSEL10)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	BTSEL03	BTSEL02	BTSEL01	BTSEL00
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit4]: Reserved bits

In case of writing	Always write "0" to this bit (these bits).
In case of reading	"0" is read.

[bit3]: BTSEL03 (Interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 49 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 49 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.3	BT3STC: UDIR BT3STC: DTIR BT3STC: OVIR
1	Interrupt request 1 from base timer ch.3	BT3STC: TGIR BT3STC: EDIR

[bit2]: BTSEL02 (Interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 48 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 48 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.2	BT2STC: UDIR BT2STC: DTIR BT2STC: OVIR
1	Interrupt request 1 from base timer ch.2	BT2STC: TGIR BT2STC: EDIR

[bit1]: BTSEL01 (Interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 47 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 47 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.1	BT1STC: UDIR BT1STC: DTIR BT1STC: OVIR
1	Interrupt request 1 from base timer ch.1	BT1STC: TGIR BT1STC: EDIR

[bit0]: BTSEL00 (Interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 46 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 46 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.0	BT0STC: UDIR BT0STC: DTIR BT0STC: OVIR
1	Interrupt request 1 from base timer ch.0	BT0STC: TGIR BT0STC: EDIR

32.3.9 Select Register 11 for DMA Transfer Request Clear by a Peripheral Function (ICSEL11)

The interrupt requests from base timer ch.4 and ch.5 are assigned to interrupt vector number 50 (decimal).

The interrupt requests from base timer ch.6 and ch.7 are assigned to interrupt vector number 51 (decimal).

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 32-11 shows the bit configuration of select register 11 for DMA transfer request clear by a peripheral function (ICSEL11).

Figure 32-11. Bit Configuration of Select Register 11 for DMA Transfer Request Clear by a Peripheral Function (ICSEL11)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	BTSEL13	BTSEL12	BTSEL11	BTSEL10
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

R/W: Read/Write

Note: Do not change this register during DMA transfer.

[bit7 to bit4]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit3, bit2]: BTSEL13, BTSEL12 (Interrupt request select bit)

These bits select either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 51 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 51 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

BTSEL13	BTSEL12	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Interrupt request 0 from base timer ch.6	BT6STC: UDIR BT6STC: DTIR BT6STC: OVIR
0	1	Interrupt request 1 from base timer ch.6	BT6STC: TGIR BT6STC: EDIR
1	0	Interrupt request 0 from base timer ch.7	BT7STC: UDIR BT7STC: DTIR BT7STC: OVIR
1	1	Interrupt request 1 from base timer ch.7	BT7STC: TGIR BT7STC: EDIR

[bit1, bit0]: BTSEL11, BTSEL10 (Interrupt request select bit)

These bits select interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 50 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 50 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

BTSEL11	BTSEL10	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Interrupt request 0 from base timer ch.4	BT4STC: UDIR BT4STC: DTIR BT4STC: OVIR
0	1	Interrupt request 1 from base timer ch.4	BT4STC: TGIR BT4STC: EDIR
1	0	Interrupt request 0 from base timer ch.5	BT5STC: UDIR BT5STC: DTIR BT5STC: OVIR
1	1	Interrupt request 1 from base timer ch.5	BT5STC: TGIR BT5STC: EDIR

32.4 An Explanation of Operations and Setting Procedure Examples

This section explains operations and setting procedure examples for activating a DMA transfer by using an interrupt request of peripheral functions.

32.4.1 Operations upon a DMA Transfer

Setting

To select an interrupt request of a peripheral function as a transfer request source of DMA transfer, the interrupt vector number selections and the peripheral function settings are required.

The setting procedure is as follows:

1. Select the interrupt vector number (IO-data request registers [IORR0 to IORR7]).
   a. Write values corresponding to the interrupt vector numbers to the bits ranging from IOS5 to IOS0.
   b. Enable the activation of a DMA transfer in the IOE bit by using interrupt request from peripheral functions (IOE = 1).
2. Select an interrupt request to be cleared with the DMA controllers (DMAC) (select registers for DMA transfer request clear by a peripheral circuitry [ICSEL0 to ICSEL11]).
3. Set the DMA controller (DMAC).
   For details, see "31. DMA Controller (DMAC)".
   a. Set a transfer request source of DMA transfer to an interrupt request of the peripheral functions.
   b. Enable DMA transfer operation and set it to a state of transfer request wait.
4. Set peripheral functions.
   See the chapters corresponding to the peripheral functions to be used.
   a. Clear the flag of an interrupt request to be used for a DMA transfer.
   b. Enable the generation of an interrupt request to be used for a DMA transfer.

Notes:

- An interrupt request flag of peripheral functions is cleared by the DMA controller (DMAC). Therefore, it is not possible to use it as an interrupt request of peripheral functions.
  Set the interrupt level to "31" (interrupt is disabled) for an interrupt request to be used as a transfer request source of DMA transfer.
  For information on the interrupt level settings, see "10. Interrupt Controller".
- When peripheral functions are set, clear an interrupt request flag first, and then enable the generation of an interrupt request.

Operation

Operations are as follows:

1. Peripheral functions are activated.
2. An interrupt request to be a DMA transfer request source is generated in the peripheral functions.
3. A DMA transfer request is generated, and the DMA controller (DMAC) is activated.
4. A clear of an interrupt request flag in the peripheral functions is requested from DMA controller (DMAC) for the block size multiplied by the number of transfers per transfer.
5. DMA transfer is finished.

Note: Set the interrupt level so that the values of the interrupt level mask register (ILM) and interrupt control registers (ICR00 to ICR47) indicate the following values when interrupt requests are generated:
ILM ≤ **ICR**
If the value of the interrupt level mask registers (ILM) is greater than the value of the interrupt control registers (ICR00 to ICR47), the interrupt request generation operation of the peripheral functions will be established and also enable a DMA transfer request generation. However, this will make interrupt request processing operation unstable.

33. Control of Built-in Program Memory

This series microcontrollers contain flash memory or mask ROM as built-in program memory. This chapter explains the register settings for using the built-in program memory.

33.1 Overview of Built-in Program Memory Controller

33.2 Register for Built-in Program Memory Controller

33.1 Overview of Built-in Program Memory Controller

This series microcontrollers contain flash memory or mask ROM as built-in program memory.

Overview

The register shown below needs to be set to use the built-in program memory.

This register needs to be set regardless of whether the microcontroller uses flash memory products or mask ROM products.

- FLASH control register (FCTLR)

When flash memory products are used, see also "34. Flash Memory".

Clock

Table 33-1 lists the clock used for the built-in program controller.

Table 33-1. Clock used for the built-in program controller

Clock Name	Description
Operation clock	Source clock (SRCCLK)

33.2 Register for Built-in Program Memory Controller

This section explains the register configuration and function of the built-in program memory controller.

Register for built-in program memory controller

Table 33-2 lists the register for the built-in program memory controller.

Table 33-2. Register for built-in program memory controller

Abbreviated Register Name	Register Name	Reference
FCTLR	FLASH control register	33.2.1

33.2.1 FLASH Control Register (FCTLR)

The FLASH control register controls access to the built-in program memory.

Figure 33-1 shows the bit configuration of the FLASH control register (FCTLR).

Figure 33-1. Bit Configuration of FLASH Control Register (FCTLR)

bit	15	14	13	12	11	10	9	8
	Reserved	FWE	Undefined	Undefined	FSZ1	FSZ0	FWC1	FWC0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	1	0	1	1

bit	7	0
	Reserved	
Attribute	R/W	
Initial value	0	

R/W: Read/Write

Notes:

Built-in program memory cannot be accessed normally if a bit other than FWC1 or FWC0 of this register is rewritten during access. Be sure to rewrite the bit other than FWC1 or FWC0 of the register while the built-in program memory is not being accessed, as follows:

- At read access: Immediately before read operation
- At command issue: When the FRDY bit of the FLASH status register (FSTR) is "1"

Do not write an instruction to change the bit other than FWC1 or FWC0 of the register value, in a program running in the built-in program memory area. Rewrite the bit other than FWC1 or FWC0 of the register with a program running in a built-in RAM or an external area.

[bit15]: Reserved bit

In case of writing	Always write "1" to this (these) bit (bits).
In case of reading	"1" is read.

[bit14]: FWE(FLASH write enable)

- With a flash memory product

  This bit sets the access mode by enabling/disabling write to flash memory.

Written Value	Explanation
0	Disables write. CPU ROM mode is set.
1	Enables write. CPU programming mode is set.

- With a mask ROM product

  This bit is reserved bit.

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit13, bit12]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit11, bit10]: FSZ1, FSZ0 (FLASH access size setting bits)

These bits set the size (bit width) of access to built-in program memory.

FS21	FS20	Access Size
0	0	Setting prohibited
0	1	16 bit
1	0	32 bit
1	1	Setting prohibited

Set FSZ [1:0] of FCTLR as "10" (32-bit) before use.

The access size that can be set varies depending on the access mode (read access or write access) of the built-in program memory.

	Read from	Written to
CPU ROM mode	32 bit	-
CPU programming mode	16 bit	16 bit

[bit9, bit8]: FWC1, FWC0(FLASH wait setting bits)

These bits set the interval (wait cycle) at which a request to read the built-in program memory is issued.

FWC1	FWC0	Wait Cycle
0	0	Setting prohibited
0	1	1
1	0	2
1	1	3

[bit7 to bit0]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

34. Flash Memory

This chapter explains the functions and operations of flash memory.

34.1 Overview

34.2 Configuration

34.3 Registers

34.4 Flash Memory Access Mode

34.5 Automatic Programming Algorithm

34.6 Explanation of Flash Memory Operation

34.7 Notes on Using Flash Memory

34.1 Overview

The size of the flash memory built in this series microcontrollers is 512 KB.

Data can be erased in units of sectors or in all sectors altogether from the CPU. Data can also be written in units of half words.

Overview

The flash memory built in this series microcontrollers allows an access mode to be selected from 3 modes: 2 CPU modes and 1 ROM writer mode.

- CPU mode

  Flash memory is used as memory for storing CPU programs and data. The following two CPU modes are available:

  - CPU programming mode

    In this mode, flash memory data can be read, written, or erased (automatic programming algorithm[1]). Because word access is disabled, programs in flash memory cannot be executed in this mode. Half word access is enabled.

  - CPU ROM mode

    In this mode, flash memory data is only read. Word access is enabled. This mode, however, does not support writing, erase, and activating the automatic programming algorithm.

- ROM writer mode

  The ROM writer can read, write, or erase flash memory data (automatic programming algorithm *)

[1]: Automatic programming algorithm = Embedded Algorithm

Note: This manual describes flash memory that is used in CPU mode.
For details of access to flash memory from the ROM writer, see the instruction manual for the ROM writer used.

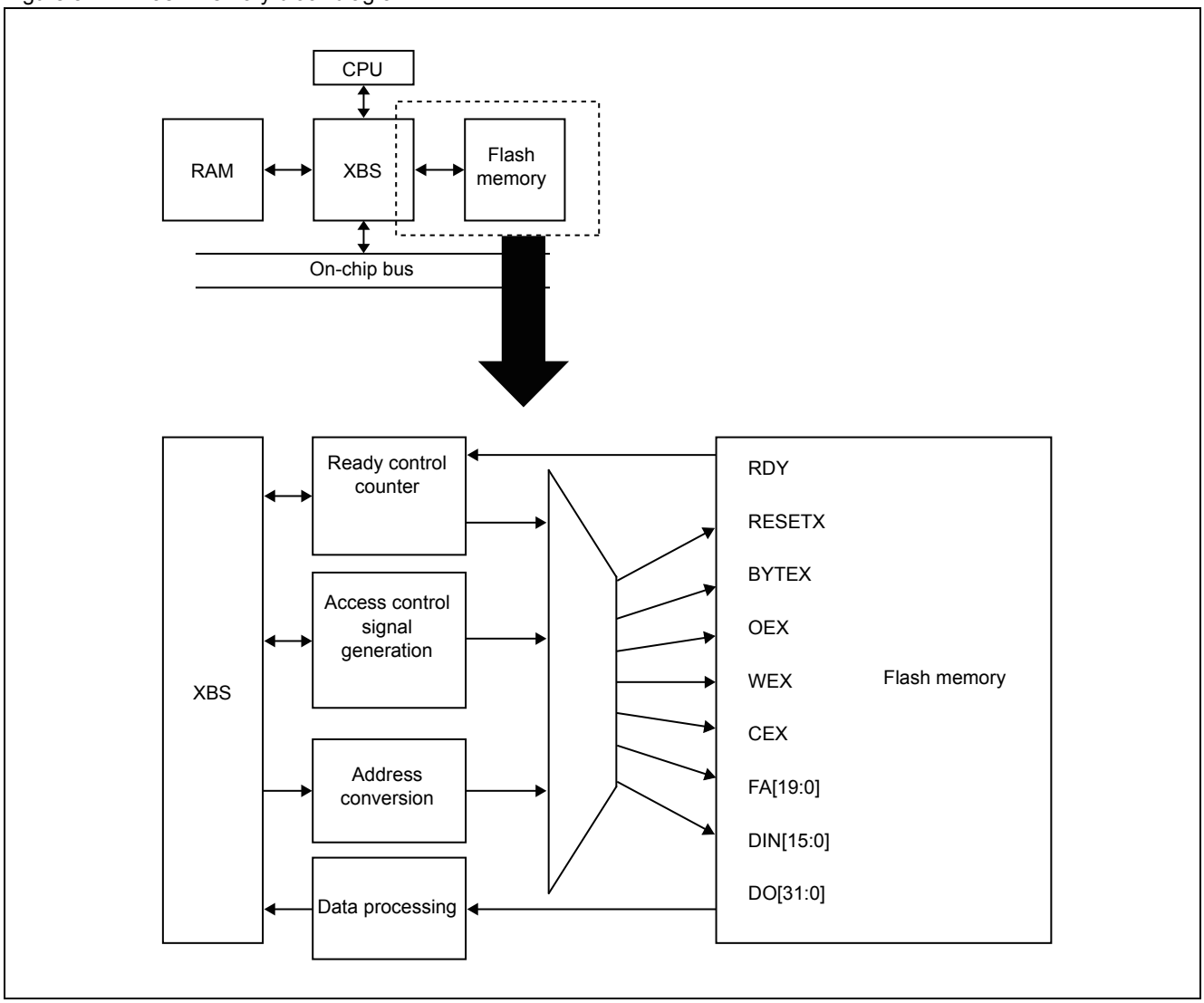
34.2 Configuration

This section explains the block configuration of flash memory.

Flash memory block diagram

Figure 34-1 is a flash memory block diagram.

Figure 34-1. Flash memory block diagram

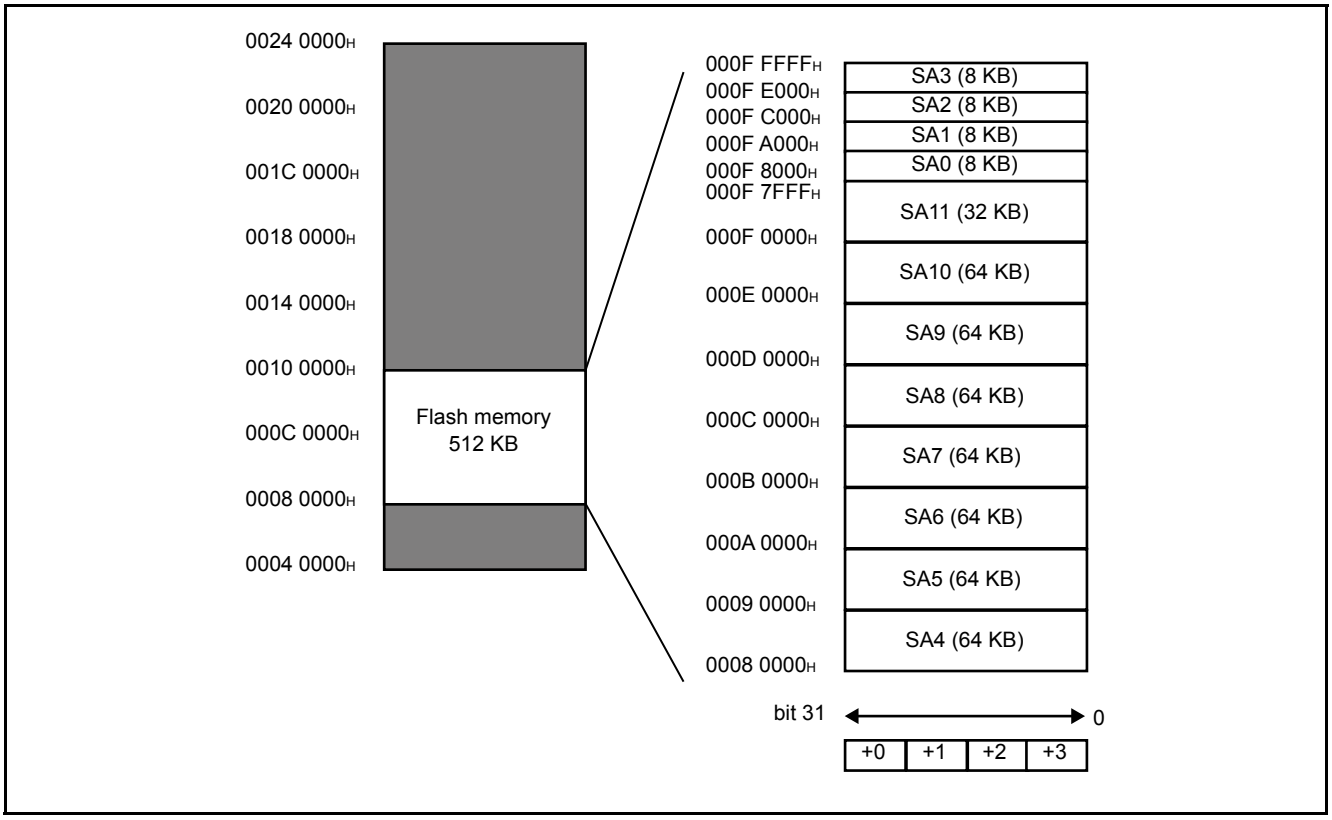


Flash memory sector configuration

The following shows the sector configuration of each capacity of flash memory.

Figure 34-2 shows the memory map of 512 KB flash memory.

Figure 34-2. Memory map (512 KB-byte flash memory)



Clock

Table 34-1 lists the clock used for flash memory.

Table 34-1. Clock used for flash memory

Clock Name	Description
Operation clock	Source clock (SRCCLK)

34.3 Registers

This section explains the configuration and functions of registers used for flash memory.

Flash memory registers

Table 34-2 lists the registers used for flash memory.

Table 34-2. Flash memory registers

Abbreviated Register Name	Register Name	Reference
FSTR	FLASH status register	34.3.1
FCTLR	FLASH control register	34.3.2

34.3.1 FLASH Status Register (FSTR)

This register indicates the state of flash memory.

Figure 34-3 shows the bit configuration of the FLASH status register (FSTR).

Figure 34-3. Bit Configuration of FLASH Status Register (FSTR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FRDY
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	1

R: Read only

[bit7 to bit1]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit0]: FRDY (Flash write enable bit)

This bit indicates whether flash memory writing or erase is in progress or complete using the automatic programming algorithm.When such operation is in progress, data cannot be written to or erased from flash memory.

Read Value	Explanation
0	In progress (writing/erase disabled)
1	Complete (writing/erase enabled)

- With a mask ROM product
  "1" is always read.

34.3.2 FLASH Control Register (FCTLR)

The FLASH control register controls access to the built-in program memory.

For details of this register, see "33.2.1 FLASH Control Register (FCTLR)".

34.4 Flash Memory Access Mode

The CPU accesses flash memory in one of the following two access modes:

- CPU programming mode
- CPU ROM mode

Overview

The access mode can be set using the FWE bit of the FLASH control register (FCTLR).

- CPU ROM mode (FWE=0)

  Flash memory data is only read in this mode. Because word access is enabled, 32-bit data can be read at one time.

  This mode, however, does not support writing or erasing commands or data, or activating the automatic programming algorithm.

  Note: This mode is set upon the release of resetting.

- CPU programming mode (FWE=1)

  Data can be read, written, or erased in this mode. Because word access is disabled in this mode, programs in flash memory cannot run when this mode is enabled. The CPU performs access as shown below.

  - In case of reading

    The CPU performs half word accesses to flash memory to read 16 bits of data at one time.

  - In case of command writing

    The CPU activates the automatic programming algorithm to write or erase flash memory data.For details of the automatic programming algorithm, see "34.5 Automatic Programming Algorithm".

  Note: CPU ROM mode is set when resetting is released during CPU operation. To enable this mode, write "1" to the FWE bit after releasing resetting. If resetting occurs after the CPU programming mode is set, the FWE bit changes to "0" and the ROM mode is restored.

34.5 Automatic Programming Algorithm

In CPU programming mode, the automatic programming algorithm is activated to write data to or erase data from flash memory.

This section explains the automatic programming algorithm.

34.5.1 Command Sequence

Writing half word (16-bit) data 1 to 6 times consecutively to flash memory activates the automatic programming algorithm. This operation is called the command. lists the command sequence.

Table 34-3. Command Sequence

Command	Writing Count	1st Time		2nd Time		3rd Time		4th Time		5th Time		6th Time	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read/Reset	1	$FXXXX_H$	$F0F0_H$	RA	RD	--	--	--	--	--	--	--	--
Read/Reset	4	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	$F0F0_H$	RA	RD	--	--	--	--
Writing	4	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	$A0A0_H$	PA	PD	--	--	--	--
Chip erase	6	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	8080_H	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	1010_H
Sector erase	6	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	8080_H	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	SA	3030_H
Sector erase suspended	1	$FXXXX_H$	$B0B0_H$	--	--	--	--	--	--	--	--	--	--
Sector erase restarting	1	$FXXXX_H$	3030_H	--	--	--	--	--	--	--	--	--	--
Continuous mode	3	$F5556_H$	$AAAA_H$	$FAAAA_H$	5555_H	$F5556_H$	2020_H	--	--	--	--	--	--
Continuous writing	2	$FXXXX_H$	$A0A0_H$	PA	PD	--	--	--	--	--	--	--	--
Reset continuous mode	2	$FXXXX_H$	9090_H	$FXXXX_H$	$F0F0_H$ or 0000_H	--	--	--	--	--	--	--	--

RA: Read address PA: Write address SA: Sector address (*) RD: Read data PA: Address PD: Write data

Notes:

- Always write half word. (An address applicable in CPU mode is written.)
- If an invalid address or invalid data is written or if an address or data is written in the incorrect order, the flash memory is reset to read mode.
- For details of the sector address, specify the low-order part of a 32-bit address space, the 4 lower bits of which indicate "2_H, "6_H ", "A_H", or "E_H ".

Read/reset command

Sending the read/reset command listed in to the target sector continuously can set the flash memory in the read/reset state.

Two types of read/reset commands, one including only 1 bus writing cycle and one including 4 bus writing cycles, are available. There are no essential differences between them.

When a read/reset command is issued, flash memory is kept in read state until another command is issued.

If a read/reset command is issued after execution of the automatic programming algorithm exceeds the timing limit, flash memory returns to the read/reset state. Read data from flash memory during a read cycle.

For details of the actual operation, see "34.6.1 Read/Reset Operation".

Note: When the device power is turned on, flash memory is automatically set in the read/reset state. In this case, a read/reset command need not be issued. Issue a read/reset command if another command fails to end normally or when the automatic programming algorithm needs to be initialized.

Program (write) command

Sending the write command listed in four times consecutively to the target sector activates the automatic programming algorithm to write data to flash memory. Data can be written in any order of addresses or even beyond sector boundaries.

Write with half word in CPU programming mode.

After finishing writing four times as described in , the automatic programming algorithm is activated and then automatic writing to flash memory begins.

After execution of the automatic write algorithm command sequence, flash memory need not be controlled externally.

For details of the actual operation, see "34.6.2 Write Operation".

Notes:

- If the fourth write command (write data cycle) is written at an odd-numbered location, data cannot be written normally. Always write it at an even-numbered location.
- 1 write command sequence can write only one piece of half word data. To write two or more data items, issue 1 write command sequence for each data item.

Chip erase command

Sending the chip erase command listed in six times consecutively to the target sector can erase all sectors in flash memory all at once.

After writing six times as described in is finished, the automatic programming algorithm is activated and then chip erase begins.

When the automatic erase algorithm is activated, the flash memory, before erasing the entire chip data, writes "0" to all cells in the chip to perform margin verification (preprogram). Therefore, the flash memory need not be written before chip erase.

During margin verification, the flash memory need not be controlled externally.

For details of the actual operation, see "34.6.3 Chip Erase".

Sector erase command

Sending the sector erase command listed in six times consecutively to the target sector can erase the sector in flash memory.

When 50μs have passed after the sixth writing as listed in is finished (timeout period), the automatic programming algorithm is activated and then sector erase begins.

To erase two or more sectors, write the erase code (3030_H) at the addresses of the sectors to be erased within 50μs (timeout period). If the next sector is not entered within the timeout period, the sector erase command may be disabled.

When the automatic erase algorithm is activated, the flash memory, before executing sector erase, writes "0" to the cells of the sector to be erased to perform margin verification (preprogram). Therefore, the flash memory need not be written before sector erase.

During margin verification, the flash memory need not be controlled externally.

For details of the actual operation, see "34.6.4 Sector Erase".

Sector erase suspend command

If the sector erase suspend command listed in is sent to the target sector during sector erase, sector erase is suspended so that data can be read from or written to sectors other than the one being erased.

If this command is issued during the timeout period after a sector erase command, the timeout is immediately finished and the erase operation is stopped. Note, however, that it takes a maximum of 20μs from when this command is issued to when sector erase is actually stopped.

For details of the actual operation, see "34.6.5 Sector Erase Suspending".

Note: This command is valid only during sector erase. The command is ignored if it is issued during chip erase or writing.

Sector erase restart command

Sending the sector erase restart command listed in consecutively to the target sector can release the sector erase suspended state and restart sector erase.

For details of the actual operation, see "34.6.6 Sector Erase Restarting".

Note: This command is valid only while sector erase is temporarily stopped with the sector erase suspend command. The command is ignored if it is issued during sector erase.

34.5.2 Execution State of Automatic Programming Algorithm

Flash memory implements writing and erase based on the automatic programming algorithm. The FRDY bit of the FLASH status register (FSTR) can be used to verify whether the automatic programming algorithm is in progress or the hardware sequence flag can be used to verify the operating state of the automatic programming algorithm.

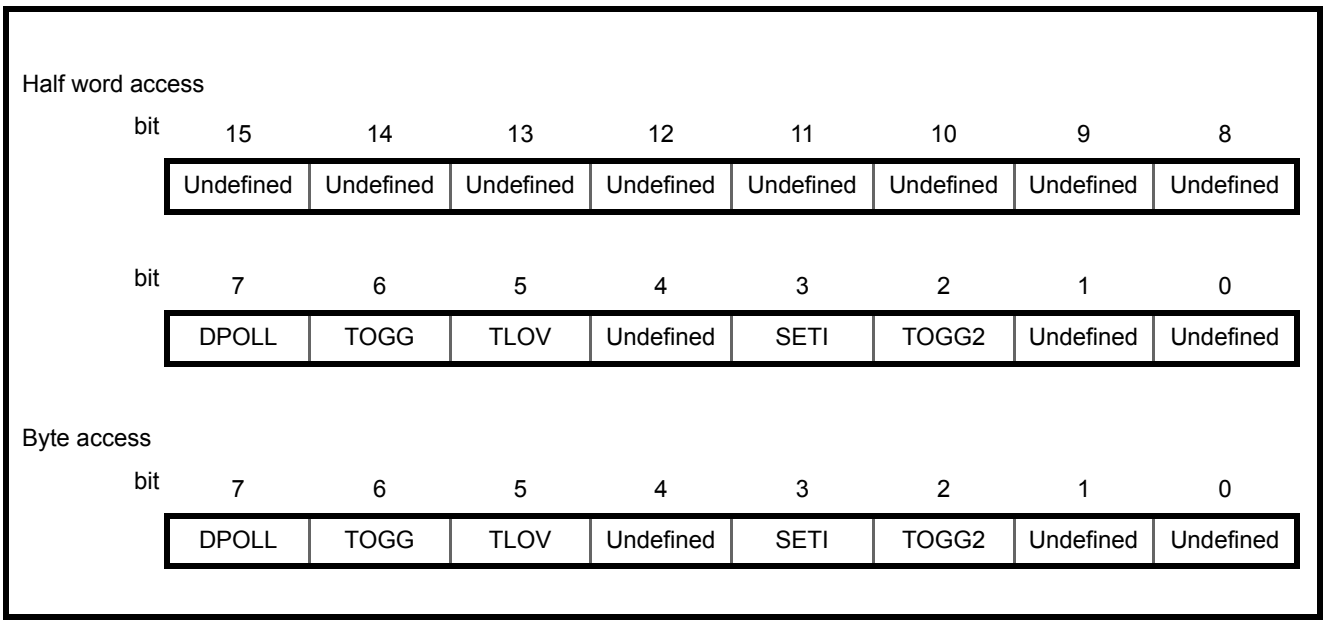
For details of the FRDY bit of the FLASH status register (FSTR), see "34.3.1 FLASH Status Register (FSTR)".

Hardware sequence flag

The hardware sequence flag shows the state of the automatic programming algorithm. The operating status can be verified by reading an arbitrary address in flash memory when the FRDY bit of the FLASH status register (FSTR) is "0".

Figure 34-4 shows the bit configuration of the hardware sequence flag.

Figure 34-4. Bit Configuration of Hardware Sequence Flag



Notes:

- Word access is disabled for reading. Be sure to execute half word or byte access for reading in CPU programming mode.
- Even if an arbitrary address is read in CPU ROM mode, the hardware sequence flag cannot be read.
- Read an odd-numbered address for reading by byte access.

■ Correspondence between bits and flash memory states

Table 34-4 lists the correspondence between hardware sequence flag bits and flash memory states.

Table 34-4. Correspondence Between Flags and Flash Memory States

State			DPOLL	TOGG	TLOV	SETI	TOGG2
In progress	Writing		Inverted data[1]	Toggle	0	0	1
	Sector/chip erase in progress		0	Toggle	0	1	Toggle
	Sector erase suspended	Read (Sector whose erase is suspended)	1	1	0	0	Toggle[2]
		Read (Sector other than one whose erase is suspended)	Data[1]	Data[1]	Data[1]	Data[1]	Data[1]
		Writing (Sector whose erase is suspended)	Inverted data[1]	Toggle[3]	1	0	1[4]
Time limit exceeded	Sector/chip erase command		Inverted data[1]	Toggle	1	0	1
	Sector erase suspended		0	Toggle	1	1	[5]
	Write operation while sector erase is suspended		0	Toggle	1	1	[5]

[1]: For details of the values to be read, see "Explanation of bits".
[2]: Continuously reading addresses involved in the sector whose erase is suspended causes toggle operation that outputs "1" and "0" alternately.
[3]: Regardless of the specified address, continuous reading causes toggle operation that outputs "1" and "0" alternately.
[4]: Data may be written to a sector while erase of another sector is suspended. In this case, reading the sector being written outputs "1", or reading the sector whose erase is suspended causes toggle operation that outputs "1" and "0" alternately.
[5]: Continuously reading a sector being written or erased while the TLOV bit is "1" causes toggle operation that outputs "1" and "0" alternately. However, reading another sector does not cause toggle operation.

- Explanation of bits

  [bit15 to bit8]: Undefined bits

  [bit7]: DPOLL: (Data polling flag bit)

  This bit indicates whether the automatic programming algorithm is in progress by the data polling function when the hardware sequence flag is read with an arbitrary address specified.

  The read value varies depending on the operating status.

  - In case of writing
    1. During writing:

       The value (inverted data) opposite the value of bit7 of the data written last is read.

       The address specified for reading the hardware sequence flag is not accessed.
    2. After writing is finished:

       The value of bit7 of the address specified for reading the hardware sequence flag is read.
  - In case of sector erase
    1. During sector erase: "0" is read from the sector being erased.
    2. After sector erase: "1" is always read.
  - In case of chip erase
    1. During chip erase: "0" is always read.
    2. After chip erase: "1" is always read.
  - In case of section erase suspended

    The sector being suspended or erased can be verified by seeing this bit and TOGG bit.
    1. When this bit is read by specifying the address of the sector being erased, in erase suspended read mode: "1" is read.
    2. When this bit is read by specifying an address other than the address of the sector being erased, in erase suspended read mode:
       The value of bit7 of the specified address is read.
    3. When this bit is read by specifying the address of the sector being erased, in erase suspended write mode:
       The value (inverted data) opposite the value of bit7 of the data in the sector being erased is read.

    Note: While the automatic programming algorithm is activated, the data at the specified address cannot be read. Read the data after checking this bit to verify that the automatic programming algorithm has ended operation.

  [bit6]: TOGG: (Toggle flag bit)

  This bit indicates whether the automatic programming algorithm is in progress when the hardware sequence flag is read with an arbitrary address specified.

  The read value varies depending on the operating state.

  - In case of writing, sector erase, or chip erase
    1. During writing, sector erase, or chip erase
       If this bit is read continuously, "1" and "0" are read alternately (toggle operation).
       The address specified for reading the hardware sequence flag is not accessed.
    2. After writing, sector erase, or chip erase is finished:
       The value of bit6 of the address specified for reading the hardware sequence flag is read.
  - In case of section erase suspended
    1. When this bit is read by specifying the address of the sector being erased: "1" is read.
    2. When this bit is read by specifying an address other than that of the sector being erased:

  The value of bit6 of the specified address is read.

  Notes:

  - If an attempt is made to write to a write-protected sector, toggle operation is performed for about 2μs and then finished without rewriting data.
  - If all of the selected sectors are write-protected during sector erase, toggle operation is performed for about 100μs and then the read/reset state is restored without erasing data.

[bit5]: TLOV: (Timing limit overrun flag bit)

This bit indicates whether the execution time of the automatic programming algorithm exceeds the duration (internal pulse count) specified in the flash memory when the hardware sequence flag is read with an arbitrary address specified.

The read value varies depending on the operating state.

- In case of writing, sector erase, or chip erase
  One of the following values is read:

Read Value	Explanation
0	Within the time limit
1	Beyond the time limit

If the DPOLL or TOGG bit indicates that the automatic programming algorithm is in progress while this bit is "1", it indicates that writing or erase has failed.

For instance, because flash memory does not allow data "0" to be rewritten to "1", an attempt to write "1" to an address at which "0" has been written locks the flash memory, preventing the automatic programming algorithms from ending. In this event, the value of the DPOLL bit is kept invalid, and "1" and "0" are read alternately from the TOGG bit.

When the time limit is exceeded under this state, the TLOV bit changes to "1". If this bit becomes "1", issue a reset command.

Note: When this bit is "1", it indicates that flash memory has not been used correctly. It does not mean that the flash memory is faulty.
Issue a reset command and then take the appropriate action.

[bit4]: Undefined bit

[bit3]: SETI (Sector erase timer flag bit)

When a sector is erased, a timeout period of 50μs is required from when the sector erase command is issued to when sector erase actually begins.

This bit indicates whether it is within the timeout period of the sector erase command when the hardware sequence flag is read with an arbitrary address specified.

The read value varies depending on the operating state.

- In case of sector erase:

  When sectors are erased, this bit can be checked before entering the next sector erase code to verify whether the next sector erase code can be accepted.

  The address specified for reading the hardware sequence flag is not accessed. Instead, the next value is read.

Read Value	Explanation
0	In a sector erase wait period The next sector erase code (3030_H) can be accepted.
1	The sector erase wait period is exceeded.[1]

[1]: If the DPOLL or TOGG bit indicates that the automatic programming algorithm is in progress while this bit is "1", erase of flash memory internal data is started. In this case, the sector erase code (3030_H) and commands other than the erase suspend command are ignored until flash memory internal data is erased completely.

- In case of section erase suspended
  1. When this bit is read by specifying the address of the sector being erased: "1" is read.
  2. When this bit is read by specifying an address other than that of the sector being erased:

  The value of bit3 of the specified address is read.

[bit2]: TOGG2: (Toggle flag bit)

This bit indicates whether erase is in progress or suspended when the hardware sequence flag is read with an arbitrary address specified. Because this bit indicates toggle operation in erase suspend read mode, verifying both this bit and TOGG bit can detect whether the read operation is performed (erase suspend read mode) while sector erase is suspended.

This bit can also be used to detect a sector in erase suspended state.

What this bit indicates differs depending on the operating state.

- In case of writing, sector erase, or chip erase (same as the TOGG bit)
  1. During writing, sector erase, or chip erase
     If this bit is read continuously, "1" and "0" are read alternately (toggle operation).
     The address specified for reading the hardware sequence flag is not accessed.
  2. After writing, sector erase, or chip erase is finished:
     The value of bit6 of the address specified for reading the hardware sequence flag is read.
- In case of section erase suspended:
  1. When this bit is read by specifying an address of the sector in erase suspended state:
     If this bit is read continuously in erase suspend read mode, "1" and "0" are read alternately (toggle operation).
  2. When this bit is read by specifying an address other than that of the sector in erase suspended state:
     In erase suspend write mode, "1" is read from this bit.

[bit1, bit0]: Undefined bits

34.6 Explanation of Flash Memory Operation

This section explains flash memory operations for each command.

Overview

Issuing a command 1 to 6 times consecutively for flash memory activates the automatic programming algorithm to perform the following operations:

- Read/Reset
- Writing
- Chip erase
- Sector erase
- Sector erase suspension
- Erase restart

The hardware sequence flag can be used to verify the execution state of the automatic programming algorithm.

For details of the commands and execution state of the automatic programming algorithm, see "34.5 Automatic Programming Algorithm".

34.6.1 Read/Reset Operation

This section explains the flash memory read/reset state.

Sending the read/reset command to the target sector continuously can set the flash memory in the read/reset state.

This state is the initial state of flash memory. Flash memory always returns to the read/reset state when the power is turned on or a command ends normally. At power on, a data read command need not be issued. Also, in the read/reset state, ordinary read access can be used to read data or implement program access from the CPU and, therefore, a read/reset command need not be issued to read data.

For details of the read/reset command, see "34.5 Automatic Programming Algorithm".

34.6.2 Write Operation

This section explains flash memory write operation.

Write operation

The procedure for write operation is as follows:

1. Write commands are continuously sent to the target sector.

   The automatic programming algorithm is activated to write data to flash memory.

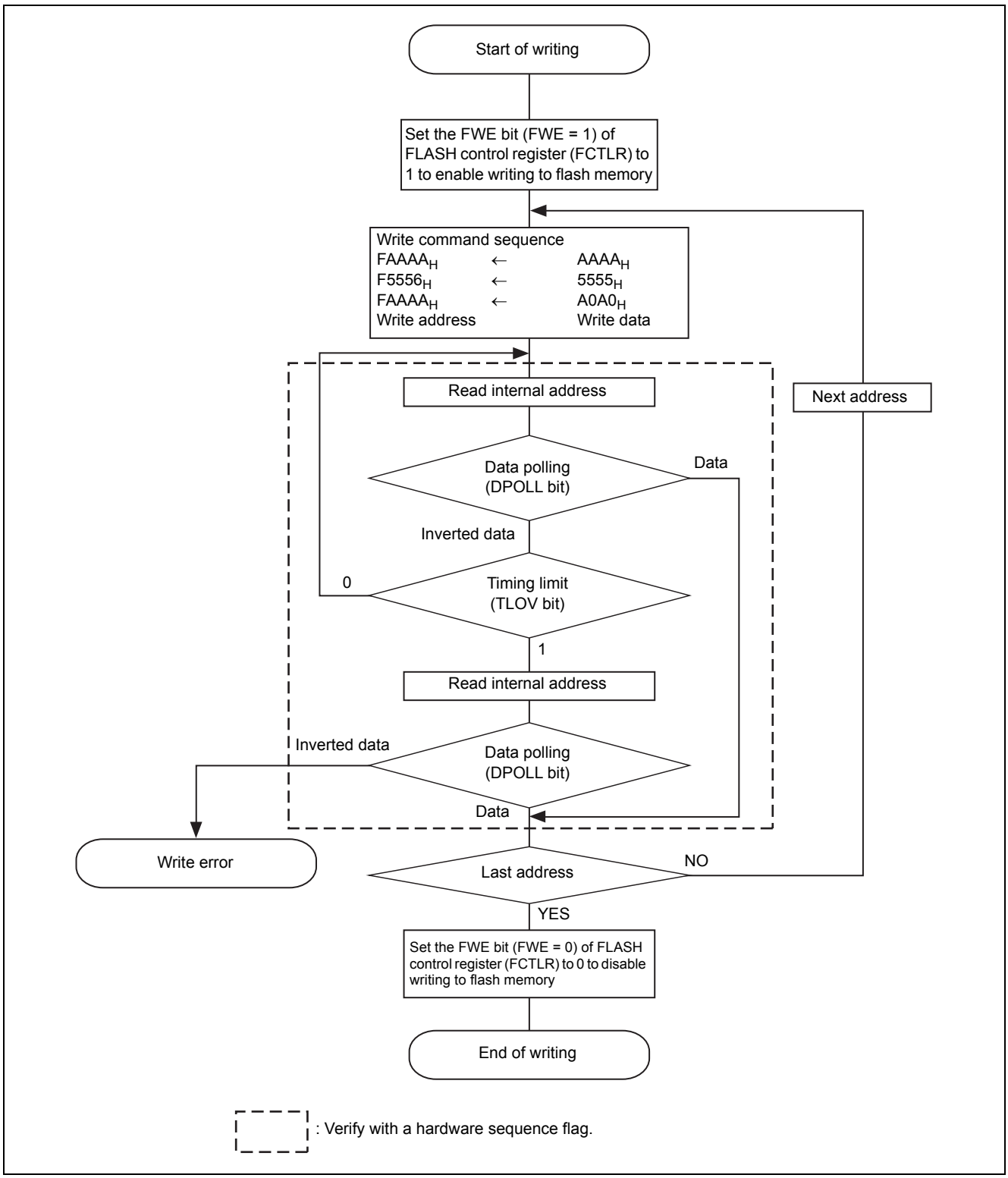
   After the write command is issued, flash memory need not be controlled externally.
2. A read access is made to the address at which data was written.

   The read data becomes a hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data matches the written value, it means that writing to the flash memory is finished.

   If writing has not been finished, the value (inverted data) opposite the value of bit7 of the data written last is read.

Figure 34-5 shows an example of the flash memory write operation.

Figure 34-5. Write Procedure Example



After writing is finished, flash memory returns to read mode and accepts no write address.

Notes:

- For details of the write command, see "34.5 Automatic Programming Algorithm".
- Because the DPOLL bit of hardware sequence flag changes the value at the same time as the TLOV bit, it must be verified again even when the TLOV bit is "1".
- Toggle operation stops simultaneously when the TOGG and TLOV bit of the hardware sequence flag change to "1". Therefore, even when the TLOV bit is "1", the TOGG bit must be verified again.
- Data can be written to flash memory in any order of addresses or even beyond sector boundaries, but 1 write command sequence can write only 1 half word data item. To write two or more data items, issue 1 write command sequence for each data item.

Notes on writing

- Data to which "0" is once written cannot be restored to "1". Rewriting "0" to "1" results in one of the following:
  - An element is judged as defective by the data polling algorithm.
  - The write time limit is exceeded and the TLOV bit of hardware sequence flag changes to "1".
  - It seems that "1" has been written.

  However, even when "1" seems to be written, the actual data remains "0" and therefore "0" is read by the read/reset mode command. To return data to "1", perform chip or sector erase.
- During write operation, the commands written to flash memory are all ignored.
- If the this device is reset during write operation, the data being written is not guaranteed.

34.6.3 Chip Erase

Flash memory sectors can be erased all at once. Erasing sectors all at once is called chip erase.

Sending chip erase commands continuously to target sectors can activate the automatic programming algorithm to erase all the sectors altogether.

For details of the chip erase command, see "34.5 Automatic Programming Algorithm".

1. Chip erase commands are continuously sent to the target sectors.

   The automatic programming algorithm is activated to write data to flash memory.
2. A read access is made to an arbitrary address.

   The read data becomes a hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", it means that chip erase is finished.

The duration required for chip erase is "sector erase time × total number of sectors + chip write time (preprogram)".

After chip erase is finished, flash memory returns to read/reset mode.

Note: When the automatic erase algorithm is activated, the flash memory, before erasing the entire chip data, writes "0" to all cells in the chip to perform margin verification (preprogram). Therefore, the flash memory need not be written before chip erase.
During margin verification, the flash memory need not be controlled externally.

34.6.4 Sector Erase

Only a specific sector that is selected from flash memory can be erased. Multiple sectors can be specified simultaneously.

The procedure for sector erase is as follows:

1. Sector erase commands are continuously sent to the target sectors.

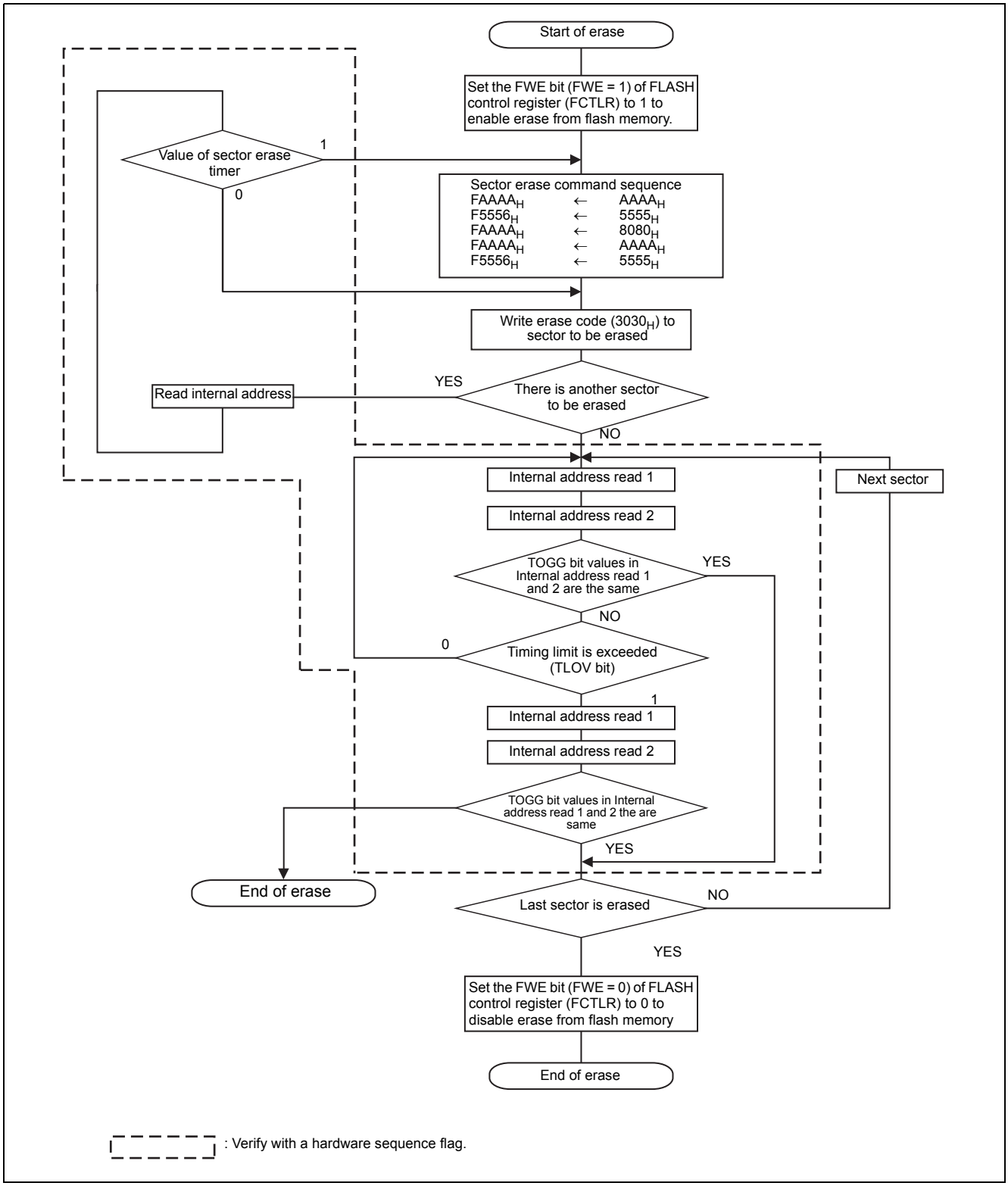
   50 μs later (timeout period), the automatic programming algorithm is activated to start sector erase.

   To erase two or more sectors, write the erase code (3030_H) at the addresses of the sectors to be erased within 50μs (timeout period). If it is written after the lapse of the timeout period, the sector erase command may be invalid.
2. A read access is made to an arbitrary address.

   The read data becomes a hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", it means that sector erase is finished.

The TOGG bit can also be used to verify whether sector erase is complete. Figure 34-6 shows an example of the sector erase procedure, in which the TOGG bit is used for verification.

Figure 34-6. Sector Erase Procedure Example



The duration required for sector erase is "(sector erase time + sector write time (preprogram)) × number of sectors".

After sector erase is finished, flash memory returns to read/reset mode.

Notes:

- For details of the sector erase command, see "34.5 Automatic Programming Algorithm".
- To specify a sector to be erased, specify an address (low-order side of 32 bits) whose 4 lower bits indicate 2_H, 6_H, A_H, or E_H.
- Because the DPOLL bit of hardware sequence flag changes the value at the same time as the TLOV bit, it must be verified again even when the TLOV bit is "1".
- The TOGG bit of the hardware sequence flag stops toggle operation simultaneously when the TLOV bit change to "1". Therefore, even when the TLOV bit is "1", the TOGG bit must be verified again.
- If a command other than the sector erase command and erase suspend command is issued during sector erase including the timeout period, flash memory is put in the read/reset state.

  In this event, flash memory is reset and, accordingly, the one or more sector erase commands that precede the issue of the command are disabled.

  To perform sector erase, issue sector erase commands again from the beginning.
- When the automatic erase algorithm is activated, the flash memory, before executing sector erase, writes "0" to the cells to be erased to perform margin verification (preprogram). Therefore, the flash memory need not be written before sector erase.

  During margin verification, the flash memory need not be controlled externally.

34.6.5 Sector Erase Suspending

Sector erase can be suspended so that data can be read from or written to sectors other than the sector being erased. Once sector erase is suspended, it is kept suspended until a sector erase restart command is issued.

If a sector erase suspend command is sent to the target sector while sector erase is stopped, sector erase is suspended so that data can be read from or written to sectors other than the sector being erased.

In this manual, reading data from another sector while sector erase is suspended is referred to as sector erase suspend read, and writing to another sector is referred to as sector erase suspend write.

Sector erase suspending

Sector erase is suspended by using the following procedure:

1. A sector erase suspend command is sent to the target sector during the period from the sector erase timeout period to sector erase.

   If the command is issued during the timeout period, the timeout period is immediately terminated and sector erase is stopped.

   If this command is issued during sector erase, it takes a maximum of 20 μs until sector erase is actually stopped.
2. A read access is made to the write address or the address at which sector erase was suspended.

   The read data becomes a hardware sequence flag. This means that sector erase is finished when "1" is read from bit7 (DPOLL bit) and bit6 (TOGG bit) of the read data.

   When sector erase is stopped, the FRDY bit of the FLASH status register (FSTR) changes to "1".

Notes:

- For details of the sector erase suspend command, see "34.5 Automatic Programming Algorithm".
- Sector erase can be suspended only in the period from sector erase timeout period to sector erase in progress. Chip erase cannot be suspended. A sector erase suspend command is ignored if it is issued while sector erase is suspended.

State after sector erase is suspended

- Sector erase suspend read mode

  After sector erase is suspended, sectors other than the one in the section erase suspended state can be read in the same way as usual. This state is referred to as sector erase suspend read mode.

  Notes:

  The sector in the sector erase suspended state cannot be read. If an attempt is made to read the sector in the sector erase suspended state, a hardware sequence flag is read. When the hardware sequence flag is read, each bit of the read data is as follows:

  - bit7 (DPOLL bit) and bit6 (TOGG bit): "1"
  - bit2 (TOGG2 bit): If this bit is read continuously, "1" and "0" are read alternately (toggle operation).
- Sector erase suspend write mode

  If a program (write) command is issued in sector erase suspend read mode, data can be written to sectors other than the one in the sector erase suspend state. This state is referred to as sector erase suspend write mode.

  Data can be written in the same way as in normal operation. Always write half word data.

  Notes:

  - No data can be written to the sector in the sector erase suspend state.
    If an attempt is made to read the sector in the sector erase suspended state in sector erase suspend write mode, a hardware sequence flag is read. When the hardware sequence flag is read, each bit of the read data is as follows:
    - bit6 (TOGG bit): If this bit is read continuously, "1" and "0" are read alternately (toggle operation).
    - bit2 (TOGG2 bit): If this bit is read continuously, "1" and "0" are read alternately (toggle operation).
  - If data is read from a sector other than the one whose erase is suspended in sector erase suspend write mode, bit 7 indicates the inverted value of the actual one.

34.6.6 Sector Erase Restarting

This section explains the operation for releasing the sector erase suspend state and restarting sector erase.

Sending a sector erase restart command to an arbitrary address while sector erase is suspended can restart sector erase.

When a sector erase restart command is issued, erase of the sector in the sector erase suspend state is restarted.

For details of the sector erase restart command, see "34.5 Automatic Programming Algorithm".

Note: The sector erase restart command is enabled only while sector erase is suspended. The sector erase restart command is ignored if it is issued during sector erase.

34.7 Notes on Using Flash Memory

Note the following points on using flash memory.

- If this device is reset during write operation, the data being written is not guaranteed.
- When the FWE bit of the FLASH control register (FCTLR) is set to 1 to enable CPU programming mode, do not run the program in the flash memory. The program fails to obtain normal values and hangs up.

  For details of the FLASH control register, see "33.2.1 FLASH Control Register (FCTLR)".
- Do not generate an interrupt request when an interrupt vector table exists in the flash memory while the FWE bit of the FLASH control register (FCTLR) is set to 1. The program fails to obtain normal values and hangs up.

  For details of the FLASH control register, see "33.2.1 FLASH Control Register (FCTLR)".
- When the FWE bit of the FLASH control register (FCTLR) is set to 1 to enable CPU programming mode, do not change to Sub-run mode or a lowerpower dissipation mode.

  For details of the FLASH control register, see "33.2.1 FLASH Control Register (FCTLR)".
- When the FWE bit of the FLASH control register (FCTLR) is set to 0 to enable CPU ROM mode (FWE = 0), do not write to flash memory.

  For details of the FLASH control register, see "33.2.1 FLASH Control Register (FCTLR)".
- When the FWE bit of the FLASH control register (FCTLR) is set to 1 to enable CPU programming mode, be sure to write half words to flash memory. Do not write data in bytes.

  For details of the FLASH control register, see "33.2.1 FLASH Control Register (FCTLR)".
- Do not write to flash memory continuously. To continue to write to flash memory, be sure to insert at least 1 "NOP" instruction.
- After writing data to flash memory, be sure to read dummy data and then read data actually required. The data read immediately after it is written cannot be guaranteed.

35. Wild Register

This chapter explains the functions and operations of the wild register.

35.1 Overview

35.2 Configuration

35.3 Registers

35.4 Explanation of Operations and Setting Procedure Examples

35.5 Notes on Using the Wild Register

35.1 Overview

The wild register has the function to replace the data at a patch target address.

This series microcontroller has 16 built-in channels of wild register that enable 16 pairs of patch target addresses and replacement data to be set.

Overview

The wild register function can be used to read memory data (instruction code/data) at the specified address by replacing it with the data in the predetermined register.

With this function, the data to be read can be corrected without rewriting data in flash memory/ROM.

35.2 Configuration

This section explains the wild register configuration.

- Wild register address registers (WRAR00 to WRAR15)
  These registers are used to specify the addresses at which replacement data used by the wild register function is stored.
- Wild register data registers (WRDR00 to WRDR15)
  These registers are used to store replacement data.
- Wild register enable register (WREN)
  This register is used to enable/disable the wild register function.

35.3 Registers

This section explains the configuration and functions of registers used for the wild register.

Registers of wild register

Table 35-1 lists the registers used for the wild register.

Table 35-1. Registers of wild register

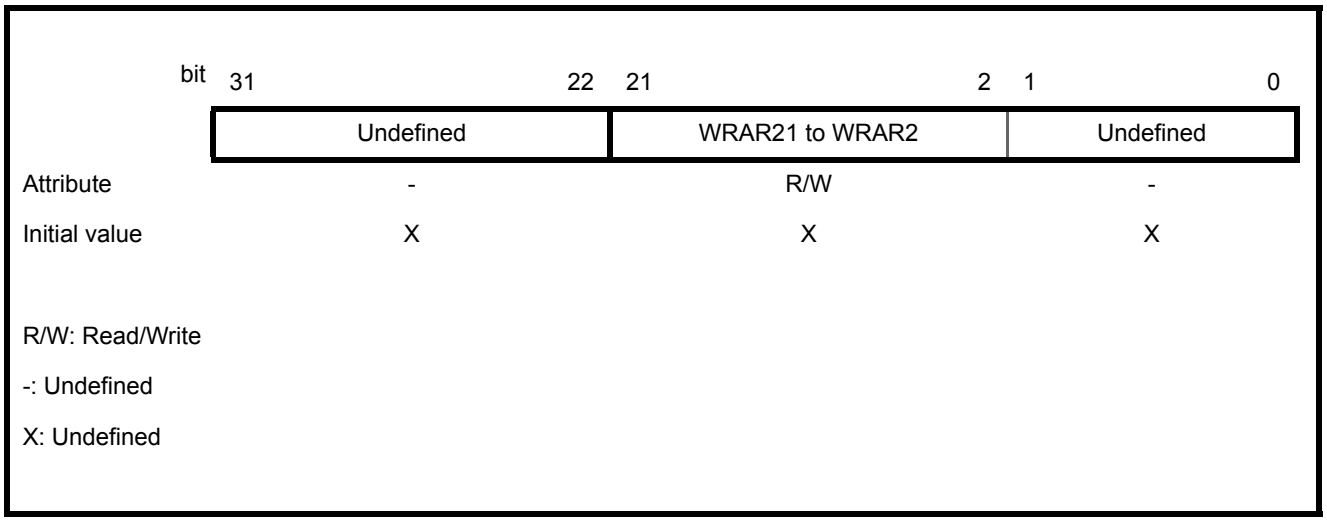
Channel	Abbreviated Register Name	Register Name	Reference
Common	WREN	Wild register enable register	35.3.3
0	WRAR00	Wild register address register 00	35.3.1
	WRDR00	Wild register data register 00	35.3.2
1	WRAR01	Wild register address register 01	35.3.1
	WRDR01	Wild register data register 01	35.3.2
2	WRAR02	Wild register address register 02	35.3.1
	WRDR02	Wild register data register 02	35.3.2
3	WRAR03	Wild register address register 03	35.3.1
	WRDR03	Wild register data register 03	35.3.2
4	WRAR04	Wild register address register 04	35.3.1
	WRDR04	Wild register data register 04	35.3.2
5	WRAR05	Wild register address register 05	35.3.1
	WRDR05	Wild register data register 05	35.3.2
6	WRAR06	Wild register address register 06	35.3.1
	WRDR06	Wild register data register 06	35.3.2
7	WRAR07	Wild register address register 07	35.3.1
	WRDR07	Wild register data register 07	35.3.2
8	WRAR08	Wild register address register 08	35.3.1
	WRDR08	Wild register data register 08	35.3.2
9	WRAR09	Wild register address register 09	35.3.1
	WRDR09	Wild register data register 09	35.3.2
10	WRAR10	Wild register address register 10	35.3.1
	WRDR10	Wild register data register 10	35.3.2
11	WRAR11	Wild register address register 11	35.3.1
	WRDR11	Wild register data register 11	35.3.2
12	WRAR12	Wild register address register 12	35.3.1
	WRDR12	Wild register data register 12	35.3.2
13	WRAR13	Wild register address register 13	35.3.1
	WRDR13	Wild register data register 13	35.3.2
14	WRAR14	Wild register address register 14	35.3.1
	WRDR14	Wild register data register 14	35.3.2
15	WRAR15	Wild register address register 15	35.3.1
	WRDR15	Wild register data register 15	35.3.2

35.3.1 Wild Register Address Register (WRAR00 to WRAR15)

These registers are used to specify the addresses at which replacement data used by the wild register function are stored. The value in WRAR21 to WRAR2 bits is compared with the actual address. When the memory at the address specified in this register is read, the actual memory is not read but instead the value set in the wild register data register (WRDR00 to WRDR15) is read.

Figure 35-1 shows the bit configuration of the wild register address register (WRAR00 to WRAR15).

Figure 35-1. Bit Configuration of Wild Register Address Register (WRAR00 to WRAR15)



Notes:

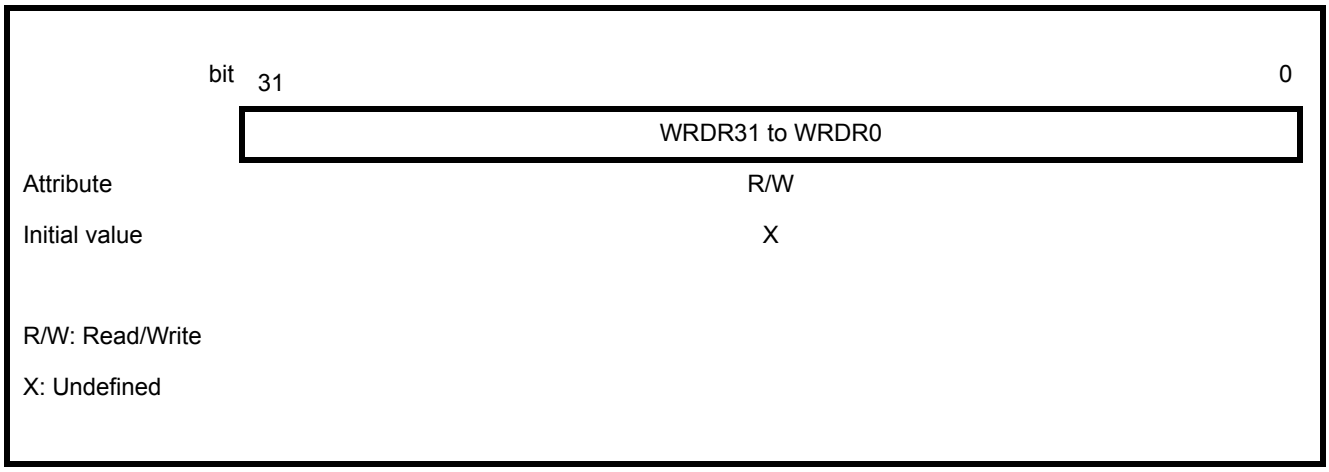
- Specify an address in units of words.
- This register cannot be read when the wild register function is enabled. If the register is read, the read value is undefined.
- Allocate the program, which sets an address in this register, to an area other than the built-in flash memory/ROM area.
- Be careful not to overlap the addresses to be set. The value read from an overlapped address is undefined.

35.3.2 Wild Register Data Register (WRDR00 to WRDR15)

These registers are used to store replacement data. When the memory at the address specified in the wild register address register (WRAR00 to WRAR15) is read, the actual memory is not read but instead the value set in this register is read.

Figure 35-2 shows the bit configuration of the wild register data register (WRDR00 to WRDR15).

Figure 35-2. Bit Configuration of Wild Register Data Register (WRDR00 to WRDR15)



Notes:

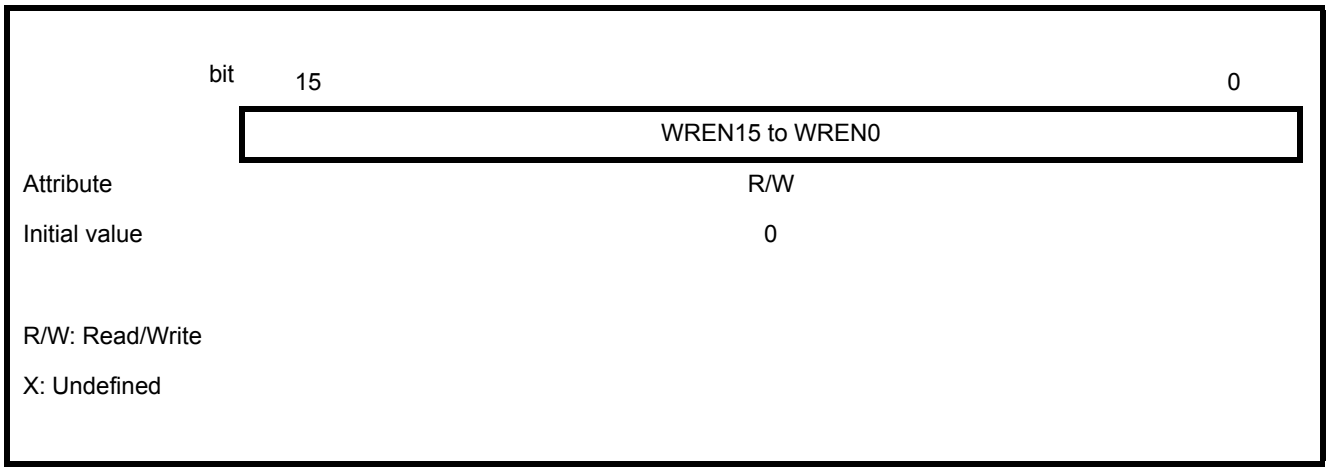
- Set word data in this register.
- This register cannot be read when the wild register function is enabled. If the register is read, the read value is undefined.

35.3.3 Wild Register Enable Register (WREN)

This register is used to enable/disable the wild register function.

Figure 35-3 shows the bit configuration of the wild register enable register (WREN).

Figure 35-3. Bit Configuration of Wild Register Enable Register (WREN)



[bit15 to bit0]: WREN15 to WREN0(Operation enable bit)

Each bit enables or disable the wild register function of the corresponding channel.

The WREN15 bit corresponds to ch.15, the WREN14 bit corresponds to ch.14 ... the WREN0 bit corresponds to ch.0.

Written Value	Explanation
0	Disables the function.
1	Enables the function.

Note: Do not enable the wild register function during execution of the automatic algorithm of flash memory.
The FRDY bit of the FLASH status register (FSTR) can be used to verify that the automatic algorithm is operating (FRDY=0).

35.4 Explanation of Operations and Setting Procedure Examples

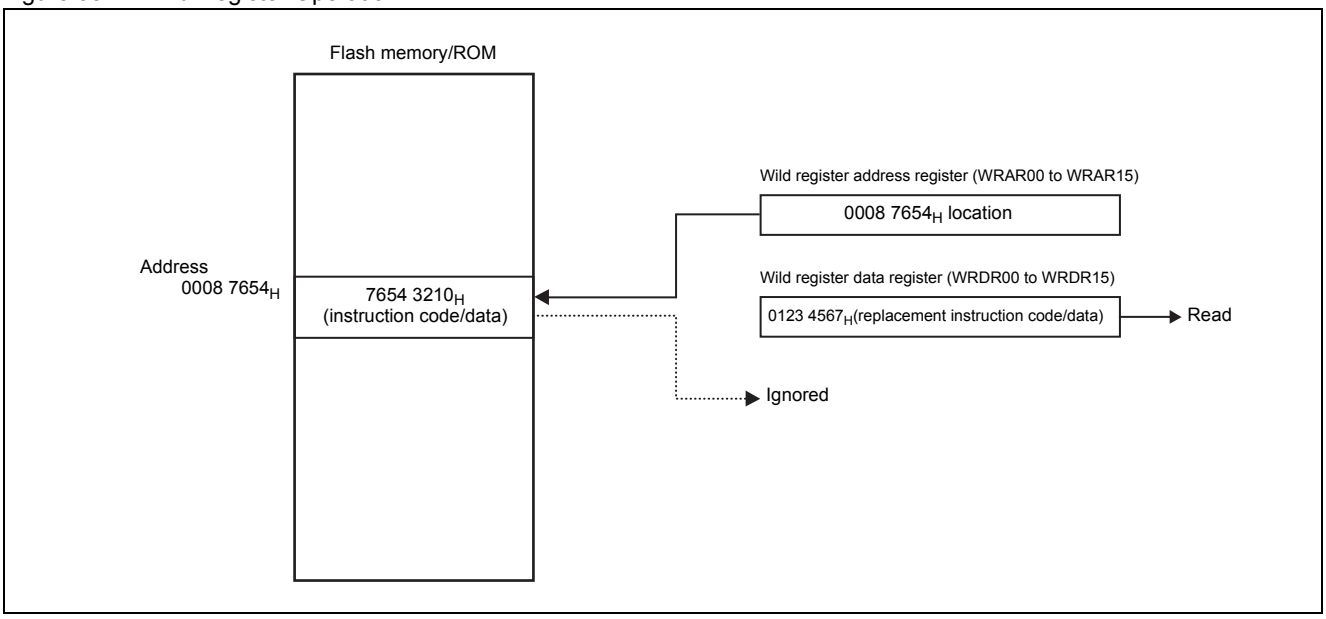
This section explains the operation of the wild register. The section also provides the setting procedure examples for operation.

35.4.1 Wild Register Operation

Figure 35-4 shows an example of wild register operation based on the following settings:

- Location 0008 7654_H is set in the wild register address register (WRAR00 to WRAR15).
- Wild register data register (WRDR00 to WRDR15) value: 0123 4567_H
- Value at location 0008 7654_H of flash memory/ROM: 7654 3210_H

Figure 35-4. Wild Register Operation



When the CPU attempts to read the data stored at location 0008 7654_H of flash memory/ROM, value "0123 4567_H" set in the wild register data register (WRDR0 to WRDR15) is read instead of "7654 3210_H" at location 0008 7654_H.

35.5 Notes on Using the Wild Register

Note the following points about using the wild register.

Notes on Programming

- Allocate the program, which sets an address in the wild register address register (WRAR00 to WRAR15), to an area other than the built-in flash memory/ROM area.
- Be careful not to overlap the addresses to be set in the wild register address register (WRAR00 to WRAR15). The value read from an overlapped address is undefined.
- Do not enable the wild register function during execution of the automatic algorithm of flash memory. The FRDY bit of the FLASH status register (FSTR) can be used to verify that the automatic algorithm is operating (FRDY=0).

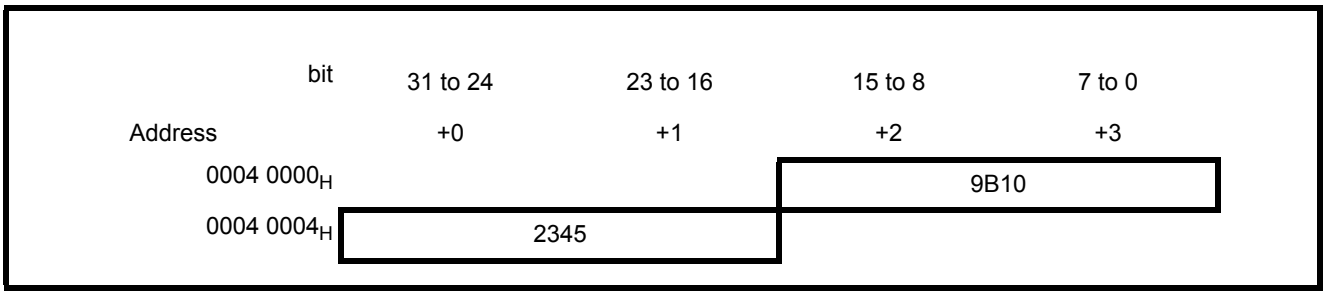
Notes on Operations

- The wild register address register (WRAR00 to WRAR15) and wild register data register (WRDR00 to WRDR15) are read in big endian mode.
- If a wild register is set at the address at which an instruction (32- or 48-bit instruction) exceeding 16 bits is stored, the CPU may not correctly interpret the instruction and malfunction. When setting a wild register at the address where a 32/48-bit instruction is allocated, do not set it inside the instruction.

  The following shows the operation of a 32/48-bit instruction.

  - 32-bit instruction (LDI:20): Example) LDI:20 #0x12345,r0($9B102345_H$)

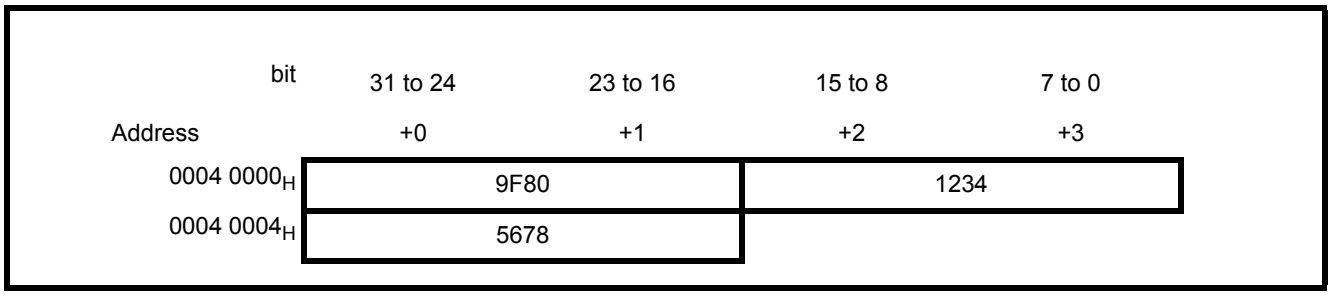
Figure 35-5. Memory Map of the Register Allocated at Location 0004 0000_H



1. When the wild register function is not used (WREN=0000)
   Data after replacement: 9B10 2345
   ❍ **0001 2345_H is set in R0**.
2. When 16 lower bits are replaced with the "INT" instruction
   (WRAR00=0004 0004, WRDR00=1FF4 ????, WREN=0001)
   Data after replacement: 9B10 **1FF4**
   ▼ **Because $1FF4_H$ is interpreted not as an instruction but as LDI:20 immediate data, 0001 $1FF4_H$ is set in R0**
3. When 16 upper bits are replaced with the INT instruction
   (WRAR00=0004 0000, WRDR00=1FF4???? , WREN=0001)
   Data after replacement: **1FF4** 2345
   ❍ **$1FF4_H$ is interpreted as an instruction.**
   ▼ **Next 2345_H is interpreted not as LDI:20 immediate data, but as an instruction.**

- 48-bit instruction (LDI:32) Example) LDI:32 #0x12345678,r0(9F8012345678$_H$)

Figure 35-6. Memory Map of the Register Allocated at Location 0004 0000$_H$/0004 0004$_H$



1. When the wild register function is not used (WREN=0000)
   Data after replacement: 9F80 1234 5678
   ❍ **12345678$_H$ is set in R0**.
2. When 16 lower bits at location 0004 0000$_H$ are replaced with the INT instruction
   (WRAR00=0004 0000, WRDR00=9F80 1FF4, WREN=0001)
   Data after replacement: 9F80 **1FF4** 5678
   ▼ **Because 1FF4$_H$ is interpreted not as an instruction but as LDI:32 immediate data, 1FF4 5678$_H$ is set in R0**
3. When 16 upper bits at location 0004 0004$_H$ are replaced with the INT instruction
   (WRAR00=0004 0004, WRDR00=1FF4 ????, WREN=0001)
   Data after replacement: 9F80 1234 **1FF4**
   ▼ **Because 1FF4$_H$ is interpreted not as an instruction but as LDI:32 immediate data, 1234 1FF4$_H$ is set in R0**
4. When 16 high-order bits at location 0004 0000$_H$ are replaced with the INT instruction (WRAR00=0004 0004,
   WRDR00=1FF4 1234, WREN=0001)
   Data after replacement: **1FF4** 1234 5678
   ❍ **1FF4$_H$ is interpreted as an instruction.**
   ▼ **Next 1234$_H$ and 5678$_H$ are interpret not as LDI:32 immediate data, but as an instruction.**

36. Serial Programming Connection

MB91F61x supports serial onboard write (Cypress standard) to flash memory.

This chapter explains the basic configuration for serial write to flash memory by using the Cypress Serial Programmer.

36.1 Cypress Serial Programmer

36.1 Cypress Serial Programmer

Cypress Serial Programmer (software) is an onboard programming tool for all Cypress-made controllers with built-in flash memory.

Two types of Serial Programmer are available according to the PC interface (RS-232C or USB) used. Choose the type according to your environment.

USB function is installed in the MB91F61x so that onboard write is possible by connecting the PC and microcontroller directly without performing USB-serial conversion.

Basic configuration of Cypress MCU programmer (Clock Asynchronous Serial Write)

Cypress MCU Programmer is used when the PC and microcontroller are connected through an RS-232C cable.
MCU Programmer writes data, through clock asynchronous serial communication, to built-in flash memory of a microcontroller installed in the user system.

Figure 36-1 shows the basic configuration of Cypress MCU Programmer, and Table 36-1 lists the system configuration.

Figure 36-1. Basic Configuration of Cypress MCU Programmer

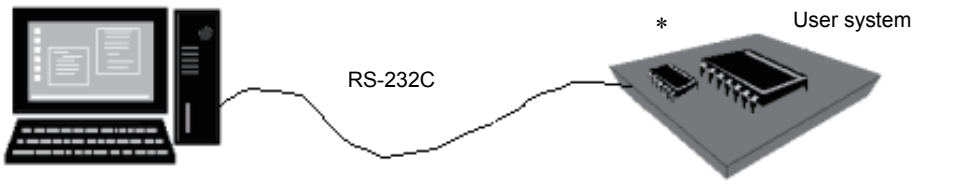


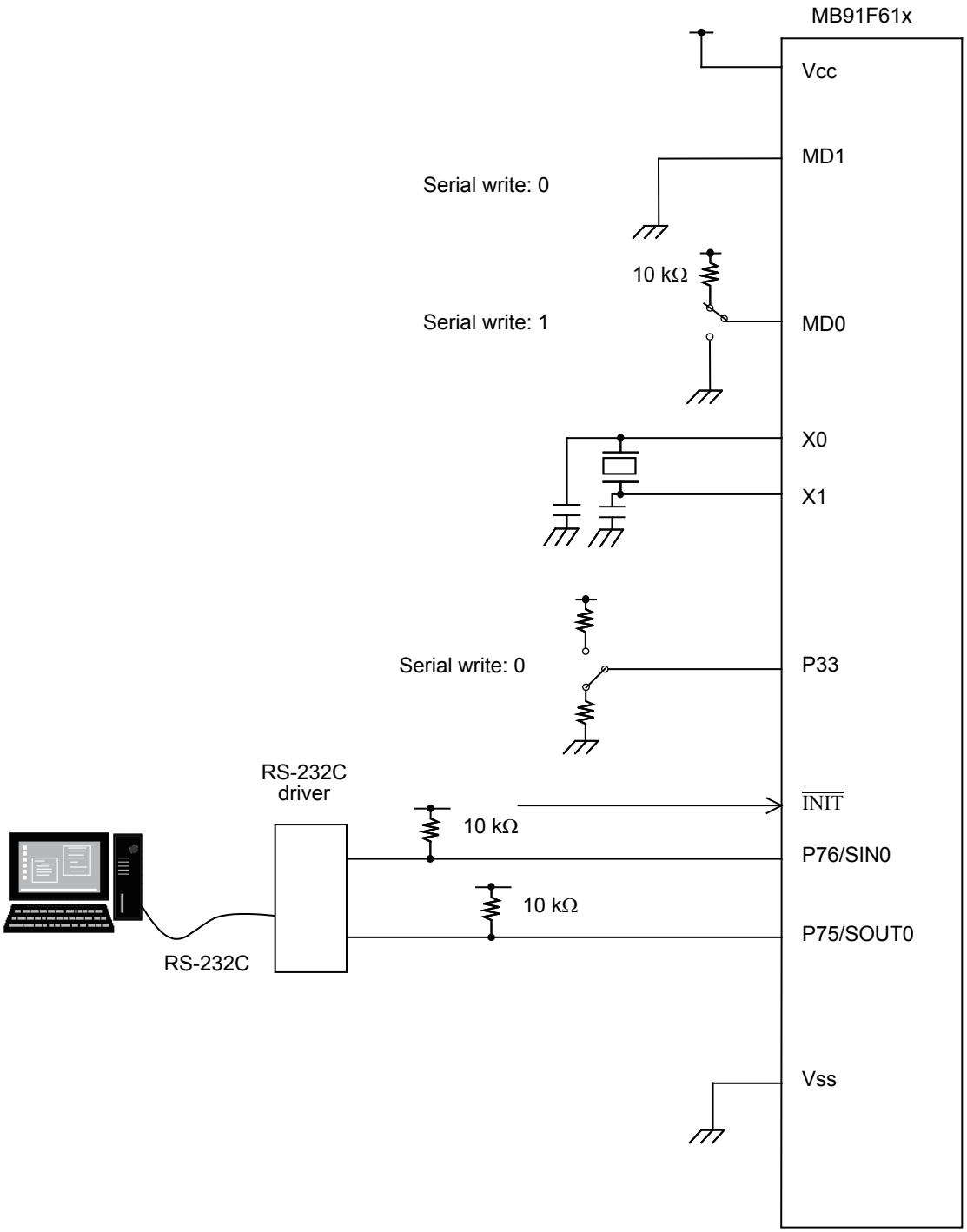
Table 36-1. System Configuration of Cypress MCU Programmer

Name	Type	Specifications
Cypress MCU Programmer	-	Software (can be downloaded from Web (registration system))*

* For registration, contact your sales representatives.

Figure 36-2 shows a connection example.

Figure 36-2. Connection Example using Cypress MCU Programmer



The pull-up resistance values shown are for example. Select the most appropriate resistance values for each system.
Use a 4 MHz or 8 MHz oscillator for writing.

Table 36-2. Oscillating Frequency and Communication Baud Rate Available for Clock Asynchronous Serial Communication

Master Oscillating Frequency	Communication Baud Rate
4 MHz	9600 bps
8 MHz	19200 bps
16 MHz	38400 bps
24 MHz	57600 bps
48 MHz	115200 bps

Basic configuration of Cypress USB programmer (Clock Synchronous Serial Write)

Cypress USB Programmer is used when the PC and microcontroller are connected through an adapter (MB2146-09A-E). USB Programmer writes data, through clock synchronous serial communication, to built-in flash memory of a microcontroller.

Figure 36-3 shows the basic configuration of Cypress USB Programmer, and Table 36-2 lists the system configuration.

Figure 36-3. Basic Configuration of Cypress USB Programmer

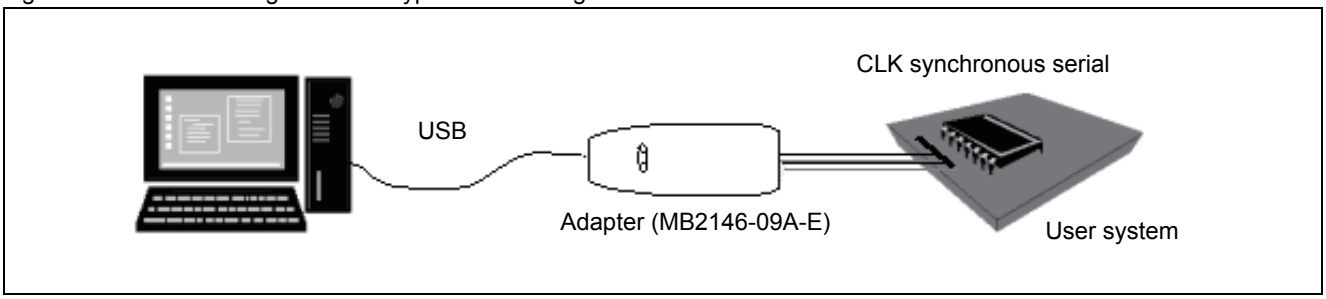


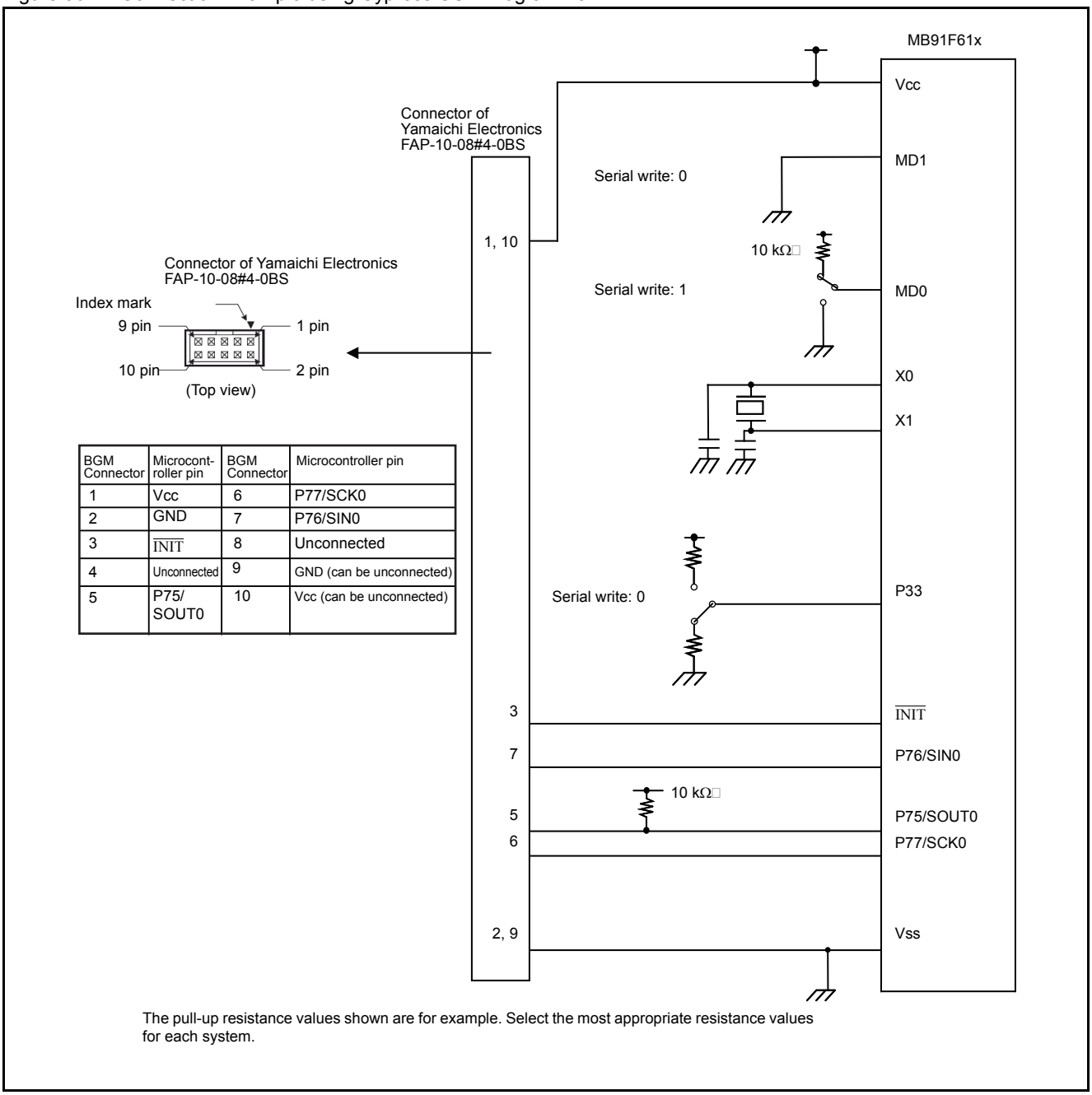
Table 36-3. System Configuration of Cypress USB Programmer

Name	Type	Specifications
Cypress USB Programmer	-	Software (can be downloaded from Web (registration system))[1]
Adapter	MB2146-09A-E	F^2MC family BGM adapter (Accessory: USB cable)

* For registration, contact your sales representatives.

Figure 36-4 shows a connection example.

Figure 36-4. Connection Example using Cypress USB Programmer



BGM Connector	Microcont-roller pin	BGM Connector	Microcontroller pin
1	Vcc	6	P77/SCK0
2	GND	7	P76/SIN0
3	$\overline{\text{INIT}}$	8	Unconnected
4	Unconnected	9	GND (can be unconnected)
5	P75/ SOUT0	10	Vcc (can be unconnected)

The pull-up resistance values shown are for example. Select the most appropriate resistance values for each system.

Basic configuration of Cypress USB DIRECT programmer (USB serial write)

Cypress USB DIRECT Programmer is used when the PC and microcontroller are connected through an USB cable. USB DIRECT Programmer writes data, through USB communication mode, to built-in flash memory of a microcontroller installed in the user system.

Figure 36-5 shows the basic configuration of Cypress USB DIRECT Programmer, and Table 36-4 lists the system configuration.

Figure 36-5. Basic Configuration of Cypress USB DIRECT Programmer

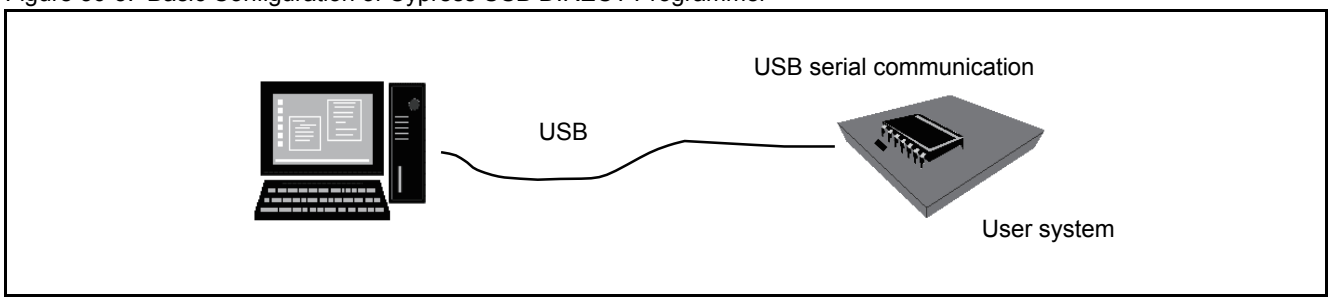


Table 36-4. System Configuration of Cypress USB DIRECT Programmer

Name	Type	Specifications
Cypress USB DIRECT Programmer	-	Software (can be downloaded from Web (registration system))[1]
USB cable	-	Sold on the market

[1]: For registration, contact your sales representatives.

For connection examples, see the manual (help section) of Cypress USB DIRECT Programmer.

Figure 36-6 shows a connection example.

Figure 36-6. Connection Example using Cypress USB DIRECT Programmer (own power supply is used)

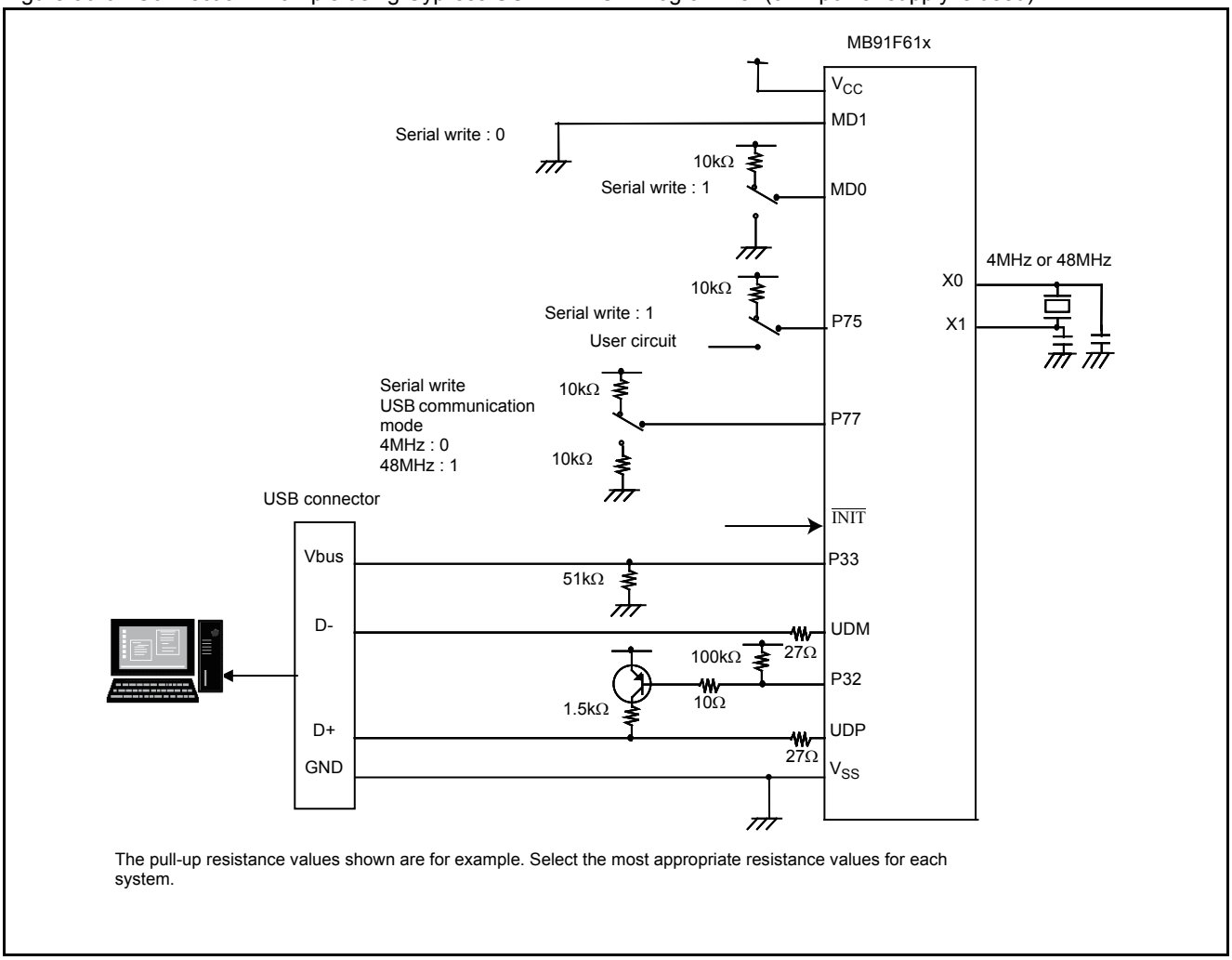
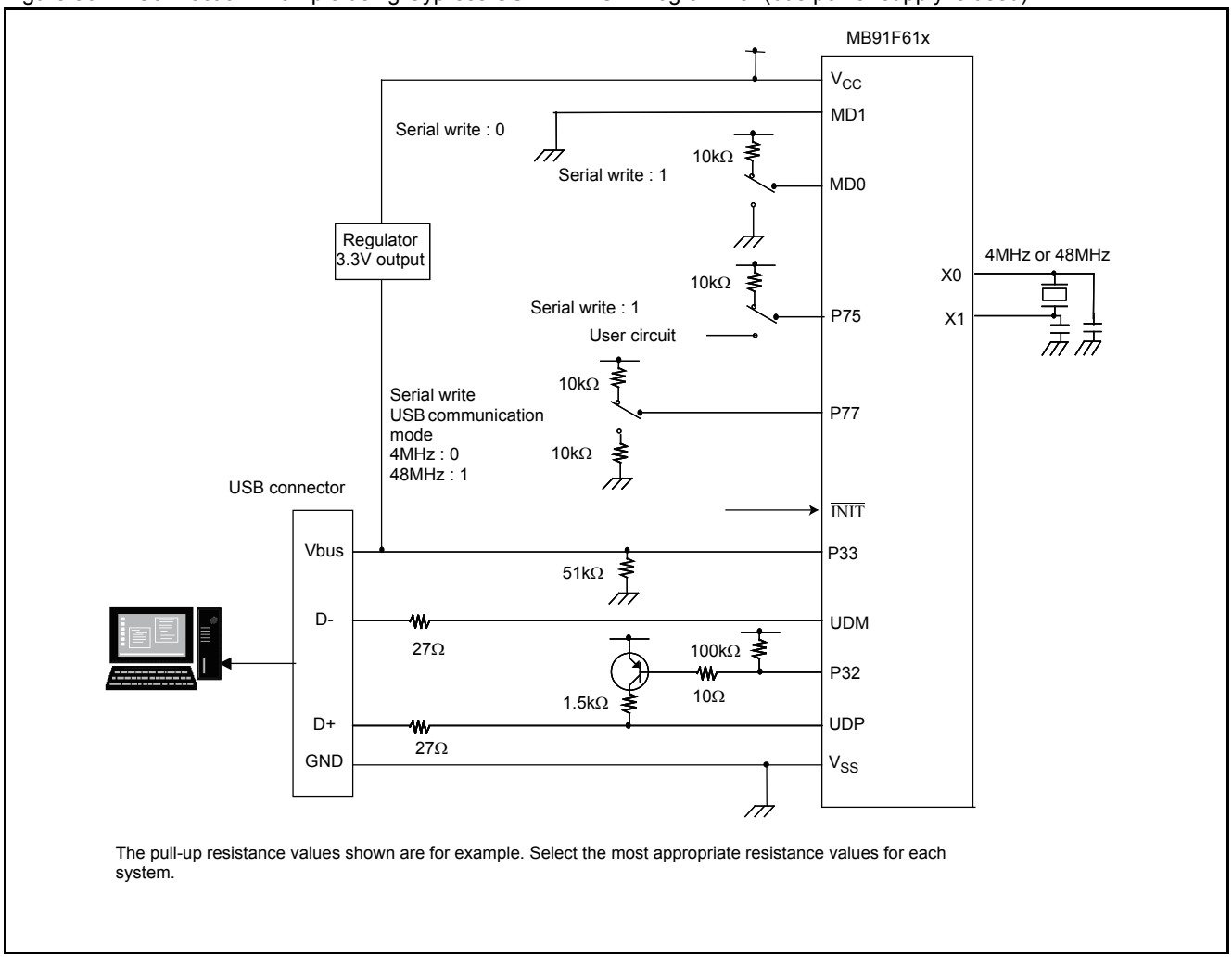


Figure 36-7 shows a connection example.

Figure 36-7. Connection Example using Cypress USB DIRECT Programmer (bus power supply is used)



36.1.1 Pins Used

Table 36-5. Pins Used

Pins	Function	Supplement
MD1,MD0	Mode pin	Resetting the system ($\overline{INIT}$: L → H) with SOUT0=H after setting MD1=L and MD0=H enters the serial write mode. When attaching a pull-up or pull-down resistor, avoid wiring.
X0,X1	Oscillation pin	See the "Datasheet" for the source clock frequencies that can be used in serial write mode. (Restrictions apply to clock asynchronous communication. For details, see .)
P75/SOUT0	Serial write mode activation pin/UART serial data output pin	After adding an external pull-up resistor and then releasing the reset state, setting the level of this pin to "H" activates the serial write mode. When the communication mode is set to UART, this pin becomes a serial data output pin when communication begins after the serial write mode is activated. When the communication mode is set to USB, the output operation can not be performed with keeping the pull-up state.
P76/SIN0	Clock synchronous/asynchronous select pin/UART serial data input pin	Setting the input level of this pin to "H" until the start of communication enables the clock asynchronous communication mode, and setting it to "L" enables the clock synchronous communication mode. This pin is used as an UART serial data input pin when communication begins after the serial write mode is activated.
P77/SCK0	Serial clock I/O pin	This pin becomes a serial clock input/output pin when the communication mode is set to clock synchronous communication. When the communication mode is USB, this pin controls the frequency for X0/X1. P77=L: X0/X1 = 4MHz P77=H: X0/X1 = 48MHz
P33	Communication mode select pin	The communication mode is determined by the input level of this pin at reset to shift to the serial write mode. Setting this pin to "H" enables the USB communication mode, and setting it to "L" enables the UART communication mode.
P32	Pull-up control pin for UDP	This pin controls the pull-up of USB side (+) when the communication mode is USB. P32=L: Connect the pull-up resistor P32=H: Disconnect the pull-up resistor
UDP	USB I/O pin	This pin becomes an input/output pin of USB side (+) when the communication mode is set to USB.
UDM	USB I/O pin	This pin becomes an input/output pin of USB side (-) when the communication mode is set to USB.
$\overline{INIT}$	Reset pin	-
V_{CC}	Supply voltage pin	For writing, supply power voltage to the microcontroller from the user system.
V_{SS}	GND pin	-

37. Handling the Device

This chapter provides notes on using this series.

37.1 Notes on Handling the Device

37.1 Notes on Handling the Device

Note the following points on using this series.

Precautions for handling the devices

Any semiconductor devices have inherently a certain rate of failure. The possibility of failure is greatly affected by the conditions in which they are used (circuit conditions, environmental conditions, etc.). This page describes precautions that must be observed to minimize the chance of failure and to obtain higher reliability from your Cypress devices.

1. Precautions for Product Design

   This section describes precautions when designing electronic equipment using semiconductor devices.

   - Absolute Maximum Ratings

     Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of absolute maximum ratings. Do not exceed these ratings.

   - Recommended Operating Conditions

     The recommended operating conditions are required in order to ensure the normal operation of the semiconductor device. All of the device's electrical characteristics are warranted when the device is operated within these ranges.

     Always use semiconductor devices within their recommended operating condition ranges. Operation outside these ranges may adversely affect reliability and could result in device failure. No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their sales representatives beforehand.

   - Processing and Protection of Pins

     These precautions must be followed when handling the pins which connect semiconductor devices to power supply and input/output functions.

     - Preventing Over-Voltage and Over-Current Conditions
       Exposure to voltage or current levels in excess of maximum ratings at any pin is likely to cause deterioration within the device, and in extreme cases leads to permanent damage of the device. Try to prevent such over-voltage or over-current conditions at the design stage.
     - Protection of Output Pins
       Shorting of output pins to supply pins or other output pins, or connection to large capacitance can cause large current flows. Such conditions if present for extended periods of time can damage the device. Therefore, avoid this type of connection.
     - Handling of Unused Input Pins
       Unconnected input pins with very high impedance levels can adversely affect stability of operation. Such pins should be connected through an appropriate resistance to a power supply pin or ground pin.

       PLL pin for OSD (recommended pin handling when PLL for OSD is not in use)

Pin no.	Pin name	Recommended handling of unused pin
45	VSSP	V_{SS} (PLL macro GND)
46	VDDP	V_{SS} (PLL macro power supply)
47	VCI	V_{SS}
48	CPO	V_{SS}

Analog OSD (recommended pin handling when analog OSD is not in use)

Pin no.	Pin name	Recommended handling of unused pin
49	VSSD	V_{SS}(DAC macro GND)
50	VDDD	V_{SS}(DAC macro power supply)
51	VREF	V_{SS}
52	VRO	V_{SS}
53	ROUT	V_{SS}
54	GOUT	V_{SS}
55	BOUT	V_{SS}

Digital OSD (recommended pin handling when digital OSD is not in use)

Pin no.	Pin name	Recommended handling of unused pin
19	R0	OPEN
20	R1	OPEN
21	R2	OPEN
22	R3	OPEN
23	R4	OPEN
24	G0	OPEN
25	G1	OPEN
26	G2	OPEN
27	G3	OPEN
28	G4	OPEN
29	G5	OPEN
32	B0	OPEN
33	B1	OPEN
34	B2	OPEN
35	B3	OPEN
36	B4	OPEN

Other OSD pins

Pin no.	Pin name	Recommended handling of unused pin
37	VOA0	OPEN
38	VOA1	OPEN
39	VOA2	OPEN
40	VOB	OPEN
41	DCKO	OPEN
42	DCKI	pull-down
43	VSYNC	pull-down
44	HSYNC	pull-down

USB (example of pin handling when USB is not in use)

Pin no.	Pin name	Recommended handling of unused pin
57	UDP	pull-down
58	UDM	pull-down

DSU pin

Pin no.	Pin name	Recommended handling of unused pin
83	$\overline{\text{TRST}}$	Reset signal input from user board
84 I	CLK	OPEN
85 I	BREAK	OPEN
86 I	CS0	OPEN
87 I	CS1	OPEN
88 I	CS2	OPEN
92 I	CD0	OPEN
93 I	CD1	OPEN
94 I	CD2	OPEN
95	ICD3	OPEN

- Latch-up

  Semiconductor devices are constructed by the formation of P-type and N-type areas on a substrate. When subjected to abnormally high voltages, internal parasitic PNPN junctions (called thyristor structures) may be formed, causing large current levels in excess of several hundred mA to flow continuously at the power supply pin. This condition is called latch-up.

  Note: The occurrence of latch-up not only causes loss of reliability in the semiconductor device, but can cause injury or damage from high heat, smoke or flame. To prevent this from happening, do the following:

  - Be sure that voltages applied to pins do not exceed the absolute maximum ratings. This should include attention to abnormal noise, surge levels, etc.
  - Be sure that abnormal current flows do not occur during the power-on sequence.

- Observance of Safety Regulations and Standards

  Most countries in the world have established standards and regulations regarding safety, protection from electromagnetic interference, etc. Customers are requested to observe applicable regulations and standards in the design of products.

- Fail-Safe Design

  Any semiconductor devices have inherently a certain rate of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

- Precautions Related to Usage of Devices

  Cypress devices are intended for use in standard applications (computers, office automation and other office equipment, industrial, communications, and measurement equipment, personal or household devices, etc.).

  CAUTION: Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.

1. Precautions for package mounting

   Package mounting may be either lead insertion type or surface mount type. In either case, for heat resistance during soldering, you should only mount under Cypress recommended conditions. For detailed information about mount conditions, contact your sales representative.

   - Lead Insertion Type
     Mounting of lead insertion type packages onto printed circuit boards may be done by two methods: direct soldering on the board, or mounting by using a socket.
     Direct mounting onto boards normally involves processes for inserting leads into through-holes on the board and using the flow soldering (wave soldering) method of applying liquid solder.
     In this case, the soldering process usually causes leads to be subjected to thermal stress in excess of the absolute ratings for storage temperature. Mounting processes should conform to Cypress recommended mounting conditions.
     If socket mounting is used, differences in surface treatment of the socket contacts and IC lead surfaces can lead to contact deterioration after long periods. For this reason it is recommended that the surface treatment of socket contacts and IC leads be verified before mounting.
   - Surface Mount Type
     Surface mount packaging has longer and thinner leads than lead-insertion packaging, and therefore leads are more easily deformed or bent. The use of packages with higher pin counts and narrower pin pitch results in increased susceptibility to open connections caused by deformed pins, or shorting due to solder bridges.
     You must use appropriate mounting techniques. Cypress recommends the solder reflow method, and has established a ranking of mounting conditions for each product. Users are advised to mount packages in accordance with Cypress ranking of recommended conditions.
   - Lead-Free Packaging

     Note: When ball grid array (BGA) packages with Sn-Ag-Cu balls are mounted using Sn-Pb eutectic soldering, junction strength may be reduced under some conditions of use.
   - Storage of Semiconductor Devices
     Because plastic chip packages are formed from plastic resins, exposure to natural environmental conditions will cause absorption of moisture. During mounting, the application of heat to a package that has absorbed moisture can cause surfaces to peel, reducing moisture resistance and causing packages to crack. To prevent, do the following:
     1. Avoid exposure to rapid temperature changes, which cause moisture to condense inside the product. Store products in locations where temperature changes are slight.
     2. Use dry boxes for product storage. Products should be stored below 70% relative humidity, and at temperatures between 5°C and 30 °C. When you open Dry Package that recommends humidity 40% to 70% relative humidity.
     3. When necessary, Cypress packages semiconductor devices in highly moisture-resistant aluminum laminate bags, with a silica gel desiccant. Devices should be sealed in their aluminum laminate bags for storage.
     4. Avoid storing packages where they are exposed to corrosive gases or high levels of dust.
   - Baking
     Packages that have absorbed moisture may be de-moisturized by baking (heat drying). Follow the Cypress recommended conditions for baking.
     Condition: 125 °C/24 h
   - Static Electricity
     Because semiconductor devices are particularly susceptible to damage by static electricity, you must take the following precautions:
     1. Maintain relative humidity in the working environment between 40% and 70%. Use of an apparatus for ion generation may be needed to remove electricity.
     2. Electrically ground all conveyors, solder vessels, soldering irons and peripheral equipment.
     3. Eliminate static body electricity by the use of rings or bracelets connected to ground through high resistance (on the level of 1 MΩ). Wearing of conductive clothing and shoes, use of conductive floor mats and other measures to minimize shock loads is recommended.
     4. Ground all fixtures and instruments, or protect with anti-static measures.
     5. Avoid the use of styrofoam or other highly static-prone materials for storage of completed board assemblies.

3. Precautions for use environment

   Reliability of semiconductor devices depends on ambient temperature and other conditions as described above. For reliable performance, do the following:

   - Humidity
     Prolonged use in high humidity can lead to leakage in devices as well as printed circuit boards. If high humidity levels are anticipated, consider anti-humidity processing.
   - Discharge of Static Electricity
     When high-voltage charges exist close to semiconductor devices, discharges can cause abnormal operation. In such cases, use anti-static measures or processing to prevent discharges.
   - Corrosive Gases, Dust, or Oil
     Exposure to corrosive gases or contact with dust or oil may lead to chemical reactions that will adversely affect the device. If you use devices in such conditions, consider ways to prevent such exposure or to protect the devices.
   - Radiation, Including Cosmic Radiation
     Most devices are not designed for environments involving exposure to radiation or cosmic radiation. Users should provide shielding as appropriate.
   - Smoke, Flame

     Note: Plastic molded devices are flammable, and therefore should not be used near combustible substances. If devices begin to smoke or burn, there is danger of the release of toxic gases.

   Customers considering the use of Cypress products in other special environmental conditions should consult with sales representatives.

Notes on handling the device

- Power pins

  Because there are multiple V_{CC} and V_{SS} pins, respective pins at the same potential are interconnected to prevent malfunctions such as latch-up. However, you must connect the pins externally to the power supply and ground lines to reduce the electro-magnetic emission levels, to prevent abnormal operation of strobe signals caused by the rise in the ground level, and to conform to the total output current rating. Furthermore, the current supply source should be connected to the V_{CC} and V_{SS} pins of the device at a low impedance.
  It is recommended to connect a ceramic bypass capacitor of approximately 0.1 μF as a bypass capacitor between the V_{CC} and V_{SS} pins near this device.
- Crystal oscillation circuit

  Noise around X0 and X1 pins causes this device to malfunction. Design the PC board so that X0, X1, the crystal oscillator, and the bypass capacitors to the ground are as close as possible.

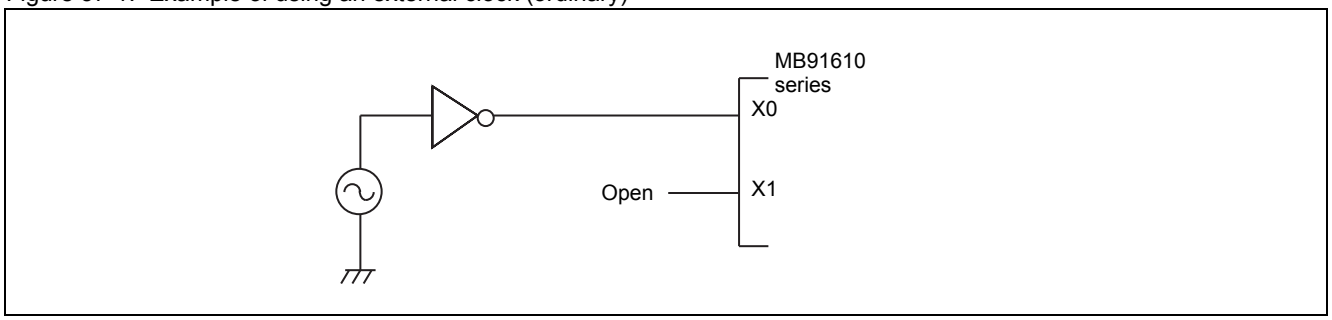
  We also strongly recommend such PC board artwork where pins X0 and X1 are surrounded by grounding because this is expected to ensure stable device operation.
- OSDC output pin

  The OSDC output pins (R0 to R4, G0 to G5, B0 to B4, VOA0 to VOA2,VOB,DCKO) are high-speed corresponded output pin.

  Adjust the signal waveform such as by inserting damping resistor on the board as needed.
- Notes on using an external clock

  When an external clock is selected, use X0 pin only. X1 pin should be opened.

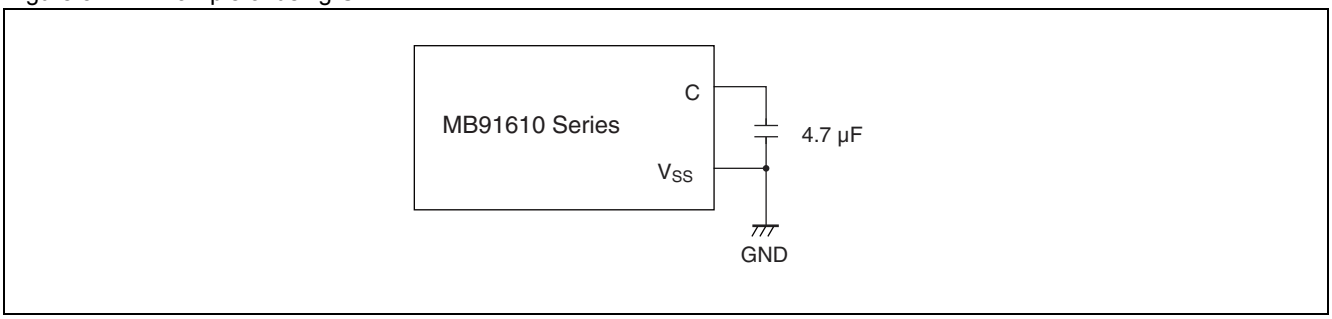
Figure 37-1. Example of using an external clock (ordinary)



- C pin

  As this series includes an internal regulator, always connect a bypass capacitor of approximately 4.7 μF to the C pin for use by the regulator.

Figure 37-2. Example of using C Pin



- MD0 and MD1 (Mode pins)

  Connect the mode pins (MD0, MD1) directly to V_{CC} or V_{SS} pins. Design the printed circuit board such that the pull-up/ down resistance stays low, as well as the distance between the mode pins and power supply or GND pins is as short as possible and the connection impedance is low, when the pins are pulled-up/down such as for switching the pin level and rewriting the Flash memory data. It is because of preventing the device erroneously switching to test mode due to noise.

- Power-on sequence
  - To guarantee an oscillation stabilization wait time of the internal regulator + oscillation circuit, immediately after power-on, leave the input to pin $\overline{INIT}$ "L" level for a time that is the sum of the regulator voltage stabilization wait time, the oscillator startup time, and the main oscillation stabilization wait time.
  - Turn power on/off in the following order.

    power-on: $V_{CC} \rightarrow AV_{CC} \rightarrow AVRH$

    power-off: $AVRH \rightarrow AV_{CC} \rightarrow V_{CC}$
  - When canceling the reset state (by changing the level at pin $\overline{INIT}$ from "L" to "H"), ensure that the power is stable.

- Caution on Operations during PLL Clock Mode

  On this microcontroller, if in case the crystal oscillator breaks off or an external reference clock input stops while the PLL clock mode is selected, a self-oscillator circuit contained in the PLL may continue its operation at its self-running frequency.

  However, Cypress will not guarantee results of operations if such failure occurs.

Notes on program status registers (PS)

During the execution of a certain instruction, the program status register (PS) is processed in advance. Therefore, the following exception handling operation might cause an interrupt handling routine to encounter a break when the debugger is in use. Alternatively, it might cause the interrupt handling routine to update displayed flags of the program status register (PS).In either case, the device performs processing before and after EIT operation in accordance with the specification because the device is designed to correctly perform re-processing after return from the EIT.

1. When one of events 1 to 3:
   a. A user interrupt is accepted.
   b. Single-step execution is performed.
   c. A break occurs due to a data event or from the emulator menu.

   Occurs immediately before the DIV0U/DIV0S instruction, the following operation might be performed:
   - The D0 and D1 flags are updated in advance.
   - The EIT handling routine (user interrupt or emulator) is executed.
   - After return from the EIT, the DIV0U/DIV0S instruction is executed to update the D0 and D1 flags to the same values as in step 1.
2. If the ORCCR/STILM/MOV Ri or PS instruction is executed to enable interrupts when a user interrupt request has been generated, the following operation will result:
   - The program status register (PS) is updated in advance.
   - The EIT processing routine (user interrupt or emulator) is executed.
   - After return from the EIT, the above instruction is executed to update the program status register (PS) to the same value as in step 1.

Debugger-related notes

- Single-step execution of the RETI instruction

  In an environment where single-step instruction execution will encounter frequent interrupts, only the appropriate interrupt processing routine for this case is executed repeatedly. As a result, the main routine and programs assigned lower interrupt levels are not executed. (For example, if base timer interrupts are enabled, single-step RETI execution will always cause a break whenever the base timer routine begins.)

  When the interrupt handling routine no longer needs to be debugged, disable the pertinent interrupt.
- Break function

  If target addresses for hardware breaks (including event breaks) are set to current system stack pointer addresses or set in the area containing stack pointers, the user program will encounter a break after executing one instruction even though it does not contain an actual data access instruction.

  To avoid this problem, do not specify (word) access to the area containing system stack pointer addresses as a target of hardware breaks (including event breaks).
- Built-in ROM (Flash memory, mask ROM)
  - Notes on using the EVA chip
    1. Do not specify any built-in ROM area as a transfer destination specified by the DMA controller (DMAC).
    2. If a built-in ROM area is specified as a DMA controller (DMAC) transfer destination, the built-in ROM area might be overwritten when a break occurs during DMAC transfer.
    3. It is, however, possible to specify a built-in ROM area as a DMA controller (DMAC) transfer source.
- Operand break

  A malfunction could occur if a stack pointer exists in an area that is specified as DSU operand breaks. Do not specify access to the area containing system stack pointer addresses as the target of data event breaks.

Notes when not using the emulator

To operate the evaluation MCU on the user system without connecting the emulator, each of the input pins on the evaluation MCU connected to the emulator interface on the user system as shown below.

Note that switching circuits or other measures may be needed on the user system.

Table 37-1. Emulator Interface Pin Treatment

Evaluation MCU Pin Name	Pin Connection
$\overline{\text{TRST}}$	Connect to the reset output circuit on the user system.
$\overline{\text{INIT}}$	Connect to the reset output circuit on the user system.
Other Pins	Open

A. Appendix

These appendixes explain the I/O map, a list of registers, the pin state in each CPU state, and a list of instructions for the FR80 family CPUs.

A.1 I/O Map

A.2 List of Registers

A.3 Interrupt Vectors

A.4 Pin State in Each CPU State

A.5 PLL for Dot Clock Generation

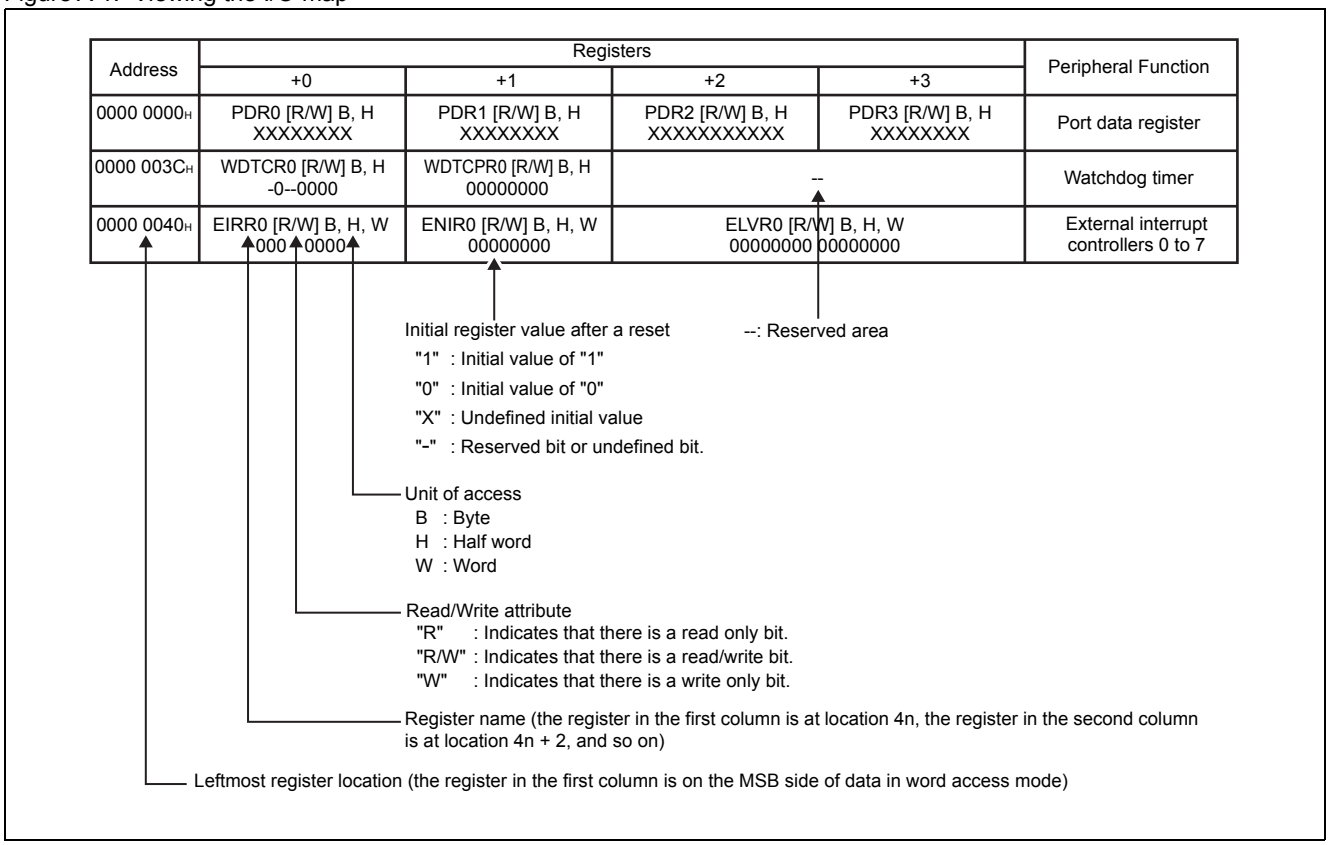
A.6 Lists of Instructions

A.1 I/O Map

This appendix outlines the relationship between memory space areas and each register for peripheral functions.

Viewing the I/O map

Figure A-1. Viewing the I/O map



Notes:

- Use the following address for data access depending on the access size.
  - Word access:The address is a multiple of 4 (the lowest 2 bits are set to "00_B").
  - Half word access:The address is a multiple of 2 (the lowest bit is set to "0_B").
  - Byte access:--
- Do not access the reserved areas.

Table A-1. I/O Map (Sheet 1 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0000$_H$	PDR0 [R/W] B,H XXXXXXXX	PDR1 [R/W] B,H XXXXXXXX	PDR2 [R/W] B,H XXXXXXXX	PDR3 [R/W] B,H XXXXXXXX	Port data register
0000 0004$_H$	-	PDR5 [R/W] B,H XXXXXXXX	-	PDR7[R/W] B,H XXXXXXXX	
0000 0008$_H$ to 0000 0010$_H$	-				
0000 0014$_H$	PDRK [R/W] B ------XX	-			
0000 0018$_H$ to 0000 001C$_H$	-				
0000 0020$_H$	RCCR [R/W] B 0---0000	RCST [R/W] B 00000000	RCSHW [R/W] B 00000000	RCDAHW [R/W] B 00000000	HDMI-CEC/ Remote controller
0000 0024$_H$	RCDBHW [R/W] B 00000000	-	RCADR1 [R/W] B ---00000	RCADR2 [R/W] B ---00000	
0000 0028$_H$	RCDTHH [R] B,H,W 00000000	RCDTHL [R] B,H,W 00000000	RCDTLH [R] B,H,W 00000000	RCDTLL [R] B,H,W 00000000	
0000 002C$_H$	RCCKD [R/W] H ---00000 00000000		-		
0000 0030$_H$ to 0000 0038$_H$	-				Reserved
0000 003C$_H$	WDTCR0[R/W] B,H 00000000	WDTCPR0[R/W] B,H 00000000	WDTCR1[R] B,H XXXX0000	WDTCPR1[R/W] B,H 00000000	Watchdog timer
0000 0040$_H$	EIRR0[R/W] B,H,W 00000000	ENIR0[R/W] B,H,W 00000000	ELVR0[R/W] B,H,W 00000000 00000000		External interrupt 0 to 7
0000 0044$_H$	DICR [R/W] B -------0	-			Delay interrupt
0000 0048$_H$	TMRLRA0 [R/W] H XXXXXXXX XXXXXXXX		TMR0 [R] H XXXXXXXX XXXXXXXX		16-bit reload timer ch.0
0000 004C$_H$	-		TMCSR0 [R/W] H --000000 --000000		
0000 0050$_H$	TMRLRA1 [R/W] H XXXXXXXX XXXXXXXX		TMR1 [R] H XXXXXXXX XXXXXXXX		16-bit reload timer ch.1
0000 0054$_H$	-		TMCSR1 [R/W] H --000000 --000000		
0000 0058$_H$	TMRLRA2 [R/W] H XXXXXXXX XXXXXXXX		TMR2 [R] H XXXXXXXX XXXXXXXX		16-bit reload timer ch.2
0000 005C$_H$	-		TMCSR2 [R/W] H --000000 --000000		
0000 0060$_H$	SCR0 [R/W] B,H,W 0--00000	SMR0 [R/W] B,H,W 000-0000	SSR0 [R,R/W] B,H,W 0-000011	ESCR0 [R/W] B,H,W -0000000	Multi-function serial interface ch.0
0000 0064$_H$	RDR0[R]/TDR0[W] B,H,W[1] -------0 00000000		BGR10[R/W]H,W 00000000	BGR00[R/W] H,W 00000000	

Table A-1. I/O Map (Sheet 2 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0068$_H$	SCR1[R/W] IBCR1[R,R/W] B,H,W[2] 0--00000	SMR1 [R/W] B,H,W 000-0000	SSR1 [R,R/W] B,H,W 0-000011	ESCR1[R/W] IBSR1[R,R/W] B,H,W[2] -0000000	Multi-function serial interface ch.1
0000 006C$_H$	RDR1[R]/TDR1[W] B,H,W[1] -------0 00000000		BGR11[R/W] H,W 00000000	BGR01[R/W] H,W 00000000	
0000 0070$_H$	ISMK1 [R/W] B,H[2] --------	ISBA1 [R/W] B,H[2] --------	-		
0000 0074$_H$	SCR2[R/W] IBCR2[R,R/W] B,H,W[2] 0--00000	SMR2 [R/W] B,H,W 000-0000	SSR2 [R,R/W] B,H,W 0-000011	ESCR2[R/W] IBSR2 [R,R/W] B,H,W[2] -0000000	Multi-function serial interface ch.2
0000 0078$_H$	RDR2[R]/TDR2[W] B,H,W[1] -------0 00000000		BGR12[R/W] H,W 00000000	BGR02[R/W] H,W 00000000	
0000 007C$_H$	ISMK2 [R/W] B,H[2] --------	ISBA2 [R/W] B,H[2] --------	-		
0000 0080$_H$	SCR3[R/W] IBCR3[R,R/W] B,H,W[2] 0--00000	SMR3 [R/W] B,H,W 000-0000	SSR3 [R,R/W] B,H,W 0-000011	ESCR3[R/W] IBSR3[R,R/W] B,H,W[2] -0000000	Multi-function serial interface ch.3
0000 0084$_H$	RDR3[R]/TDR3[W] B,H,W[1] -------0 00000000		BGR13[R/W] H,W 00000000	BGR03[R/W] H,W 00000000	
0000 0088$_H$	ISMK3 [R/W] B,H[2] --------	ISBA3 [R/W] B,H[2] --------	-		
0000 008C$_H$ to 0000 00BC$_H$	-				Reserved
0000 00C0$_H$	RDRM0 [R]/ TDRM0[W] B,H,W 00000000	RDRM1 [R]/ TDRM1[W] B,H,W 00000000	RDRM2 [R]/ TDRM2[W] B,H,W 00000000	RDRM3 [R]/ TDRM3[W] B,H,W 00000000	Multi-function serial interface data register (mirror)
0000 00C4$_H$	-				
0000 00C8$_H$	SSEL0123 [R/W] B ------00	-			Multi-function serial interface serial clock selection
0000 00CC$_H$	-				Reserved
0000 00D0$_H$	SCR8 [R/W] IBCR8 [R,R/W] B,H,W[2] 0--00000	SMR8 [R/W] B,H,W 000-0000	SSR8 [R,R/W] B,H,W 0-000011	ESCR8 [R/W] IBSR8 [R,R/W] B,H,W[2] -0000000	Multi-function serial interface ch. 8 (FIFO)
0000 00D4$_H$	RDR8[R]/TDR8[W] B,H,W[1] -------0 00000000		BGR18 [R/W] H,W 00000000	BGR08 [R,R/W] H,W 00000000	
0000 00D8$_H$	ISMK8 [R/W] B,H[2] --------	ISBA8 [R/W] B,H[2] --------	-		
0000 00DC$_H$	FCR18 [R/W] B,H,W ---00100	FCR08 [R,R/W] B,H,W -0000000	FBYTE28 [R/W] B,H,W 00000000	FBYTE18 [R/W] B,H,W 00000000	

Table A-1. I/O Map (Sheet 3 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 00E0$_H$	SCR9 [R/W] IBCR9 [R,R/W] B,H,W[2] 0--00000	SMR9 [R/W] B,H,W 000-0000	SSR9 [R,R/W] B,H,W 0-000011	ESCR9 [R/W] IBSR9[R,R/W] B,H,W[2] -0000000	Multi-function serial interface ch. 9 (FIFO)
0000 00E4$_H$	RDR9[R]/TDR9[W] B,H,W[1] -------0 00000000		BGR19 [R/W] H,W 00000000	BGR09 [R/W] H,W 00000000	
0000 00E8$_H$	ISMK9 [R/W] B,H[2] --------	ISBA9 [R/W] B,H[2] --------	-		
0000 00EC$_H$	FCR19 [R/W] B,H,W ---00100	FCR09 [R,R/W] B,H,W -0000000	FBYTE29 [R/W] B,H,W 00000000	FBYTE19 [R/W] B,H,W 00000000	
0000 00F0$_H$	SCR10 [R/W] IBCR10 [R,R/W] B,H,W[2] 0--00000	SMR10 [R/W] B,H,W 000-0000	SSR10 [R,R/W] B,H,W 0-000011	ESCR10 [R/W] IBSR10 [R,R/W] B,H,W[2] -0000000	Multi-function serial interface ch.10 (FIFO)
0000 00F4$_H$	RDR10[R]/TDR10[W] B,H,W[1] -------0 00000000		BGR110 [R/W] H,W 00000000	BGR010 [R/W] H,W 00000000	
0000 00F8$_H$	ISMK10 [R/W] B,H[2] --------	ISBA10 [R/W] B,H[2] --------	-		
0000 00FC$_H$	FCR110 [R/W] B,H,W ---00100	FCR010 [R,R/W] B,H,W -0000000	FBYTE210 [R/W] B,H,W 00000000	FBYTE110 [R/W] B,H,W 00000000	
0000 0100$_H$	SCR11 [R/W] IBCR11 [R,R/W] B,H,W[2] 0--00000	SMR11 [R/W] B,H,W 000-0000	SSR11 [R,R/W] B,H,W 0-000011	ESCR11 [R/W] IBSR11 [R,R/W] B,H,W[2] -0000000	Multi-function serial interface ch.11 (FIFO)
0000 0104$_H$	RDR11[R]/TDR11[W] B,H,W[1] -------0 00000000		BGR111 [R/W] H,W 00000000	BGR011 [R/W] H,W 00000000	
0000 0108$_H$	ISMK11 [R/W] B,H[2] --------	ISBA11 [R/W] B,H[2] --------	-		
0000 010C$_H$	FCR111 [R/W] B,H,W ---00100	FCR011 [R,R/W] B,H,W -0000000	FBYTE211 [R/W] B,H,W 00000000	FBYTE111 [R/W] B,H,W 00000000	
0000 0110$_H$	EIRR1[R/W] B,H,W 00000000	ENIR1[R/W] B,H,W 00000000	ELVR1[R/W] B,H,W 00000000 00000000		External interrupt 8 to 15
0000 0114$_H$ to 0000 011C$_H$	-				Reserved
0000 0120$_H$	ADCR0[R/W] B,H 000-0000	ADSR0[R,R/W] B,H 00---000	-		A/D converter
0000 0124$_H$	SCCR0[R,R/W] B,H 1000-000	SFNS0[R/W] B,H ----0000	SCFD0[R] B,H XXXXXXXX XX-XXXXX		
0000 0128$_H$	-			SCIS00[R/W] B 00000000	
0000 012C$_H$	PCCR0[R,R/W] B,H 1000-000	PFNS0[R/W] B,H ------00	PCFD0[R] B,H XXXXXXXX XXXXXXXX		
0000 0130$_H$	PCIS0[R/W] B 00000000	-	CMPD0[R/W] B,H 00000000	CMPCR0[R/W] B,H 00000000	
0000 0134$_H$	-			ADSS00[R/W] B 00000000	
0000 0138$_H$	ADST00[R/W] B,H 00100000	ADST10[R/W] B,H 00100000	ADCT0[R/W] B -----111	-	
0000 013C$_H$	-				Reserved

Table A-1. I/O Map (Sheet 4 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0140$_H$	BT0TMR[R]H 00000000 00000000		BT0TMCR[R/W] B,H -0000000 00000000		Base timer ch.0
0000 0144$_H$	-	BT0STC[R/W]B 0000-000	-		
0000 0148$_H$	BT0PCSR/BT0PRLL[R/W]H XXXXXXXX XXXXXXXX		BT0PDUT/BT0PRLH/BT0DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 014C$_H$	-				
0000 0150$_H$	BT1TMR[R]H 00000000 00000000		BT1TMCR[R/W] B,H -0000000 00000000		Base timer ch.1
0000 0154$_H$	-	BT1STC[R/W]B 0000-000	-		
0000 0158$_H$	BT1PCSR/BT1PRLL[R/W]H XXXXXXXX XXXXXXXX		BT1PDUT/BT1PRLH/BT1DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 015C$_H$	-				
0000 0160$_H$	BT2TMR[R]H 00000000 00000000		BT2TMCR [R/W] B,H -0000000 00000000		Base timer ch.2
0000 0164$_H$	-	BT2STC[R/W]B 0000-000	-		
0000 0168$_H$	BT2PCSR/BT2PRLL[R/W]H XXXXXXXX XXXXXXXX		BT2PDUT/BT2PRLH/BT2DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 016C$_H$	-				
0000 0170$_H$	BT3TMR[R]H 00000000 00000000		BT3TMCR[R/W] B,H -0000000 00000000		Base timer ch.3
0000 0174$_H$	-	BT3STC[R/W]B 0000-000	-		
0000 0178$_H$	BT3PCSR/BT3PRLL[R/W]H XXXXXXXX XXXXXXXX		BT3PDUT/BT3PRLH/BT3DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 017C$_H$	BTSEL0123 [R/W] B 00000000	-			
0000 0180$_H$ to 0000 01A8$_H$	-				Reserved
0000 01AC$_H$	ADCHE [R/W] B,H,W -------- -------- -------- 11111111				A/D channel enable
0000 01B0$_H$	IRPR0H [R] B 000-----	-	IRPR1H [R] B,H 000-000-	IRPR1L [R] B,H 000-000-	Interrupt request batch read function
0000 01B4$_H$	-	IRPR2L [R] B,H,W 000-----	IRPR3H [R] B,H,W 0000----	-	
0000 01B8$_H$	IRPR4H [R] B,H,W 0000----	-	IRPR5H [R] B,H,W 0000----	IRPR5L [R] B,H,W 0000----	
0000 01BC$_H$	-			IRPR7L [R] B,H,W 0000----	
0000 01C0$_H$ to 0000 01FC$_H$	-				Reserved

Table A-1. I/O Map (Sheet 5 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0200$_H$	CPCLR0 [R/W] W 11111111 11111111 11111111 11111111				32-bit Free-run timer ch.0
0000 0204$_H$	TCDT0 [R/W] W 00000000 00000000 00000000 00000000				
0000 0208$_H$	TCCSH0 [R/W] B,H 0-----00	TCCSL0 [R/W] B,H -1-00000	-		
0000 020C$_H$	IPCP0 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				32-bit Input capture ch.0 to ch.3
0000 0210$_H$	IPCP1 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0214$_H$	IPCP2 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0218$_H$	IPCP3 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 021C$_H$	-	ICS01 [R/W] B 00000000	-	ICS23 [R/W] B 00000000	
0000 0220$_H$ to 0000 0230$_H$	-				Reserved
0000 0234$_H$	OCCP0 [R/W] W 00000000 00000000 00000000 00000000				32-bit Output compare ch.0 to ch.3
0000 0238$_H$	OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
0000 023C$_H$	OCCP2 [R/W] W 00000000 00000000 00000000 00000000				
0000 0240$_H$	OCCP3 [R/W] W 00000000 00000000 00000000 00000000				
0000 0244$_H$	OCSH1 [R/W] B,H,W ---0--00	OCSL0 [R/W] B,H,W 0000--00	OCSH3 [R/W] B,H,W ---0--00	OCSL2 [R/W] B,H,W 0000--00	
0000 0248$_H$ to 0000 031C$_H$	-				Reserved
0000 0320$_H$	FCTLR[R/W] H -0--1011 --------		-	FSTR[R] B -------1	Flash memory control
0000 0324$_H$ to 0000 0334$_H$	-				Reserved
0000 0338$_H$	-		WREN[R/W] B,H 00000000 00000000		Wild register
0000 033C$_H$	-				
0000 0340$_H$ to 0000 037C$_H$	-				Reserved

Table A-1. I/O Map (Sheet 6 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0380$_H$	WRAR00[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild register
0000 0384$_H$	WRDR00[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0388$_H$	WRAR01[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 038C$_H$	WRDR01[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0390$_H$	WRAR02[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 0394$_H$	WRDR02[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0398$_H$	WRAR03[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 039C$_H$	WRDR03[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03A0$_H$	WRAR04[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03A4$_H$	WRDR04[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03A8$_H$	WRAR05[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03AC$_H$	WRDR05[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03B0$_H$	WRAR06[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03B4$_H$	WRDR06[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03B8$_H$	WRAR07[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03BC$_H$	WRDR07[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03C0$_H$	WRAR08[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03C4$_H$	WRDR08[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03C8$_H$	WRAR09[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03CC$_H$	WRDR09[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table A-1. I/O Map (Sheet 7 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 03D0$_H$	WRAR10[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild register
0000 03D4$_H$	WRDR10[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03D8$_H$	WRAR11[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03DC$_H$	WRDR11[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03E0$_H$	WRAR12[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03E4$_H$	WRDR12[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03E8$_H$	WRAR13[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03EC$_H$	WRDR13[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03F0$_H$	WRAR14[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03F4$_H$	WRDR14[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03F8$_H$	WRAR15[R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0000 03FC$_H$	WRDR15[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0400$_H$	DDR0 [R/W] B,H 00000000	DDR1 [R/W] B,H 00000000	DDR2 [R/W] B,H 00000000	DDR3 [R/W] B,H 00000000	Data direction register
0000 0404$_H$	-	DDR5 [R/W] B,H 00000000	-	DDR7[R/W] B,H 00000000	
0000 0408$_H$ to 0000 0410$_H$	-				
0000 0414$_H$	DDRK [R/W] B ------00	-			
0000 0418$_H$ to 0000 041C$_H$	-				
0000 0420$_H$	PCR0 [R/W] B,H 00000000	PCR1 [R/W] B,H 00000000	-		Pull-up control register
0000 0424$_H$	-	PCR5 [R/W] B 00000000	-	PCR7[R/W] B,H 00000000	
0000 0428$_H$ to 0000 043C$_H$	-				

Table A-1. I/O Map (Sheet 8 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0440$_H$	ICR00 [R,R/W] B,H,W ---11111	ICR01 [R,R/W] B,H,W ---11111	ICR02 [R,R/W] B,H,W ---11111	ICR03 [R,R/W] B,H,W ---11111	Interrupt control
0000 0444$_H$	ICR04 [R,R/W] B,H,W ---11111	ICR05 [R,R/W] B,H,W ---11111	ICR06 [R,R/W] B,H,W ---11111	ICR07 [R,R/W] B,H,W ---11111	
0000 0448$_H$	ICR08 [R,R/W] B,H,W ---11111	ICR09 [R,R/W] B,H,W ---11111	ICR10 [R,R/W] B,H,W ---11111	ICR11 [R,R/W] B,H,W ---11111	
0000 044C$_H$	ICR12 [R,R/W] B,H,W ---11111	ICR13 [R,R/W] B,H,W ---11111	ICR14 [R,R/W] B,H,W ---11111	ICR15 [R,R/W] B,H,W ---11111	
0000 0450$_H$	ICR16 [R,R/W] B,H,W ---11111	ICR17 [R,R/W] B,H,W ---11111	ICR18 [R,R/W] B,H,W ---11111	ICR19 [R,R/W] B,H,W ---11111	
0000 0454$_H$	ICR20 [R,R/W] B,H,W ---11111	ICR21 [R,R/W] B,H,W ---11111	ICR22 [R,R/W] B,H,W ---11111	ICR23 [R,R/W] B,H,W ---11111	
0000 0458$_H$	ICR24 [R,R/W] B,H,W ---11111	ICR25 [R,R/W] B,H,W ---11111	ICR26 [R,R/W] B,H,W ---11111	ICR27 [R,R/W] B,H,W ---11111	
0000 045C$_H$	ICR28 [R,R/W] B,H,W ---11111	ICR29 [R,R/W] B,H,W ---11111	ICR30 [R,R/W] B,H,W ---11111	ICR31 [R,R/W] B,H,W ---11111	
0000 0460$_H$	ICR32 [R,R/W] B,H,W ---11111	ICR33 [R,R/W] B,H,W ---11111	ICR34 [R,R/W] B,H,W ---11111	ICR35 [R,R/W] B,H,W ---11111	
0000 0464$_H$	ICR36 [R,R/W] B,H,W ---11111	ICR37 [R,R/W] B,H,W ---11111	ICR38 [R,R/W] B,H,W ---11111	ICR39 [R,R/W] B,H,W ---11111	
0000 0468$_H$	ICR40 [R,R/W] B,H,W ---11111	ICR41 [R,R/W] B,H,W ---11111	ICR42 [R,R/W] B,H,W ---11111	ICR43 [R,R/W] B,H,W ---11111	
0000 046C$_H$	ICR44 [R,R/W] B,H,W ---11111	ICR45 [R,R/W] B,H,W ---11111	ICR46 [R,R/W] B,H,W ---11111	ICR47 [R,R/W] B,H,W ---11111	
0000 0470$_H$ to 0000 047C$_H$	-				Reserved
0000 0480$_H$	RSTRR [R] B,H,W 11XX---X[3]	RSTCR [R/W] B,H,W 000----0	STBCR [R/W] B,H,W 0000--11	SLPRR [R/W] B,H,W 00000000	Reset control/Power consumption control
0000 0484$_H$	-				
0000 0488$_H$	DIVR0 [R/W] B,H 000--011	-	DIVR2 [R/W] B 0011----	-	Clock division control
0000 048C$_H$	-				
0000 0490$_H$	IORR0 [R/W] B,H,W -0000000	IORR1 [R/W] B,H,W -0000000	IORR2 [R/W] B,H,W -0000000	IORR3 [R/W] B,H,W -0000000	Peripheral DMA transmission request control
0000 0494$_H$	IORR4 [R/W] B,H,W -0000000	IORR5 [R/W] B,H,W -0000000	IORR6 [R/W] B,H,W -0000000	IORR7 [R/W] B,H,W -0000000	
0000 0498$_H$ to 0000 049C$_H$	-				Reserved
0000 04A0$_H$	PFR0 [R/W] B,H 00000000	PFR1 [R/W] B,H 00000000	PFR2 [R/W] B,H 00000000	PFR3 [R/W] B,H 00000000	Port function register
0000 04A4$_H$	-	PFR5 [R/W] B,H 00000000	-	PFR7[R/W] B,H 00000000	
0000 04A8$_H$ to 0000 04B0$_H$	-				
0000 04B4$_H$	PFRK [R/W] B,H ----0000	-			

Table A-1. I/O Map (Sheet 9 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 04B8$_H$	EPFR0 [R/W] B,H ---00-00	EPFR1 [R/W] B,H ---00-00	-		Extended port function register
0000 04BC$_H$	-		EPFR6 [R/W] B,H -00-00-0	EPFR7 [R/W] B,H ----0-0-	
0000 04C0$_H$	EPFR8 [R/W] B,H ----0-0-	EPFR9 [R/W] B,H ----00-0	EPFR10 [R/W] B,H ----0---	-	
0000 04C4$_H$	-		EPFR14 [R/W] B,H ----0-0-	EPFR15 [R/W] B,H ----0-0-	
0000 04C8$_H$	EPFR16 [R/W] B,H ----0-0-	EPFR17 [R/W] B,H ----0-0-	-	EPFR19 [R/W] B,H -------1	
0000 04CC$_H$	EPFR20 [R/W] B,H ---0--0-	EPFR21 [R/W] B,H ---0--0-	EPFR22 [R/W] B,H ---0--0-	EPFR23 [R/W] B,H ---0--0-	
0000 04D0$_H$, 0000 04D4$_H$	-				
0000 04D8$_H$	-	EPFR33 [R/W] B,H ---0--0-	EPFR34 [R/W] B,H --0-----	-	
0000 04DC$_H$	-				
0000 04E0$_H$ to 0000 04EC$_H$	-				Reserved
0000 04F0$_H$	ICSEL0[R/W] B,H,W -----000	ICSEL1[R/W] B,H,W -----000	-		DMA start request clear select function
0000 04F4$_H$	ICSEL4[R/W] B,H,W ------00	-	ICSEL6[R/W] B,H,W ------00	ICSEL7[R/W] B,H,W -------0	
0000 04F8$_H$	ICSEL8[R/W] B,H,W ------00	-	ICSEL10[R/W] B,H,W ----0000	ICSEL11[R/W] B,H,W ----0000	
0000 04FC$_H$	-				
0000 0500$_H$ to 0000 050C$_H$	-				Reserved
0000 0510$_H$	CSELR [R/W] B,H,W 001---00	CMONR [R] B,H,W 001---00	MTMCR [R/W] B,H,W 00001111	STMCR [R/W]B,H,W 0000-111	Clock generation/ Main timer/ Sub timer
0000 0514$_H$	PLLCR [R/W] B,H --000000 11110000		CSTBR [R/W] B -0000000	-	
0000 0518$_H$	WCRD [R] B,H --000000	WCRL [R/W] B,H --000000	WCCR [R,R/W] B 00--0000	-	Clock counter
0000 051C$_H$	UCCR [R/W] B -----001	-			USB clock generation
0000 0520$_H$ to 0000 0BFC$_H$	-				Reserved

Table A-1. I/O Map (Sheet 10 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0C00$_H$	DCCR0 [R/W] W 0----000 --00--00 00000000 0-000000				DMAC
0000 0C04$_H$	DCSR0 [R,R/W] H 0------- -----000		DTCR0 [R/W] H 00000000 00000000		
0000 0C08$_H$	DSAR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C0C$_H$	DDAR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C10$_H$	DCCR1 [R/W] W 0----000 --00--00 00000000 0-000000				
0000 0C14$_H$	DCSR1 [R,R/W] H 0------- -----000		DTCR1 [R/W] H 00000000 00000000		
0000 0C18$_H$	DSAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C1C$_H$	DDAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table A-1. I/O Map (Sheet 11 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0C20$_H$	DCCR2 [R/W] W 0----000 --00--00 00000000 0-000000				DMAC
0000 0C24$_H$	DCSR2 [R,R/W] H 0------- -----000		DTCR2 [R/W] H 00000000 00000000		
0000 0C28$_H$	DSAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C2C$_H$	DDAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C30$_H$	DCCR3 [R/W] W 0----000 --00--00 00000000 0-000000				
0000 0C34$_H$	DCSR3 [R,R/W] H 0------- -----000		DTCR3 [R/W] H 00000000 00000000		
0000 0C38$_H$	DSAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C3C$_H$	DDAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C40$_H$	DCCR4 [R/W] W 0----000 --00--00 00000000 0-000000				
0000 0C44$_H$	DCSR4 [R,R/W] H 0------- -----000		DTCR4 [R/W] H 00000000 00000000		
0000 0C48$_H$	DSAR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C4C$_H$	DDAR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C50$_H$	DCCR5 [R/W] W 0----000 --00--00 00000000 0-000000				
0000 0C54$_H$	DCSR5 [R,R/W] H 0------- -----000		DTCR5 [R/W] H 00000000 00000000		
0000 0C58$_H$	DSAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C5C$_H$	DDAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C60$_H$	DCCR6 [R/W] W 0----000 --00--00 00000000 0-000000				
0000 0C64$_H$	DCSR6 [R,R/W] H 0------- -----000		DTCR6 [R/W] H 00000000 00000000		
0000 0C68$_H$	DSAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C6C$_H$	DDAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C70$_H$	DCCR7 [R/W] W 0----000 --00--00 00000000 0-000000				
0000 0C74$_H$	DCSR7 [R,R/W] H 0------- -----000		DTCR7 [R/W] H 00000000 00000000		

Table A-1. I/O Map (Sheet 12 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0C78$_H$	DSAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMAC
0000 0C7C$_H$	DDAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C80$_H$ to 0000 0DF0$_H$	-				
0000 0DF4$_H$	-			DILVR [R,R/W] B ---11111	
0000 0DF8$_H$	DMACR [R/W] W 0------- -------- 0------- --------				
0000 0DFC$_H$ to 0000 0F3C$_H$	-				Reserved
0000 0F40$_H$	BT4TMR[R]H 00000000 00000000		BT4TMCR[R/W] B,H -0000000 00000000		Base timer ch.4
0000 0F44$_H$	-	BT4STC[R/W]B 0000-000	-		
0000 0F48$_H$	BT4PCSR/BT4PRLL [R/W]H XXXXXXXX XXXXXXXX		BT4PDUT/BT4PRLH/BT4DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0F4C$_H$	-				
0000 0F50$_H$	BT5TMR[R]H 00000000 00000000		BT5TMCR[R/W] B,H -0000000 00000000		Base timer ch.5
0000 0F54$_H$	-	BT5STC[R/W]B 0000-000	-		
0000 0F58$_H$	BT5PCSR/BT5PRLL [R/W]H XXXXXXXX XXXXXXXX		BT5PDUT/BT5PRLH/BT5DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0F5C$_H$	-				
0000 0F60$_H$	BT6TMR[R]H 00000000 00000000		BT6TMCR[R/W] B,H -0000000 00000000		Base timer ch.6
0000 0F64$_H$	-	BT6STC[R/W]B 0000-000	-		
0000 0F68$_H$	BT6PCSR/BT6PRLL [R/W]H XXXXXXXX XXXXXXXX		BT6PDUT/BT6PRLH/BT6DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0F6C$_H$	-				
0000 0F70$_H$	BT7TMR[R]H 00000000 00000000		BT7TMCR[R/W] B,H -0000000 00000000		Base timer ch.7
0000 0F74$_H$	-	BT7STC[R/W]B 0000-000	-		
0000 0F78$_H$	BT7PCSR/BT7PRLL [R/W]H XXXXXXXX XXXXXXXX		BT7PDUT/BT7PRLH/BT7DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0F7C$_H$	BTSEL4567 [R/W] B 00000000	-			
0000 0F80$_H$ to 0000 0FF8$_H$	-				Reserved

Table A-1. I/O Map (Sheet 13 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 0FFC$_H$	-		BTSSSR[W] H XXXXXXXX XXXXXXXX		Base Timer I/O Select Function
0000 1000$_H$ to 0000 20FC$_H$	-				Reserved
0000 2100$_H$	HCNT1[R/W] B,H -----001	HCNT0[R/W] B,H 00000000	-		USB function/HOST
0000 2104$_H$	HERR[R/W] B,H 00000011	HIRQ[R/W] B,H 0-000000	-		
0000 2108$_H$	HFCOMP[R/W] B,H 00000000	HSTATE[R,R/W] B,H ---10010	-		
0000 210C$_H$	HRTIMER1[R/W] B,H 00000000	HRTIMER0[R/W] B,H 00000000	-		
0000 2110$_H$	HADR[R/W] B,H -0000000	HRTIMER2[R/W] B,H ------00	-		
0000 2114$_H$	HEOF1[R/W] B,H --000000	HEOF0[R/W] B,H 00000000	-		
0000 2118$_H$	HFRAME1[R/W] B,H -----000	HFRAME0[R/W] B,H 00000000	-		
0000 211C$_H$	-	HTOKEN[R/W] B 00000000	-		
0000 2120$_H$	-	UDCC[R/W] B 1010--00	-		

Table A-1. I/O Map (Sheet 14 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 2124$_H$	EP0C[R/W] H ------0- -1000000		-		USB function/HOST
0000 2128$_H$	EP1C[R/W] H 01100001 00000000		-		
0000 212C$_H$	EP2C[R/W] H 0110000- -1000000		-		
0000 2130$_H$	EP3C[R/W] H 0110000- -1000000		-		
0000 2134$_H$	EP4C[R/W] H 0110000- -1000000		-		
0000 2138$_H$	EP5C[R/W] H 0110000- -1000000		-		
0000 213C$_H$	TMSP[R] H -----000 00000000		-		
0000 2140$_H$	UDCIE[R,R/W] B,H --000000	UDCS[R/W] B,H --000000	-		
0000 2144$_H$	EP0IS[R/W] H 10---1-- --------		-		
0000 2148$_H$	EP00S[R,R/W] H 100--00- -XXXXXXX		-		
0000 214C$_H$	EP1S[R,R/W] H 100-000X XXXXXXXX		-		
0000 2150$_H$	EP2S[R,R/W] H 100-000- -XXXXXXX		-		
0000 2154$_H$	EP3S[R,R/W] H 100-000- -XXXXXXX		-		
0000 2158$_H$	EP4S[R,R/W] H 100-000- -XXXXXXX		-		
0000 215C$_H$	EP5S[R,R/W] H 100-000- -XXXXXXX		-		
0000 2160$_H$	EP0DTH [R/W] B,H XXXXXXXX	EP0DTL [R/W] B,H XXXXXXXX	-		
0000 2164$_H$	EP1DTH [R/W] B,H XXXXXXXX	EP1DTL [R/W] B,H XXXXXXXX	-		
0000 2168$_H$	EP2DTH [R/W] B,H XXXXXXXX	EP2DTL [R/W] B,H XXXXXXXX	-		
0000 216C$_H$	EP3DTH [R/W] B,H XXXXXXXX	EP3DTL [R/W] B,H XXXXXXXX	-		
0000 2170$_H$	EP4DTH [R/W] B,H XXXXXXXX	EP4DTL [R/W] B,H XXXXXXXX	-		
0000 2174$_H$	EP5DTH [R/W] B,H XXXXXXXX	EP5DTL [R/W] B,H XXXXXXXX	-		
0000 2178$_H$ to 0000 217C$_H$	-				
0000 2180$_H$ to 0000 21A0$_H$	-				Reserved
0000 21A4$_H$	DREQSEL [R/W] B,H 00111011	USBSEL [R/W] B,H -------0	USBEN [R/W] B -------0	-	DMA transfer request selector/ USB enable

Table A-1. I/O Map (Sheet 15 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 21A8$_H$ to 0000 3FFC$_H$	-				Reserved
0000 4000$_H$	MOSD_VADR [W] W -------- -------0 ---00000 --000000				OSDC (MAIN)
0000 4004$_H$	MOSD_CDS1 [W] W 00000000 ---00000 00000000 00000000				
0000 4008$_H$	MOSD_CDS2 [W] W -------- 0000-000 --000000 00000000				
0000 400C$_H$	MOSD_LDS1 [W] W 0000-000 00000000 ----0000 00000000				
0000 4010$_H$	MOSD_LDS2 [W] W -------- ---00000 --000000 00000000				
0000 4014$_H$	MOSD_SCOC [W] W ------00 0000---- ---0---0 XXXX----				
0000 4018$_H$	MOSD_HVDP [W] W -----000 00000000 -----000 00000000				
0000 401C$_H$	MOSD_TSBC [W] W -------- -------- -------0 00000000				
0000 4020$_H$	MOSD_GRCC [W] W -------0 00000000 -------0 00000000				
0000 4024$_H$	MOSD_SBCC [W] W -----000 ------00 --000000 00000000				
0000 4028$_H$	MOSD_SCBC [W] W -------- --00--00 ---0-000 00000000				
0000 402C$_H$	MOSD_WPC1 [W] W -----000 00000000 -----000 00000000				

Table A-1. I/O Map (Sheet 16 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 4030$_H$	MOSD_WPC2 [W] W ----0000 00000000 ----0000 00000000				OSDC (MAIN)
0000 4034$_H$	MOSD_SPC1 [W] W ---0-000 ------00 --000000 00000000				
0000 4038$_H$	MOSD_SPC2 [W] W ----0000 00000000 -----000 00000000				
0000 403C$_H$	MOSD_SYNC [W] W -------- --000000 -------- -0-0----				
0000 4040$_H$	MOSD_CBC0 [W] W --000000 00000000 --000000 00000000				
0000 4044$_H$	MOSD_CBC1 [W] W --000000 00000000 --000000 00000000				
0000 4048$_H$	MOSD_CBC2 [W] W --000000 00000000 --000000 00000000				
0000 404C$_H$	MOSD_CBC3 [W] W --000000 00000000 --000000 00000000				
0000 4050$_H$	MOSD_CBC4 [W] W --000000 00000000 --000000 00000000				
0000 4054$_H$	MOSD_CBC5 [W] W --000000 00000000 --000000 00000000				
0000 4058$_H$	MOSD_CBC6 [W] W --000000 00000000 --000000 00000000				
0000 405C$_H$	MOSD_CBC7 [W] W --000000 00000000 --000000 00000000				
0000 4060$_H$	MOSD_IOTC [W] W -------0 0----00- -------- -----XXX				
0000 4064$_H$	MOSD_CDP1 [W] W -----000 00000000 -----000 00000000				
0000 4068$_H$	MOSD_CDP2 [W] W ----0000 00000000 ----0000 00000000				
0000 406C$_H$	MOSD_INTC [R/W] W -------- -------- -----XXX -----XXX				
0000 4070$_H$	MOSD_SBC0 [W] W 00000000 00000000 00000000 00000000				
0000 4074$_H$	MOSD_SBC1 [W] W 00000000 00000000 00000000 00000000				
0000 4078$_H$	MOSD_SBC2 [W] W 00000000 00000000 00000000 00000000				
0000 407C$_H$	MOSD_SBC3 [W] W 00000000 00000000 00000000 00000000				
0000 4080$_H$ to 0000 40FC$_H$	-				Reserved

Table A-1. I/O Map (Sheet 17 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
0000 4100$_H$	SOSD_VADR [W] W -------- -------0 ---00000 --000000				OSDC (SUB)
0000 4104$_H$	SOSD_CDS1 [W] W 00000000 ---00000 00000000 00000000				
0000 4108$_H$	SOSD_CDS2 [W] W -------- 0000-000 --000000 00000000				
0000 410C$_H$	SOSD_LDS1 [W] W 0000-000 00000000 ----0000 00000000				
0000 4110$_H$	SOSD_LDS2 [W] W -------- ---00000 --000000 00000000				
0000 4114$_H$	SOSD_SCOC [W] W ------00 0000---- ---0---0 XX-X---X				
0000 4118$_H$	SOSD_HVDP [W] W -----000 00000000 -----000 00000000				
0000 411C$_H$	SOSD_TSBC [W] W -------- -------- -------0 00000000				
0000 4120$_H$	SOSD_GRCC [W] W -------0 00000000 -------0 00000000				
0000 4124$_H$	-				
0000 4128$_H$	SOSD_SCBC [W] W -------- --00--00 ---0-000 00000000				
0000 412C$_H$	SOSD_WPC1 [W] W -----000 00000000 -----000 00000000				
0000 4130$_H$	SOSD_WPC2 [W] W ----0000 00000000 ----0000 00000000				
0000 4134$_H$	SOSD_SPC1 [W] W ---0-000 ------00 --000000 00000000				
0000 4138$_H$	SOSD_SPC2 [W] W ----0000 00000000 -----000 00000000				
0000 413C$_H$ to 0000 4168$_H$	-				
0000 416C$_H$	SOSD_INTC [R/W] W -------- -------- -----XXX -----XXX				
0000 4170$_H$	SOSD_SBC0 [W] W 00000000 00000000 00000000 00000000				
0000 4174$_H$	SOSD_SBC1 [W] W 00000000 00000000 00000000 00000000				
0000 4178$_H$	SOSD_SBC2 [W] W 00000000 00000000 00000000 00000000				
0000 417C$_H$	SOSD_SBC3 [W] W 00000000 00000000 00000000 00000000				
0000 4180$_H$ to 0000 41FC$_H$	-				Reserved

Table A-1. I/O Map (Sheet 18 of 18)

Address	Register				Block
	+ 0	+ 1	+ 2	+ 3	
$0000\ 4200_H$ to $0000\ 43FC_H$	MOSD_PLn [W] W *n: 0 to 127 00000000 00000000 00000000 00000000				OSDC (MAIN)
$0000\ 4400_H$	MOSD_OSDC [W] W -------- --XX--XX ------XX ---X---X				
$0000\ 4404_H$	MOSD_PLLC [W] W --000000 00000000 00000000 ---00000				
$0000\ 4408_H$ to $0000\ FFFC_H$	-				Reserved

[1] : Byte access is available only when accessing the lower 8 bits within 9 bits.

[2] : The register of I^2C can not be read immediate after reset.

[3] : Value just after reset by $\overline{INIT}$ pin.
Do not access the reserved areas.

A.2 List of Registers

This appendix lists the registers that can be used with this series.

The list is sorted in alphabetical order by abbreviated register name of this series.

Abbreviated Name	Register Name	Address	Reference
ADCHE	A/D channel enable register	0000 01AC$_H$	13.4.6
ADCR0	A/DC control register 0	0000 0120$_H$	23.4.1
ADCT0	Compare time setting register 0	0000 013A$_H$	23.4.15
ADSR0	A/DC status register 0	0000 0121$_H$	23.4.2
ADSS00	Sampling time select register 00	0000 0137$_H$	23.4.14
ADST00	Sampling time setting register 00	0000 0138$_H$	23.4.13
ADST10	Sampling time setting register 10	0000 0139$_H$	23.4.13
BGR00	Baud rate generator register 00	0000 0067$_H$	24.4.6, 24.13.6
BGR01	Baud rate generator register 01	0000 006F$_H$	24.4.6, 24.13.6, 24.21.8
BGR02	Baud rate generator register 02	0000 007B$_H$	24.4.6, 24.13.6, 24.21.8
BGR03	Baud rate generator register 03	0000 0087$_H$	24.4.6, 24.13.6, 24.21.8
BGR08	Baud rate generator register 08	0000 00D7$_H$	24.4.6, 24.13.6, 24.21.8
BGR09	Baud rate generator register 09	0000 00E7$_H$	24.4.6, 24.13.6, 24.21.8
BGR010	Baud rate generator register 010	0000 00F7$_H$	24.4.6, 24.13.6, 24.21.8
BGR011	Baud rate generator register 011	0000 0107$_H$	24.4.6, 24.13.6, 24.21.8
BGR10	Baud rate generator register 10	0000 0066$_H$	24.4.6, 24.13.6
BGR11	Baud rate generator register 11	0000 006E$_H$	24.4.6, 24.13.6, 24.21.8
BGR12	Baud rate generator register 12	0000 007A$_H$	24.4.6, 24.13.6, 24.21.8
BGR13	Baud rate generator register 13	0000 0086$_H$	24.4.6, 24.13.6, 24.21.8
BGR18	Baud rate generator register 18	0000 00D6$_H$	24.4.6, 24.13.6, 24.21.8
BGR19	Baud rate generator register 19	0000 00E6$_H$	24.4.6, 24.13.6, 24.21.8
BGR110	Baud rate generator register 110	0000 00F6$_H$	24.4.6, 24.13.6, 24.21.8
BGR111	Baud rate generator register 111	0000 0106$_H$	24.4.6, 24.13.6, 24.21.8
BT0DTBF	Base timer 0 data buffer register	0000 014A$_H$	22.8.4.2
BT0PCSR	Base timer 0 cycle setting register	0000 0148$_H$	22.8.1.2, 22.8.3.2
BT0PDUT	Base timer 0 duty setting register	0000 014A$_H$	22.8.1.3
BT0PRLH	Base timer 0 H width setting register	0000 014A$_H$	22.8.2.3
BT0PRLL	Base timer 0 L width setting register	0000 0148$_H$	22.8.2.2
BT0STC	Base timer 0 status control register	0000 0145$_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT0TMCR	Base timer 0 timer control register	0000 0142$_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT0TMR	Base timer 0 timer register	0000 0140$_H$	22.8.1.4, 22.8.2.4, 22.8.3.3
BT1DTBF	Base timer 1 data buffer register	0000 015A$_H$	22.8.4.2
BT1PCSR	Base timer 1 cycle setting register	0000 0158$_H$	22.8.1.2, 22.8.3.2
BT1PDUT	Base timer 1 duty setting register	0000 015A$_H$	22.8.1.3
BT1PRLH	Base timer 1 H width setting register	0000 015A$_H$	22.8.2.3
BT1PRLL	Base timer 1 L width setting register	0000 0158$_H$	22.8.2.2
BT1STC	Base timer 1 status control register	0000 0155$_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT1TMCR	Base timer 1 timer control register	0000 0152$_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT1TMR	Base timer 1 timer register	0000 0150$_H$	22.8.1.4, 22.8.2.4, 22.8.3.3
BT2DTBF	Base timer 2 data buffer register	0000 016A$_H$	22.8.4.2

Abbreviated Name	Register Name	Address	Reference
BT2PCSR	Base timer 2 cycle setting register	0000 0168_H	22.8.1.2, 22.8.3.2
BT2PDUT	Base timer 2 duty setting register	0000 $016A_H$	22.8.1.3
BT2PRLH	Base timer 2 H width setting register	0000 $016A_H$	22.8.2.3
BT2PRLL	Base timer 2 L width setting register	0000 0168_H	22.8.2.2
BT2STC	Base timer 2 status control register	0000 0165_H	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT2TMCR	Base timer 2 timer control register	0000 0162_H	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT2TMR	Base timer 2 timer register	0000 0160_H	22.8.1.4, 22.8.2.4, 22.8.3.3
BT3DTBF	Base timer 3 data buffer register	0000 $017A_H$	22.8.4.2
BT3PCSR	Base timer 3 cycle setting register	0000 0178_H	22.8.1.2, 22.8.3.2
BT3PDUT	Base timer 3 duty setting register	0000 $017A_H$	22.8.1.3
BT3PRLH	Base timer 3 H width setting register	0000 $017A_H$	22.8.2.3
BT3PRLL	Base timer 3 L width setting register	0000 0178_H	22.8.2.2
BT3STC	Base timer 3 status control register	0000 0175_H	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT3TMCR	Base timer 3 timer control register	0000 0172_H	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT3TMR	Base timer 3 timer register	0000 0170_H	22.8.1.4, 22.8.2.4, 22.8.3.3
BT4DTBF	Base timer 4 data buffer register	0000 $0F4A_H$	22.8.4.2
BT4PCSR	Base timer 4 cycle setting register	0000 $0F48_H$	22.8.1.2, 22.8.3.2
BT4PDUT	Base timer 4 duty setting register	0000 $0F4A_H$	22.8.1.3
BT4PRLH	Base timer 4 H width setting register	0000 $0F4A_H$	22.8.2.3
BT4PRLL	Base timer 4 L width setting register	0000 $0F48_H$	22.8.2.2
BT4STC	Base timer 4 status control register	0000 $0F45_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT4TMCR	Base timer 4 timer control register	0000 $0F42_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT4TMR	Base timer 4 timer register	0000 $0F40_H$	22.8.1.4, 22.8.2.4, 22.8.3.3
BT5DTBF	Base timer 5 data buffer register	0000 $0F5A_H$	22.8.4.2
BT5PCSR	Base timer 5 cycle setting register	0000 $0F58_H$	22.8.1.2, 22.8.3.2
BT5PDUT	Base timer 5 duty setting register	0000 $0F5A_H$	22.8.1.3
BT5PRLH	Base timer 5 H width setting register	0000 $0F5A_H$	22.8.2.3
BT5PRLL	Base timer 5 L width setting register	0000 $0F58_H$	22.8.2.2
BT5STC	Base timer 5 status control register	0000 $0F55_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT5TMCR	Base timer 5 timer control register	0000 $0F52_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT5TMR	Base timer 5 timer register	0000 $0F50_H$	22.8.1.4, 22.8.2.4, 22.8.3.3
BT6DTBF	Base timer 6 data buffer register	0000 $0F6A_H$	22.8.4.2
BT6PCSR	Base timer 6 cycle setting register	0000 $0F68_H$	22.8.1.2, 22.8.3.2
BT6PDUT	Base timer 6 duty setting register	0000 $0F6A_H$	22.8.1.3
BT6PRLH	Base timer 6 H width setting register	0000 $0F6A_H$	22.8.2.3
BT6PRLL	Base timer 6 L width setting register	0000 $0F68_H$	22.8.2.2
BT6STC	Base timer 6 status control register	0000 $0F65_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT6TMCR	Base timer 6 timer control register	0000 $0F62_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT6TMR	Base timer 6 timer register	0000 $0F60_H$	22.8.1.4, 22.8.2.4, 22.8.3.3
BT7DTBF	Base timer 7 data buffer register	0000 $0F7A_H$	22.8.4.2
BT7PCSR	Base timer 7 cycle setting register	0000 $0F78_H$	22.8.1.2, 22.8.3.2
BT7PDUT	Base timer 7 duty setting register	0000 $0F7A_H$	22.8.1.3
BT7PRLH	Base timer 7 H width setting register	0000 $0F7A_H$	22.8.2.3

Abbreviated Name	Register Name	Address	Reference
BT7PRLL	Base timer 7 L width setting register	0000 0F78$_H$	22.8.2.2
BT7STC	Base timer 7 status control register	0000 0F75$_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT7TMCR	Base timer 7 timer control register	0000 0F72$_H$	22.8.1.1, 22.8.2.1, 22.8.3.1, 22.8.4.1
BT7TMR	Base timer 7 timer register	0000 0F70$_H$	22.8.1.4, 22.8.2.4, 22.8.3.3
BTSEL0123	Base timer io select register for ch.0/1/2/3	0000 017C$_H$	21.4.1
BTSEL4567	Base timer io select register for ch.4/5/6/7	0000 0F7C$_H$	21.4.2
BTSSSR	Base timer same time soft start register	0000 0FFE$_H$	21.4.3
CMONR	Clock source monitor register	0000 0511$_H$	4.4.2
CMPCR0	A/D comparison control register 0	0000 0133$_H$	23.4.12
CMPD0	A/D comparison data setting register 0	0000 0132$_H$	23.4.11
CPCLR0	Compare clear register 0	0000 0200$_H$	17.4.1
CSELR	Clock source select register	0000 0510$_H$	4.4.1
CSTBR	Clock stabilization time select register	0000 0516$_H$	4.4.3
DCCR0	DMA channel control register 0	0000 0C00$_H$	31.3.5
DCCR1	DMA channel control register 1	0000 0C10$_H$	31.3.5
DCCR2	DMA channel control register 2	0000 0C20$_H$	31.3.5
DCCR3	DMA channel control register 3	0000 0C30$_H$	31.3.5
DCCR4	DMA channel control register 4	0000 0C40$_H$	31.3.5
DCCR5	DMA channel control register 5	0000 0C50$_H$	31.3.5
DCCR6	DMA channel control register 6	0000 0C60$_H$	31.3.5
DCCR7	DMA channel control register 7	0000 0C70$_H$	31.3.5
DCSR0	DMA channel status register 0	0000 0C04$_H$	31.3.6
DCSR1	DMA channel status register 1	0000 0C14$_H$	31.3.6
DCSR2	DMA channel status register 2	0000 0C24$_H$	31.3.6
DCSR3	DMA channel status register 3	0000 0C34$_H$	31.3.6
DCSR4	DMA channel status register 4	0000 0C44$_H$	31.3.6
DCSR5	DMA channel status register 5	0000 0C54$_H$	31.3.6
DCSR6	DMA channel status register 6	0000 0C64$_H$	31.3.6
DCSR7	DMA channel status register 7	0000 0C74$_H$	31.3.6
DDAR0	DMA destination address register 0	0000 0C0C$_H$	31.3.3
DDAR1	DMA destination address register 1	0000 0C1C$_H$	31.3.3
DDAR2	DMA destination address register 2	0000 0C2C$_H$	31.3.3
DDAR3	DMA destination address register 3	0000 0C3C$_H$	31.3.3
DDAR4	DMA destination address register 4	0000 0C4C$_H$	31.3.3
DDAR5	DMA destination address register 5	0000 0C5C$_H$	31.3.3
DDAR6	DMA destination address register 6	0000 0C6C$_H$	31.3.3
DDAR7	DMA destination address register 7	0000 0C7C$_H$	31.3.3
DDR0	Port data direction register 0	0000 0400$_H$	13.4.1
DDR1	Port data direction register 1	0000 0401$_H$	13.4.1
DDR2	Port data direction register 2	0000 0402$_H$	13.4.1
DDR3	Port data direction register 3	0000 0403$_H$	13.4.1
DDR5	Port data direction register 5	0000 0405$_H$	13.4.1
DDR7	Port data direction register 7	0000 0407$_H$	13.4.1

Abbreviated Name	Register Name	Address	Reference
DDRK	Port data direction register K	0000 0414_H	13.4.1
DICR	Delayed interrupt control register	0000 0044_H	12.3.1
DILVR	DMA-halt by interrupt level register	0000 $0DF7_H$	31.3.7
DIVR0	Divide clock configuration register 0	0000 0488_H	5.4.1
DIVR2	Divide clock configuration register 2	0000 $048A_H$	5.4.2
DMACR	DMA control register	0000 $0DF8_H$	31.3.1
DREQSEL	DREQ select register	0000 $21A4_H$	26.3.1
DSAR0	DMA source address register 0	0000 $0C08_H$	31.3.2
DSAR1	DMA source address register 1	0000 $0C18_H$	31.3.2
DSAR2	DMA source address register 2	0000 $0C28_H$	31.3.2
DSAR3	DMA source address register 3	0000 $0C38_H$	31.3.2
DSAR4	DMA source address register 4	0000 $0C48_H$	31.3.2
DSAR5	DMA source address register 5	0000 $0C58_H$	31.3.2
DSAR6	DMA source address register 6	0000 $0C68_H$	31.3.2
DSAR7	DMA source address register 7	0000 $0C78_H$	31.3.2
DTCR0	DMA transfer count register 0	0000 $0C06_H$	31.3.4
DTCR1	DMA transfer count register 1	0000 $0C16_H$	31.3.4
DTCR2	DMA transfer count register 2	0000 $0C26_H$	31.3.4
DTCR3	DMA transfer count register 3	0000 $0C36_H$	31.3.4
DTCR4	DMA transfer count register 4	0000 $0C46_H$	31.3.4
DTCR5	DMA transfer count register 5	0000 $0C56_H$	31.3.4
DTCR6	DMA transfer count register 6	0000 $0C66_H$	31.3.4
DTCR7	DMA transfer count register 7	0000 $0C76_H$	31.3.4
EIRR0	External interrupt request register 0	0000 0040_H	14.4.2
EIRR1	External interrupt request register 1	0000 0110_H	14.4.2
ELVR0	External interrupt request level register 0	0000 0042_H	14.4.1
ELVR1	External interrupt request level register 1	0000 0112_H	14.4.1
ENIR0	Enable interrupt request register 0	0000 0041_H	14.4.3
ENIR1	Enable interrupt request register 1	0000 0111_H	14.4.3
EP0C	EP0 control register	0000 2124_H	27.3.4
EP0DTH	EP0 data register upper	0000 2160_H	27.3.12
EP0DTL	EP0 data register lower	0000 2161_H	27.3.12
EP0IS	EP0I status register	0000 2144_H	27.3.9
EP0OS	EP0O status register	0000 2148_H	27.3.10
EP1C	EP1 control register	0000 2128_H	27.3.5
EP1DTH	EP1 data register upper	0000 2164_H	27.3.12
EP1DTL	EP1 data register lower	0000 2165_H	27.3.12
EP1S	EP1 status register	0000 $214C_H$	27.3.11
EP2C	EP2 control register	0000 $212C_H$	27.3.5
EP2DTH	EP2 data register upper	0000 2168_H	27.3.12
EP2DTL	EP2 data register lower	0000 2169_H	27.3.12
EP2S	EP2 status register	0000 2150_H	27.3.11
EP3C	EP3 control register	0000 2130_H	27.3.5

Abbreviated Name	Register Name	Address	Reference
EP3DTH	EP3 data register upper	0000 216C$_H$	27.3.12
EP3DTL	EP3 data register lower	0000 216D$_H$	27.3.12
EP3S	EP3 status register	0000 2154$_H$	27.3.11
EP4C	EP4 control register	0000 2134$_H$	27.3.5
EP4DTH	EP4 data register upper	0000 2170$_H$	27.3.12
EP4DTL	EP4 data register lower	0000 2171$_H$	27.3.12
EP4S	EP4 status register	0000 2158$_H$	27.3.11
EP5C	EP5 control register	0000 2138$_H$	27.3.5
EP5DTH	EP5 data register upper	0000 2174$_H$	27.3.12
EP5DTL	EP5 data register lower	0000 2175$_H$	27.3.12
EP5S	EP5 status register	0000 215C$_H$	27.3.11
EPFR0	Extended port function register 0	0000 04B8$_H$	13.4.3
EPFR1	Extended port function register 1	0000 04B9$_H$	13.4.3
EPFR2	Extended port function register 2	0000 04BA$_H$	13.4.3
EPFR3	Extended port function register 3	0000 04BB$_H$	13.4.3
EPFR4	Extended port function register 4	0000 04BC$_H$	13.4.3
EPFR5	Extended port function register 5	0000 04BD$_H$	13.4.3
EPFR6	Extended port function register 6	0000 04BE$_H$	13.4.3
EPFR7	Extended port function register 7	0000 04BF$_H$	13.4.3
EPFR8	Extended port function register 8	0000 04C0$_H$	13.4.3
EPFR9	Extended port function register 9	0000 04C1$_H$	13.4.3
EPFR10	Extended port function register 10	0000 04C2$_H$	13.4.3
EPFR11	Extended port function register 11	0000 04C3$_H$	13.4.3
EPFR12	Extended port function register 12	0000 04C4$_H$	13.4.3
EPFR13	Extended port function register 13	0000 04C5$_H$	13.4.3
EPFR14	Extended port function register 14	0000 04C6$_H$	13.4.3
EPFR15	Extended port function register 15	0000 04C7$_H$	13.4.3
EPFR16	Extended port function register 16	0000 04C8$_H$	13.4.3
EPFR17	Extended port function register 17	0000 04C9$_H$	13.4.3
EPFR18	Extended port function register 18	0000 04CA$_H$	13.4.3
EPFR19	Extended port function register 19	0000 04CB$_H$	13.4.3
EPFR20	Extended port function register 20	0000 04CC$_H$	13.4.3
EPFR21	Extended port function register 21	0000 04CD$_H$	13.4.3
EPFR22	Extended port function register 22	0000 04CE$_H$	13.4.3
EPFR23	Extended port function register 23	0000 04CF$_H$	13.4.3
EPFR24	Extended port function register 24	0000 04D0$_H$	13.4.3
EPFR25	Extended port function register 25	0000 04D1$_H$	13.4.3
EPFR26	Extended port function register 26	0000 04D2$_H$	13.4.3
EPFR27	Extended port function register 27	0000 04D3$_H$	13.4.3
EPFR28	Extended port function register 28	0000 04D4$_H$	13.4.3
EPFR29	Extended port function register 29	0000 04D5$_H$	13.4.3
EPFR30	Extended port function register 30	0000 04D6$_H$	13.4.3
EPFR31	Extended port function register 31	0000 04D7$_H$	13.4.3

Abbreviated Name	Register Name	Address	Reference
EPFR32	Extended port function register 32	0000 $04D8_H$	13.4.3
EPFR33	Extended port function register 33	0000 $04D9_H$	13.4.3
EPFR34	Extended port function register 34	0000 $04DA_H$	13.4.3
ESCR0	Extended serial control register 0	0000 0063_H	24.4.4, 24.13.4
ESCR1	Extended serial control register 1	0000 $006B_H$	24.4.4, 24.13.4
ESCR2	Extended serial control register 2	0000 0077_H	24.4.4, 24.13.4
ESCR3	Extended serial control register 3	0000 0083_H	24.4.4, 24.13.4
ESCR8	Extended serial control register 8	0000 $00D3_H$	24.4.4, 24.13.4
ESCR9	Extended serial control register 9	0000 $00E3_H$	24.4.4, 24.13.4
ESCR10	Extended serial control register 10	0000 $00F3_H$	24.4.4, 24.13.4
ESCR11	Extended serial control register 11	0000 0103_H	24.4.4, 24.13.4
FBYTE18	FIFO byte register 18	0000 $00DF_H$	24.4.9, 24.13.9, 24.21.11
FBYTE19	FIFO byte register 19	0000 $00EF_H$	24.4.9, 24.13.9, 24.21.11
FBYTE110	FIFO byte register 110	0000 $00FF_H$	24.4.9, 24.13.9, 24.21.11
FBYTE111	FIFO byte register 111	0000 $010F_H$	24.4.9, 24.13.9, 24.21.11
FBYTE28	FIFO byte register 28	0000 $00DE_H$	24.4.9, 24.13.9, 24.21.11
FBYTE29	FIFO byte register 29	0000 $00EE_H$	24.4.9, 24.13.9, 24.21.11
FBYTE210	FIFO byte register 210	0000 $00FE_H$	24.4.9, 24.13.9, 24.21.11
FBYTE211	FIFO byte register 211	0000 $010E_H$	24.4.9, 24.13.9, 24.21.11
FCR08	FIFO control register 08	0000 $00DD_H$	24.4.8, 24.13.8, 24.21.10
FCR09	FIFO control register 09	0000 $00ED_H$	24.4.8, 24.13.8, 24.21.10
FCR010	FIFO control register 010	0000 $00FD_H$	24.4.8, 24.13.8, 24.21.10
FCR011	FIFO control register 011	0000 $010D_H$	24.4.8, 24.13.8, 24.21.10
FCR18	FIFO control register 18	0000 $00DC_H$	24.4.7, 24.13.7, 24.21.9
FCR19	FIFO control register 19	0000 $00EC_H$	24.4.7, 24.13.7, 24.21.9
FCR110	FIFO control register 110	0000 $00FC_H$	24.4.7, 24.13.7, 24.21.9
FCR111	FIFO control register 111	0000 $010C_H$	24.4.7, 24.13.7, 24.21.9
FCTLR	FLASH control register	0000 0320_H	33.2.1, 34.3.2
FSTR	FLASH status register	0000 0323_H	34.3.1
HADR	Host address register	0000 2110_H	28.3.7
HCNT0	Host control register 0	0000 2101_H	28.3.1
HCNT1	Host control register 1	0000 2100_H	28.3.1
HEOF0	EOF setting register 0	0000 2115_H	28.3.8
HEOF1	EOF setting register 1	0000 2114_H	28.3.8
HERR	Host error status register	0000 2104_H	28.3.3
HFCOMP	SOF interrupt FRAME comparison register	0000 2108_H	28.3.5
HFRAME0	FRAME setting register 0	0000 2119_H	28.3.9
HFRAME1	FRAME setting register 1	0000 2118_H	28.3.9
HIRQ	Host interrupt register	0000 2105_H	28.3.2
HRTIMER0	Retry timer setting register 0	0000 $210D_H$	28.3.6
HRTIMER1	Retry timer setting register 1	0000 $210C_H$	28.3.6
HRTIMER2	Retry timer setting register 2	0000 2111_H	28.3.6
HSTATE	Host status register	0000 2109_H	28.3.4

Abbreviated Name	Register Name	Address	Reference
HTOKEN	Host token end point register	0000 211D$_H$	28.3.10
IBCR1	I^{2}C bus control register 1	0000 0068$_H$	24.21.1
IBCR2	I^{2}C bus control register 2	0000 0074$_H$	24.21.1
IBCR3	I^{2}C bus control register 3	0000 0080$_H$	24.21.1
IBCR8	I^{2}C bus control register 8	0000 00D0$_H$	24.21.1
IBCR9	I^{2}C bus control register 9	0000 00E0$_H$	24.21.1
IBCR10	I^{2}C bus control register 10	0000 00F0$_H$	24.21.1
IBCR11	I^{2}C bus control register 11	0000 0100$_H$	24.21.1
IBSR1	I^{2}C bus status register 1	0000 006B$_H$	24.21.3
IBSR2	I^{2}C bus status register 2	0000 0077$_H$	24.21.3
IBSR3	I^{2}C bus status register 3	0000 0083$_H$	24.21.3
IBSR8	I^{2}C bus status register 8	0000 00D3$_H$	24.21.3
IBSR9	I^{2}C bus status register 9	0000 00E3$_H$	24.21.3
IBSR10	I^{2}C bus status register 10	0000 00F3$_H$	24.21.3
IBSR11	I^{2}C bus status register 11	0000 0103$_H$	24.21.3
ICR00	Interrupt control register 00	0000 0440$_H$	10.3.1
ICR01	Interrupt control register 01	0000 0441$_H$	10.3.1
ICR02	Interrupt control register 02	0000 0442$_H$	10.3.1
ICR03	Interrupt control register 03	0000 0443$_H$	10.3.1
ICR04	Interrupt control register 04	0000 0444$_H$	10.3.1
ICR05	Interrupt control register 05	0000 0445$_H$	10.3.1
ICR06	Interrupt control register 06	0000 0446$_H$	10.3.1
ICR07	Interrupt control register 07	0000 0447$_H$	10.3.1
ICR08	Interrupt control register 08	0000 0448$_H$	10.3.1
ICR09	Interrupt control register 09	0000 0449$_H$	10.3.1
ICR10	Interrupt control register 10	0000 044A$_H$	10.3.1
ICR11	Interrupt control register 11	0000 044B$_H$	10.3.1
ICR12	Interrupt control register 12	0000 044C$_H$	10.3.1
ICR13	Interrupt control register 13	0000 044D$_H$	10.3.1
ICR14	Interrupt control register 14	0000 044E$_H$	10.3.1
ICR15	Interrupt control register 15	0000 044F$_H$	10.3.1
ICR16	Interrupt control register 16	0000 0450$_H$	10.3.1
ICR17	Interrupt control register 17	0000 0451$_H$	10.3.1
ICR18	Interrupt control register 18	0000 0452$_H$	10.3.1
ICR19	Interrupt control register 19	0000 0453$_H$	10.3.1
ICR20	Interrupt control register 20	0000 0454$_H$	10.3.1
ICR21	Interrupt control register 21	0000 0455$_H$	10.3.1
ICR22	Interrupt control register 22	0000 0456$_H$	10.3.1
ICR23	Interrupt control register 23	0000 0457$_H$	10.3.1
ICR24	Interrupt control register 24	0000 0458$_H$	10.3.1
ICR25	Interrupt control register 25	0000 0459$_H$	10.3.1
ICR26	Interrupt control register 26	0000 045A$_H$	10.3.1
ICR27	Interrupt control register 27	0000 045B$_H$	10.3.1

Abbreviated Name	Register Name	Address	Reference
ICR28	Interrupt control register 28	0000 $045C_H$	10.3.1
ICR29	Interrupt control register 29	0000 $045D_H$	10.3.1
ICR30	Interrupt control register 30	0000 $045E_H$	10.3.1
ICR31	Interrupt control register 31	0000 $045F_H$	10.3.1
ICR32	Interrupt control register 32	0000 0460_H	10.3.1
ICR33	Interrupt control register 33	0000 0461_H	10.3.1
ICR34	Interrupt control register 34	0000 0462_H	10.3.1
ICR35	Interrupt control register 35	0000 0463_H	10.3.1
ICR36	Interrupt control register 36	0000 0464_H	10.3.1
ICR37	Interrupt control register 37	0000 0465_H	10.3.1
ICR38	Interrupt control register 38	0000 0466_H	10.3.1
ICR39	Interrupt control register 39	0000 0467_H	10.3.1
ICR40	Interrupt control register 40	0000 0468_H	10.3.1
ICR41	Interrupt control register 41	0000 0469_H	10.3.1
ICR42	Interrupt control register 42	0000 $046A_H$	10.3.1
ICR43	Interrupt control register 43	0000 $046B_H$	10.3.1
ICR44	Interrupt control register 44	0000 $046C_H$	10.3.1
ICR45	Interrupt control register 45	0000 $046D_H$	10.3.1
ICR46	Interrupt control register 46	0000 $046E_H$	10.3.1
ICR47	Interrupt control register 47	0000 $046F_H$	10.3.1
ICS01	Input capture status control register 01	0000 $021D_H$	18.4.1
ICS23	Input capture status control register 23	0000 $021F_H$	18.4.1
ICSEL0	Select Register 0 for DMA Transfer Request Clear by a Peripheral function	0000 $04F0_H$	32.3.2
ICSEL1	Select Register 1 for DMA Transfer Request Clear by a Peripheral function	0000 $04F1_H$	32.3.3
ICSEL4	Select Register 4 for DMA Transfer Request Clear by a Peripheral function	0000 $04F4_H$	32.3.4
ICSEL6	Select Register 6 for DMA Transfer Request Clear by a Peripheral function	0000 $04F6_H$	32.3.5
ICSEL7	Select Register 7 for DMA Transfer Request Clear by a Peripheral function	0000 $04F7_H$	32.3.6
ICSEL8	Select Register 8 for DMA Transfer Request Clear by a Peripheral function	0000 $04F8_H$	32.3.7
ICSEL10	Select Register 10 for DMA Transfer Request Clear by a Peripheral function	0000 $04FA_H$	32.3.8
ICSEL11	Select Register 11 for DMA Transfer Request Clear by a Peripheral function	0000 $04FB_H$	32.3.9
IORR0	IO-data request register 0	0000 0490_H	32.3.1
IORR1	IO-data request register 1	0000 0491_H	32.3.1
IORR2	IO-data request register 2	0000 0492_H	32.3.1
IORR3	IO-data request register 3	0000 0493_H	32.3.1
IORR4	IO-data request register 4	0000 0494_H	32.3.1
IORR5	IO-data request register 5	0000 0495_H	32.3.1
IORR6	IO-data request register 6	0000 0496_H	32.3.1
IORR7	IO-data request register 7	0000 0497_H	32.3.1
IPCP0	Input capture data register 0	0000 $020C_H$	18.4.2
IPCP1	Input capture data register 1	0000 0210_H	18.4.2

Abbreviated Name	Register Name	Address	Reference
IPCP2	Input capture data register 2	0000 0214_H	18.4.2
IPCP3	Input capture data register 3	0000 0218_H	18.4.2
IRPR0H	Interrupt request batch-read register 0 upper	0000 $01B0_H$	11.3.1
IRPR1H	Interrupt request batch-read register 1 upper	0000 $01B2_H$	11.3.2
IRPR3H	Interrupt request batch-read register 3 upper	0000 $01B6_H$	11.3.4
IRPR4H	Interrupt request batch-read register 4 upper	0000 $01B8_H$	11.3.5
IRPR5H	Interrupt request batch-read register 5 upper	0000 $01BA_H$	11.3.6
IRPR1L	Interrupt request batch-read register 1 lower	0000 $01B3_H$	11.3.2
IRPR2L	Interrupt request batch-read register 2 lower	0000 $01B5_H$	11.3.3
IRPR5L	Interrupt request batch-read register 5 lower	0000 $01BB_H$	11.3.7
IRPR7L	Interrupt request batch-read register 7 lower	0000 $01BF_H$	11.3.8
ISBA1	7 bit slave address register 1	0000 0071_H	24.21.7
ISBA2	7 bit slave address register 2	0000 $007D_H$	24.21.7
ISBA3	7 bit slave address register 3	0000 0089_H	24.21.7
ISBA8	7-bit slave address register 8	0000 $00D9_H$	24.21.7
ISBA9	7-bit slave address register 9	0000 $00E9_H$	24.21.7
ISBA10	7-bit slave address register 10	0000 $00F9_H$	24.21.7
ISBA11	7-bit slave address register 11	0000 0109_H	24.21.7
ISMK1	7-bit slave address mask register 1	0000 0070_H	24.21.6
ISMK2	7-bit slave address mask register 2	0000 $007C_H$	24.21.6
ISMK3	7-bit slave address mask register 3	0000 0088_H	24.21.6
ISMK8	7-bit slave address mask register 8	0000 $00D8_H$	24.21.6
ISMK9	7-bit slave address mask register 9	0000 $00E8_H$	24.21.6
ISMK10	7-bit slave address mask register 10	0000 $00F8_H$	24.21.6
ISMK11	7-bit slave address mask register 11	0000 0108_H	24.21.6
MTMCR	Main timer control register	0000 0512_H	6.3.1
MOSD_CBC0	Character background character code set (Main/OSDC Operation)	0000 4040_H	30.4.14
MOSD_CBC1	Character background character code set (Main/OSDC Operation)	0000 4044_H	30.4.14
MOSD_CBC2	Character background character code set (Main/OSDC Operation)	0000 4048_H	30.4.14
MOSD_CBC3	Character background character code set (Main/OSDC Operation)	0000 $404C_H$	30.4.14
MOSD_CBC4	Character background character code set (Main/OSDC Operation)	0000 4050_H	30.4.14
MOSD_CBC5	Character background character code set (Main/OSDC Operation)	0000 4054_H	30.4.14
MOSD_CBC6	Character background character code set (Main/OSDC Operation)	0000 4058_H	30.4.14
MOSD_CBC7	Character background character code set (Main/OSDC Operation)	0000 $405C_H$	30.4.14
MOSD_CDP1	Display period control 1 (Main/OSDC Operation)	0000 4064_H	30.4.16
MOSD_CDP2	Display period control 2 (Main/OSDC Operation)	0000 4068_H	30.4.16
MOSD_CDS1	Character data set 1 (Main/OSDC Operation)	0000 4004_H	30.4.3
MOSD_CDS2	Character data set 2 (Main/OSDC Operation)	0000 4008_H	30.4.3
MOSD_GRCC	Graphics color control (Main/OSDC Operation)	0000 4020_H	30.4.8

Abbreviated Name	Register Name	Address	Reference
MOSD_HVDP	Screen display position control (Main/OSDC Operation)	0000 4018$_H$	30.4.6
MOSD_INTC	Interrupt control (Main/OSDC Operation)	0000 406C$_H$	30.4.17
MOSD_IOTC	I/O pin control (Main/OSDC Operation)	0000 4060$_H$	30.4.15
MOSD_LDS1	Line control data set 1 (Main/OSDC Operation)	0000 400C$_H$	30.4.4
MOSD_LDS2	Line control data set 2 (Main/OSDC Operation)	0000 4010$_H$	30.4.4
MOSD_OSDC	OSDC operation control (Main/OSDC Operation)	0000 4400$_H$	30.4.20
MOSD_PL0 to MOSD_PL127	Palette control (Main/OSDC Operation)	0000 4200$_H$ to 0000 43FC$_H$	30.4.19
MOSD_PLLC	PLL clock control (Main/OSDC Operation)	0000 4404$_H$	30.4.21
MOSD_SBC0	Shaded background frame color control (Main/OSDC Operation)	0000 4070$_H$	30.4.18
MOSD_SBC1	Shaded background frame color control (Main/OSDC Operation)	0000 4074$_H$	30.4.18
MOSD_SBC2	Shaded background frame color control (Main/OSDC Operation)	0000 4078$_H$	30.4.18
MOSD_SBC3	Shaded background frame color control (Main/OSDC Operation)	0000 407C$_H$	30.4.18
MOSD_SBCC	Screen background character control (Main/OSDC Operation)	0000 4024$_H$	30.4.9
MOSD_SCBC	Screen background control (Main/OSDC Operation)	0000 4028$_H$	30.4.10
MOSD_SCOC	Screen output control (Main/OSDC Operation)	0000 4014$_H$	30.4.5
MOSD_SPC1	Sprite character control 1 (Main/OSDC Operation)	0000 4034$_H$	30.4.12
MOSD_SPC2	Sprite character control 2 (Main/OSDC Operation)	0000 4038$_H$	30.4.12
MOSD_SYNC	Synchronization control (Main/OSDC Operation)	0000 403C$_H$	30.4.13
MOSD_TSBC	Transparent color control (Main/OSDC Operation)	0000 401C$_H$	30.4.7
MOSD_VADR	VRAM write address set (Main/OSDC Operation)	0000 4000$_H$	30.4.2
MOSD_WPC1	Window period control 1 (Main/OSDC Operation)	0000 402C$_H$	30.4.11
MOSD_WPC2	Window period control 2 (Main/OSDC Operation)	0000 4030$_H$	30.4.11
OCCP0	Output compare register 0	0000 0234$_H$	19.4.1
OCCP1	Output compare register 1	0000 0238$_H$	19.4.1
OCCP2	Output compare register 2	0000 023C$_H$	19.4.1
OCCP3	Output compare register 3	0000 0240$_H$	19.4.1
OCSH1	Compare control register upper 1	0000 0244$_H$	19.4.2
OCSH3	Compare control register upper 3	0000 0246$_H$	19.4.2
OCSL0	Compare control register lower 0	0000 0245$_H$	19.4.3
OCSL2	Compare control register lower 2	0000 0247$_H$	19.4.3
PCCR0	Priority conversion control register 0	0000 012C$_H$	23.4.7
PCFD0	Priority conversion FIFO data register 0	0000 012E$_H$	23.4.9
PCIS0	Priority conversion input select register 0	0000 0130$_H$	23.4.10
PCR0	Pull-up resistor control register 0	0000 0420$_H$	13.4.5
PCR1	Pull-up resistor control register 1	0000 0421$_H$	13.4.5

Abbreviated Name	Register Name	Address	Reference
PCR5	Pull-up resistor control register 5	0000 0425_H	13.4.5
PCR7	Pull-up resistor control register 7	0000 0427_H	13.4.5
PDR0	Port data register 0	0000 0000_H	13.4.4
PDR1	Port data register 1	0000 0001_H	13.4.4
PDR2	Port data register 2	0000 0002_H	13.4.4
PDR3	Port data register 3	0000 0003_H	13.4.4
PDR5	Port data register 5	0000 0005_H	13.4.4
PDR7	Port data register 7	0000 0007_H	13.4.4
PDRK	Port data register K	0000 0014_H	13.4.4
PFNS0	Priority conversion FIFO number setting register 0	0000 $012D_H$	23.4.8
PFR0	Port function register 0	0000 $04A0_H$	13.4.2
PFR1	Port function register 1	0000 $04A1_H$	13.4.2
PFR2	Port function register 2	0000 $04A2_H$	13.4.2
PFR3	Port function register 3	0000 $04A3_H$	13.4.2
PFR5	Port function register 5	0000 $04A5_H$	13.4.2
PFR7	Port function register 7	0000 $04A7_H$	13.4.2
PLLCR	PLL configuration register	0000 0514_H	4.4.4
RCADR1	Device Address Configuration Register 1	0000 0026_H	29.2.3
RCADR2	Device Address Configuration Register 2	0000 0027_H	29.2.3
RCCKD	Clock Divider Register	0000 $002C_H$	29.2.8
RCCR	Remote Control Reception Control Register	0000 0020_H	29.2.1
RCDAHW	High Width Configuration Register A	0000 0023_H	29.2.5
RCDBHW	High Width Configuration Register B	0000 0024_H	29.2.6
RCDTHH	Data Store Register HH	0000 0028_H	29.2.7
RCDTHL	Data Store Register HL	0000 0029_H	29.2.7
RCDTLH	Data Store Register LH	0000 $002A_H$	29.2.7
RCDTLL	Data Store Register LL	0000 $002B_H$	29.2.7
RCSHW	Start Bit High Width Configuration Register	0000 0022_H	29.2.4
RCST	Remote Control Reception Interrupt Control Register	0000 0021_H	29.2.2
RDR0	Received data register 0	0000 0064_H	24.4.5, 24.13.5
RDR1	Received data register 1	0000 $006C_H$	24.4.5, 24.13.5, 24.21.5
RDR2	Received data register 2	0000 0078_H	24.4.5, 24.13.5, 24.21.5
RDR3	Received data register 3	0000 0084_H	24.4.5, 24.13.5, 24.21.5
RDR8	Received data register 8	0000 $00D4_H$	24.4.5, 24.13.5, 24.21.5
RDR9	Received data register 9	0000 $00E4_H$	24.4.5, 24.13.5, 24.21.5
RDR10	Received data register 10	0000 $00F4_H$	24.4.5, 24.13.5, 24.21.5
RDR11	Received data register 11	0000 0104_H	24.4.5, 24.13.5, 24.21.5
RDRM0	Received data mirror register 0	0000 $00C0_H$	24.13.11
RDRM1	Received data mirror register 1	0000 $00C1_H$	24.13.11
RDRM2	Received data mirror register 2	0000 $00C2_H$	24.13.11
RDRM3	Received data mirror register 3	0000 $00C3_H$	24.13.11
RSTCR	Reset control register	0000 0481_H	9.4.2
RSTRR	Reset result register	0000 0480_H	9.4.1

Abbreviated Name	Register Name	Address	Reference
SCCR0	Scan conversion control register 0	0000 0124_H	23.4.3
SCFD0	Scan conversion FIFO data register 0	0000 0126_H	23.4.5
SCIS00	Scan conversion input select register 00	0000 $012B_H$	23.4.6
SCR0	Serial control register 0	0000 0060_H	24.4.1, 24.13.1
SCR1	Serial control register 1	0000 0068_H	24.4.1, 24.13.1
SCR2	Serial control register 2	0000 0074_H	24.4.1, 24.13.1
SCR3	Serial control register 3	0000 0080_H	24.4.1, 24.13.1
SCR8	Serial control register 8	0000 $00D0_H$	24.4.1, 24.13.1
SCR9	Serial control register 9	0000 $00E0_H$	24.4.1, 24.13.1
SCR10	Serial control register 10	0000 $00F0_H$	24.4.1, 24.13.1
SCR11	Serial control register 11	0000 0100_H	24.4.1, 24.13.1
SFNS0	Scan conversion FIFO number setting register 0	0000 0125_H	23.4.4
SLPRR	Sleep rate configuration register	0000 0483_H	8.3.2
SMR0	Serial mode register 0	0000 0061_H	24.4.2, 24.13.2
SMR1	Serial mode register 1	0000 0069_H	24.4.2, 24.13.2, 24.21.2
SMR2	Serial mode register 2	0000 0075_H	24.4.2, 24.13.2, 24.21.2
SMR3	Serial mode register 3	0000 0081_H	24.4.2, 24.13.2, 24.21.2
SMR8	Serial mode register 8	0000 $00D1_H$	24.4.2, 24.13.2, 24.21.2
SMR9	Serial mode register 9	0000 $00E1_H$	24.4.2, 24.13.2, 24.21.2
SMR10	Serial mode register 10	0000 $00F1_H$	24.4.2, 24.13.2, 24.21.2
SMR11	Serial mode register 11	0000 0101_H	24.4.2, 24.13.2, 24.21.2
SOSD_CDS1	Character data set 1 (Sub Operation)	0000 4104_H	30.5.3
SOSD_CDS2	Character data set 2 (Sub Operation)	0000 4108_H	30.5.3
SOSD_GRCC	Graphics color control (Sub Operation)	0000 4120_H	30.5.8
SOSD_HVDP	Screen display position control (Sub Operation)	0000 4118_H	30.5.6
SOSD_INTC	Interrupt control (Sub Operation)	0000 $416C_H$	30.5.12
SOSD_LDS1	Line control data set 1 (Sub Operation)	0000 $410C_H$	30.5.4
SOSD_LDS2	Line control data set 2 (Sub Operation)	0000 4110_H	30.5.4
SOSD_SBC0	Shaded background frame color control (Sub Operation)	0000 4170_H	30.5.13
SOSD_SBC1	Shaded background frame color control (Sub Operation)	0000 4174_H	30.5.13
SOSD_SBC2	Shaded background frame color c30.5.4 ontrol (Sub Operation)	0000 4178_H	30.5.13
SOSD_SBC3	Shaded background frame color control (Sub Operation)	0000 $417C_H$	30.5.13
SOSD_SCBC	Screen background control (Sub Operation)	0000 4128_H	30.5.9
SOSD_SCOC	Screen output control (Sub Operation)	0000 4114_H	30.5.5
SOSD_SPC1	Sprite character control 1 (Sub Operation)	0000 4134_H	30.5.11
SOSD_SPC2	Sprite character control 2 (Sub Operation)	0000 4138_H	30.5.11
SOSD_TSBC	Transparent color control (Sub Operation)	0000 $411C_H$	30.5.7
SOSD_VADR	VRAM write address set (Sub Operation)	0000 4100_H	30.5.2
SOSD_WPC1	Window period control 1 (Sub Operation)	0000 $412C_H$	30.5.10
SOSD_WPC2	Window period control 2 (Sub Operation)	0000 4130_H	30.5.10
SSEL0123	Serial mode select register 0123	0000 $00C8_H$	24.13.10

Abbreviated Name	Register Name	Address	Reference
SSR0	Serial status register 0	0000 0062_H	24.4.3, 24.13.3
SSR1	Serial status register 1	0000 $006A_H$	24.4.3, 24.13.3, 24.21.4
SSR2	Serial status register 2	0000 0076_H	24.4.3, 24.13.3, 24.21.4
SSR3	Serial status register 3	0000 0082_H	24.4.3, 24.13.3, 24.21.4
SSR8	Serial status register 8	0000 $00D2_H$	24.4.3, 24.13.3, 24.21.4
SSR9	Serial status register 9	0000 $00E2_H$	24.4.3, 24.13.3, 24.21.4
SSR10	Serial status register 10	0000 $00F2_H$	24.4.3, 24.13.3, 24.21.4
SSR11	Serial status register 11	0000 0102_H	24.4.3, 24.13.3, 24.21.4
STBCR	Standby mode control register	0000 0482_H	8.3.1
STMCR	Sub timer control register	0000 0513_H	7.3.1
TCCSH0	Timer status control register upper 0	0000 0208_H	17.4.3
TCCSL0	Timer status control register lower 0	0000 0209_H	17.4.3
TCDT0	Timer data register 0	0000 0204_H	17.4.2
TDR0	Transmitted data register 0	0000 0064_H	24.4.5, 24.13.5
TDR1	Transmitted data register 1	0000 $006C_H$	24.4.5, 24.13.5, 24.21.5
TDR2	Transmitted data register 2	0000 0078_H	24.4.5, 24.13.5, 24.21.5
TDR3	Transmitted data register 3	0000 0084_H	24.4.5, 24.13.5, 24.21.5
TDR8	Transmitted data register 8	0000 $00D4_H$	24.4.5, 24.13.5, 24.21.5
TDR9	Transmitted data register 9	0000 $00E4_H$	24.4.5, 24.13.5, 24.21.5
TDR10	Transmitted data register 10	0000 $00F4_H$	24.4.5, 24.13.5, 24.21.5
TDR11	Transmitted data register 11	0000 0104_H	24.4.5, 24.13.5, 24.21.5
TDRM0	Transmitted data mirror register 0	0000 $00C0_H$	24.13.11
TDRM1	Transmitted data mirror register 1	0000 $00C1_H$	24.13.11
TDRM2	Transmitted data mirror register 2	0000 $00C2_H$	24.13.11
TDRM3	Transmitted data mirror register 3	0000 $00C3_H$	24.13.11
TMCSR0	Timer control status register 0	0000 $004E_H$	20.4.1
TMCSR1	Timer control status register 1	0000 0056_H	20.4.1
TMCSR2	Timer control status register 2	0000 $005E_H$	20.4.1
TMR0	16-bit timer register 0	0000 $004A_H$	20.4.3
TMR1	16-bit timer register 1	0000 0052_H	20.4.3
TMR2	16-bit timer register 2	0000 $005A_H$	20.4.3
TMRLRA0	16-bit timer reload register A0	0000 0048_H	20.4.2
TMRLRA1	16-bit timer reload register A1	0000 0050_H	20.4.2
TMRLRA2	16-bit timer reload register A2	0000 0058_H	20.4.2
TMSP	Time stamp register	0000 $213C_H$	27.3.6
UCCR	USB clock setting register	0000 $051C_H$	25.3.1
UDCC	UDC control register	0000 2121_H	27.3.3
UDCIE	UDC interrupt request register	0000 2140_H	27.3.8
UDCS	UDC status register	0000 2141_H	27.3.7
USBEN	USB request register	0000 $21A6_H$	27.3.2
USBSEL	USB select register	0000 $21A5_H$	27.3.1
WCCR	Watch counter control register	0000 $051A_H$	16.3.2
WCRD	Watch counter read register	0000 0518_H	16.3.3

Abbreviated Name	Register Name	Address	Reference
WCRL	Watch counter reload register	0000 0519$_H$	16.3.1
WDTCPR0	Watchdog timer clear pattern register 0	0000 003D$_H$	15.3.2
WDTCPR1	Watchdog timer clear pattern register 1	0000 003F$_H$	15.3.4
WDTCR0	Watchdog timer control register 0	0000 003C$_H$	15.3.1
WDTCR1	Watchdog timer control register 1	0000 003E$_H$	15.3.3
WRAR00	Wild register address register 00	0000 0380$_H$	35.3.1
WRAR01	Wild register address register 01	0000 0388$_H$	35.3.1
WRAR02	Wild register address register 02	0000 0390$_H$	35.3.1
WRAR03	Wild register address register 03	0000 0398$_H$	35.3.1
WRAR04	Wild register address register 04	0000 03A0$_H$	35.3.1
WRAR05	Wild register address register 05	0000 03A8$_H$	35.3.1
WRAR06	Wild register address register 06	0000 03B0$_H$	35.3.1
WRAR07	Wild register address register 07	0000 03B8$_H$	35.3.1
WRAR08	Wild register address register 08	0000 03C0$_H$	35.3.1
WRAR09	Wild register address register 09	0000 03C8$_H$	35.3.1
WRAR10	Wild register address register 10	0000 03D0$_H$	35.3.1
WRAR11	Wild register address register 11	0000 03D8$_H$	35.3.1
WRAR12	Wild register address register 12	0000 03E0$_H$	35.3.1
WRAR13	Wild register address register 13	0000 03E8$_H$	35.3.1
WRAR14	Wild register address register 14	0000 03F0$_H$	35.3.1
WRAR15	Wild register address register 15	0000 03F8$_H$	35.3.1
WRDR00	Wild register data register 00	0000 0384$_H$	35.3.2
WRDR01	Wild register data register 01	0000 038C$_H$	35.3.2
WRDR02	Wild register data register 02	0000 0394$_H$	35.3.2
WRDR03	Wild register data register 03	0000 039C$_H$	35.3.2
WRDR04	Wild register data register 04	0000 03A4$_H$	35.3.2
WRDR05	Wild register data register 05	0000 03AC$_H$	35.3.2
WRDR06	Wild register data register 06	0000 03B4$_H$	35.3.2
WRDR07	Wild register data register 07	0000 03BC$_H$	35.3.2
WRDR08	Wild register data register 08	0000 03C4$_H$	35.3.2
WRDR09	Wild register data register 09	0000 03CC$_H$	35.3.2
WRDR10	Wild register data register 10	0000 03D4$_H$	35.3.2
WRDR11	Wild register data register 11	0000 03DC$_H$	35.3.2
WRDR12	Wild register data register 12	0000 03E4$_H$	35.3.2
WRDR13	Wild register data register 13	0000 03EC$_H$	35.3.2
WRDR14	Wild register data register 14	0000 03F4$_H$	35.3.2
WRDR15	Wild register data register 15	0000 03FC$_H$	35.3.2
WREN	Wild register enable register	0000 033A$_H$	35.3.3

A.3 Interrupt Vectors

This appendix explains the interrupt vector table for this series. This table specifies how interrupt sources are assigned to interrupt vectors and interrupt control registers (ICR00 to ICR47).

Interrupt source (Peripheral resource)	Interrupt number		Interrupt level setting register	Offset	Address of TBR default
	Decimal	Hexa-decimal			
Reset	0	00	-	$3FC_H$	000F FFFC$_H$
System reserved	1	01	-	$3F8_H$	000F FFF8$_H$
System reserved	2	02	-	$3F4_H$	000F FFF4$_H$
System reserved	3	03	-	$3F0_H$	000F FFF0$_H$
System reserved	4	04	-	$3EC_H$	000F FFEC$_H$
System reserved	5	05	-	$3E8_H$	000F FFE8$_H$
System reserved	6	06	-	$3E4_H$	000F FFE4$_H$
System reserved	7	07	-	$3E0_H$	000F FFE0$_H$
System reserved	8	08	-	$3DC_H$	000F FFDC$_H$
INTE instruction	9	09	-	$3D8_H$	000F FFD8$_H$
Instruction break exception	10	0A	-	$3D4_H$	000F FFD4$_H$
Operand break	11	0B	-	$3D0_H$	000F FFD0$_H$
Step trace trap	12	0C	-	$3CC_H$	000F FFCC$_H$
System reserved	13	0D	-	$3C8_H$	000F FFC8$_H$
Undefined instruction exception	14	0E	-	$3C4_H$	000F FFC4$_H$
-	15	0F	15 (F_H) fixed	$3C0_H$	000F FFC0$_H$
External interrupt request ch.0 to ch.7	16	10	ICR00	$3BC_H$	000F FFBC$_H$
External interrupt request ch.8 to ch.15	17	11	ICR01	$3B8_H$	000F FFB8$_H$
Reserved	18	12	ICR02	$3B4_H$	000F FFB4$_H$
Reserved	19	13	ICR03	$3B0_H$	000F FFB0$_H$
16-bit reload timer ch.0 to ch.2	20	14	ICR04	$3AC_H$	000F FFAC$_H$
Reception interrupt request of UART/CSIO ch.0	21	15	ICR05	$3A8_H$	000F FFA8$_H$
Transmission interrupt request of UART/CSIO ch.0 Transmission bus idle interrupt request of UART/CSIO ch.0	22	16	ICR06	$3A4_H$	000F FFA4$_H$
Reception interrupt request of UART/CSIO/ I^2C ch.1	23	17	ICR07	$3A0_H$	000F FFA0$_H$
Transmission interrupt request of UART/CSIO/ I^2C ch.1 Transmission bus idle interrupt request of UART/CSIO ch.1	24	18	ICR08	$39C_H$	000F FF9C$_H$
Status interrupt request of I^2C ch.1	25	19	ICR09	398_H	000F FF98$_H$
Reception interrupt request of UART/CSIO/I^2C ch.2	26	1A	ICR10	394_H	000F FF94$_H$
Transmission interrupt request of UART/CSIO/I^2C ch.2 Transmission bus idle interrupt request of UART/CSIO ch.2	27	1B	ICR11	390_H	000F FF90$_H$
Status interrupt request of I^2C ch.2	28	1C	ICR12	$38C_H$	000F FF8C$_H$
Reception interrupt request of UART/CSIO/I^2C ch.3	29	1D	ICR13	388_H	000F FF88$_H$
Transmission interrupt request of UART/CSIO/I^2C ch.3 Transmission bus idle interrupt request of UART/CSIO ch.3 Status interrupt request of I^2C ch.3	30	1E	ICR14	384_H	000F FF84$_H$
Reserved	31	1F	ICR15	380_H	000F FF80$_H$

Interrupt source (Peripheral resource)	Interrupt number		Interrupt level setting register	Offset	Address of TBR default
	Decimal	Hexa-decimal			
Reserved	32	20	ICR16	$37C_H$	000F FF7C$_H$
Reserved	33	21	ICR17	378_H	000F FF78$_H$
Reserved	34	22	ICR18	374_H	000F FF74$_H$
Reserved	35	23	ICR19	370_H	000F FF70$_H$
Reserved	36	24	ICR20	$36C_H$	000F FF6C$_H$
Reserved	37	25	ICR21	368_H	000F FF68$_H$
Reserved	38	26	ICR22	364_H	000F FF64$_H$
Reception interrupt request of UART/CSIO/I^{2}C ch.8 to ch.11 Transmission interrupt request of UART/CSIO/I^{2}C ch.8 to ch.11 Transmission bus idle interrupt request of UART/CSIO ch.8 to ch.11 Transmission FIFO interrupt request UART/CSIO/I^{2}C ch.8 to ch.11 Status interrupt request of I^{2}C ch.8 to ch.11	39	27	ICR23	360_H	000F FF60$_H$
HDMI-CEC/Remote control reception	40	28	ICR24	$35C_H$	000F FF5C$_H$
Main timer/Sub timer/Watch counter	41	29	ICR25	358_H	000F FF58$_H$
10-bit A/D converter ■ Scan conversion interrupt request ■ Priority conversion interrupt request ■ FIFO overrun interrupt request ■ Conversion result compare interrupt request	42	2A	ICR26	354_H	000F FF54$_H$
32-bit free run timer ch.0	43	2B	ICR27	350_H	000F FF50$_H$
32-bit input capture ch.0 to ch.3	44	2C	ICR28	$34C_H$	000F FF4C$_H$
32-bit output compare ch.0 to ch.3	45	2D	ICR29	348_H	000F FF48$_H$
Base timer ch.0	46	2E	ICR30	344_H	000F FF44$_H$
Base timer ch.1	47	2F	ICR31	340_H	000F FF40$_H$
Base timer ch.2	48	30	ICR32	$33C_H$	000F FF3C$_H$
Base timer ch.3	49	31	ICR33	338_H	000F FF38$_H$
Base timer ch.4, ch.5	50	32	ICR34	334_H	000F FF34$_H$
Base timer ch.6, ch.7	51	33	ICR35	330_H	000F FF30$_H$
Reserved	52	34	ICR36	$32C_H$	000F FF2C$_H$
OSDC (MAIN)	53	35	ICR37	328_H	000F FF28$_H$
USB function (DRQ of End Point 1 to 5)	54	36	ICR38	324_H	000F FF24$_H$
USB function (DRQI of End Point 0, DRQO and each status/ USB HOST (each status)	55	37	ICR39	320_H	000F FF20$_H$
OSDC (SUB)	56	38	ICR40	$31C_H$	000F FF1C$_H$
DMA controller (DMAC) ch.0	57	39	ICR41	318_H	000F FF18$_H$
DMA controller (DMAC) ch.1	58	3A	ICR42	314_H	000F FF14$_H$
DMA controller (DMAC) ch.2	59	3B	ICR43	310_H	000F FF10$_H$
DMA controller (DMAC) ch.3	60	3C	ICR44	$30C_H$	000F FF0C$_H$
DMA controller (DMAC) ch.4 to ch.7	61	3D	ICR45	308_H	000F FF08$_H$
System reserved	62	3E	ICR46	304_H	000F FF04$_H$
Delay interrupt	63	3F	ICR47	300_H	000F FF00$_H$
System reserved (Used by REALOS)	64	40	-	$2FC_H$	000F FEFC$_H$

Interrupt source (Peripheral resource)	Interrupt number		Interrupt level setting register	Offset	Address of TBR default
	Decimal	Hexa-decimal			
System reserved (Used by REALOS)	65	41	-	$2F8_H$	$000F\ FEF8_H$
Used by INT instruction	66 to 255	42 to FF	-	$2F4_H$ to 000_H	$000F\ FEF4_H$ to $000F\ FC00_H$

*: USB interrupt source

Number		USB interrupt source	Details
Decimal	Hexadecimal		
54	36	USB function (DRQ of End Point 1 to 5)	DRQ (End Point1 to 5)
55	37	USB function (DRQI, DRQO of End Point 0 and each status)	DRQI, DRQO, SPK, SUSP, SOF, BRST, CONF, WKUP
		USB HOST (Each status)	DIRQ, URIRQ, RWKIRQ, CNNIRQ, SOFIRQ, CMPIRQ

A.4 Pin State in Each CPU State

This appendix lists the pin state for each CPU state.

Pin state

The terms used in the table have the meanings as follows.

- $\overline{\text{INIT}}$ = "L" Period

  This is the period in which the $\overline{\text{INIT}}$ pin is at the "L" level.
- $\overline{\text{INIT}}$ = "H" Period

  The $\overline{\text{INIT}}$ pin is in the state immediately following a transition from the "L" to "H" level.
- SLVL1

  This bit is a standby level setting bit in the standby mode control register (STBCR).
- Input enabled

  This indicates that the input function can be used.
- Input disabled

  This indicates that the input function cannot be used.
- Output Hi-Z

  The pin is placed in Hi-Z by preventing the transistor from driving the pin.
- Last state maintained

  The output state immediately before this mode is entered is maintained.

  If any built-in peripheral function is active, output is performed according to that peripheral function.

  Any output that is performed for operation such as for a port is maintained.
- Internal input "0" fixed

  External input is cut off at the input gate immediately next to the pin, and "0" is sent to the CPU.
- Input enabled when the selection of interrupt function is enabled

  The pin functions are set for an external interrupt request input pin to enable input only when external interrupt requests are enabled.

List of pin status

Pin name	Function	During initialization		Sleep Mode	Standby Mode	
		$\overline{INIT}$ = "L"	$\overline{INIT}$ = "H"		SLVL1 = 0	SLVL1 = 1
INIT	INIT	-	-	Input enabled	Input enabled	Input enabled
X0	X0	Input enabled	Input enabled		Hi-Z/ Input enabled	Hi-Z/ Input enabled
X1	X1	Input enabled	Input enabled		"H" output/ Input enabled	"H" output/ Input enabled
X0A	X0A (When $\overline{INIT}$ input, see PK1. When port selected, input disabled)	Input disabled	Input disabled		Hi-Z/ Input enabled	Hi-Z/ Input enabled
X1A	X1A (When $\overline{INIT}$ input, see PK0. When port selected, input disabled)	Input disabled	Input disabled		"H" output/ Input enabled	"H" output/ Input enabled
MD0	MD0	Input enabled	Input enabled		Input enabled	Input enabled
MD1	MD1	Input enabled	Input enabled			
P00	P00/TIOA0/SOUT0_1/IN0	Output Hi-Z	Output Hi-Z Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0"
P01	P01/TIOB0/SIN0_1/IN1					
P02	P02/TIOA1/SCK0_1/IN2					
P03	P03/TIOB1/IN3					
P04	P04/TIOA2/SOUT1					
P05	P05/TIOB2/SIN1					
P06	P06/TIOA3/SCK1					
P07	P07/TIOB3					
P10	P10/TIOA4/SOUT2/INT0	Output Hi-Z	Output Hi-Z Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0" Input enabled when interrupt function selected and enabled
P11	P11/TIOB4/SIN2/INT1					
P12	P12/TIOA5/SCK2/INT2					
P13	P13/TIOB5/INT3					
P14	P14/TIOA6/SOUT3/INT4					
P15	P15/TIOB6/SIN3/INT5					
P16	P16/TIOA7/SCK3/INT6					
P17	P17/TIOB7/INT7					
P20	P20/SOUT8	Output Hi-Z	Output Hi-Z Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0"
P21	P21/SIN8					
P22	P22/SCK8					
P23	P23/RCIN_1					
P24	P24/SOUT9/OUT0					
P25	P25/SIN9/OUT1					
P26	P26/SCK9/OUT2					
P27	P27/OUT3					

Pin name	Function	During initialization		Sleep Mode	Standby Mode	
		$\overline{INIT}$ = "L"	$\overline{INIT}$ = "H"		SLVL1 = 0	SLVL1 = 1
P30	P30/SOUT10/INT8	Output Hi-Z	Output Hi-Z Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0" Input enabled when interrupt function selected and enabled
P31	P31/SIN10/INT9					
P32	P32/SCK10/INT10					
P33	P33/INT11					
P34	P34/SOUT11/INT12					
P35	P35/SIN11/INT13					
P36	P36/SCK11/INT14					
P37	P37/INT15					
P50	P50	Output Hi-Z	Output Hi-Z Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0"
P51	P51					
P52	P52					
P53	P53					
P54	P54/RCIN					
P55	P55/ADTRG					
P56	P56/FRCK					
P57	P57					
P70	P70/AN0/OUT0_1	Output Hi-Z	Output Hi-Z Input enabled*	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0"
P71	P71/AN1/OUT1_1					
P72	P72/AN2/TMO0/OUT2_1					
P73	P73/AN3/TMO1/OUT3_1					
P74	P74/AN4/TMO2					
P75	P75/AN5/SOUT0/TMI0					
P76	P76/AN6/SIN0/TMI1					
P77	P77/AN7/SCK0/TMI2					
*: Analog input has a priority (digital input is disconnected)						
PK0	PK0	Output Hi-Z	Output Hi-Z Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/Internal input fixed at "0"
PK1	PK1					
UDP	UDP(USB)	Output Hi-Z	Output Hi-Z Input enabled	Maintain previous state/ Input enabled	Maintain previous state	Output Hi-Z/ Internal input fixed at "0"
UDM	UDM(USB)					
DCKI	DCKI	Input state	Input enabled	Input enabled	Input state	Input state
DCKO	DCKO	L output	L output/ DCK output	L output/ DCK output	L output (OSDC stop)	L output (OSDC stop)
VSYNC	VSYNC	Input state	Input enabled	Input enabled	Input state	Input state
HSYNC	HSYNC					

Pin name	Function	During initialization		Sleep Mode	Standby Mode	
		$\overline{INIT}$ = "L"	$\overline{INIT}$ = "H"		SLVL1 = 0	SLVL1 = 1
R4 to R0	R4 to R0	L output	L output/ R output	L output/ R output	L output (OSDC stop)	L output (OSDC stop)
G5 to G0	G5 to G0		L output/ G output	L output/ G output		
B4 to B0	B4 to B0		L output/ B output	L output/ B output		
VOA2 to VOA0	VOA2 to VOA0		L output/ VOA output	L output/ VOA output		
VOB	VOB		L output/ VOB output	L output/ VOB output		
ROUT	ROUT		L output/ROUT output	L output/ ROUT output		
GOUT	GOUT		L output/GOUT output	L output/ GOUT output		
BOUT	BOUT		L output/BOUT output	L output/ BOUT output		
HWDE	HWDE	Input state	Input enabled	Input enabled	Input state	Input state

List of pin status (serial write mode)

Pin name	Function	During Initialization	During Asynchronous Write Operation	During Synchronous Write Operation
		$\overline{\text{INIT}}$ = “L”	$\overline{\text{INIT}}$ = “H”	
INIT	INIT	-	-	-
X0	X0	Input enabled	Input enabled	Input enabled
X1	X1	Input enabled	Input enabled	Input enabled
X0A	X0A (When $\overline{\text{INIT}}$ input, see PK1. When port selected, input disabled)	Input disabled	Input disabled	Input disabled
X1A	X1A (When $\overline{\text{INIT}}$ input, see PK0. When port selected, input disabled)	Input disabled	Input disabled	Input disabled
MD0	MD0	Input enabled	Input enabled	Input enabled
MD1	MD1	Input enabled	Input enabled	Input enabled
P00	P00/TIOA0/SOUT0_1/IN0	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
P01	P01/TIOB0/SIN0_1/IN1			
P02	P02/TIOA1/SCK0_1/IN2			
P03	P03/TIOB1/IN3			
P04	P04/TIOA2/SOUT1			
P05	P05/TIOB2/SIN1			
P06	P06/TIOA3/SCK1			
P07	P07/TIOB3			
P10	P10/TIOA4/SOUT2/INT0	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
P11	P11/TIOB4/SIN2/INT1			
P12	P12/TIOA5/SCK2/INT2			
P13	P13/TIOB5/INT3			
P14	P14/TIOA6/SOUT3/INT4			
P15	P15/TIOB6/SIN3/INT5			
P16	P16/TIOA7/SCK3/INT6			
P17	P17/TIOB7/INT7			
P20	P20/SOUT8	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
P21	P21/SIN8			
P22	P22/SCK8			
P23	P23/RCIN_1			
P24	P24/SOUT9/OUT0			
P25	P25/SIN9/OUT1			
P26	P26/SCK9/OUT2			
P27	P27/OUT3			

Pin name	Function	During Initialization	During Asynchronous Write Operation	During Synchronous Write Operation
		$\overline{INIT}$ = "L"	$\overline{INIT}$ = "H"	
P30	P30/SOUT10/INT8	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
P31	P31/SIN10/INT9			
P32	P32/SCK10/INT10			
P33	P33/INT11			
P34	P34/SOUT11/INT12			
P35	P35/SIN11/INT13			
P36	P36/SCK11/INT14			
P37	P37/INT15			
P50	P50	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
P51	P51			
P52	P52			
P53	P53			
P54	P54/RCIN			
P55	P55/ADTRG			
P56	P56/FRCK			
P57	P57			
P70	P70/AN0/OUT0_1	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
P71	P71/AN1/OUT1_1			
P72	P72/AN2/TMO0/OUT2_1			
P73	P73/AN3/TMO1/OUT3_1			
P74	P74/AN4/TMO2			
P75	P75/AN5/SOUT0/TMI0		Output	Output
P76	P76/AN6/SIN0/TMI1		Output Hi-Z Input enabled	Output Hi-Z Input enabled
P77	P77/AN7/SCK0/TMI2			
PK0	PK0	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
PK1	PK1			
UDP	UDP (USB)	Output Hi-Z	Output Hi-Z Input enabled	Output Hi-Z Input enabled
UDM	UDM (USB)			
DCKI	DCKI	Input state	Input enabled	Input enabled
DCKO	DCKO	L output	L output	L output
VSYNC	VSYNC	Input state	Input enabled	Input enabled
HSYNC	HSYNC			
R4 to R0	R4 to R0	L output	L output	L output
G5 to G0	G5 to G0			
B4 to B0	B4 to B0			
VOA2 to VOA0	VOA2 to VOA0			
VOB	VOB			

Pin name	Function	During Initialization	During Asynchronous Write Operation	During Synchronous Write Operation
		$\overline{\text{INIT}}$ = “L”	$\overline{\text{INIT}}$ = “H”	
ROUT	ROUT	L output	L output	L output
GOUT	GOUT			
BOUT	BOUT			
HWDE	HWDE	Input state	Input enabled	Input enabled

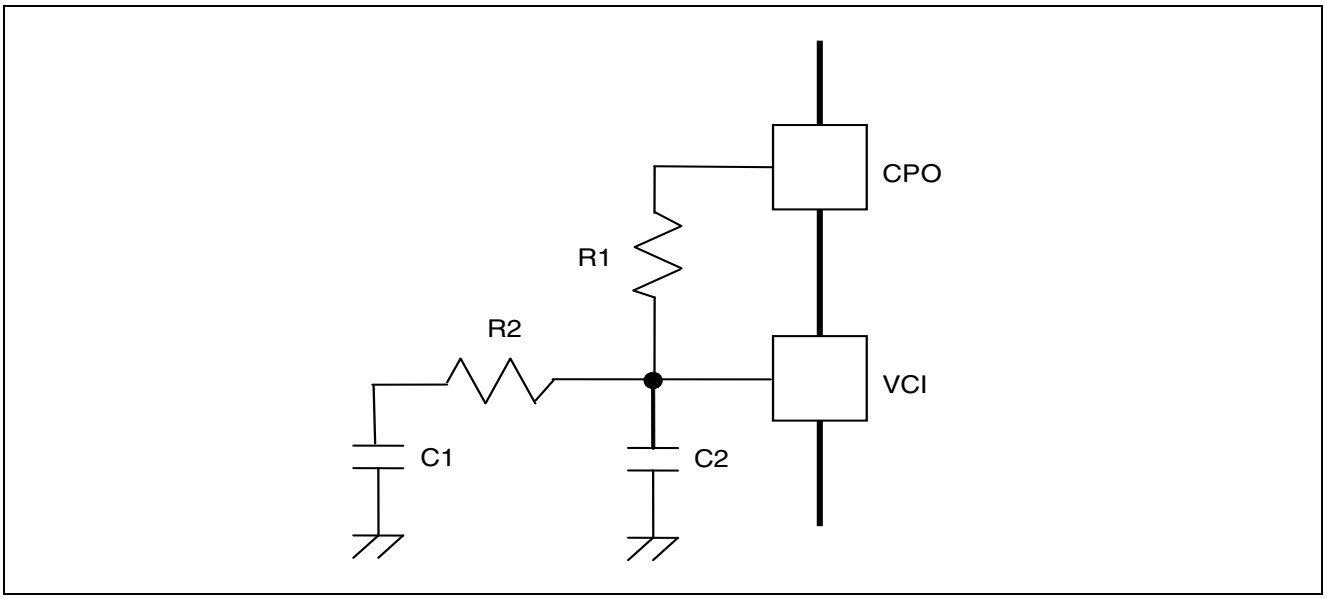
A.5 PLL for Dot Clock Generation

LPF configuration example and setting example of PLL for dot clock generation are shown below.

PLL

The built-in PLL for dot clock generation has VCO corresponding to the oscillation frequency.

It is required to connect LPF shown in the following figure to the CPO pin.



The constant of LPF is depending on the oscillation frequency. For the recommended value, contact your sales representative.

The following is an example configuration.

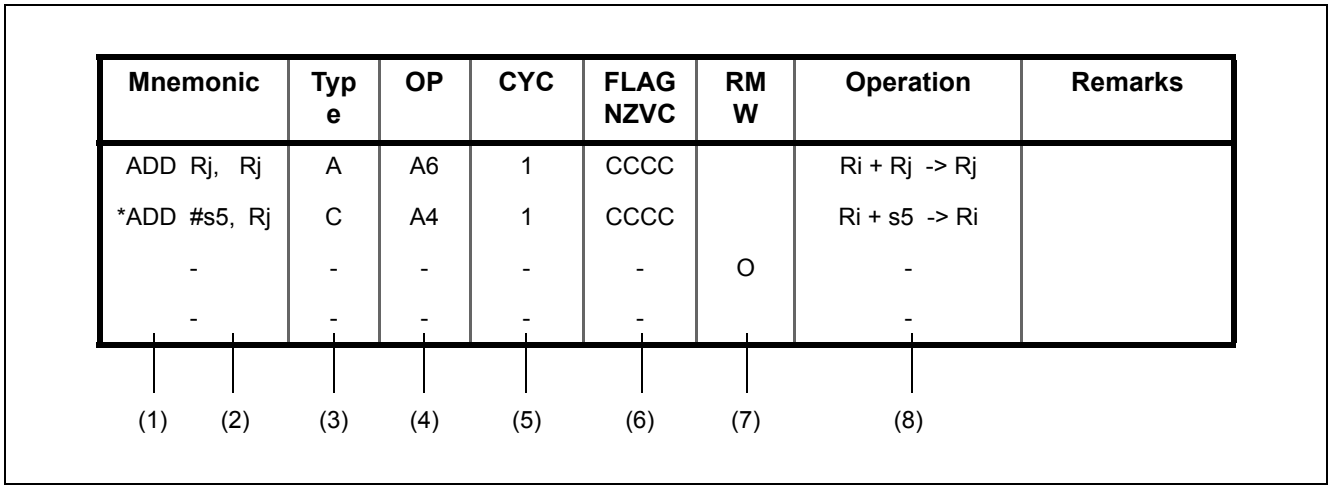
No.	HSYNC (KHz)	PLL (MHz)	VCO	External LPF			
				R1 (KΩ)	R2 (Ω)	C1 (μF)	C2 (pF)
1	15.734	32.223	VCO1	3.3	510	0.33	100

A.6 Lists of Instructions

This section lists and maps instructions for the FR80 family CPUs.

A.6.1 Instruction List

This section explains the symbols used in the instruction tables and instruction rules.



Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ADD Rj, Rj	A	A6	1	CCCC		Ri + Rj -> Rj	
*ADD #s5, Rj	C	A4	1	CCCC		Ri + s5 -> Ri	
-	-	-	-	-	O	-	
-	-	-	-	-		-	

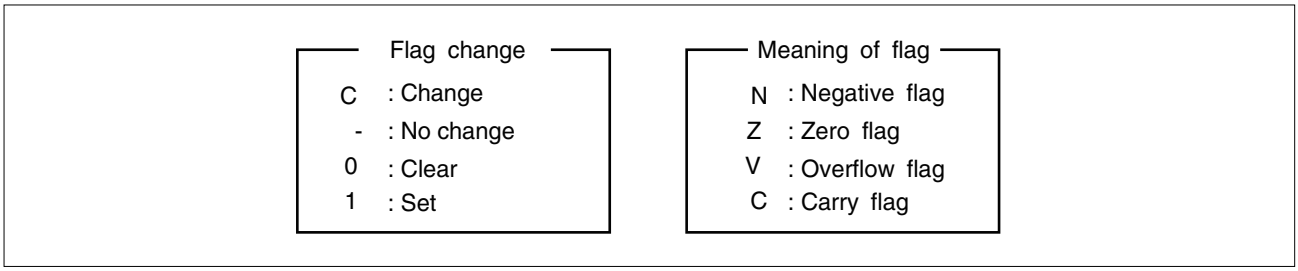
(1) (2) (3) (4) (5) (6) (7) (8)

1. The instruction name is shown.

   Instructions marked with* are extended instructions implemented either by extending existing instructions or by coding from scratch using the assembler, which are not native to the CPU.
2. A specifiable Addressing mode is shown in the operand by the sign.

   Please refer to-the sign of the Addressing mode (next item) for the meaning of the sign.
3. The instruction format is shown.
4. The hexadecimal number is displayed to the instruction code (Not written in assembler extended instructions).
5. The number of machine cycles is shown.
   a. It is a memory access cycle, and there is a possibility to postpone by the Ready function.
      The minimum number of cycles is 1.
   b. It is a memory access cycle, and there is a possibility to postpone by the Ready function. When the immediately succeeding instruction references the register to be subject to LD operation, however, an interlock is applied and the number of execution cycles is incremented by 1.
      When the number of uncompleted LD instructions reaches 4, the interlock is applied, continuing from that point until the first LD instruction is completed, and the number of execution cycles is incremented by a given number (number-of-memory-access cycles-number-of-cycles-from-instruction-issuance-until-completion-of-first-LD instruction).
   c. If the succeeding instruction references MDH, an interlock is applied and the number of execution cycles is incremented to become 2. Otherwise, the number of cycles is 1.
   d. If no read-ahead instruction has been executed on the prefetch buffer, the number of cycles is 2. The minimum number of cycles is 1.

6. The flag change is shown.

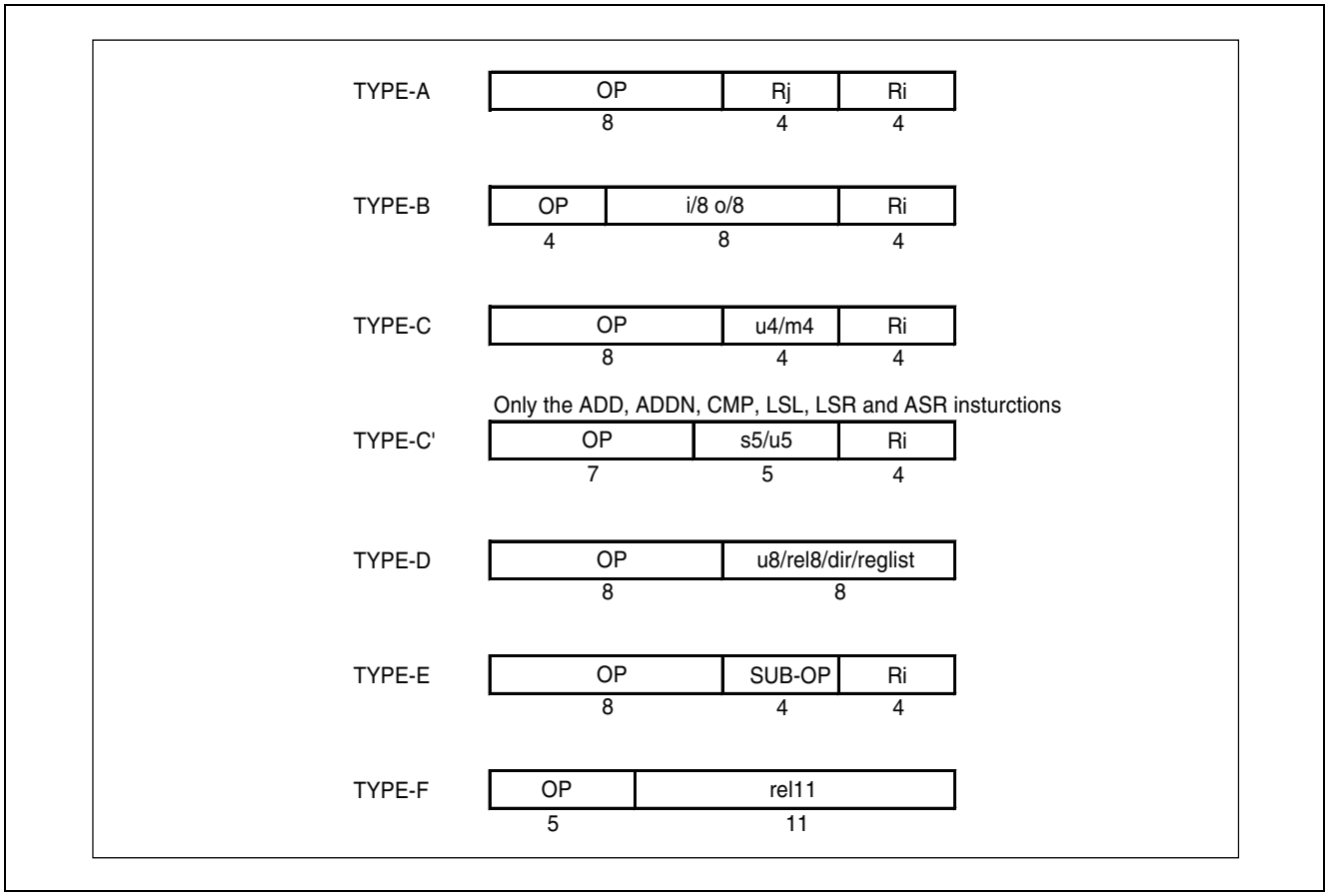


7. O is applied for RMW type of instructions.
8. The instruction operation is written.

Addressing Mode Symbols

Ri	: register direct (R0 to R15, AC, FP, SP)
Rj	: register direct (R0 to R15, AC, FP, SP)
R13	: register direct (R13, AC)
Ps	: Register direct (program status register)
Rs	: register direct (TBR, RP, SSP, USP, MDH, MDL)
#i4	: 4-bit value immediately (zero extension:0 to 15, negative extension:-16 to -1)
#i8	: Unsigned 8-bit value immediately (0 to 255)
#i20	: Unsigned 20-bit value immediately (-0X80000 to 0XFFFFF) Attention: -0X7FFFF to -1 is treated as 0X7FFFF to 0XFFFFF.
#i32	: Unsigned 32-bit value immediately (-0X80000000 to 0XFFFFFFFF) Attention: 0X80000000 to -1 is treated as 0X80000000 to 0XFFFFFFFF.
#s5	: Signed 5-bit immediate value (-16 to 15)
#s10	: Signed 10-bit immediate value (only multiples of 4, - 512 to 508)
#u4	: Unsigned 4-bit value immediately (0 to 15)
#u5	: Unsigned 5-bit value immediately (0 to 31)
#u8	: Unsigned 8-bit value immediately (0 to 255)
#u10	: Unsigned 10-bit value immediately (Only the multiple of 4, 0 to 1020)
@dir8	: Unsigned 8-bit direct address (0 to 0XFF)
@dir9	: Unsigned 9-bit direct address (Only the multiple of 2, 0 to 0X1FE)
@dir10	: Unsigned 10-bit direct address (Only the multiple of 4, 0 to 0X3FC)
label9	: Signed 9-bit branch address (only multiples of 2, -0X100 to 0XFC)
label12	: Signed 12-bit branch address (only multiples of 2, -0X800 to 0X7FC)
label20	: Signed 20-bit branch address (-0X80000 to 0X7FFFF)
label32	: Signed 32-bit branch address (-0X80000000 to 0X7FFFFFFF)
@Ri	: Register indirect (R0 to R15, AC, FP, SP)
@Rj	: Register indirect (R0 to R15, AC, FP, SP)
@(R13,Rj)	: Relativity is register indirect (Rj: R0 to R15, AC, FP, SP)
(R14,disp10)	: Relative indirectly register (Only the multiple of disp10:-0X200 to 0X1FC 4)
@(R14,disp9)	: Relative indirectly register (Only the multiple of disp9: -0X100 to 0XFE 2)
(R14,disp8)	: Relativity is register indirect (disp8: -0X80 to 0X7F)
@(R15,udisp6)	: Relative indirectly register (Only the multiple of 4, udisp6: 0 to 60)
@Ri+	: Register indirect with post increment (R0 to R15, AC, FP, SP)
@R13+	: Register indirect with post increment (R13, AC)
@SP+	: Stack pop
@-SP	: Stack push
(reglist)	: Register list

Instruction format



Operation column

The symbols listed below are used in the operation column of the instruction tables and in operations for the instruction rules.

extu( )	It represents a zero extension operation. The empty higher bits are padded with "0"s.
extn( )	It represents a negative extension operation. The empty higher bits are padded with "1"s.
exts( )	It represents a signed extension operation. If the MSB of the data in ( ) is "0", a zero extension operation is performed; if the MSB is "1", a negative extension operation is performed.
&	It represents a logical multiplication (AND) of each bit.
|	It represents a logical addition (OR) of each bit.
^	It represents an exclusive disjunction (EXOR) of each bit.
( )	Parentheses indicate indirect addressing. A value is read from or written to memory at the address indicated by the register or expression in ( ).
{ }	Curly brackets explicitly indicate the priority of operations. { } is used because ( ) is used for indirect addressing.

if (condition) then {expression}
or
if (condition) then {expression 1} else {expression 2}
	Each represents conditional execution. If the condition is satisfied, the expression following "then" is executed. If the condition is not satisfied, the expression following "else" is executed. One or more expressions enclosed in { } can be scripted.

[m:n]	Bits are retrieved from bit m to bit n for an operation.

A.6.2 Instruction Tables

This section explains the instructions for the FR80 family CPUs.

There is a total of 162 instructions for the FR80 family CPUs. They are categorized into the following 15 types:

- Add-subtract instructions
- Comparison operation instructions
- Logical operation instructions
- Bit operation instructions
- Multiplication and division instructions
- Shift operation instructions
- Immediate value data transfer instructions
- Memory load instructions
- Memory store instructions
- Register-to-register transfer instructions/Dedicated register transfer instructions
- Non-delayed branch instructions
- Delayed branch instructions
- Direct addressing instructions
- Bit search instructions
- Other instructions

Table A-2. Add-subtract Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ADD Rj, Ri	A	A6	1	CCCC	-	Ri + Rj → Ri	
*ADD #s5, Ri	C'	-	1	CCCC	-	Ri + s5 → Ri	The upper 1 bit of s5 is considered as a sign in the assembler.
ADD #i4, Ri	C	A4	1	CCCC	-	Ri + extu(i4) → Ri	i4 is a zero extension.
ADD2 #i4, Ri	C	A5	1	CCCC	-	Ri + extn(i4) → Ri	i4 is a negative extension.
ADDC Rj, Ri	A	A7	1	CCCC	-	Ri + Rj + C → Ri	Addition with a carry
ADDN Rj, Ri	A	A2	1	----	-	Ri + Rj → Ri	
*ADDN #s5, Ri	C'	-	1	----	-	Ri + s5 → Ri	The upper 1 bit of s5 is considered as a sign in the assembler.
ADDN #i4, Ri	C	A0	1	----	-	Ri + extu(i4) → Ri	i4 is a zero extension.
ADDN2 #i4, Ri	C	A1	1	----	-	Ri + extn(i4) → Ri	i4 is a negative extension.
SUB Rj, Ri	A	AC	1	CCCC	-	Ri-Rj → Ri	
SUBC Rj, Ri	A	AD	1	CCCC	-	Ri-Rj-C → Ri	Subtraction with a carry
SUBN Rj, Ri	A	AE	1	----	-	Ri-Rj → Ri	

Table A-3. Comparison Operation Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
CMP Rj, Ri	A	AA	1	CCCC	-	Ri-Rj	
*CMP #s5, Ri	C'	-	1	CCCC	-	Ri-s5	The upper 1 bit of s5 is considered as a sign in the assembler.
CMP #i4, Ri	C	A8	1	CCCC	-	Ri-extu(i4)	i4 is a zero extension.
CMP2 #i4, Ri	C	A9	1	CCCC	-	Ri-extn(i4)	i4 is a negative extension.

Table A-4. Logical Operation Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
AND Rj, Ri	A	82	1	CC--	-	Ri & Rj → Ri	Word
AND Rj, @Ri	A	84	1 + 2a	CC--	O	(Ri) & Rj → (Ri)	Word
ANDH Rj, @Ri	A	85	1 + 2a	CC--	O	(Ri) & Rj → (Ri)	Half word
ANDB Rj, @Ri	A	86	1 + 2a	CC--	O	(Ri) & Rj → (Ri)	Byte
OR Rj, Ri	A	92	1	CC--	-	Ri | Rj → Ri	Word
OR Rj, @Ri	A	94	1 + 2a	CC--	O	(Ri) | Rj → (Ri)	Word
ORH Rj, @Ri	A	95	1 + 2a	CC--	O	(Ri) | Rj → (Ri)	Half word
ORB Rj, @Ri	A	96	1 + 2a	CC--	O	(Ri) | Rj → (Ri)	Byte
EOR Rj, Ri	A	9A	1	CC--	-	Ri ^ Rj → Ri	Word
EOR Rj, @Ri	A	9C	1 + 2a	CC--	O	(Ri) ^ Rj → (Ri)	Word
EORH Rj, @Ri	A	9D	1 + 2a	CC--	O	(Ri) ^ Rj → (Ri)	Half word
EORB Rj, @Ri	A	9E	1 + 2a	CC--	O	(Ri) ^ Rj → (Ri)	Byte

Table A-5. Bit Operation Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
BANDL #u4, @Ri	C	80	1 + 2a	----	O	(Ri) & {$F0_H$ + u4} → (Ri)	Lower 4 bits
BANDH #u4, @Ri	C	81	1 + 2a	----	O	(Ri) & {u4<<4 + $0F_H$} → (Ri)	Higher 4 bits
*BAND #u8, @Ri[1]	-	-	-	----	O	(Ri) &= u8	
BORL #u4, @Ri	C	90	1 + 2a	----	O	(Ri) | u4 → (Ri)	Lower 4 bits
BORH #u4, @Ri	C	91	1 + 2a	----	O	(Ri) | {u4<<4} → (Ri)	Higher 4 bits
*BOR #u8, @Ri[2]	-	-	-	----	O	(Ri) | = u8	
BEORL #u4, @Ri	C	98	1 + 2a	----	O	(Ri) ^ u4 → (Ri)	Lower 4 bits
BEORH #u4, @Ri	C	99	1 + 2a	----	O	(Ri) ^ {u4<<4} → (Ri)	Higher 4 bits
*BEOR #u8, @Ri[3]	-	-	-	----	O	(Ri) ^ = u8	
BTSTL #u4, @Ri	C	88	2 + a	0C--	-	(Ri) & u4	Lower 4 bits
BTSTH #u4, @Ri	C	89	2 + a	CC--	-	(Ri) & {u4<<4}	Higher 4 bits

[1]: The assembler generates BANDL or BANDH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BANDL and BANDH are occasionally generated.

[2]: The assembler generates BORL or BORH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BORL and BORH are occasionally generated.

[3]: The assembler generates BEORL or BEORH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BEORL and BEORH are occasionally generated.

Table A-6. Multiplication and Division Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
MUL Rj, Ri	A	AF	5	CCC-	-	Ri x Rj → MDH,MDL	32 x 32 bits = 64 bits
MULU Rj, Ri	A	AB	5	CCC-	-	Ri x Rj → MDH,MDL	Unsigned
MULH Rj, Ri	A	BF	3	CC--	-	Ri x Rj → MDL	16 x 16 bits = 32 bits
MULUH Rj, Ri	A	BB	3	CC--	-	Ri x Rj → MDL	Unsigned
DIV0S Ri	E	97-4	1	----	-	With the given instruction sequence MDL / Ri → MDL MDL % Ri → MDH	Step operation 32 / 32 bits = 32 bits
DIV0U Ri	E	97-5	1	----	-		
DIV1 Ri	E	97-6	1	-C-C	-		
DIV2 Ri	E	97-7	c	-C-C	-		
DIV3	E	9F-6	1	----	-		
DIV4S	E	9F-7	1	----	-		
*DIV Ri[1]	-	-	36	-C-C	-	MDL/Ri → MDL, MDL %Ri → MDH	
*DIVU Ri[2]	-	-	36	-C-C	-	MDL/Ri → MDL, MDL %Ri → MDH	

[1]: DIV0S, DIV1×32, DIV2, DIV3, and DIV4S are generated. The instruction code length becomes 72 bytes.

[2]: DIV0U, DIV1×32 are generated. The instruction code length becomes 66 bytes.

Table A-7. Shift Operation Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LSL Rj, Ri	A	B6	1	CC-C	-	Ri << Rj → Ri	Logical shift
*LSL #u5, Ri (u5: 0 to 31)	C'	-	1	CC-C	-	Ri << u5 → Ri	
LSL #u4, Ri	C	B4	1	CC-C	-	Ri << u4 → Ri	
LSL2 #u4, Ri	C	B5	1	CC-C	-	Ri << {u4 + 16} → Ri	
LSR Rj, Ri	A	B2	1	CC-C	-	Ri >> Rj → Ri	Logical shift
*LSR #u5, Ri (u5: 0 to 31)	C'	-	1	CC-C	-	Ri >> u5 → Ri	
LSR #u4, Ri	C	B0	1	CC-C	-	Ri >> u4 → Ri	
LSR2 #u4, Ri	C	B1	1	CC-C	-	Ri >> {u4 + 16} → Ri	
ASR Rj, Ri	A	BA	1	CC-C	-	Ri >> Rj → Ri	Arithmetic shift
*ASR #u5, Ri (u5: 0 to 31)	C'	-	1	CC-C	-	Ri >> u5 → Ri	
ASR #u4, Ri	C	B8	1	CC-C	-	Ri >> u4 → Ri	
ASR2 #u4, Ri	C	B9	1	CC-C	-	Ri >> {u4 + 16} → Ri	

Table A-8. Immediate Value Data Transfer Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LDI:32 #i32, Ri	H	9F-8	d	----	-	i32 → Ri	
LDI:20 #i20, Ri	G	9B	d	----	-	extu(i20) → Ri	The higher 12 bits are a zero extension.
LDI:8 #i8, Ri	B	C0	1	----	-	extu(i8) → Ri	The higher 24 bits are a zero extension.
*LDI {i8 | i20 | i32}, Ri[1]	-	-	-	-	-	{i8 | i20 | i32} → Ri	

[1]: When the immediate value is an absolute value, i8, i20, i32 is selected automatically by the assembler.
When the immediate value is a relative value or includes an externally referenced symbol, i32 is selected.

Table A-9. Memory load instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LD @Rj, Ri	A	04	b	----	-	(Rj) → Ri	Word
LD @(R13, Rj), Ri	A	00	b	----	-	(R13 + Rj) → Ri	
LD @(R14, disp10), Ri	B	2	b	----	-	(R14 + o8 x 4) → Ri	
LD @(R15, udisp6), Ri	C	03	b	----	-	(R15 + u4 x 4) → Ri	
LD @R15+, Ri	E	07-0	b	----	-	(R15) → Ri, R15 + 4 → R15	
LD @R15+, Rs	E	07-8	b	----	-	(R15) → Rs, R15 + 4 → R15	Rs : Special register
LD @R15+, PS	E	07-9	1 + a	CCCC	-	(R15) → PS, R15 + 4 → R15	Word

Table A-9. Memory load instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LDUH @Rj, Ri	A	05	b	----	-	extu((Rj)) → Ri	Half word zero extension
LDUH @(R13, Rj), Ri	A	01	b	----	-	extu((R13 + Rj)) → Ri	
LDUH @(R14, disp9), Ri	B	04	b	----	-	extu((R14 + o8 x 2)) → Rj	
LDUB @Rj, Ri	A	06	b	----	-	extu((Rj)) → Ri	Byte zero extension
LDUB @(R13, Rj), Ri	A	02	b	----	-	extu((R13 + Rj)) → Ri	
LDUB @(R14, disp8), Ri	B	6	b	----	-	extu((R14 + o8)) → Ri	

- The relationship between the instruction format TYPE-B o8/TYPE-C u4 fields and disp8 to disp10 in assembler code is as follows:

  o8 = disp8

  o8 = disp9 >> 1

  o8 = disp10 >> 2

  u4 = udisp6 >> 2

Table A-10. Memory Store Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ST Ri, @Rj	A	14	a	----	-	Ri → (Rj)	Word
ST Ri, @(R13, Rj)	A	10	a	----	-	Ri → (R13 + Rj)	
ST Ri, @(R14, disp10)	B	3	a	----	-	Ri → (R14 + o8 x 4)	
ST Ri, @(R15, udisp6)	C	13	a	----	-	Ri → (R15 + u4 x 4)	
ST Ri, @-R15	E	17-0	a	----	-	R15 - 4 → R15, Ri → (R15)	
ST Rs, @-R15	E	17-8	a	----	-	R15 - 4 → R15, Rs → (R15)	Rs : Special register
ST PS, @-R15	E	17-9	a	----	-	R15 - 4 → R15, PS → (R15)	Word
STH Ri, @Rj	A	15	a	----	-	Ri → (Rj)	Half word
STH Ri, @(R13, Rj)	A	11	a	----	-	Ri → (R13 + Rj)	
STH Ri, @(R14, disp9)	B	5	a	----	-	Ri → (R14 + o8 x 2)	
STB Ri, @Rj	A	16	a	----	-	Ri → (Rj)	Byte
STB Ri, @(R13, Rj)	A	12	a	----	-	Ri → (R13 + Rj)	
STB Ri, @(R14, disp8)	B	7	a	----	-	Ri → (R14 + o8)	

- The relationship between the instruction format TYPE-B o8/TYPE-C u4 fields and disp8 to disp10 in assembler code is as follows:

  o8 = disp8

  o8 = disp9 >> 1

  o8 = disp10 >> 2

  u4 = udisp6 >> 2

Table A-11. Register-to-register Transfer Instructions/dedicated Register Transfer Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
MOV Rj, Ri	A	8B	1	----	-	Rj → Ri	Transfer between general-purpose registers
MOV Rs, Ri	A	B7	1	----	-	Rs → Ri	Rs: special register
MOV Ri, Rs	A	B3	1	----	-	Ri → Rs	Rs: special register
MOV PS, Ri	E	17-1	1	----	-	PS → Ri	PS: Program status
MOV Ri, PS	E	07-1	1	CCCC	-	Ri → PS	PS: Program status

Table A-12. Non-delayed Branch Instructions (Sheet 1 of 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
JMP @Ri	E	97-0	2	----	-	Ri → PC
CALL label12	F	D0	2	----	-	PC + 2 → RP, PC + 2 + exts(rel11 x 2) → PC
CALL @Ri	E	97-1	2	----	-	PC + 2 → RP, Ri → PC
RET	E	97-2	2	----	-	RP → PC
INT #u8	D	1F	1 + 3a	----	-	SSP-4 → SSP, PS → (SSP), SSP-4 → SSP, PC + 2 → (SSP), 0 → CCR:I, 0 → CCR:S, (TBR + 3FC-u8 x 4) → PC
INTE	E	9F-3	1 + 3a	----	-	SSP-4 → SSP, PS → (SSP), SSP-4 → SSP, PC + 2 → (SSP), 0 → CCR:S, 4 → ILM, (TBR + 3D8) → PC
RETI	E	97-3	1 + 2b	----	-	(SSP) → PC, SSP + 4 → SSP, (SSP) → PS, SSP + 4 → SSP
BRA label9	D	E0	2	----	-	PC + 2 + exts(rel8 x 2) → PC
BNO label9	D	E1	1	----	-	Non-branch
BEQ label9	D	E2	2/1	----	-	if (Z==1) then PC + 2 + exts(rel8 x 2) → PC
BNE label9	D	E3	2/1	----	-	if (Z==0) then PC + 2 + exts(rel8 x 2) → PC
BC label9	D	E4	2/1	----	-	if (C==1) then PC + 2 + exts(rel8 x 2) → PC
BNC label9	D	E5	2/1	----	-	if (C==0) then PC + 2 + exts(rel8 x 2) → PC
BN label9	D	E6	2/1	----	-	if (N==1) then PC + 2 + exts(rel8 x 2) → PC
BP label9	D	E7	2/1	----	-	if (N==0) then PC + 2 + exts(rel8 x 2) → PC
BV label9	D	E8	2/1	----	-	if (V==1) then PC + 2 + exts(rel8 x 2) → PC
BNV label9	D	E9	2/1	----	-	if (V==0) then PC + 2 + exts(rel8 x 2) → PC
BLT label9	D	EA	2/1	----	-	if (V ^ N==1) then PC + 2 + exts(rel8 x 2) → PC
BGE label9	D	EB	2/1	----	-	if (V ^ N==0) then PC + 2 + exts(rel8 x 2) → PC

Table A-12. Non-delayed Branch Instructions (Sheet 2 of 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
BLE label9	D	EC	2/1	----	-	if ({V ^ N} | Z==1) then PC + 2 + exts(rel8 x 2) → PC
BGT label9	D	ED	2/1	----	-	if ({V ^ N} | Z==0) then PC + 2 + exts(rel8 x 2) → PC
BLS label9	D	EE	2/1	----	-	if (C or Z==1) then PC + 2 + exts(rel8 x 2) → PC
BHI label9	D	EF	2/1	----	-	if (C or Z==0) then PC + 2 + exts(rel8 x 2) → PC

- "2/1" in the CYC column represents 2 in cases of branching and 1 in cases of no branching.
- The stack flag (S) must be "0" when RETI is executed.
- The relationship between the instruction format TYPE-D rel8/TYPE-F rel11 fields and label9/label12 in assembler code is as follows:

  rel8 = (label9-PC-2)/2

  rel11 = (label12-PC-2)/2

Table A-13. Delayed Branch Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
JMP:D @Ri	E	9F-0	1	----	-	Ri → PC
CALL:D label12	F	D8	1	----	-	PC + 4 → RP, PC + 2 + exts(rel11 x 2) → PC
CALL:D @Ri	E	9F - 1	1	----	-	PC + 4 → RP, Ri → PC
RET:D	E	9F - 2	1	----	-	RP → PC
BRA:D label9	D	F0	1	----	-	PC + 2 + exts(rel8 x 2) → PC
BNO:D label9	D	F1	1	----	-	Non-branch
BEQ:D label9	D	F2	1	----	-	if (Z==1) then PC + 2 + exts(rel8 x 2) → PC
BNE:D label9	D	F3	1	----	-	if (Z==0) then PC + 2 + exts(rel8 x 2) → PC
BC:D label9	D	F4	1	----	-	if (C==1) then PC + 2 + exts(rel8 x 2) → PC
BNC:D label9	D	F5	1	----	-	if (C==0) then PC + 2 + exts(rel8 x 2) → PC
BN:D label9	D	F6	1	----	-	if (N==1) then PC + 2 + exts(rel8 x 2) → PC
BP:D label9	D	F7	1	----	-	if (N==0) then PC + 2 + exts(rel8 x 2) → PC
BV:D label9	D	F8	1	----	-	if (V==1) then PC + 2 + exts(rel8 x 2) → PC
BNV:D label9	D	F9	1	----	-	if (V==0) then PC + 2 + exts(rel8 x 2) → PC
BLT:D label9	D	FA	1	----	-	if (V ^ N==1) then PC + 2 + exts(rel8 x 2) → PC
BGE:D label9	D	FB	1	----	-	if (V ^ N==0) then PC + 2 + exts(rel8 x 2) → PC
BLE:D label9	D	FC	1	----	-	if ({V ^ N} | Z==1) then PC + 2 + exts(rel8 x 2) → PC
BGT:D label9	D	FD	1	----	-	if ({V ^ N} | Z==0) then PC + 2 + exts(rel8 x 2) → PC
BLS:D label9	D	FE	1	----	-	if (C or Z==1) then PC + 2 + exts(rel8 x 2) → PC
BHI:D label9	D	FF	1	----	-	if (C or Z==0) then PC + 2 + exts(rel8 x 2) → PC

- Branching for a delayed branch instruction occurs after the next instruction (delay slot) is executed.
- The relationship between the instruction format TYPE-D rel8/TYPE-F rel11 fields and label9/label12 in assembler code is as follows:

  rel8 = (label9-PC-2)/2

  rel11 = (label12-PC-2)/2

Table A-14. Direct Addressing Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
DMOV @dir10, R13	D	08	b	----	-	(dir10) → R13	Word
DMOV R13, @dir10	D	18	a	----	-	R13 → (dir10)	
DMOV @dir10, @R13+	D	0C	1 + 2a	----	-	(dir10) → (R13), R13+=4	
DMOV @R13+, @dir10	D	1C	1 + 2a	----	-	(R13) → (dir10), R13+=4	
DMOV @dir10, @-R15	D	0B	1 + 2a	----	-	R15-=4, (R15) → (dir10)	
DMOV @R15+, @dir10	D	1B	1 + 2a	----	-	(R15) → (dir10), R15+=4	
DMOVH @dir9, R13	D	09	b	----	-	(dir9) → R13	Half word
DMOVH R13, @dir9	D	19	a	----	-	R13 → (dir9)	
DMOVH @dir9, @R13+	D	0D	1 + 2a	----	-	(dir9) → (R13), R13+=2	
DMOVH @R13+, @dir9	D	1D	1 + 2a	----	-	(R13) → (dir9), R13+=2	
DMOVB @dir8, R13	D	0A	b	----	-	(dir8) → R13	Byte
DMOVB R13, @dir8	D	1A	a	----	-	R13 → (dir8)	
DMOVB @dir8, @R13+	D	0E	1 + 2a	----	-	(dir8) → (R13), R13++	
DMOVB @R13+, @dir8	D	1E	1 + 2a	----	-	(R13) → (dir8), R13++	

- The relationship between the instruction format TYPE-D dir8 field and dir8, dir9, and dir10 in assembler code is as follows:
  dir8 = dir8
  dir8 = dir9 >> 1
  dir8 = dir10 >> 2

Table A-15. Bit Search Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
SRCH0 Ri	E	97-C	1	----	-	search_zero (Ri) → Ri	Searches for the first 0 bit from MSB to LSB
SRCH1 Ri	E	97-D	1	----	-	search_one (Ri) → Ri	Searches for the first 1 bit from MSB to LSB
SRCHC Ri	E	97-E	1	----	-	search_change (Ri) → Ri	Searches for the first change from MSB to LSB

Table A-16. Other Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
NOP	E'	9F-A	1	----	-	No change	
ANDCCR #u8	D	83	1	CCCC	-	CCR & u8 → CCR	
ORCCR #u8	D	93	1	CCCC	-	CCR | u8 → CCR	
STILM #u8	D	87	1	----	-	u8 → ILM	ILM immediate value set
ADDSP #s10	D	A3	1	----	-	R15 += s10	
EXTSB Ri	E	97-8	1	----	-	exts (Ri[7:0]) → Ri	Signed extension 8 → 32
EXTUB Ri	E	97-9	1	----	-	extu (Ri[7:0]) → Ri	Zero extension 8 → 32
EXTSH Ri	E	97-A	1	----	-	exts (Ri[15:0]) → Ri	Signed extension 16 → 32
EXTUH Ri	E	97-B	1	----	-	extu (Ri[15:0]) → Ri	Zero extension 16 → 32
LDM0(reglist)	D	8C	[1]	----	-	(R15) → reglist, R15 increment	Load Multi R0 to R7
LDM1(reglist)	D	8D	[1]	----	-	(R15) → reglist, R15 increment	Load Multi R8 to R15
*LDM(reglist)[3]	-	-	-	----	-	(R15) → reglist, R15 increment	Load Multi R0 to R15
STM0(reglist)	D	8E	[2]	----	-	R15 decrement, reglist → (R15)	Store Multi R0 to R7
STM1(reglist)	D	8F	[2]	----	-	R15 decrement, reglist → (R15)	Store Multi R8 to R15
*STM(reglist)[4]	-		-	----	-	R15 decrement, reglist → (R15)	Store Multi R0 to R15
ENTER #u10	D	0F	1 + a	----	-	R14 → (R15-4), R15 - 4 → R14, R15-extu (u8 x 4) → R15	Function entry processing
LEAVE	E	9F - 9	b	----	-	R14 + 4 → R15, (R15-4) → R14	Function exit processing
XCHB @Rj, Ri	A	8A	2a	----	O	Ri → TEMP, extu((Rj)) → Ri, TEMP → (Rj)	Byte data for semaphore management

[1]: The number of execution cycles for LDM0 (reglist) and LDM1 (reglist) becomes b x n cycles when the number of registers specified is n.

[2]: The number of execution cycles for STM0 (reglist) and STM1 (reglist) becomes a x n cycles when the number of registers specified is n.

[3]: If reglist specifies any of R0 to R7, LDM0 is generated.
If it specifies any of R8 to R15, LDM1 is generated. Both LDM0 and LDM1 are occasionally generated.

[4]: If reglist specifies any of R0 to R7, STM0 is generated.
If it specifies any of R8 to R15, STM1 is generated. Both STM1 and STM0 are occasionally generated.

- In the ADDSP instruction, the relationship between the instruction format TYPE-D s8 field and s10 in assembler code is as follows:

  s8 = s10 >> 2

- In the ENTER instruction, the relationship between the instruction format TYPE-D u8 field and u10 in assembler code is as follows:

  u8 = u10 >> 2

A.6.3 List of Instructions That Can Be Specified for Delay Slots

This section lists instructions that can be specified for delay slots in delayed branch instructions.

- Add-subtract instructions

ADD Rj, Ri	ADD #14, Ri	ADD2 #i4, Ri
ADDC Rj, Ri	ADDN Rj, Ri	ADDN #i4, Ri
ADDN2 #i4, Ri	SUB Rj, Ri	SUBC Rj, Ri
SUBN Rj, Ri		

- Comparison operation instructions

CMP Rj, Ri	CMP #i4, Ri	CMP2 #i4, Ri

- Logical operation instructions

AND Rj, Ri	OR Rj, Ri	EOR Rj, Ri

- Multiplication and division instructions

DIV0S Ri	DIV0U Ri	DIV1 Ri
DIV2 Ri	DIV3	DIV4S

- Shift operation instructions

LSL Rj, Ri	LSL #u4, Ri	LSL2 #u4, Ri
LSR Rj, Ri	LSR #u4, Ri	LSR2 #u4, Ri
ASR Rj, Ri	ASR #u4, Ri	ASR2 #u4, Ri

- Immediate value data transfer instruction

LDI:8 #i8, Ri

- Memory load instructions

LD @Rj, Ri	LD @(R13, Rj), Ri	LD @(R14, disp10), Ri
LD @(R15, udisp6), Ri	LD @R15+, Ri	LD @R15+, Rs
LDUH @Rj, Ri	LDUH @(R13, Rj), Ri	LDUH @(R14, disp9), Ri
LDUB @Rj, Ri	LDUB @(R13, Rj), Ri	LDUB @(R14, disp8), Ri

- Memory store instructions

ST Ri, @Rj	ST Ri, @(R13, Rj)	ST Ri, @(R14, disp10)
ST Ri, @(R15, udisp6)	ST Ri, @-R15	ST Rs, @-R15
ST PS, @-R15		
STH Ri, @Rj	STH Ri, @(R13, Rj)	STH Ri, @(R14, disp9)
STB Ri, @Rj	STB Ri, @(R13, Rj)	STB Ri, @(R14, disp8)

- Register-to-register transfer instructions

MOV Rj, Ri	MOV Rs, Ri	MOV Ri, Rs
MOV PS, Ri	MOV Ri, PS	

- Direct addressing instructions

DMOV @dir10, R13	DMOV R13, @dir10	DMOVH @dir9, R13
DMOVH R13, @dir9	DMOVB @dir8, R13	DMOVB R13, @dir8

- Bit search instructions

SRCH0 Ri	SRCH1 Ri	SRCHC Ri

- Other instructions

NOP	ANDCCR #u8	ORCCR #u8
STILM #u8	ADDSP #s10	EXTSB Ri
EXTUB Ri	EXTSH Ri	EXTUH Ri
LEAVE		

Main changes in this edition

Changes (For details, refer to main body.)	
1.2 MB91610 Series Product Configuration	Table 1-1. MB91610 series product configuration was changed as indicated by the shading below. (Built-in RAM capacity → Built-in RAM capacity (Instruction execution enabled))
1.3 MB91610 Series Block Diagram	Figure 1-1. MB91610 series block diagram was changed as indicated by the shading below. (RAM → Built-in RAM (Instruction execution enabled))
3.1 Memory Space Memory map	Figure 3-1. Memory map was changed as indicated by the shading below. (Built-in RAM area → Built-in RAM area (Instruction execution enabled))
3.11.4 I Flag	The following description was added to the end of the page. (If an interrupt is received while executing an instruction to set I flag to "0", there is a delay for 1 cycle from execution of an instruction for I flag and ILM to change. Therefore, I flag becomes "0" although processing moves to the interrupt processing routine. At this time, if multiple interrupts are generated, I flag can not receive any interrupt because it is "0", and processing of multiple interrupts is not performed. I flag itself is updated when executing an instruction. Therefore, a value of I flag after update is saved to the stack, and when the value of the stack is returned, the value of I flag after update is reflected to PS register. To receive a new interrupt within the interrupt routine, it is required to set software to make I flag to "1" at the beginning of the interrupt routine.)
9.4.1 Reset Result Register (RSTRR)	The following "Figure 9-2. Bit configuration of the reset result register (RSTRR)" was corrected as indicated by the shading below. (X : Not initialized.→ X : Each bit is initialized by a specific reset factor. It is not initialized by other factors.)
	The following description was added to the under of Figure 9-1. (Please refer to "Flow of reset result determination" of "9.5.2 Reset Resource" for the determination of the Reset Result.)
	The following <Notes> was corrected as indicated by the shading below. (″ If this register is read, all the bits are cleared. ″ The value when the power supply is turned on is undefined.)

Changes (For details, refer to main body.)	
23.6.4 Activating the DMA Controller (DMAC)	The following description was added above "Figure 23-25. DMA transfer operation (through scan conversion interrupt requests)". (..... number for DMA transfer, see "31. DMA Controller (DMAC)". ″ In single conversion mode To perform DMA transfer, set the same value to the DMA block size and the interrupt generation FIFO stage number, and perform the next A/D activation after DMA is completed. ″ In repeat conversion mode To perform DMA transfer, set 1 to the DMA block size, and 1 for the interrupt generation FIFO stage number.)
	"Figure 23-25. DMA transfer operation (through scan conversion interrupt requests)" was corrected. The following <Note> was added to the lower part of "Figure 23-25. DMA transfer operation (through scan conversion interrupt requests)". (″ Set the same value to the DMA block size and the interrupt generation FIFO stage number. ″ Perform the next A/D activation after performing DMA transfer of all FIFO data.)
	"Figure 23-26. DMA retransfer operation" was corrected . The following <Note> was added to the lower part of "Figure 23-26. DMA retransfer operation". (Set 1 to block size of DMA, and 1 for the interrupt generation FIFO stage number.)
24.1 Characteristics of Multi-function Serial Interface	The following <Notes> was corrected as indicated by the shading below. (″ The operation mode must be set first. Otherwise, the part of registers of the same channel will be initialized when the operation mode is changed. For the registers to be initialized, see the notes for serial mode register (SMR) of each operation mode.)
24.4.2 Serial Mode Register (SMR)	The following <Note> was corrected as indicated by the shading below. (The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed. ″ Serial Control Register (SCR) ″ Extended Serial Control Register (ESCR))
24.10 Notes on UART Mode	Added "Notes on UART Mode" as 24.10.
24.13.2 Serial Mode Register (SMR)	The following <Note> was corrected as indicated by the shading below. (The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed. ″ Serial Control Register (SCR) ″ Extended Serial Control Register (ESCR))
24.16.1 Setting Baud Rate	The following <Note> was corrected as indicated by the shading below. (″ When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the SCINV bit settings. When the reload value is odd-numbered,the "L" width is the same as the "H" width. - When SPI is set to "0" and SCINV is set to "0", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer. - When SPI is set to "0" and SCINV is set to "1", the "L" width of the serial clock is one peripheral clock (PCLK) cycle longer. - When SPI is set to "1" and SCINV is set to "0", the "L" width of the serial clock is one peripheral clock (PCLK) cycle longer. - When SPI is set to "1" and SCINV is set to "1", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer.)
24.18 Notes on CSIO Mode	Added "Notes on CSIO Mode" as 24.18.
24.21.2 Serial Mode Register (SMR)	The following <Note> was corrected as indicated by the shading below. (The operation mode must be set first. Otherwise, the following registers of the same channel will be initialized when the operation mode is changed. ″ I^2C Bus Control Register (IBCR) ″ I^2C Bus Status Register (IBSR) Note, however, that when IBCR and SMR are written simultaneously with 16-bit write access, IBCR reflects the written content.)

Changes (For details, refer to main body.)	
24.23.1 Example of I^2C Flowcharts	The figure was corrected as "Figure 24-100. Master Reception Interrupt Process" and "Figure 24-106. Master Transmission Interrupt Process".
24.24 Notes on I^2C Mode	Added "Notes on I^2C Mode" as 24.24.

Note: Please see “Document Revision History” about later revised information.

Revision History

Document Revision History

Document Title: MB91610 Series 32-bit Microcontroller FR80 Hardware Manual Document Number: 002-04478				
Revision	**ECN#**	**Issue Date**	**Origin of Change**	**Description of Change**
**	-	01/30/2014	MKEA	Initial release
*A	5602719	02/08/2017	TOYO	Migrated Spansion document “CM71-10148-3E” into Cypress template format.