



The following document contains information on Cypress products. Although the document is marked with the name "Spansion" and "Fujitsu", the company that originally developed the specification, Cypress will continue to offer these products to new and existing customers.

Continuity of Specifications

There is no change to this document as a result of offering the device as a Cypress product. Any changes that have been made are the result of normal document improvements and are noted in the document history page, where supported. Future revisions will occur when appropriate, and changes will be noted in a document history page.

Continuity of Ordering Part Numbers

Cypress continues to support existing part numbers. To order these products, please use only the Ordering Part Numbers listed in this document.

For More Information

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

About Cypress

Cypress (NASDAQ: CY) delivers high-performance, high-quality solutions at the heart of today's most advanced embedded systems, from automotive, industrial and networking platforms to highly interactive consumer and mobile devices. With a broad, differentiated product portfolio that includes NOR flash memories, F-RAM™ and SRAM, Traveo™ microcontrollers, the industry's only PSoC® programmable system-on-chip solutions, analog and PMIC Power Management ICs, CapSense® capacitive touch-sensing controllers, and Wireless BLE Bluetooth® Low-Energy and USB connectivity solutions, Cypress is committed to providing its customers worldwide with consistent innovation, best-in-class support and exceptional system value.

FR80
32-BIT MICROCONTROLLER
MB91605A Series
HARDWARE MANUAL

FR80

32-BIT MICROCONTROLLER

MB91605A Series

HARDWARE MANUAL

For the information for microcontroller supports, see the following web site.

This web site includes the "**Customer Design Review Supplement**" which provides the latest cautions on system development and the minimal requirements to be checked to prevent problems before the system development.

<http://edevise.fujitsu.com/micom/en-support/>

FUJITSU SEMICONDUCTOR

Preface

Thank you for your continued use of Fujitsu semiconductor products.

Read this manual and "Data Sheet" thoroughly before using the products in the MB91605A series.

■ Purpose of this manual and intended readers

This manual explains the functions and operations of the MB91605A series and describes how it is used. The manual is intended for engineers engaged in the actual development of products using the MB91605A series.

■ Trademarks

FR is an abbreviation for the FUJITSU RISC controller, which is a product of Fujitsu Semiconductor Limited.

Other system and product names used in this manual are the trademarks of the related companies and organization.

■ Sample programs and development environment

Fujitsu offers sample programs free of charge for using the peripheral functions of the FR80 family. Fujitsu also makes available descriptions of the development environment required for the MB91605A series. Feel free to use the sample programs to verify the operational specifications and usage of this Fujitsu microcontroller.

- Microcontroller support information:

<http://jp.fujitsu.com/microelectronics/products/micom/support/>

* Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system.

Fujitsu is not responsible for any damage that may result from use of a sample program.

- The contents of this document are subject to change without notice.
Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU SEMICONDUCTOR device; FUJITSU SEMICONDUCTOR does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU SEMICONDUCTOR assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU SEMICONDUCTOR or any third party or does FUJITSU SEMICONDUCTOR warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU SEMICONDUCTOR assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
Please note that FUJITSU SEMICONDUCTOR will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance 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.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

Copyright ©2009-2010 FUJITSU SEMICONDUCTOR LIMITED All rights reserved.

How to Use This Manual

■ Finding a function

The following methods can be used to search for the explanation of the 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 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 "APPENDIX A I/O Map," and "APPENDIX B List of Registers."

- **Search from the index**

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

■ About the chapters

Basically, this manual explains one peripheral function per chapter.

However, each timer function/operation mode concerning the timer functions or other peripheral functions that can switch the operation mode is covered in separate chapters.

- Base timer

The following is explained in four chapters:

- Base timer (16/32-bit reload timer)
- Base timer (16-bit PWM timer)
- Base timer (16/32-bit PWC timer)
- Base timer (16-bit PPG timer)

- Multifunction serial interface

The following is explained in three chapters:

- Multifunction serial interface (UART)
- Multifunction serial interface (CSIO)
- Multifunction serial interface (I²C)

■ 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

CHAPTER 1 Overview	1
1.1 MB91605A Series Overview	2
1.2 MB91605A Series Product Configuration	5
1.3 MB91605A Series Block Diagram	6
1.4 Package Dimensions	7
CHAPTER 2 Pins of the MB91605A Series	9
2.1 Pin Assignment Diagram	10
2.2 Pin Functions	11
2.3 I/O Circuit Types	20
CHAPTER 3 CPU	25
3.1 Memory Space	26
3.2 Features of the Internal Architecture	28
3.3 Operation Modes	29
3.4 Pipeline	30
3.5 Overview of Instructions	32
3.5.1 Arithmetic Operation	32
3.5.2 Load and Store	32
3.5.3 Branch	33
3.5.4 Logical Operation and Bit Operation	33
3.5.5 Direct Addressing	33
3.5.6 Bit Search	33
3.5.7 Other	33
3.6 Basic Programming Model	34
3.7 Registers	35
3.7.1 General-purpose Registers (R0 to R15)	35
3.7.2 Program Status Register (PS)	36
3.7.3 Program Counter (PC)	41
3.7.4 Table Base Register (TBR)	42
3.7.5 Return Pointer (RP)	43
3.7.6 System Stack Pointer (SSP)	44
3.7.7 User Stack Pointer (USP)	45
3.7.8 Multiply & Divide Registers	46
3.8 Data Configuration	47
3.8.1 Bit Ordering	47
3.8.2 Byte Ordering	48
3.8.3 Word Alignment	49
3.9 Addressing	50
3.9.1 Direct Addressing Areas	50
3.9.2 20-bit Addressing Area	51
3.9.3 32-bit Addressing Area	51
3.9.4 Vector Table Initial Area	51
3.10 Branch Instructions	52

3.10.1	Operation with Delay Slots	52
3.10.2	Operation without Delay Slots	54
3.11	EIT (Exception, Interrupt, Trap)	55
3.11.1	EIT Sources	55
3.11.2	Return from EIT	55
3.11.3	Interrupt Level	56
3.11.4	I Flag	57
3.11.5	Interrupt Level Mask Register (ILM)	58
3.11.6	Level Mask for Interrupts/NMI	58
3.11.7	Interrupt Control Register (ICR)	59
3.11.8	System Stack Pointer (SSP)	59
3.11.9	Interrupt Stack	59
3.11.10	Table Base Register (TBR)	60
3.11.11	EIT Vector Table	60
3.11.12	Multi-EIT Processing	61
3.11.13	Operation	62
3.11.14	INT Instruction Operation	63
3.11.15	INTE Instruction Operation	64
3.11.16	Step Trace Trap Operation	64
3.11.17	Undefined Instruction Exception Operation	65
3.11.18	RETI Instruction Operation	65
3.11.19	Delay Slots and EIT	65
CHAPTER 4	Instruction Cache Memory	67
4.1	Overview	68
4.2	Configuration	69
4.3	Registers	71
4.3.1	Instruction/Data Cache Area Configuration Register (CARR)	71
4.3.2	Instruction Cache Control Register (ICHCR)	73
4.3.3	Instruction Cache Size Register (ISIZE)	75
4.3.4	Instruction Cache Function Register (IFUNC)	76
4.4	Explanation of Operations	78
4.4.1	Initial State	78
4.4.2	Instruction Cache Area	78
4.4.3	Invalidating Instruction Cache	79
4.4.4	Enabling Instruction Cache Operation	79
4.4.5	Instruction Cache Lock Function	79
4.4.6	Cache Content in each Cache Operation State	81
CHAPTER 5	Data Cache Memory	83
5.1	Overview	84
5.2	Configuration	85
5.3	Registers	87
5.3.1	Data Cache Control Register (DCHCR)	87
5.3.2	Data Cache Size Register (DSIZE)	88
5.3.3	Data Cache Function Register (DFUNC)	89
5.4	Explanation of Operations	91
5.4.1	Initial State	91
5.4.2	Data Cache Area	91

MB91605A Series

5.4.3	Invalidating Data Cache	92
5.4.4	Enabling Data Cache Operation	92
5.4.5	Cache Content in each Cache Operation State	92
5.4.6	RAM Operation Mode when Cache is Off	93
CHAPTER 6	Operation Mode	95
6.1	Overview	96
6.2	Configuration	97
6.3	Registers	98
6.3.1	Bus Mode Data Register (BMODR)	98
6.3.2	Mode Register (MODR)	99
6.4	Explanation of Operations	100
6.4.1	How to Set the Operation Mode	100
6.4.2	Details of Each Mode	100
CHAPTER 7	Clock Generating Parts	101
7.1	Overview	102
7.2	Configuration	103
7.2.1	Clock Generating Parts	103
7.2.2	Source Clock (SRCCLK) Selection Block	105
7.3	Pins	106
7.4	Registers	107
7.4.1	Clock Source Select Register (CSELR)	108
7.4.2	Clock Source Monitor Register (CMONR)	111
7.4.3	Clock Stabilization Time Select Register (CSTBR)	113
7.4.4	PLL Configuration Register (PLLCR)	115
7.5	Explanation of Operations	118
7.5.1	Explanation of Clock Source Operations	118
7.5.2	Switching the Source Clock (SRCCLK)	120
7.5.3	Multiple Rate for Generating the PLL Clock (PLLCLK)	122
CHAPTER 8	Clock Division Control Part	125
8.1	Overview	126
8.2	Internal Clocks	127
8.3	Configuration	129
8.4	Registers	130
8.4.1	Divide Clock Configuration Register 0 (DIVR0)	131
8.4.2	Divide Clock Configuration Register 1 (DIVR1)	133
8.4.3	Divide Clock Configuration Register 2 (DIVR2)	135
8.5	Division Rate	137
CHAPTER 9	Main Timer	139
9.1	Overview	140
9.2	Configuration	141
9.3	Registers	142
9.3.1	Main Timer Control Register (MTMCR)	143
9.4	Interrupts	147
9.5	An Explanation of Operations and Setting Procedure Examples	148
9.5.1	Main Timer Operation	148

9.5.2	Transition to Stop Mode	150
CHAPTER 10	Low-power Dissipation Mode	151
10.1	Overview	152
10.2	Configuration	153
10.3	Registers	155
10.3.1	Standby Mode Control Register (STBCR)	156
10.3.2	Sleep Rate Configuration Register (SLPRR)	159
10.4	An Explanation of Operations and Setting Procedure Examples	161
10.4.1	Operation When Clock Control Is Set	162
10.4.2	Operation in Doze Mode	164
10.4.3	Operation in Sleep Mode	165
10.4.4	Operation in Main Timer Mode	168
10.4.5	Operation in Stop Mode	170
10.5	Notes on Use	172
CHAPTER 11	Reset	173
11.1	Overview	174
11.2	Configuration	175
11.3	Pins	177
11.4	Registers	178
11.4.1	Reset Result Register (RSTRR)	179
11.4.2	Reset Control Register (RSTCR)	181
11.5	Explanation of Operations	183
11.5.1	Reset Types	183
11.5.2	Reset Resource	184
11.5.3	Operation of Reset	186
11.6	Operating State and Transition	191
CHAPTER 12	Interrupt Controller	195
12.1	Overview	196
12.2	Configuration	197
12.3	Registers	198
12.3.1	Interrupt Control Register (ICR00 to ICR47)	199
12.4	An Explanation of Operations and Setting Procedure Examples	201
12.4.1	Explanation of Operations of Interrupt Controller	201
12.5	Notes on Use	203
CHAPTER 13	NMI Input	205
13.1	Overview	206
13.2	Explanation of Operations	207
CHAPTER 14	Interrupt Request Batch-Read Function	209
14.1	Overview	210
14.2	Configuration	211
14.3	Registers	212
14.3.1	Interrupt Request Batch-Read Register 0 Upper (IRPR0H)	213
14.4	Notes on Use	215

MB91605A Series

CHAPTER 15 Delay Interrupt	217
15.1 Overview	218
15.2 Configuration	219
15.3 Registers	220
15.3.1 Delayed Interrupt Control Register (DICR)	221
15.4 An Explanation of Operations and Setting Procedure Examples	222
15.4.1 Explanation of Delay Interrupt Operations	222
15.5 Notes on Use	223
 CHAPTER 16 External Bus Interface	 225
16.1 Overview	226
16.2 Configuration	227
16.3 Registers	229
16.3.1 SRAM/FLASH Mode Register (MCMRx)	231
16.3.2 SRAM/FLASH Timing Register (MCTRx)	234
16.3.3 SRAM/FLASH Area Register (MCARx)	237
16.3.4 SDRAM Mode Register (SDMRx)	239
16.3.5 SDRAM Refresh Timer Register (SDRTRx)	242
16.3.6 SDRAM Power Down Count Register (SDPDRx)	244
16.3.7 SDRAM Timing Register (SDTRx)	245
16.3.8 SDRAM Command Register (SDCMRx)	248
16.3.9 Clock Control Register (CLKCTL)	250
16.4 Register Setting Examples	251
16.4.1 Read Access Examples of SRAM/FLASH Area	252
16.4.2 Write Access Examples of SRAM/FLASH Area	254
16.4.3 Address Area Setting Examples of SRAM/FLASH Area	258
16.5 Notes on Use	259
 CHAPTER 17 I/O Ports	 261
17.1 Overview	262
17.2 Configuration	264
17.3 Pins	265
17.4 Registers	266
17.4.1 Port Data Registers (PDR2 to PDRE)	268
17.4.2 Data Direction Registers (DDR2 to DDRE)	269
17.4.3 Port Function Registers (PFR2 to PFRE)	271
17.4.4 Pull-up Control Register (PCR)	285
17.4.5 ADER Control Register (ADER)	286
 CHAPTER 18 External Interrupt Controllers	 287
18.1 Overview	288
18.2 Configuration	289
18.3 Pins	291
18.4 Registers	292
18.4.1 External Interrupt Request Level Registers (ELVR0 to ELVR2)	293
18.4.2 External Interrupt Request Registers (EIRR0 to EIRR2)	296
18.4.3 Enable Interrupt Request Registers (ENIR0 to ENIR2)	298
18.5 Explanation of Operations and Setting Procedure Examples	300
18.5.1 Operations of the External Interrupt Controllers	300

18.5.2	Return from Standby Mode	303
18.5.3	Return from Sleep Mode	305
CHAPTER 19	Watchdog Timer	307
19.1	Overview	308
19.2	Configuration	309
19.3	Registers	311
19.3.1	Watchdog Timer Control Register 0 (WDTCR0)	312
19.3.2	Watchdog Timer Clear Pattern Register 0 (WDT CPR0)	315
19.4	Explanation of Operations and Setting Procedure Examples	316
19.4.1	Operations of the Watchdog Timer	316
CHAPTER 20	16-bit Reload Timer	319
20.1	Overview	320
20.2	Configuration	321
20.3	Pins	323
20.4	Registers	324
20.4.1	Timer Control Status Register (TMCSR0 to TMCSR2)	325
20.4.2	16-bit Timer Reload Register A (TMRLRA0 to TMRLRA2)	331
20.4.3	16-bit Timer Register (TMR0 to TMR2)	332
20.5	Interrupts	333
20.6	An Explanation of Operations and Setting Procedure Examples	334
20.6.1	Operation in Interval Timer Mode	335
20.6.2	Operations in Event Counter Mode	348
20.6.3	Operation in Cascade Mode	353
20.7	Notes on Use	355
CHAPTER 21	Base Timer I/O Select Function	357
21.1	Overview	358
21.2	Configuration	359
21.3	Pins	360
21.4	Registers	362
21.4.1	Base Timer IO Select Register for Ch.0/1/2/3 (BTSEL0123)	363
21.4.2	Base Timer IO Select Register for Ch.4/5/6/7 (BTSEL4567)	366
21.4.3	Base Timer IO Select Register for Ch.8/9/A/B (BTSEL89AB)	369
21.4.4	Base Timer Same Time Soft Start Register (BTSSSR)	372
21.5	I/O Mode	376
21.5.1	I/O Mode 0 (16-bit Timer Standard Mode)	376
21.5.2	I/O Mode 1 (Timer Full Mode)	378
21.5.3	I/O Mode 2 (External Trigger Shared Mode)	380
21.5.4	I/O Mode 3 (Other Channel Trigger Shared Mode)	382
21.5.5	Operations in I/O Mode 4 (Timer Activation/Stop Mode)	384
21.5.6	Operations in I/O Mode 5 (Same Time Software Activation Mode)	387
21.5.7	Operations in I/O Mode 6 (Software Activation Timer Activation/Stop Mode)	389
21.5.8	Operations in I/O Mode 7 (Timer Activation Mode)	391
21.5.9	Operations in I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode)	393

MB91605A Series

CHAPTER 22 Base Timer	395
22.1 Overview of the Base Timer	396
22.2 Block Diagrams of the Base Timer	398
22.3 Base Timer's Registers	406
22.4 Operations of the Base Timer	415
22.5 32-bit Mode Operations	417
22.6 Notes of Using the Base Timer	419
22.7 Base Timer Interrupts	421
22.8 Base Timer Description by Function Mode	422
22.8.1 PWM Function	423
22.8.2 PPG Function	437
22.8.3 Reload Timer Function	452
22.8.4 PWC Function	464
CHAPTER 23 10-bit A/D Converter	479
23.1 Overview	480
23.2 Configuration	481
23.3 Pins	482
23.4 Registers	483
23.4.1 A/D Control Register (ADCTH, ADCTL)	484
23.4.2 Software Conversion Analog Input Select Register (ADCH)	486
23.4.3 A/D Conversion Result Registers ch.0 to ch.11 (ADAT0 to ADAT11)	487
23.5 Interrupt	488
23.6 Explanation of Operations	489
CHAPTER 24 Multi-function Serial Interface	491
24.1 Characteristics of Multi-function Serial Interface	493
24.2 UART (Asynchronous Serial Interface)	494
24.3 Overview of UART (Asynchronous Serial Interface)	495
24.4 Registers of UART (Asynchronous Serial Interface)	497
24.4.1 Serial Control Register (SCR)	503
24.4.2 Serial Mode Register (SMR)	505
24.4.3 Serial Status Register (SSR)	508
24.4.4 Extended Serial Control Register (ESCR)	511
24.4.5 Reception Data Register / Transmission Data Register (RDR/TDR)	513
24.4.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	516
24.4.7 FIFO Control Register 1 (FCR1)	518
24.4.8 FIFO Control Register 0 (FCR0)	520
24.4.9 FIFO Byte Register (FBYTE1/FBYTE2)	523
24.5 Interrupts of UART	525
24.5.1 Occurrence of Reception Interrupts and Flag Set Timing	527
24.5.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing	528
24.5.3 Occurrence of Transmission Interrupts and Flag Set Timing	529
24.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing	530
24.6 Operation of UART	531
24.7 Dedicated Baud Rate Generator	536
24.7.1 Setting Baud Rate	537
24.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)	541
24.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)	543

24.10 Notes on UART Mode	547
24.11 CSIO (Clock Synchronous Serial Interface)	548
24.12 Overview of CSIO (Clock Synchronous Serial Interface)	549
24.13 Registers of CSIO (Clock Synchronous Serial Interface)	550
24.13.1 Serial Control Register (SCR)	556
24.13.2 Serial Mode Register (SMR)	559
24.13.3 Serial Status Register (SSR)	562
24.13.4 Extended Serial Control Register (ESCR)	564
24.13.5 Reception Data Register / Transmission Data Register (RDR/TDR)	566
24.13.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	569
24.13.7 FIFO Control Register 1 (FCR1)	570
24.13.8 FIFO Control Register 0 (FCR0)	572
24.13.9 FIFO Byte Register (FBYTE1/FBYTE2)	575
24.13.10 Serial Mode Select Registers (SSEL89AB)	577
24.13.11 Received Data Mirror Registers/Transmitted Data Mirror Registers (RDRM/TDRM)	579
24.14 Interrupts of CSIO (Clock Synchronous Serial Interface)	580
24.14.1 Occurrence of Reception Interrupts and Flag Set Timing	582
24.14.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing	583
24.14.3 Occurrence of Transmission Interrupts and Flag Set Timing	585
24.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing	586
24.15 Operation of CSIO (Clock Synchronous Serial Interface)	587
24.16 Dedicated Baud Rate Generator	606
24.16.1 Setting Baud Rate	607
24.17 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)	610
24.18 Notes on CSIO Mode	612
24.19 I ² C Interface	613
24.20 Overview of I ² C Interface	614
24.21 Registers of I ² C Interface	615
24.21.1 I ² C Bus Control Register (IBCR)	622
24.21.2 Serial Mode Register (SMR)	627
24.21.3 I ² C Bus Status Register (IBSR)	629
24.21.4 Serial Status Register (SSR)	633
24.21.5 Reception Data Register / Transmission Data Register (RDR/TDR)	635
24.21.6 7-bit Slave Address Mask Register (ISMK)	637
24.21.7 7-bit Slave Address Register (ISBA)	638
24.21.8 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	639
24.21.9 FIFO Control Register 1 (FCR1)	640
24.21.10 FIFO Control Register 0 (FCR0)	642
24.21.11 FIFO Byte Register (FBYTE1/FBYTE2)	646
24.22 Interrupts of I ² C Interface	648
24.22.1 Operation of I ² C Interface Communication	650
24.22.2 Master Mode	651
24.22.3 Slave Mode	668
24.22.4 Bus Error	672
24.23 Dedicated Baud Rate Generator	673
24.23.1 Example of I ² C Flowcharts	675
24.24 Notes on I ² C Mode	687

MB91605A Series

CHAPTER 25 DMA Controller (DMAC)	689
25.1 Overview	690
25.2 Configuration	691
25.3 Registers	692
25.3.1 Global Configuration Register (GCFR)	694
25.3.2 Channel Configuration Registers (CCFR0 to CCFR3)	695
25.3.3 Channel Status Registers (CSTR0 to CSTR3)	697
25.3.4 Channel Control Registers (CCTR0 to CCTR3)	701
25.3.5 Source Base Address Registers (SBA0 to SBA3)	704
25.3.6 Destination Base Address Registers (DBA0 to DBA3)	706
25.3.7 Primary Index Registers (PIX0 to PIX3)	708
25.3.8 Secondary Index Registers (SIX0 to SIX3)	709
25.3.9 Byte Count Limit Registers (BCL0 to BCL3)	710
25.3.10 Alternate Pointer Registers (APR0 to APR3)	711
25.3.11 DMA-halt by NMI Flag Register (DNMIR)	712
25.3.12 DMA-Halt by Interrupt Level Register (DILVR)	713
25.4 Explanation of Operations	714
25.4.1 Settings	714
25.4.2 Activation	720
25.4.3 Operation	721
25.5 Control Flow	735
25.6 Notes on Use	736
 CHAPTER 26 Select Function for DMA Transfer Request Generation/ Clear by a Peripheral Function	 739
26.1 Overview	740
26.2 Configuration	741
26.3 Registers	743
26.3.1 IO-Data Request Registers (IORR0 to IORR3)	745
26.3.2 Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0)	749
26.3.3 Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1)	751
26.3.4 Select Register 2 for DMA Transfer Request Clear by a Peripheral Function (ICSEL2)	753
26.3.5 Select Register 3 for DMA Transfer Request Clear by a Peripheral Function (ICSEL3)	755
26.4 An Explanation of Operations and Setting Procedure Examples	757
26.4.1 Operations upon a DMA Transfer	757
 CHAPTER 27 Remote Control Reception	 759
27.1 Overview	760
27.2 Registers	761
27.2.1 Remote Control Reception Control Register (RCCR)	762
27.2.2 Remote Control Reception Interrupt Control Register (RCST)	764
27.2.3 Device Address Configuration Registers 1, 2 (RCADR1, RCADR2)	766
27.2.4 Start Bit High Width Configuration Register (RCSHW)	767
27.2.5 High Width Configuration Register A(RCDAHW)	767
27.2.6 High Width Configuration Register B(RCDAHW)	768
27.2.7 Data Storage Registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL)	768
27.2.8 Clock Division Configuration Register (RCCKD)	769
27.3 Explanation of Operations and Setting Procedure Examples	770

CHAPTER 28	Serial Programming Connection Example	773
28.1	Pins Used	774
28.2	Serial Programming Connection Example	775
28.2.1	Synchronous Serial Programming Connection Example	775
28.2.2	Asynchronous Serial Programming Connection Example	776
CHAPTER 29	Handling the Device	777
29.1	Notes on Handling the Device	778
APPENDIX A	I/O Map	784
APPENDIX B	List of Registers	800
APPENDIX C	Interrupt Vectors	819
APPENDIX D	Pin State in Each CPU State	823
APPENDIX E	Lists of Instructions	830
E.1	Instruction List	830
E.2	Instruction Tables	834
E.3	List of Instructions That Can Be Specified for Delay Slots	847
APPENDIX F	Program Loader Mode	849
INDEX		863
Index of Pins		875

Main changes in this edition

Page	Changes (For details, refer to main body.)	
27	CHAPTER 3 CPU 3.1 Memory Space	Corrected Figure 3.1-1.
100	CHAPTER 6 Operation Mode	Deleted "6.4.2 Activation Sequence".
	CHAPTER 6 Operation Mode 6.4.2 Details of Each Mode	The following description was added to "■ User mode/external ROM external bus". (In user mode/external ROM external bus, the operation mode is unaffected when the MD pins (MD1,MD0) change after RST is canceled.)
		The following description was added "■ Serial writer mode". (Fix always the MD pins (MD1, MD0) in serial writer mode. If MD pins are changed, the operation mode is affected.)
228	CHAPTER 16 External Bus Interface 16.2 Configuration	Corrected Figure 16.2-2.
254	CHAPTER 16 External Bus Interface 16.4.2 Write Access Examples of SRAM/FLASH Area	Corrected Figure 16.4-3.
256		Corrected Figure 16.4-5.
257		Corrected Figure 16.4-6.
547	CHAPTER 24 Multi-function Serial Interface	Added "Notes on UART Mode" as 24.10.
612		Added "Notes on CSIO Mode" as 24.18.
687		Added "Notes on I ² C Mode" as 24.24.
823	Appendix D	Added "■ Pin state" below summary of "Appendix D".
824 to 829		The following description should be added to the end of table D-1. (P:In the case of port connecting F:In the case of specified function using)

The vertical lines marked in the left side of the page show the changes.

CHAPTER 1 Overview

This chapter explains the features and basic specifications of the MB91605A series.

- 1.1 MB91605A Series Overview
- 1.2 MB91605A Series Product Configuration
- 1.3 MB91605A Series Block Diagram
- 1.4 Package Dimensions

1.1 MB91605A Series Overview

The MB91605A series is a microcontroller that uses 32-bit RISC CPUs, and has a built-in peripheral control functions required for high performance/high speed CPU processing in embedded control functions. The MB91605A series is based on the FR80 family CPUs with enhanced bus access to support use at higher speed.

■ FR80 family CPUs

- 32-bit RISC, load/store architecture, 5-stage pipeline
- 80 MHz operating frequency [Using PLL: Source oscillation 16 MHz: multiplied by 5]
- 16-bit fixed-length (basic instructions), 1 instruction per cycle
- Instructions for memory-to-memory transfer, bit processing, barrel shift, etc.: Instructions suitable for embedded applications
- Function entry/exit instructions and multi-load/store instructions for register contents: High-level language support instructions
- Register interlock function - Simplifies assembler coding
- Support for multipliers at the built-in instruction level
 - Signed 32-bit multiplication: 5 cycles
 - Signed 16-bit multiplication: 3 cycles
- Interrupt (saving PC and PS): 6 cycles, 16 priority levels
- Simultaneous access to program and data enabled by Harvard architecture
- Addition of prefetch function for instruction using the 4-word instruction queue in the CPU
- Instruction level compatibility with FR family

■ Bus interface

- Operating frequency: Settable to 1/1 to 1/4 of on-chip bus frequency
- Basic bus cycle
 - Read : 1 cycle
 - Write: 3 cycles
- Address data multiplex bus support
- Unused address pins may be used as general I/O pins
- May be set to write disabled (except read only area and SDRAM area)
- A programmable auto wait cycle generator for each area (maximum of 15 cycles)
- Area may be set in units at minimum of 1 Mbyte
- Chip select output support for completely separate 8 areas

■ DMAC controller (DMA Controller)

- Number of channels: 4
- Two transfer triggers: Internal peripheral/software
- Addressing mode: 32-bit full address specification (increment/decrement/fixed)
- Transfer mode: Burst transfer/Block transfer
- Transfer data size: Selectable among 1, 2, 4, and 32 bytes

■ 16-bit reload timer (including 1 channel for REALOS)

- Number of channels: 3
- Internal clock: Selectable from clocks divided by 2, 4, 8, 16, 32, and 64

■ Multi-function serial interface

- 8 channels with 16-byte FIFO, 4 channels without FIFO
- For each channel, the usage method is selectable from the following three methods

< UART >

- Full duplex double-buffer
- Selectable with/without parity
- Dedicated baud rate generator built in
- External clock usable as serial clock
- Variety of error detection functions (parity error, framing error, overrun error)

< CSIO >

- Transfer method: Clock synchronous (up to 10 Mbps)
- Full duplex double-buffer
- Dedicated baud rate generator built in
- Overrun error detection function

< I²C >

- Supports standard mode (up to 100Kbps) / high-speed mode (up to 400Kbps)

■ Interrupt controller

- External interrupts: Total of 25 (24 external interrupt pins + 1 $\overline{\text{NMI}}$ pin)
- Interrupt from internal peripheral function
- 16 programmable priority levels except $\overline{\text{NMI}}$ pin
- Available as wake up in STOP mode

■ A/D converter

- Number of channels: 12
- 10-bit resolution
- Successive approximation conversion type: Conversion time: Approximately 8.1 μ s
- Conversion mode: One shot conversion mode, scan conversion mode
- Activation trigger: Software/external trigger

■ Base timer

- Number of channels: 12
- For each channel, the usage method is selectable from the following
 - 16/32-bit reload timer (32-bit timer is used when used as two channel units)
 - 16-bit PWM timer
 - 16/32-bit PWC timer (32-bit timer is used when used as two channel units)
 - 16-bit PPG timer
- 4 channel simultaneous activation mode available

■ HDMI-CEC/ remote control reception

- Number of channels: 1
- HDMI-CEC receive function (with auto ACK response function)
- Remote control reception function (with 4-byte receive buffer)

■ Other interval timers

- Watchdog timer: 1 channel built in

■ I/O ports

- Up to 92 ports

■ Other features

- Built-in oscillating circuit as clock source
- $\overline{\text{INIT}}$ is provided as reset pin
- Watchdog timer reset, software reset available
- Supports stop mode and sleep mode as low power consumption mode
- Gear function
 - Time base timer built in
- Power supply voltage: Two power supplies $3.3\text{V} \pm 0.3\text{V}$ and $1.8\text{V} \pm 0.15\text{V}$

1.2 MB91605A Series Product Configuration

This section describes the MB91605A series product types.

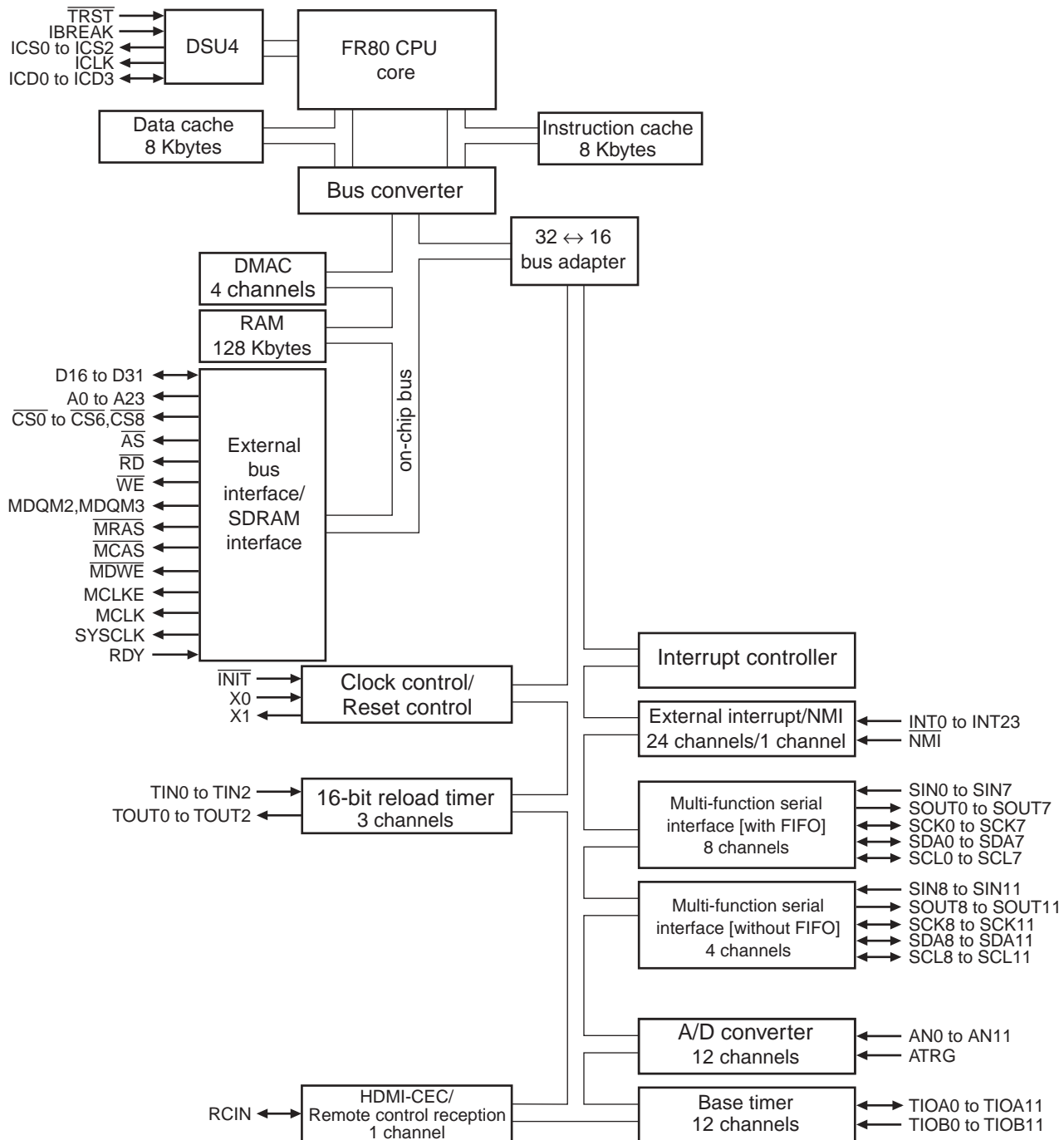
Table 1.2-1 MB91605A Series Product Configuration

Product name	MB91605A
Items	
RAM size	128 Kbytes
Instruction cache size	8 Kbytes
Data cache size	8 Kbytes
Package	Type: LQFP-176 Package code: FPT-176P-M07 Pin pitch: 0.50mm Size: 24.0mm x 24.0mm

1.3 MB91605A Series Block Diagram

Figure 1.3-1 is a block diagram of the MB91605A series.

Figure 1.3-1 MB91605A Series Block Diagram

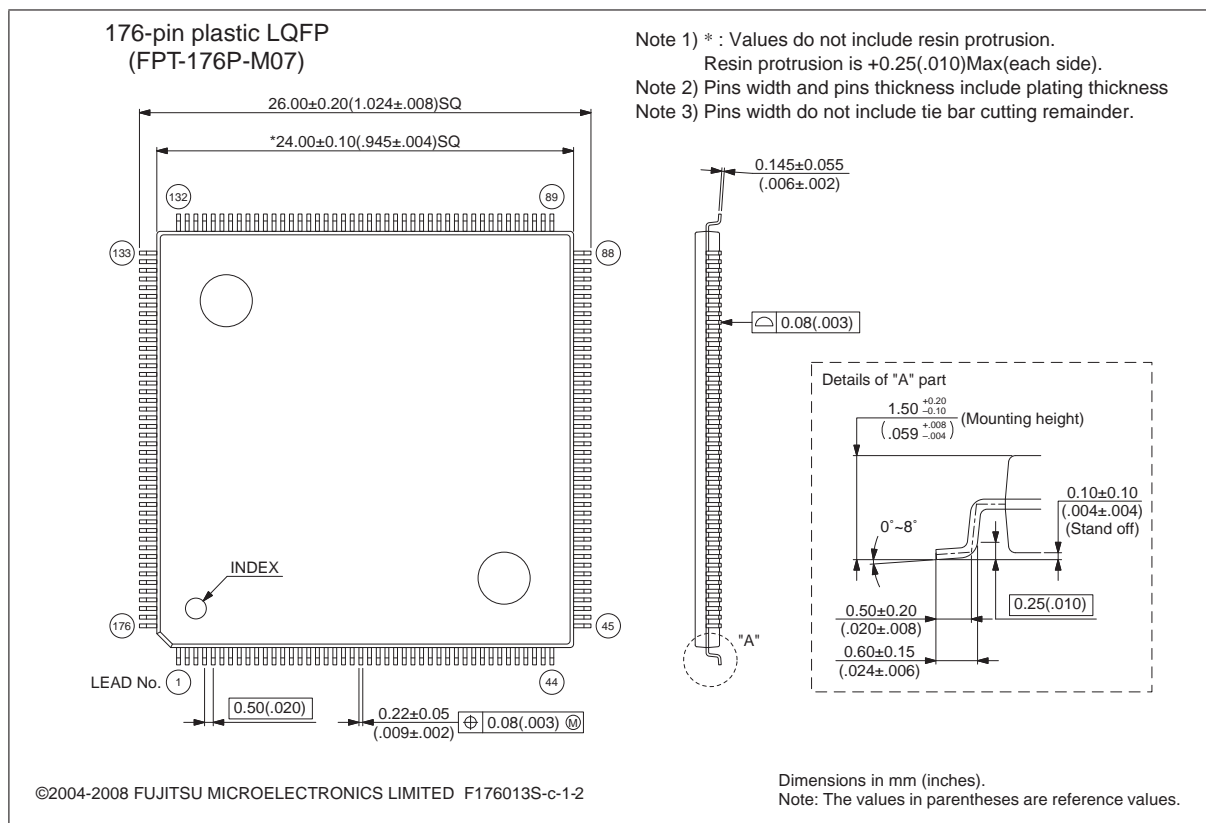


1.4 Package Dimensions

The dimensions of the packages used for the MB91605A Series are shown below.

Figure 1.4-1 Package dimensions (FPT-176P-M07)

<p>176-pin plastic LQFP</p> <p>(FPT-176P-M07)</p>	Lead pitch	0.50 mm
	Package width × package length	24.0 × 24.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Code (Reference)	P-LQFP-0176-2424-0.50



Please confirm the latest Package dimension by following URL.

<http://edevic.fujitsu.com/package/en-search/>

CHAPTER 2 Pins of the MB91605A Series

This chapter explains the pins and multiplexed pin settings of the MB91605A series.

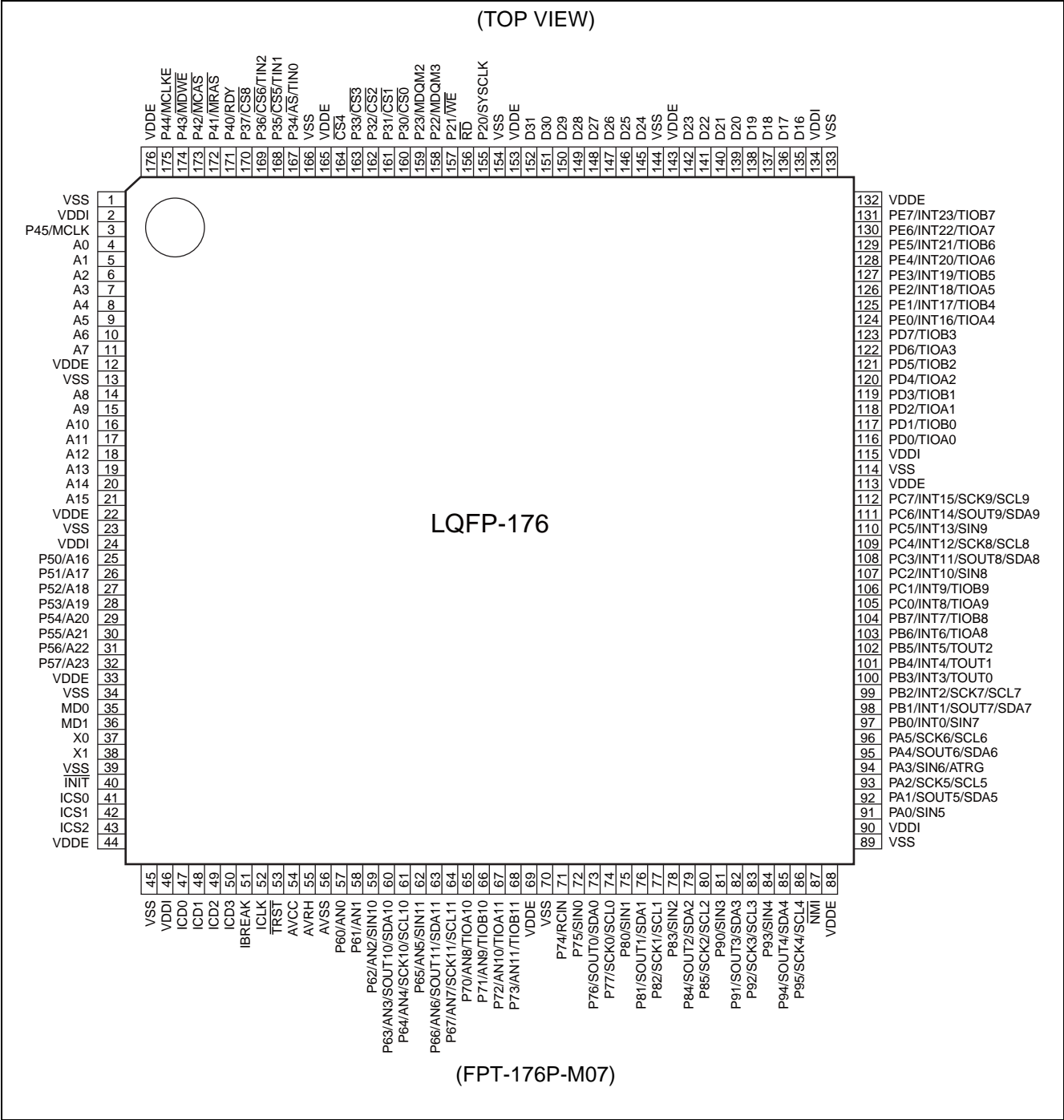
- 2.1 Pin Assignment Diagram
- 2.2 Pin Functions
- 2.3 I/O Circuit Types

2.1 Pin Assignment Diagram

1 type of package are available for the MB91605A series.

■ LQFP-176

Figure 2.1-1 Pin assignment diagram in the LQFP-176 series



2.2 Pin Functions

Table 2.2-1 lists the pin functions of the MB91605A series.

■ Pin function list

Table 2.2-1 Pin functions (1 / 9)

Pin no.	Pin name	I/O circuit type	Description
3	P45	A	General-purpose I/O port
	MCLK		Clock output pin for external bus interface
4 to 11	A0 to A7	B	Address bus output pins for external bus interface (bit0 to bit7)
14 to 21	A8 to A15	B	Address bus output pins for external bus interface (bit8 to bit15)
25 to 32	P50 to P57	A	General-purpose I/O ports
	A16 to A23		Address bus output pins for external bus interface (bit16 to bit23)
35, 36	MD0, MD1	C	Mode setting pins
37	X0	D	Clock (oscillator) input pin
38	X1	D	Clock (oscillator) I/O pin
40	$\overline{\text{INIT}}$	E	External reset input pin
41 to 43	ICS0 to ICS2	B	Development tool status output pins
47 to 50	ICD0 to ICD3	F	Development tool data I/O pins
51	IBREAK	G	Development tool break input pin
52	ICLK	B	Development tool clock output pin
53	$\overline{\text{TRST}}$	E	Development tool reset input pin
57	P60	H	General-purpose I/O port
	AN0		A/D converter ch.0 analog input pin
58	P61	H	General-purpose I/O port
	AN1		A/D converter ch.1 analog input pin
59	P62	H	General-purpose I/O port
	AN2		A/D converter ch.2 analog input pin
	SIN10		Multi-function serial interface ch.10 serial data input pin

Table 2.2-1 Pin functions (2 / 9)

Pin no.	Pin name	I/O circuit type	Description
60	P63	H	General-purpose I/O port
	AN3		A/D converter ch.3 analog input pin
	SOUT10		Multi-function serial interface ch.10 serial data output pin
	SDA10		Multi-function serial interface ch.10 I ² C data I/O pin
61	P64	H	General-purpose I/O port
	AN4		A/D converter ch.4 analog input pin
	SCK10		Multi-function serial interface ch.10 serial clock I/O pin
	SCL10		Multi-function serial interface ch.10 I ² C clock I/O pin
62	P65	H	General-purpose I/O port
	AN5		A/D converter ch.5 analog input pin
	SIN11		Multi-function serial interface ch.11 serial data input pin
63	P66	H	General-purpose I/O port
	AN6		A/D converter ch.6 analog input pin
	SOUT11		Multi-function serial interface ch.11 serial data output pin
	SDA11		Multi-function serial interface ch.11 I ² C data I/O pin
64	P67	H	General-purpose I/O port
	AN7		A/D converter ch.7 analog input pin
	SCK11		Multi-function serial interface ch.11 serial clock I/O pin
	SCL11		Multi-function serial interface ch.11 I ² C clock I/O pin
65	P70	H	General-purpose I/O port
	AN8		A/D converter ch.8 analog input pin
	TIOA10		Base timer ch.10 timer I/O pin
66	P71	H	General-purpose I/O port
	AN9		A/D converter ch.9 analog input pin
	TIOB10		Base timer ch.10 timer input pin
67	P72	H	General-purpose I/O port
	AN10		A/D converter ch.10 analog input pin
	TIOA11		Base timer ch.11 timer I/O pin

Table 2.2-1 Pin functions (3 / 9)

Pin no.	Pin name	I/O circuit type	Description
68	P73	H	General-purpose I/O port
	AN11		A/D converter ch.11 analog input pin
	TIOB11		Base timer ch.11 timer input pin
71	P74	A	General-purpose I/O port
	RCIN		HDMI-CEC/Remote control I/O pin
72	P75	A	General-purpose I/O port
	SIN0		Multi-function serial interface ch.0 serial data input pin
73	P76	A	General-purpose I/O port
	SOUT0		Multi-function serial interface ch.0 serial data output pin
	SDA0		Multi-function serial interface ch.0 I ² C data I/O pin
74	P77	A	General-purpose I/O port
	SCK0		Multi-function serial interface ch.0 serial clock I/O pin
	SCL0		Multi-function serial interface ch.0 I ² C clock I/O pin
75	P80	A	General-purpose I/O port
	SIN1		Multi-function serial interface ch.1 serial data input pin
76	P81	A	General-purpose I/O port
	SOUT1		Multi-function serial interface ch.1 serial data output pin
	SDA1		Multi-function serial interface ch.1 I ² C data I/O pin
77	P82	A	General-purpose I/O port
	SCK1		Multi-function serial interface ch.1 serial clock I/O pin
	SCL1		Multi-function serial interface ch.1 I ² C clock I/O pin
78	P83	A	General-purpose I/O port
	SIN2		Multi-function serial interface ch.2 serial data input pin
79	P84	A	General-purpose I/O port
	SOUT2		Multi-function serial interface ch.2 serial data output pin
	SDA2		Multi-function serial interface ch.2 I ² C data I/O pin
80	P85	A	General-purpose I/O port
	SCK2		Multi-function serial interface ch.2 serial clock I/O pin
	SCL2		Multi-function serial interface ch.2 I ² C clock I/O pin

Table 2.2-1 Pin functions (4 / 9)

Pin no.	Pin name	I/O circuit type	Description
81	P90	A	General-purpose I/O port
	SIN3		Multi-function serial interface ch.3 serial data input pin
82	P91	A	General-purpose I/O port
	SOUT3		Multi-function serial interface ch.3 serial data output pin
	SDA3		Multi-function serial interface ch.3 I ² C data I/O pin
83	P92	A	General-purpose I/O port
	SCK3		Multi-function serial interface ch.3 serial clock I/O pin
	SCL3		Multi-function serial interface ch.3 I ² C clock I/O pin
84	P93	A	General-purpose I/O port
	SIN4		Multi-function serial interface ch.4 serial data input pin
85	P94	A	General-purpose I/O port
	SOUT4		Multi-function serial interface ch.4 serial data output pin
	SDA4		Multi-function serial interface ch.4 I ² C data I/O pin
86	P95	A	General-purpose I/O port
	SCK4		Multi-function serial interface ch.4 serial clock I/O pin
	SCL4		Multi-function serial interface ch.4 I ² C clock I/O pin
87	NMI	E	NMI input pin
91	PA0	A	General-purpose I/O port
	SIN5		Multi-function serial interface ch.5 serial data input pin
92	PA1	A	General-purpose I/O port
	SOUT5		Multi-function serial interface ch.5 serial data output pin
	SDA5		Multi-function serial interface ch.5 I ² C data I/O pin
93	PA2	A	General-purpose I/O port
	SCK5		Multi-function serial interface ch.5 serial clock I/O pin
	SCL5		Multi-function serial interface ch.5 I ² C clock I/O pin
94	PA3	A	General-purpose I/O port
	SIN6		Multi-function serial interface ch.6 serial data input pin
	ATRG		A/D converter external trigger input pin

Table 2.2-1 Pin functions (5 / 9)

Pin no.	Pin name	I/O circuit type	Description
95	PA4	A	General-purpose I/O port
	SOUT6		Multi-function serial interface ch.6 serial data output pin
	SDA6		Multi-function serial interface ch.6 I ² C data I/O pin
96	PA5	A	General-purpose I/O port
	SCK6		Multi-function serial interface ch.6 serial clock I/O pin
	SCL6		Multi-function serial interface ch.6 I ² C clock I/O pin
97	PB0	I	General-purpose I/O port
	INT0		External interrupt input pin
	SIN7		Multi-function serial interface ch.7 serial data input pin
98	PB1	I	General-purpose I/O port
	INT1		External interrupt input pin
	SOUT7		Multi-function serial interface ch.7 serial data output pin
	SDA7		Multi-function serial interface ch.7 I ² C data I/O pin
99	PB2	I	General-purpose I/O port
	INT2		External interrupt input pin
	SCK7		Multi-function serial interface ch.7 serial clock I/O pin
	SCL7		Multi-function serial interface ch.7 I ² C clock I/O pin
100 to 102	PB3 to PB5	I	General-purpose I/O ports
	INT3 to INT5		External interrupt input pins
	TOUT0 to TOUT2		16-bit reload timer ch.0 to ch.2 output pins
103	PB6	I	General-purpose I/O port
	INT6		External interrupt input pin
	TIOA8		Base timer ch.8 timer output pin
104	PB7	I	General-purpose I/O port
	INT7		External interrupt input pin
	TIOB8		Base timer ch.8 timer input pin
105	PC0	I	General-purpose I/O port
	INT8		External interrupt input pin
	TIOA9		Base timer ch.9 timer I/O pin

Table 2.2-1 Pin functions (6 / 9)

Pin no.	Pin name	I/O circuit type	Description
106	PC1	I	General-purpose I/O port
	INT9		External interrupt input pin
	TIOB9		Base timer ch.9 timer input pin
107	PC2	I	General-purpose I/O port
	INT10		External interrupt input pin
	SIN8		Multi-function serial interface ch.8 serial data input pin
108	PC3	I	General-purpose I/O port
	INT11		External interrupt input pin
	SOUT8		Multi-function serial interface ch.8 serial data output pin
	SDA8		Multi-function serial interface ch.8 I ² C data I/O pin
109	PC4	I	General-purpose I/O port
	INT12		External interrupt input pin
	SCK8		Multi-function serial interface ch.8 serial clock I/O pin
	SCL8		Multi-function serial interface ch.8 I ² C clock I/O pin
110	PC5	I	General-purpose I/O port
	INT13		External interrupt input pin
	SIN9		Multi-function serial interface ch.9 serial data input pin
111	PC6	I	General-purpose I/O port
	INT14		External interrupt input pin
	SOUT9		Multi-function serial interface ch.9 serial data output pin
	SDA9		Multi-function serial interface ch.9 I ² C data I/O pin
112	PC7	I	General-purpose I/O port
	INT15		External interrupt input pin
	SCK9		Multi-function serial interface ch.9 serial clock I/O pin
	SCL9		Multi-function serial interface ch.9 I ² C clock I/O pin
116	PD0	A	General-purpose I/O port
	TIOA0		Base timer ch.0 timer output pin
117	PD1	A	General-purpose I/O port
	TIOB0		Base timer ch.0 timer input pin

Table 2.2-1 Pin functions (7 / 9)

Pin no.	Pin name	I/O circuit type	Description
118	PD2	A	General-purpose I/O port
	TIOA1		Base timer ch.1 timer I/O pin
119	PD3	A	General-purpose I/O port
	TIOB1		Base timer ch.1 timer input pin
120	PD4	A	General-purpose I/O port
	TIOA2		Base timer ch.2 timer output pin
121	PD5	A	General-purpose I/O port
	TIOB2		Base timer ch.2 timer input pin
122	PD6	A	General-purpose I/O port
	TIOA3		Base timer ch.3 timer I/O pin
123	PD7	A	General-purpose I/O port
	TIOB3		Base timer ch.3 timer input pin
124	PE0	I	General-purpose I/O port
	INT16		External interrupt input pin
	TIOA4		Base timer ch.4 timer output pin
125	PE1	I	General-purpose I/O port
	INT17		External interrupt input pin
	TIOB4		Base timer ch.4 timer input pin
126	PE2	I	General-purpose I/O port
	INT18		External interrupt input pin
	TIOA5		Base timer ch.5 timer I/O pin
127	PE3	I	General-purpose I/O port
	INT19		External interrupt input pin
	TIOB5		Base timer ch.5 timer input pin
128	PE4	I	General-purpose I/O port
	INT20		External interrupt input pin
	TIOA6		Base timer ch.6 timer output pin
129	PE5	I	General-purpose I/O port
	INT21		External interrupt input pin
	TIOB6		Base timer ch.6 timer input pin

Table 2.2-1 Pin functions (8 / 9)

Pin no.	Pin name	I/O circuit type	Description
130	PE6	I	General-purpose I/O port
	INT22		External interrupt input pin
	TIOA7		Base timer ch.7 timer I/O pin
131	PE7	I	General-purpose I/O port
	INT23		External interrupt input pin
	TIOB7		Base timer ch.7 timer input pin
135 to 142	D16 to D23	J	Data bus I/O pins for external bus interface (bit16 to bit23)
145 to 152	D24 to D31	J	Data bus I/O pins for external bus interface (bit24 to bit31)
155	P20	A	General-purpose I/O port
	SYSCLK		System clock output pin
156	\overline{RD}	B	External bus interface read strobe output pin
157	P21	A	General-purpose I/O port
	\overline{WE}		External bus interface write strobe output pin
158, 159	P22, P23	A	General-purpose I/O ports
	MDQM3, MDQM2		External bus interface byte enable output pins MDQM3:D[31:24], MDQM2:D[23:16]
160 to 163	P30 to P33	A	General-purpose I/O ports
	$\overline{CS0}$ to $\overline{CS3}$		External bus interface chip select output pins
164	$\overline{CS4}$	B	External bus interface chip select output pin
167	P34	A	General-purpose I/O port
	\overline{AS}		External bus interface address strobe output pin
	TIN0		16-bit reload timer ch.0 input pin
168, 169	P35, P36	A	General-purpose I/O ports
	$\overline{CS5}$, $\overline{CS6}$		External bus interface chip select output pins
	TIN1, TIN2		16-bit reload timer ch.1, ch.2 input pins
170	P37	A	General-purpose I/O port
	$\overline{CS8}$		SDRAM interface chip select output pin
171	P40	A	General-purpose I/O port
	RDY		External bus interface ready input pin
172	P41	A	General-purpose I/O port
	\overline{MRAS}		RAS strobe output pin for SDRAM interface

Table 2.2-1 Pin functions (9 / 9)

Pin no.	Pin name	I/O circuit type	Description
173	P42	A	General-purpose I/O port
	$\overline{\text{MCAS}}$		CAS strobe output pin for SDRAM interface
174	P43	A	General-purpose I/O port
	$\overline{\text{MDWE}}$		Write strobe output pin for SDRAM interface
175	P44	A	General-purpose I/O port
	MCLKE		Clock enable output pin for SDRAM interface
12, 22, 33, 44, 69, 88, 113, 132, 143, 153, 165, 176	VDDE	—	3.3 V power supply pins
2, 24, 46, 90, 115, 134	VDDI	—	1.8 V power supply pins
1, 13, 23, 34, 39, 45, 70, 89, 114, 133, 144, 154, 166	VSS	—	GND pins
54	AVCC	—	A/D converter power supply pin
56	AVSS	—	A/D converter analog GND pin
55	AVRH	—	A/D converter reference voltage pin

2.3 I/O Circuit Types

Table 2.3-1 lists the I/O circuit types for the MB91605A series.

■ I/O circuit types

Table 2.3-1 I/O circuit types (1 / 4)

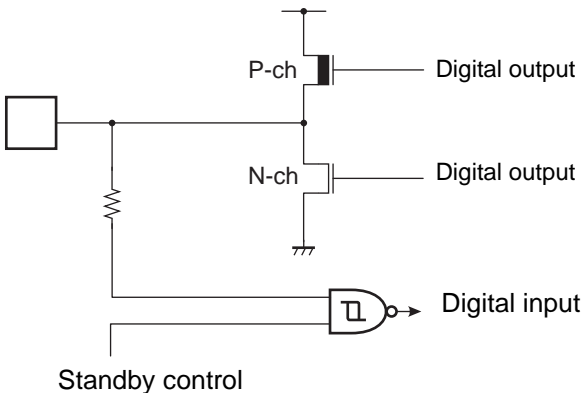
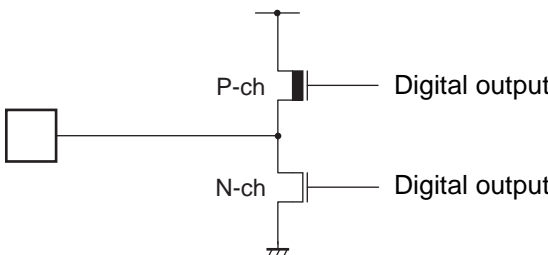
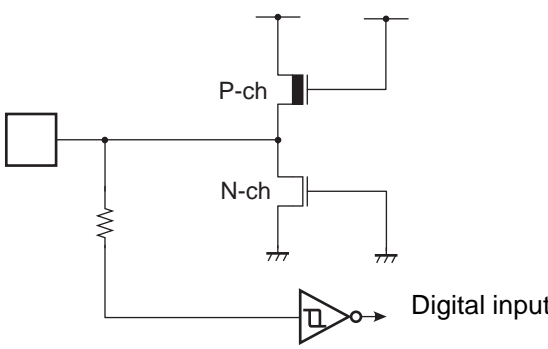
Type	Circuit type	Remarks
A		<ul style="list-style-type: none">• CMOS level output• CMOS level hysteresis input With standby control
B		CMOS level output
C		CMOS level hysteresis input

Table 2.3-1 I/O circuit types (2 / 4)

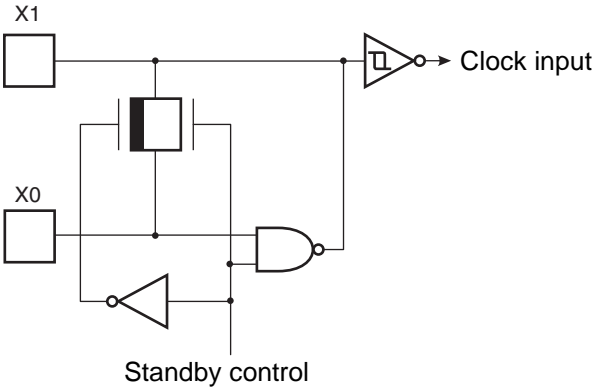
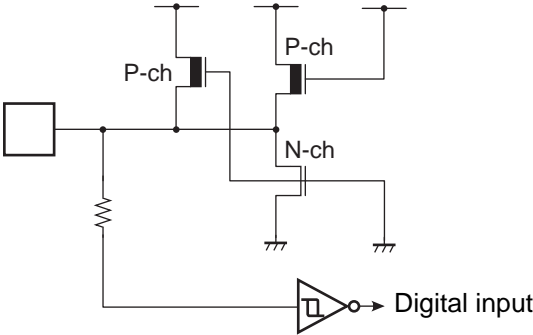
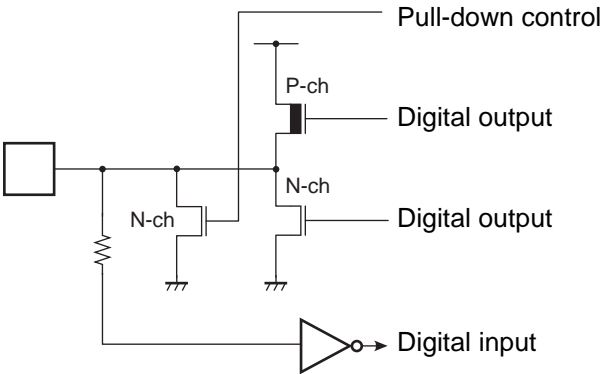
Type	Circuit type	Remarks
D		<p>Oscillation feedback resistor : approx. 1 MΩ (built-in)</p>
E		<ul style="list-style-type: none"> With pull-up resistor CMOS hysteresis input Pull-up resistor value = approx. 33 kΩ (Typ)
F		<ul style="list-style-type: none"> CMOS input/output With pull-down control

Table 2.3-1 I/O circuit types (3 / 4)

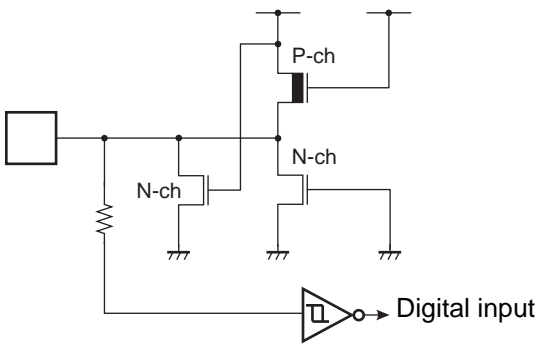
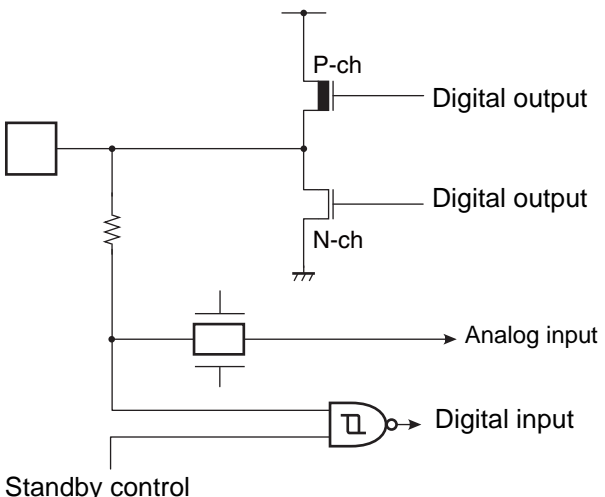
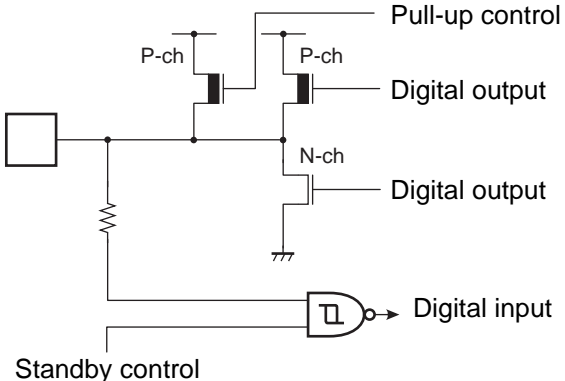
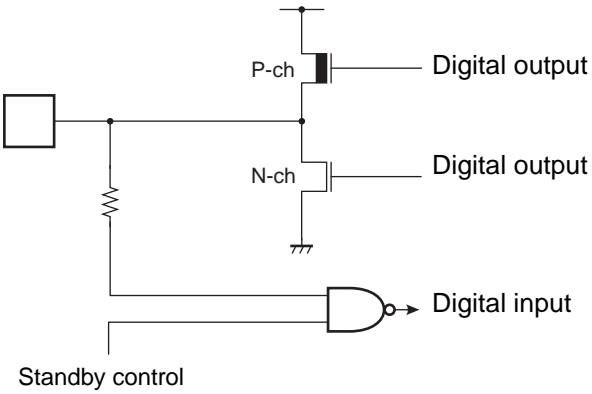
Type	Circuit type	Remarks
G		<ul style="list-style-type: none">CMOS hysteresis inputWith pull-down resistor
H		<ul style="list-style-type: none">CMOS level outputCMOS level hysteresis inputWith standby controlWith analog input
I		<ul style="list-style-type: none">With pull-up controlPull-up resistor value = approx. 33 kΩ(Typ)CMOS level outputCMOS level hysteresis inputWith standby control

Table 2.3-1 I/O circuit types (4 / 4)

Type	Circuit type	Remarks
J		<ul style="list-style-type: none">• CMOS level input/output• With standby control

CHAPTER 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.

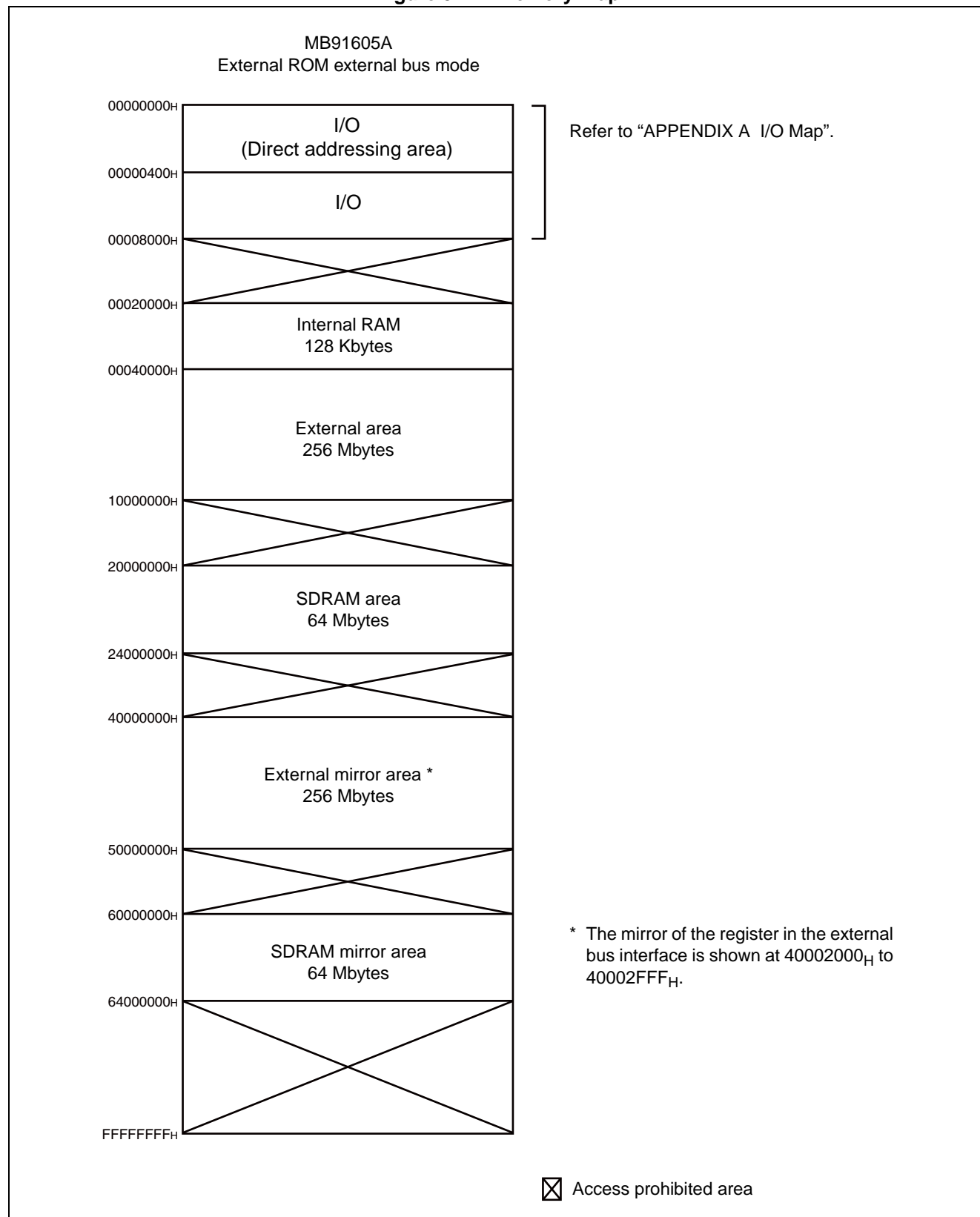
The direct addressing areas vary as follows depending on the size of the data accessed:

- 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 3.1-1 shows a memory map of the MB91605A series.

Figure 3.1-1 Memory map



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 \times 32-bit multiplication: 5 cycles
 - 16-bit \times 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 mode / external ROM external bus
- Serial writer mode

Table 3.3-1 lists the operation modes of this series.

Table 3.3-1 Operation modes

MD Pin		Operation Mode
MD1	MD0	
0	0	User mode/external ROM external bus
	1	Serial writer 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.

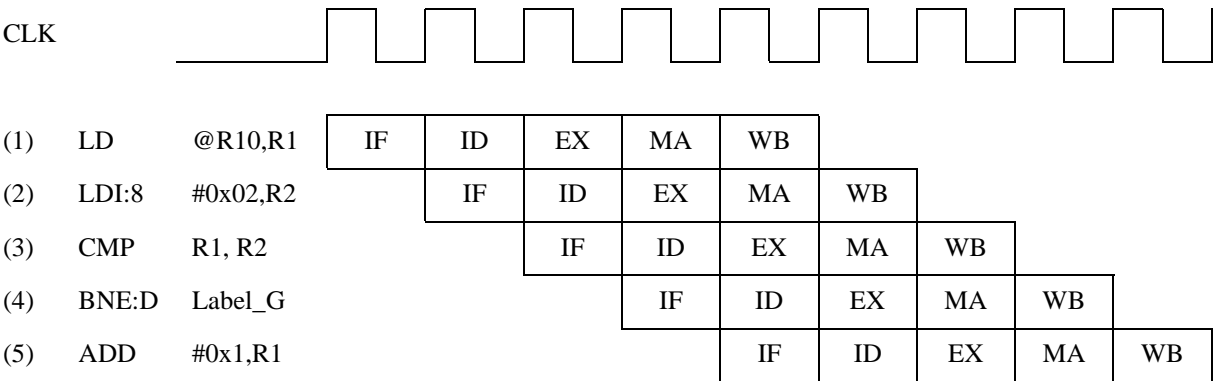
A five-stage instruction pipeline method is used in executing 1 instruction per cycle. The pipeline consists of the following stages:

- 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:

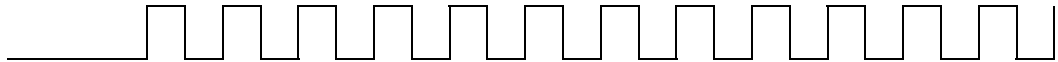


Example 1: The instructions are executed in sequence because the data that uses R1 to write the (1) LD instruction is returned in the (3) CMP instruction within 1 cycle.

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:

CLK



(1)	LD	@R10,R1	IF	ID	EX	MA	MA	MA	WB										
(2)	LDI:8	#0x02,R2		IF	ID	EX	MA	WB											
(3)	CMP	R1, R2			IF	ID	ID	ID	EX	MA	WB								
(4)	BNE:D	Label_G						IF	ID	EX	MA	WB							
(5)	ADD	#0x1,R1							IF	ID	EX	MA	WB						

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 \times 32-bit multiplication instruction, 16-bit \times 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.

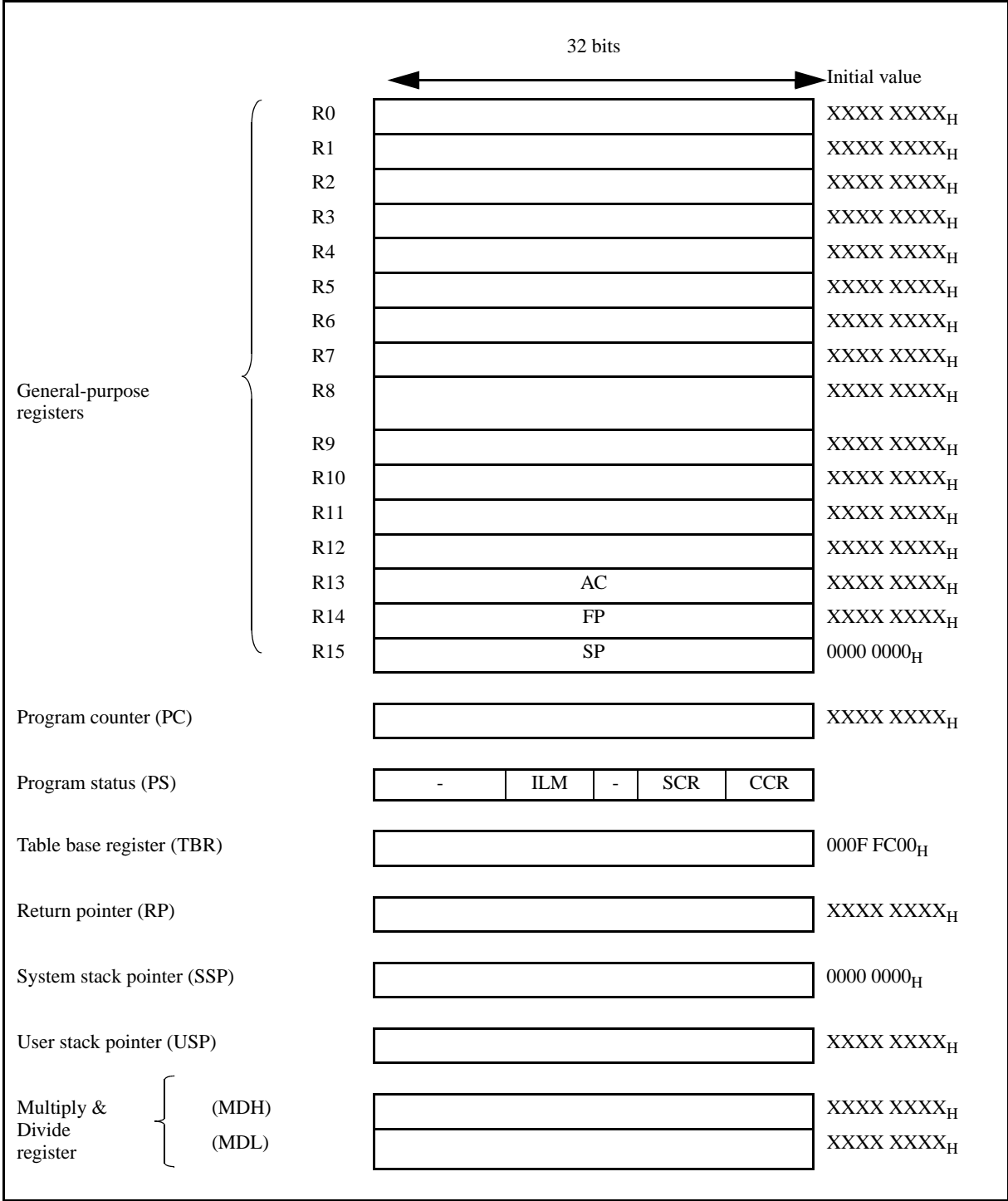
3.5.7 Other

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 3.6-1 shows the basic programming model.

Figure 3.6-1 Basic programming model



3.7 Registers

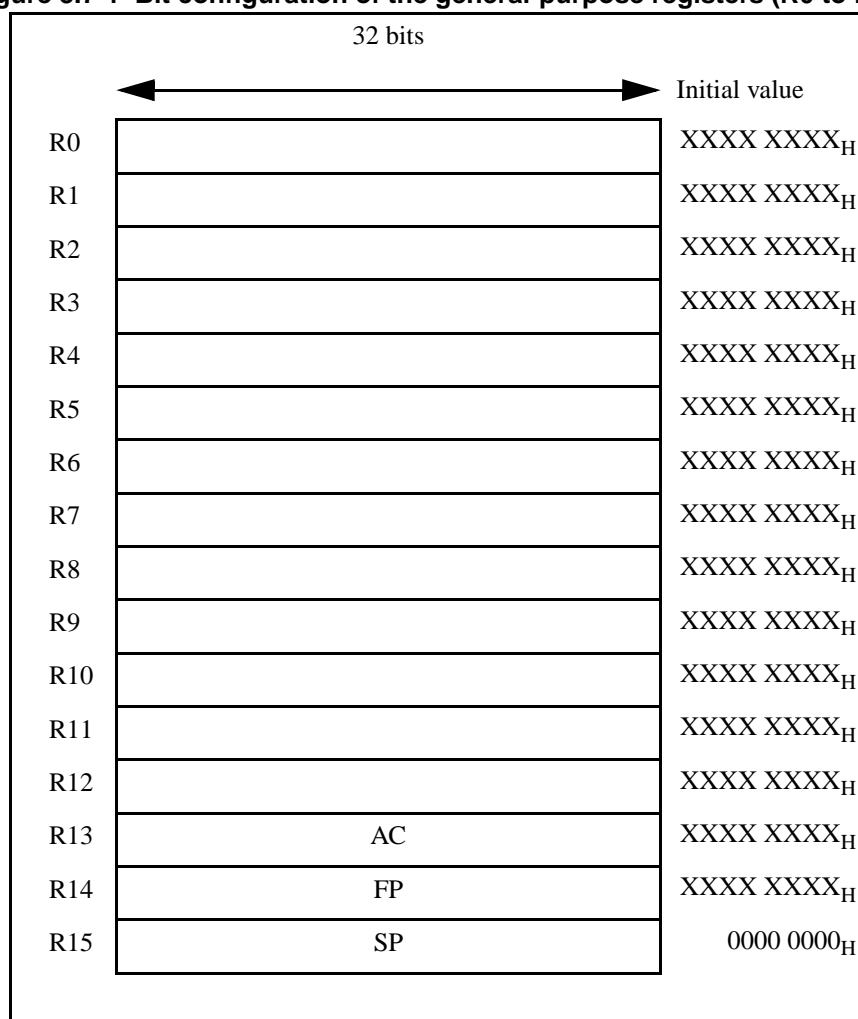
The register configuration consists of general-purpose registers and dedicated registers for specific purposes.

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 3.7-1 shows the bit configuration of the general-purpose registers (R0 to R15).

Figure 3.7-1 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 Figure 3.7-1.

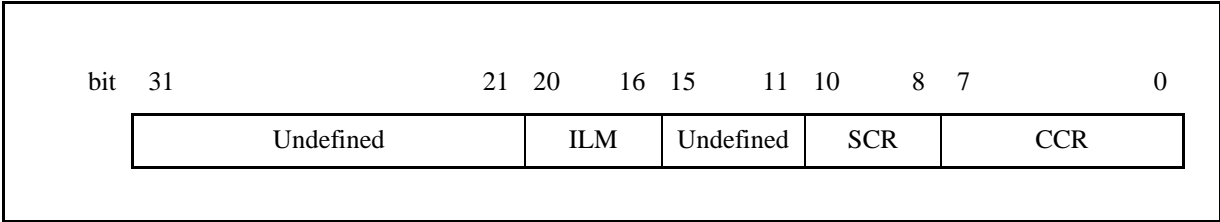
- R13: Virtual accumulator (AC)
- R14: Frame pointer (FP)
- R15: Stack pointer (SP)

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 3.7-2 shows the bit configuration of the program status register (PS).

Figure 3.7-2 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 3.7-3 shows the bit configuration of the condition code register (CCR).

Figure 3.7-3 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 3.7-4 shows the bit configuration of the system condition register (SCR).

Figure 3.7-4 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 3.7-5 shows the bit configuration of the interrupt level mask register (ILM).

Figure 3.7-5 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.

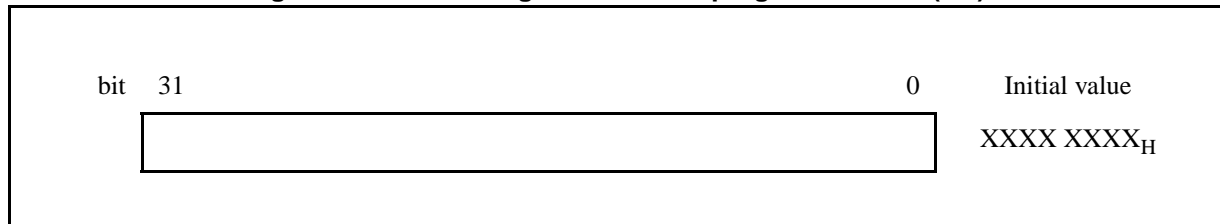
These bits are initialized to "15" (01111_B) by a reset.

3.7.3 Program Counter (PC)

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

Figure 3.7-6 shows the bit configuration of the program counter (PC).

Figure 3.7-6 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.

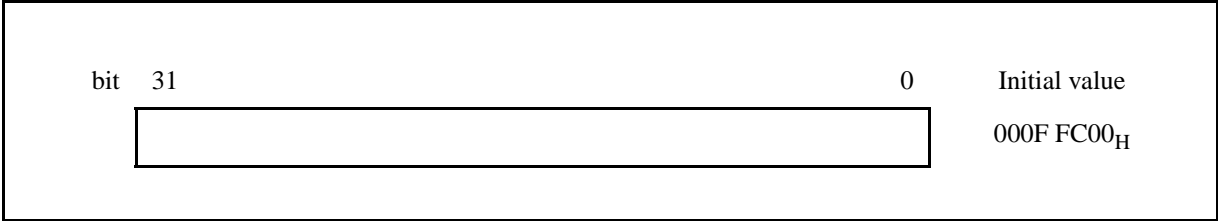
The initial value following a reset is undefined, and the program start address is set by a reset vector fetch.

3.7.4 Table Base Register (TBR)

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

Figure 3.7-7 shows the bit configuration of the table base register (TBR).

Figure 3.7-7 Bit configuration of the table base register (TBR)



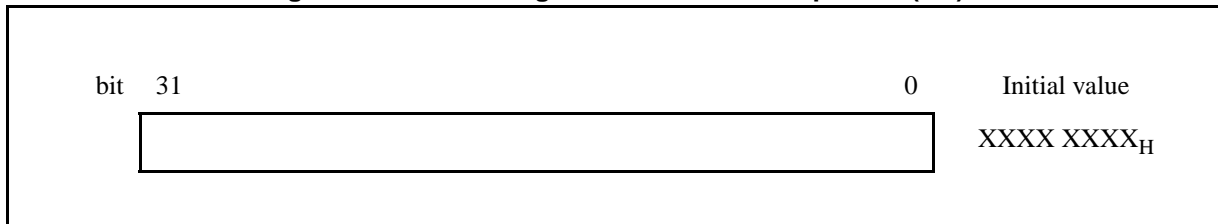
The initial value following a reset is "000F FC00_H".

3.7.5 Return Pointer (RP)

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

Figure 3.7-8 shows the bit configuration of the return pointer (RP).

Figure 3.7-8 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.

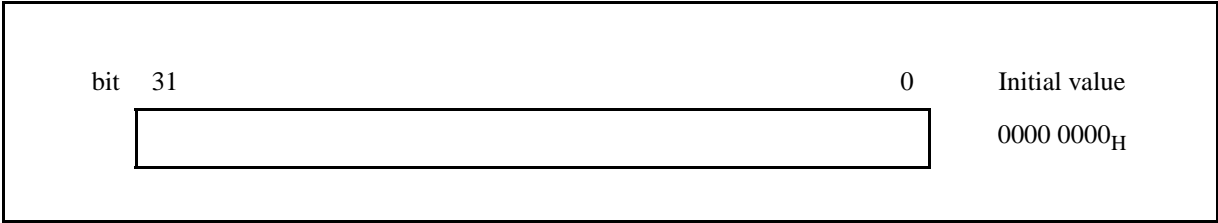
The register contents are transferred to the program counter (PC) when the RET instruction is executed.

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 3.7-9 shows the bit configuration of the system stack pointer (SSP).

Figure 3.7-9 Bit configuration of the system stack pointer (SSP)



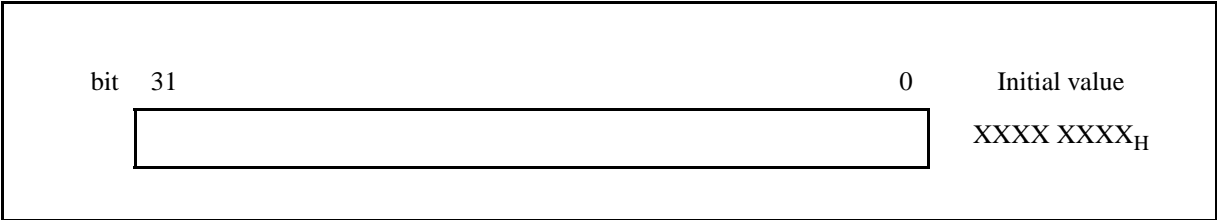
The initial value following a reset is "0000 0000_H".

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 3.7-10 shows the bit configuration of the user stack pointer (USP).

Figure 3.7-10 Bit configuration of the user stack pointer (USP)

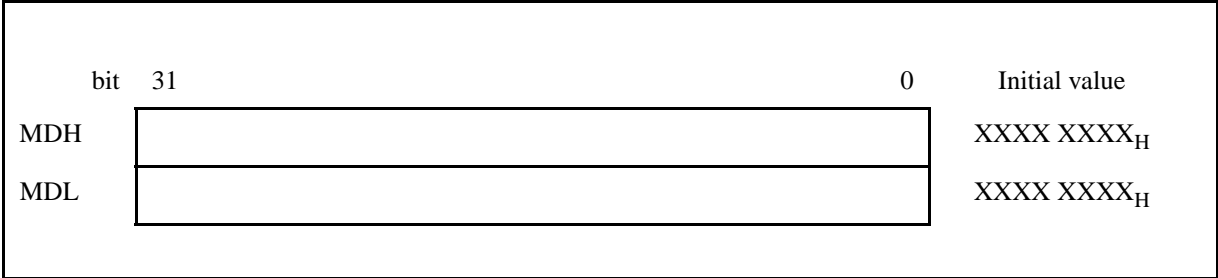


The initial value following a reset is undefined.
This pointer cannot be used in the RETI instruction.

3.7.8 Multiply & Divide Registers

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

Figure 3.7-11 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:

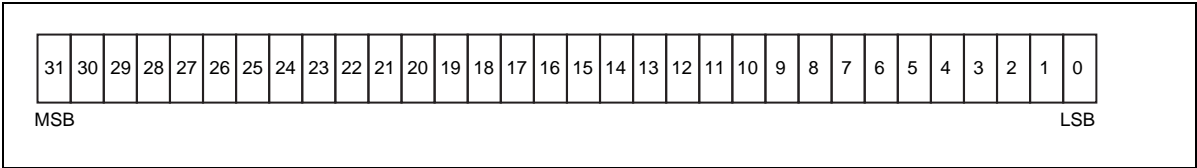
- Bit Ordering
- Byte Ordering

3.8.1 Bit Ordering

The FR80 family CPUs use little endian for bit ordering.

Figure 3.8-1 shows the bit ordering.

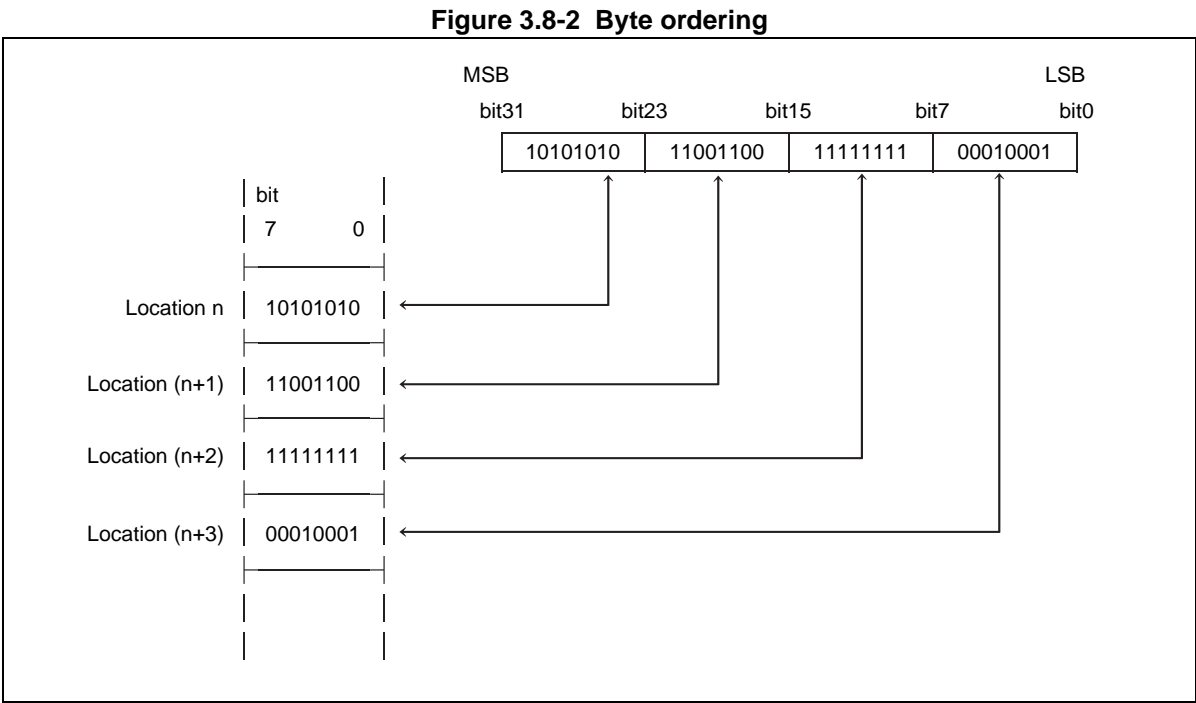
Figure 3.8-1 Bit ordering



3.8.2 Byte Ordering

The FR80 family CPUs use big endian for byte ordering.

Figure 3.8-2 shows the 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").

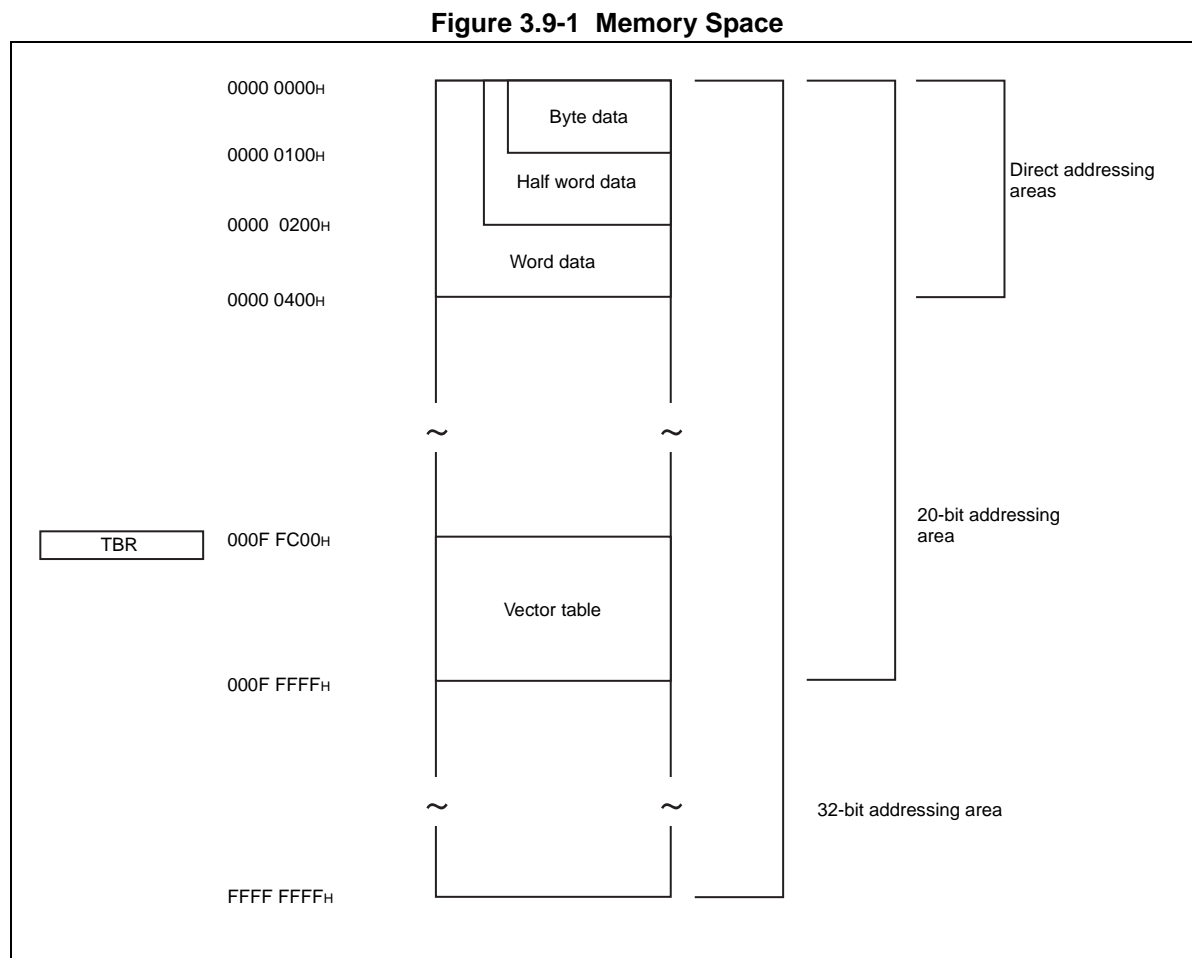
Byte access: ---

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 3.9-1 shows the 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
		<hr/>
		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
		<hr/>
		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/NMI

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

■ Undefined instruction exception

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.

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 reexecuted 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 reexecuted 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)
- NMI
- 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 3.11-1 lists the assignment of each level.

Table 3.11-1 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)	When the interrupt level mask register (ILM) is set, user interrupts must be disabled.
01101	15	NMI (for users)	
10000	16	Interrupt request	
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

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
↓	STILM #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/NMI

When an interrupt or NMI 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 ≥ Level mask value

3.11.7 Interrupt Control Register (ICR)

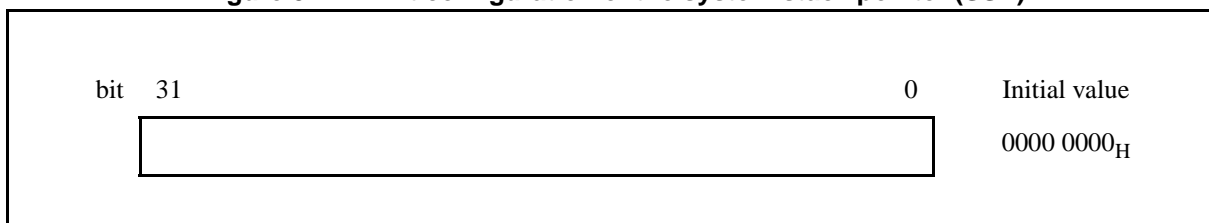
See "CHAPTER 12 Interrupt Controller".

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 3.11-1 shows the bit configuration of the system stack pointer (SSP).

Figure 3.11-1 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".

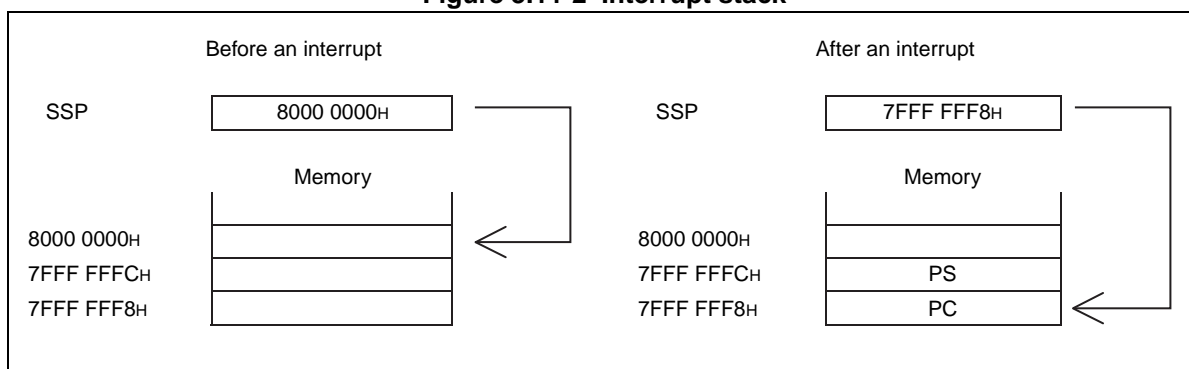
This pointer operates as general-purpose register R15 when the S flag of the condition code register (CCR) is "0".

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 3.11-2 shows the interrupt stack.

Figure 3.11-2 Interrupt stack

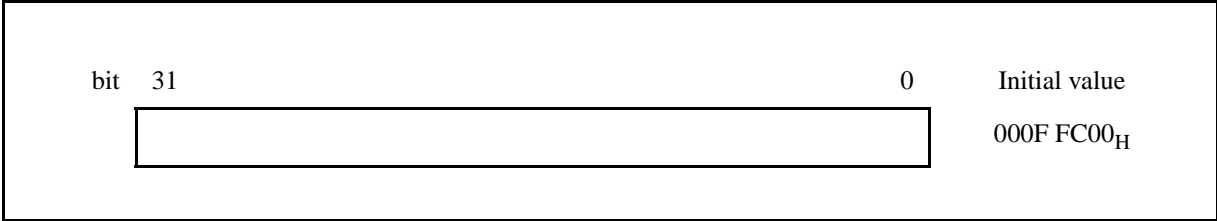


3.11.10 Table Base Register (TBR)

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

Figure 3.11-3 shows the bit configuration of the table base register (TBR).

Figure 3.11-3 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:

$$\begin{aligned} \text{vctadr} &= \text{TBR} + \text{vctofs} \\ &= \text{TBR} + (0\text{x}3\text{FC} - 4 \times \text{vct}) \end{aligned}$$

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 3.11-2 outlines the priority in which EIT sources are accepted.

Table 3.11-2 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	NMI (for users)	ILM = 15	15
7	NMI (for an emulator)	All sources of lower priority	4
8	Step trace trap	ILM=15	4

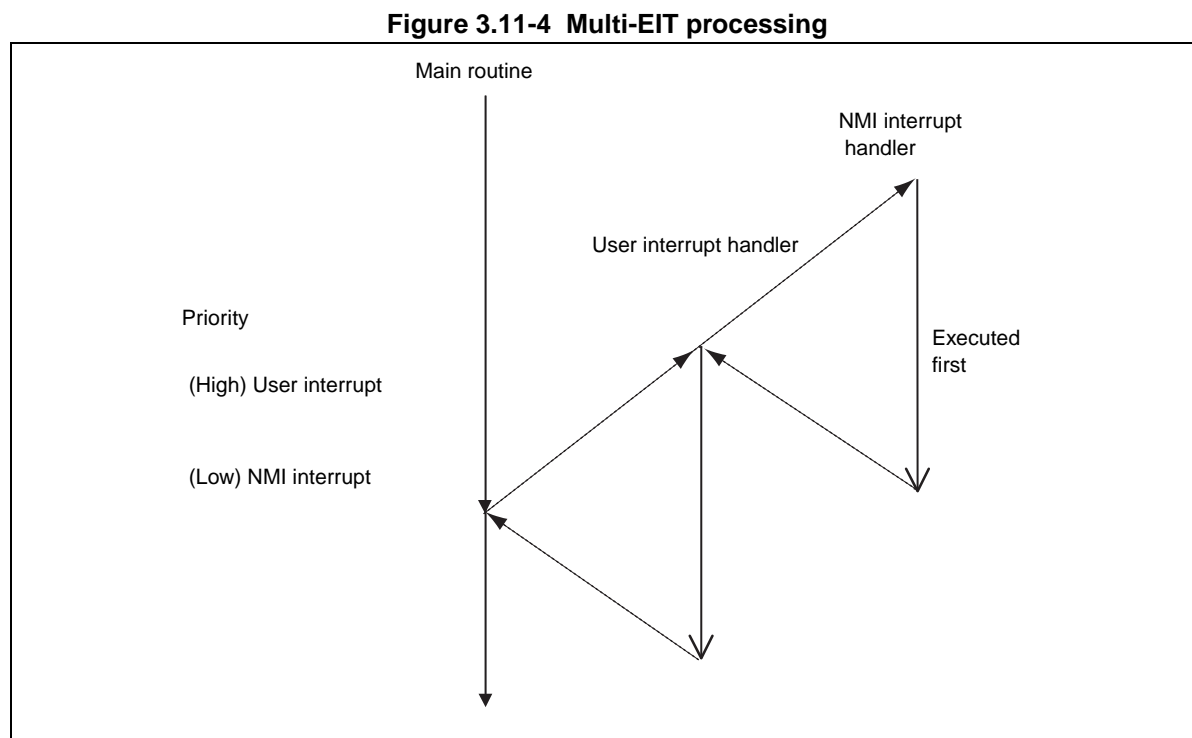
With additional consideration given to the masking of other sources after an EIT source is accepted, the sequence of execution of the handlers of EIT sources that occur at one time is as shown below.

Table 3.11-3 lists the sequence of execution.

Table 3.11-3 Sequence of EIT handler execution

Priority of Acceptance	Source
1	Reset
2	Other than undefined instructions
3	INTE instruction
4	Step trace trap
5	NMI (for users)
6	INT instruction
7	User interrupt

Figure 3.11-4 shows 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 and NMI operation

The sequence in which a generated user or user NMI 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 levels used for comparison are as shown below:
 - The value retained by the corresponding interrupt control register (ICR)
 - User NMI: a predetermined constant number
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.

When the selected interrupt request is NMI, the interrupt request is accepted regardless of the I flag value.

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 or user NMI request is accepted, the CPU operates as follows using the interrupt number corresponding to the accepted interrupt request.

* 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"	→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 and user NMI 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.

CHAPTER 4 Instruction Cache Memory

This chapter describes the functions and operations of the instruction cache memory.

- 4.1 Overview
- 4.2 Configuration
- 4.3 Registers
- 4.4 Explanation of Operations

4.1 Overview

This type of product has an instruction cache memory with the following configuration:

- Cache size: 8 Kbytes (4 Kbytes per way)
 - Mapping method: 2-way set associative
 - Line size: 16 bytes (8 instructions)
 - Total number of entries: 512 entries (256 lines per way)
-

■ Overview

The cache is filled in single line unit and a 16-byte worth of read operation is performed per miss-access.

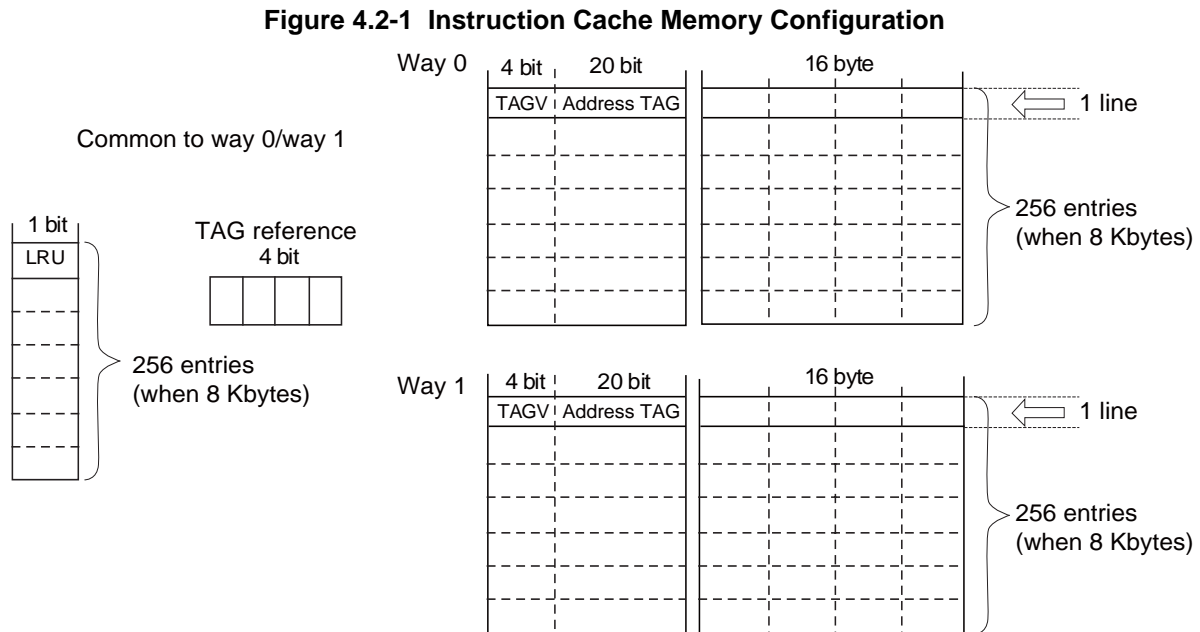
Writing to the control bit can invalidate the entire entry.

With this type of product, an instruction entered in cache can be locked in way units (no entry unit lock function is available).

The memory area that can be cached can be set per specific area using the CARR register also used by the data cache.

4.2 Configuration

Figure 4.2-1 shows the configuration of instruction cache memory.



■ TAGV (valid bit)

Indicates that an instruction is stored when the address tag of the corresponding line is valid.

It consists of four bits per line and a 4-bit tag reference value is written when an instruction is fetched.

When searching for TAG, an entry having TAGV with the same value as TAG reference value is considered a valid entry.

All entries become "1111" when TAG is initialized (see "4.4.3 Invalidating Instruction Cache").

■ Address TAG

Indicates the high-order 20 bits (when 8 Kbytes) of the memory address of the stored instruction in the corresponding line.

An entry with TAGV matching the TAG reference value and the content of address TAG matching the access required address from the CPU is considered a hit.

■ LRU

Indicates which way of the corresponding line was last accessed.

When "0", it indicates way 0 was last accessed. When "1", it indicates way 1 was last accessed.

Update the entries from way not indicated by LRU.

Always update from way 0, after the cache is flushed

Always update the entries in unlocked ways without referencing the LRU, when entries in one of the ways are locked.

■ TAG reference

A 4-bit flag indicating the same value as the valid TAGV.

It becomes "1111" when reset and "0000" is set by flushing at the start of cache use. Thereafter, it is incremented by 1 after each flush. If it is flushed when it is "1110", it returns to "0000".

When setting "0000", the TAG is initialized by sequentially writing "1111" to TAGV of all entries. Initialization requires 256 cycles (same as number of lines when 8 Kbytes). Accesses during this period are treated as access to non-cache area. Except during TAG initialization, flushing updates the TAG reference only and completes in one cycle.

4.3 Registers

This section describes the configurations and functions of registers used for instruction cache memory.

■ List of registers for instruction cache memory

Table 4.3-1 shows a list of registers for instruction cache memory.

Table 4.3-1 Instruction Cache Memory Registers

Register Abbreviation	Register Name	See
CARR*	Instruction/data cache area configuration register	4.3.1
ICHCR	Instruction cache control register	4.3.2
ISIZE	Instruction cache size register	4.3.3
IFUNC	Instruction cache function register	4.3.4

* Also used as data cache.

4.3.1 Instruction/Data Cache Area Configuration Register (CARR)

This is an 8-bit register used to configure the instruction cache and data cache area.

Write to this register only when ENAB=0 (cache disabled) in ICHCR/DCHCR register.

Write to this register is ignored and the register value is unaffected while either of the caches is enabled.

The following areas are disabled cache areas regardless of the value of this register.

- 0000 0000_H to 0000 FFFF_H : Built-in I/O areas

The following areas are enabled cache areas regardless of the value of this register.

- 0010 0000_H to 3FFF FFFF_H : Built-in/external memory areas

In emulator mode, all areas are cache disabled regardless of this register. Therefore, during emulator mode, the status of instruction cache is retained. If you rewrite a cache enabled instruction area during emulator mode, be sure to flush the instruction cache during emulator mode.

Figure 4.3-1 shows the bit configuration of the instruction/data cache area configuration register (CARR).

Figure 4.3-1 Bit configuration of the Instruction/Data Cache Area Configuration Register (CARR)

bit	7	6	5	4	3	2	1	0
	CEA7	CEA6	CEA5	CEA4	CEA3	CEA2	CEA1	CEA0
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 allowed								

[bit 7, bit 6]: CEA7, CEA6 (cache enable area bits 7, 6)

Configure the instruction/data cache area as follows:

CEA7	CEA6	Configured cache area	Control
0	0	4000 0000 _H to FFFF FFFF _H	Cache enabled
0	1	4000 0000 _H to 7FFF FFFF _H 8000 0000 _H to BFFF FFFF _H C000 0000 _H to FFFF FFFF _H	Cache disabled Cache enabled Cache disabled
1	0	4000 0000 _H to 7FFF FFFF _H 8000 0000 _H to FFFF FFFF _H	Cache enabled Cache disabled
1	1	4000 0000 _H to FFFF FFFF _H	Cache disabled

Each bit is initialized to "0" at reset.

[bit 5 to bit 0]: CEA5 to CEA0 (cache enable area bits 5 to 0)

Configure the instruction/data cache area as follows:

	Configured cache area	Control
CEA5	000C 0000 _H to 000F FFFF _H	0=Cache/1=Non-cache
CEA4	0008 0000 _H to 000B FFFF _H	0=Cache/1=Non-cache
CEA3	0004 0000 _H to 0007 FFFF _H	0=Cache/1=Non-cache
CEA2	0003 0000 _H to 0003 FFFF _H	0=Cache/1=Non-cache
CEA1	0002 0000 _H to 0002 FFFF _H	0=Cache/1=Non-cache
CEA0	0001 0000 _H to 0001 FFFF _H	0=Cache/1=Non-cache

Each bit is initialized to "0" at reset.

4.3.2 Instruction Cache Control Register (ICHCR)

This is an 8-bit register that controls the operation of the instruction cache.

Write to this register does not affect the cache operation of the instruction fetched during the four cycles starting from the write instruction.

Furthermore, since this CPU has an instruction pre-fetch function, instructions after 5 or more cycles may also be unaffected by the change in cache operation depending on the consumption of the pre-fetch buffer.

After reset, be sure to flush the instruction cache before enabling it.

Figure 4.3-2 shows the bit configuration of the instruction Cache Control Register (ICHCR)

Figure 4.3-2 Bit Configuration of Instruction Cache Control Register (ICHCR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	W1LK	W0LK	FLSH	ENAB
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	X	X	X	X	0	0	0	0
R/W: Read/write allowed								
X: Undefined								

[bit 7 to bit 4]: Reserved bits

Write	Always write "0".
Read	Value is undefined.

[bit 3]: W1LK (way 1 lock bit)

This bit controls locking of all way 1 entries as follows:

Written Value	Lock Operation of Instruction Cache Way 1
0	Way 1 is not locked. (Initial value)
1	Way 1 valid entries are locked.

The way 1 valid entries are locked starting from the cycle after writing W1LK=1.

This enables valid entries in way 1 to be used as though they are ROM.

The following table shows the operation when cache area is accessed while way 1 is locked.

Way 1 TAG Search	Way 0 TAG Search	Way 0 Lock Status	Operation
Hit	-	-	Provides way 1 instruction
-	Hit	-	Provides way 0 instruction
Entry is empty	Entry is empty	-	Updates way 0
	Miss	-	Updates way 1
Miss	Miss	Unlocked	Updates way 0
		Locked	Updates neither way

*1: Way 1 is never hit or missed when way 0 entry is empty.

*2: "Entry is empty" indicates that the entry has never been filled since it was flushed.

This bit can be rewritten during cache operation. Furthermore, flush with FLSH=1 is valid even when locked.

[bit 2]: W0LK (way 0 lock bit)

This bit controls locking of all way 0 entries as follows:

Written Value	Lock Operation of Instruction Cache Way 0
0	Way 0 is not locked. (Initial value)
1	Way 0 valid entries are locked.

The way 0 valid entries are locked starting from the cycle after writing W0LK=1.

This enables valid entries in way 0 to be used as though they are ROM.

The following table shows the operation when cache area is accessed while way 0 is locked.

Way 0 TAG Search	Way 1 TAG Search	Way 1 Lock Status	Operation
Hit	-	-	Provides way 0 instruction
-	Hit	-	Provides way 1 instruction
Entry is empty	Entry is empty	-	Updates way 0
Miss	Entry is empty	-	Updates way 1
	Miss	Unlocked	Updates way 1

*1: Way 1 is never hit or missed when way 0 entry is empty.

*2: "Entry is empty" indicates that the entry has never been filled since it was flushed.

This bit can be rewritten during cache operation. Furthermore, flush with FLSH=1 is valid even when locked.

[bit 1]: FLSH (flush bit)

This bit invalidates all instruction cache entries.

The TAG reference value is incremented and all entries are invalidated at the cycle after writing FLSH=1. Flushing is possible even during cache operation.

At the first write after reset and every 15 writes thereafter, the TAG reference value is reset to "0000". Only during this time, "1111" is sequentially written in TAGV of all entries and TAG is initialized. Initialization requires 256 cycles (same as number of lines when 8 Kbytes). Accesses during this period are treated as access to non-cache area. Except during TAG initialization, flushing updates the TAG reference only and completes in one cycle.

[bit 0]: ENAB (instruction cache enable bit)

This bit controls the enabling/disabling of instruction cache operation as follows:

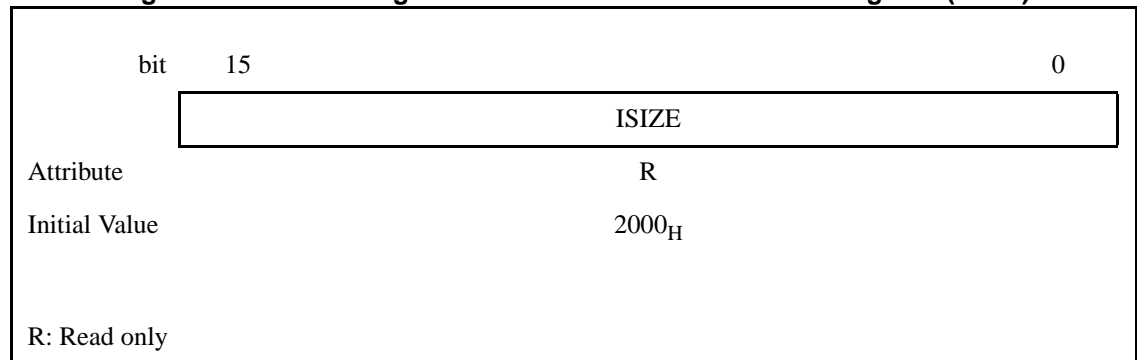
Written Value	Operation of Instruction Cache
0	Disables operation. (Initial value)
1	Enables operation.

4.3.3 Instruction Cache Size Register (ISIZE)

This is a 16-bit register that shows the size of the instruction cache.

Figure 4.3-3 shows the bit configuration of the instruction cache size register (ISIZE)

Figure 4.3-3 Bit Configuration of Instruction Cache Size Register (ISIZE)

**[bit 15 to bit 0]: ISIZE (instruction cache size bits)**

The following values are read depending on the size of the instruction cache.

Read Value	Instruction Cache Size
0800 _H	2 Kbytes
1000 _H	4 Kbytes
2000 _H	8 Kbytes
4000 _H	16 Kbytes
8000 _H	32 Kbytes

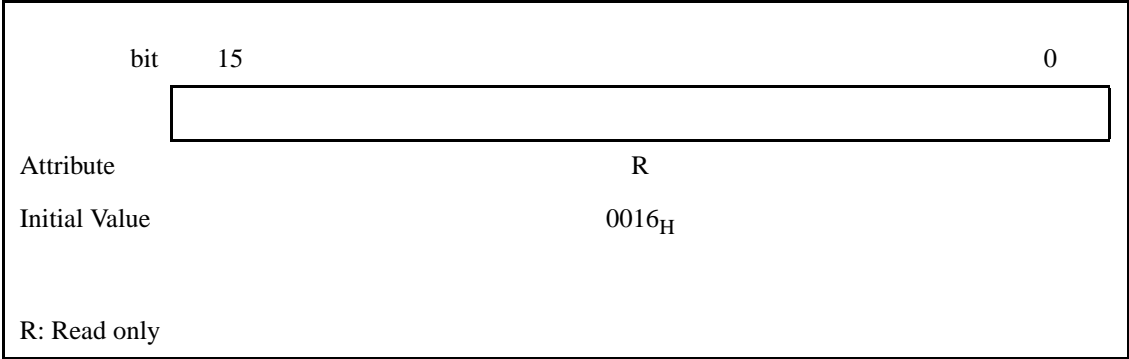
This register is read only. Writing to this register has no affect on the register value.

4.3.4 Instruction Cache Function Register (IFUNC)

This is a 16-bit register that shows the functions supported by instruction cache.

Figure 4.3-4 shows the bit configuration of the instruction cache function register (IFUNC).

Figure 4.3-4 Bit Configuration of Instruction Cache Function Register (IFUNC)



The following values are read for each bit according to each function of the instruction cache.
This register is read only. Writing to this register has no affect on the register value.

[bit 15 to bit 8]: Reserved bits

The read value is always "0".

[bit 7, bit 6]: RAM mode

These bits indicate the support content of the RAM mode function.

Read Value	RAM Mode Function
00	Unsupported (no RAM mode) (this type of product)
01	RAM is available when instruction cache is off
10	Way unit RAM mode function
11	Reserved

[bit 5, bit 4]: Number of ways

These bits indicate the number of instruction cache ways.

Read Value	Number of Ways, Method
00	Direct map
01	2-way set associative (this type of product)
10	4-way set associative
11	8-way set associative

[bit 3, bit 2]: Lock function

These bits indicate the support content of the entry lock function.

Read Value	Lock Function
00	Unsupported (no lock function)
01	Way unit global lock function (this type of product)
10	Entry unit and global lock function
11	Reserved

[bit 1, bit 0]: Flush function

These bits indicate the support content of the entry disable function.

Read Value	Disabling Function
00	Software disabling
01	1 cycle hardware disabling
10	n cycle hardware disabling (this type of product)
11	Reserved

4.4 Explanation of Operations

This section describes the operation of the instruction cache memory.

4.4.1 Initial State

After this instruction cache is reset, cache operation is disabled and the TAG contents are undefined (see "4.4.6 Cache Content in each Cache Operation State").

Before enabling cache operation, set the cache area and disable the cache.

4.4.2 Instruction Cache Area

This instruction cache uses only the following areas as cache area (CARR register is shared with data cache).

1. 0001 0000_H to 0001 FFFF_H (however, only when CEA0=0 in CARR register)
2. 0002 0000_H to 0002 FFFF_H (however, only when CEA1=0 in CARR register)
3. 0003 0000_H to 0003 FFFF_H (however, only when CEA2=0 in CARR register)
4. 0004 0000_H to 0007 FFFF_H (however, only when CEA3=0 in CARR register)
5. 0008 0000_H to 000B FFFF_H (however, only when CEA4=0 in CARR register)
6. 000C 0000_H to 000F FFFF_H (however, only when CEA5=0 in CARR register)
7. 0010 0000_H to 3FFF FFFF_H
8. 4000 0000_H to 7FFF FFFF_H (however, only when CEA6=0 in CARR register)
9. 8000 0000_H to BFFF FFFF_H (however, only when CEA7=0 in CARR register)
10. C000 0000_H to FFFF FFFF_H (however, only when CEA7, CEA6=00 in CARR register)

Write to CARR register only when ENAB=0 (cache disabled) in ICHCR/DCHCR register.

Write to CARR register is ignored and the register value is unaffected while either of the caches is enabled.

However, in emulator mode, all areas are cache disabled regardless of the CARR register. Therefore, during emulator mode, the status of instruction cache is retained. If you rewrite a cache enabled instruction area during emulator mode, be sure to flush the instruction cache during emulator mode.

4.4.3 Invalidating Instruction Cache

The FLSH bit of ICHCR register is used to invalidate (flush) the content of this instruction cache.

The instruction cache must be flushed in the following cases:

- After reset, before starting the use of instruction cache
- After instruction cache operation is stopped, before resuming use of instruction cache
- When content of instruction cache memory is rewritten

At the first flush operation after reset and every 15 flush operations thereafter, the TAG of all lines are initialized during the same cycles (256 cycles if 8 Kbytes) as the number of lines. Access during this period is treated as access to non-cache area or in the same manner as access while cache operation is disabled and the cache is not accessed even if it is an access to a cache enabled area. Normal cache operation is resumed after TAG initialization has completed.

For all other flush operations, all entries are invalidated in one cycle and cache operation is not suppressed.

4.4.4 Enabling Instruction Cache Operation

Write "1" in the ENAB bit of ICHCR register to enable the operation of this instruction cache.

If instruction cache operation is enabled during TAG initialization, no cache operation is performed until initialization completes.

4.4.5 Instruction Cache Lock Function

The content of this instruction cache can be locked in way units using the W1LK and W0LK bits of the ICHCR register. This function enables permanent high-speed access of once fetched instruction code.

Locks can be set independently in way units so that instructions to be accessed at high-speed can be kept in one way while performing normal cache operation in the other way.

Use the W1LK bit to lock all way 1 entries and the W0LK bit to lock all way 0 entries. All valid entries of the corresponding way are locked starting from the cycle after writing "0" in each bit.

If there is an access hit to a locked entry, an instruction code is supplied to the CPU as with normal cache hit.

If there is a miss-access to a locked entry, that entry is not updated. In this case, the unlocked way entry is updated regardless of the value of LRU.

If there is an access to an unused entry in the locked way, instruction code is filled in the entry in the same manner as in the unlocked state and then that entry is locked.

If both ways are locked, an access that does not hit an entry in either way operates in the same manner as access to a non-cache area and the cache is not filled.

The following procedure is effective when using the lock function:

1. Flush the cache.
2. Enable way 0 lock function.
3. Branch to the program to be locked and execute.

After flushing the cache, the first cache area access is made from way 0 because entries of both ways are unused. Therefore, the first access is fetched to way 0 and is unlocked even when way 1 rather than way 0 is locked in step (2) above. In that case, way 1 is locked only when there is a miss-access to way 0 entry.

Programs that do not need to be locked as long as lines are available are also fetched to the same way and locked. Therefore, to suppress fetching of program that must not be locked, place the program in non-cache area.

The following table shows the operation when cache area is accessed while way 0 is locked.

Way 0 TAG Search	Way 1 TAG Search	Way 1 Lock Status	Operation
Hit	-	-	Provides way 0 instruction
-	Hit	-	Provides way 1 instruction
Entry is empty	Entry is empty	-	Updates way 0
Miss	Entry is empty	-	Updates way 1
	Miss	Unlocked	Updates way 1
		Locked	Updates neither way

The following table shows the operation when cache area is accessed while way 1 is locked.

Way 1 TAG Search	Way 0 TAG Search	Way 0 Lock Status	Operation
Hit	-	-	Provides way 1 instruction
-	Hit	-	Provides way 0 instruction
Entry is empty	Entry is empty	-	Updates way 0
	Miss	-	Updates way 1
Miss	Miss	Unlocked	Updates way 0
		Locked	Updates neither way

*1: Way 1 is never hit or missed when way 0 entry is empty.

*2: "Entry is empty" indicates that the entry has never been filled since it was flushed.

Enabling/disabling of way lock can be changed during cache operation. Furthermore, flush with FLSH=1 is valid even when locked.

4.4.6 Cache Content in each Cache Operation State

The following table shows the cache hardware content during each state.

	After Reset	When Cache is Disabled	When Flushed
Cache Memory	Content undefined	Retains previous state	Retains previous state
Address TAG	Content undefined	Retains previous state	Retains previous state
TAGV bit	Content undefined	Retains previous state Able to Update with FLSH	All entries become [1111] only when TAG reference becomes [0000] Otherwise, retains previous state
TAG reference	1111	Retains previous state Able to update with FLSH	Reset to [0000] when [1110][1111] Otherwise, previous value is incremented by 1
LRU bit	Content undefined	Retains previous state	Retains previous state
W1LK bit	0=Unlocked	Retains previous state	Retains previous state
W0LK bit	0=Unlocked	Retains previous state	Retains previous state
FLSH bit	0=Not erased	Retains previous state Flushed when 1 is written	1=Erased Returns to 0 at next cycle
ENAB bit	0=Cache disabled	0=Cache disabled	Retains previous state

CHAPTER 5 Data Cache Memory

This chapter describes the functions and operations of the data cache memory.

- 5.1 Overview
- 5.2 Configuration
- 5.3 Registers
- 5.4 Explanation of Operations

5.1 Overview

This type of product has a data cache memory with the following configuration:

- Cache size: 8 Kbytes (4 Kbytes per way)
 - Mapping method: 2-way set associative
 - Replacement method: Write through (update both cache and memory upon write hit)
 - Line size: 16 bytes (4 words)
 - Total number of entries: 512 entries (256 lines per way)
-

■ Overview

The cache is filled in single line unit and a 16-byte worth read operation is performed per miss-access.

When an entry is hit by write access, write through operation, which rewrites both the cache and memory, is performed.

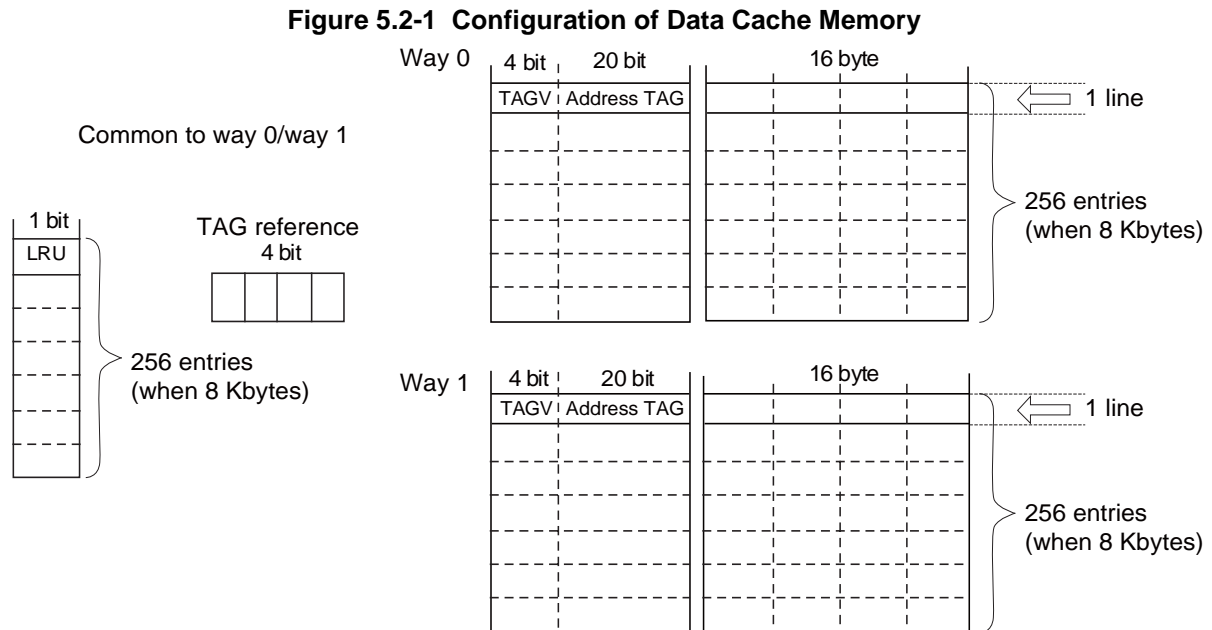
Writing to the control bit can invalidate the entire entry.

With this type of product, the entered data cannot be locked in cache.

The memory area that can be cached can be set per specific area using the CARR register also used by the data cache.

5.2 Configuration

Figure 5.2-1 shows the configuration of the data cache memory.



■ TAGV (valid bit)

Indicates that data is stored when the address tag of the corresponding line is valid.

It consists of four bits per line and a 4-bit tag reference value is written when data is fetched.

When searching for TAG, an entry having TAGV with the same value as TAG reference value is considered a valid entry.

All entries become "1111" when TAG is initialized (see "5.4.3 Invalidating Data Cache").

■ Address TAG

Indicates the high-order 20 bits (when 8 Kbytes) of the memory address of the stored data in the corresponding line.

An entry with TAGV matching the TAG reference value and the content of address TAG matching the access required address from the CPU is considered a hit.

■ LRU

Indicates which way of the corresponding line was last accessed.

When "0", it indicates way 0 was last accessed. When "1", it indicates way 1 was last accessed.

Update the entries from way not indicated by LRU.

Always update from way 0 after the cache is flushed.

■ TAG reference

A 4-bit flag indicating the same value as the valid TAGV.

It becomes "1111" when reset and "0000" is set by flushing at the start of cache use. Thereafter, it is incremented by 1 after each flush. If it is flushed when it is "1110", it returns to "0000".

When setting "0000", the TAG is initialized by sequentially writing "1111" to TAGV of all entries.

Initialization requires 256 cycles (same as number of lines). Accesses during this period are treated as access to non-cache area. Except during TAG initialization, flushing updates the TAG reference only and completes in one cycle.

5.3 Registers

This section describes the configurations and functions of registers used for data cache memory.

■ List of registers for data cache memory

Table 5.3-1 shows a list of registers for data cache memory.

Table 5.3-1 Data Cache Memory Registers

Register Abbreviation	Register Name	See
CARR*	Instruction/data cache area configuration register	4.3.1
DCHCR	Data cache control register	5.3.1
DSIZE	Data cache size register	5.3.2
DFUNC	Data cache function register	5.3.3

* : Shared with instruction cache.

5.3.1 Data Cache Control Register (DCHCR)

This is an 8-bit register that controls the operation of the data cache.

Write to this register has effect starting from data access performed by instruction immediately following a write instruction.

After reset, be sure to flush the data cache before enabling it.

Figure 5.3-1 shows the bit configuration of the data cache control register (DCHCR).

Figure 5.3-1 Bit Configuration of Data Cache Control Register (DCHCR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FLSH	ENAB
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	X	X	X	X	X	X	0	0
R/W: Read/write allowed								
X: Undefined								

[bit 7 to bit 2]: Reserved bits

Write	Always write "0".
Read	Value is undefined.

[bit 1]: FLSH (flush bit)

This bit invalidates all data cache entries.

The TAG reference value is incremented and all entries are invalidated at the cycle after writing FLSH=1. Flushing is possible even during cache operation.

At the first write after reset and every 15 writes thereafter, the TAG reference value is reset to "0000". Only during this time, "1111" is sequentially written in TAGV of all entries and TAG is initialized. Initialization requires 256 cycles (same as number of lines when 8 Kbytes). Accesses during this period are treated as access to non-cache area. Except during TAG initialization, flushing updates the TAG reference only and completes in one cycle.

[bit 0]: ENAB (data cache enable bit)

This bit controls the enabling/disabling of data cache operation as follows:

Written Value	Operation of Data Cache
0	Disables operation. (Initial value)
1	Enables operation.

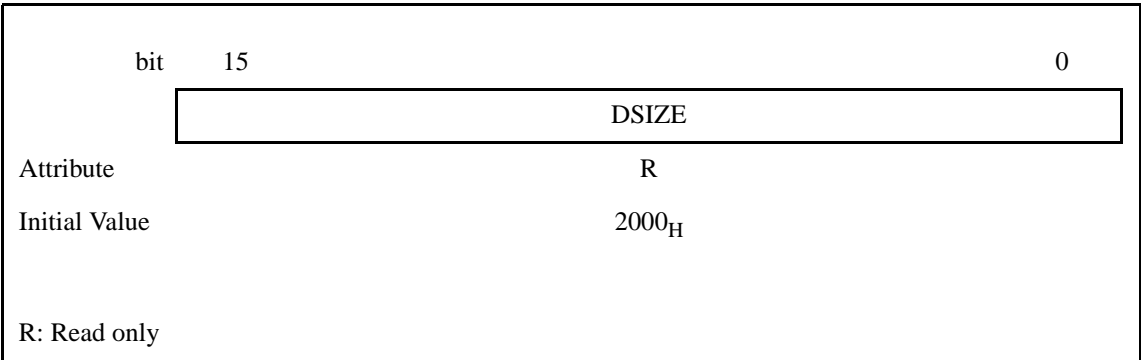
This bit is initialized to "0" at reset.

5.3.2 Data Cache Size Register (DSIZE)

This is a 16-bit register that shows the size of the data cache.

Figure 5.3-2 shows the bit configuration of the data cache size register (DSIZE).

Figure 5.3-2 Bit Configuration of Data Cache Size Register (DSIZE)



[bit 15 to bit 0]: DSIZE (data cache size bit)

The following values are read depending on the size of the data cache.

Read Value	Data cache size
0800 _H	2 Kbytes
1000 _H	4 Kbytes
2000 _H	8 Kbytes
4000 _H	16 Kbytes
8000 _H	32 Kbytes

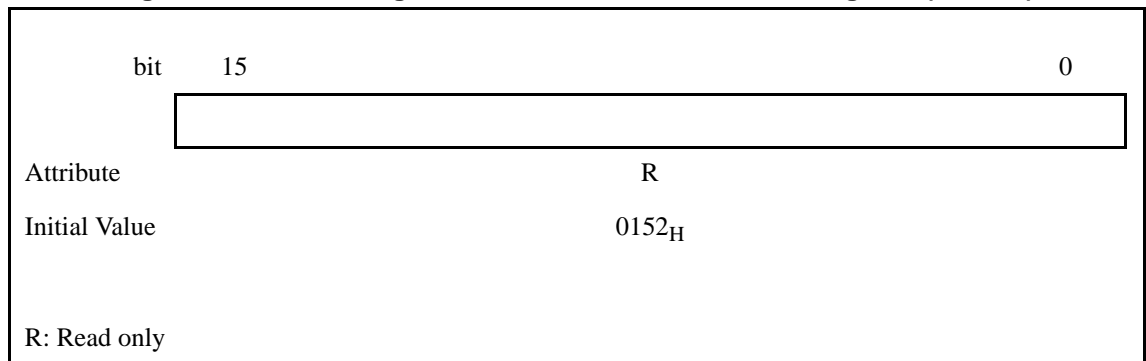
This register is read only. Writing to this register has no affect on the register value.

5.3.3 Data Cache Function Register (DFUNC)

This is a 16-bit register that shows the functions supported by data cache.

Figure 5.3-3 shows the bit configuration of the data cache function register (DFUNC).

Figure 5.3-3 Bit Configuration of Data Cache Function Register (DFUNC)



The following values are read for each bit according to each function of the data cache.

This register is read only. Writing to this register has no affect on the register value.

[bit 15 to bit 11]: Reserved bits

The read value is always "0".

[bit 10]: Copy back function

This bit indicates the content of the copy back function.

Read Value	Copy Back Function
0	Unsupported (this type of product)
1	Supported

[bit 9]: Write allocate function

This bit indicates the content of the write allocate function.

Read Value	Write Allocate Function
0	Unsupported (this type of product)
1	Supported

[bit 8]: Write through function

This bit indicates the content of the write through function.

Read Value	Write Through Function
0	Unsupported
1	Supported (this type of product)

[bit 7, bit 6]: RAM mode

These bits indicate the support content of the RAM mode function.

Read Value	RAM Mode Function
00	Unsupported (no RAM mode)
01	RAM is available when data cache is off (this type of product)
10	Way unit RAM mode function
11	Reserved

[bit 5, bit 4]: Number of ways

These bits indicate the number of data cache ways.

Read Value	Number of Ways, Method
00	Direct map
01	2-way set associative (this type of product)
10	4-way set associative
11	8-way set associative

[bit 3, bit 2]: Lock function

These bits indicate the support content of the entry lock function.

Read Value	Lock Function
00	Unsupported (no lock function) (this type of product)
01	Way unit global lock function
10	Entry unit and global lock function
11	Reserved

[bit 1, bit 0]: Flush function

These bits indicate the support content of the entry disable function.

Read Value	Disabling Function
00	Software disabling
01	1 cycle hardware disabling
10	n cycle hardware disabling (this type of product)
11	Reserved

5.4 Explanation of Operations

This section describes the operation of data cache memory.

5.4.1 Initial State

After this data cache is reset, cache operation is disabled and the TAG contents are undefined (see "4.4.6 Cache Content in each Cache Operation State").

Before enabling cache operation, set the cache area and disable the cache.

5.4.2 Data Cache Area

This data cache uses only the following areas as cache area (CARR register is also used as instruction cache).

1. 0001 0000_H to 0001 FFFF_H (however, only when CEA0=0 in CARR register)
2. 0002 0000_H to 0002 FFFF_H (however, only when CEA1=0 in CARR register)
3. 0003 0000_H to 0003 FFFF_H (however, only when CEA2=0 in CARR register)
4. 0004 0000_H to 0007 FFFF_H (however, only when CEA3=0 in CARR register)
5. 0008 0000_H to 000B FFFF_H (however, only when CEA4=0 in CARR register)
6. 000C 0000_H to 000F FFFF_H (however, only when CEA5=0 in CARR register)
7. 0010 0000_H to 3FFF FFFF_H
8. 4000 0000_H to 7FFF FFFF_H (however, only when CEA6=0 in CARR register)
9. 8000 0000_H to BFFF FFFF_H (however, only when CEA7=0 in CARR register)
10. C000 0000_H to FFFF FFFF_H (however, only when CEA7, CEA6=00 in CARR register)

However, in emulator mode, all areas are cache disabled for read and cache enabled for write regardless of the CARR register. Therefore, in emulator mode, the content of data cache is retained for read and if cache is hit for write, both the data in the cache and memory are updated (LRU is not updated).

Furthermore, cache is flushed when TAG reference becomes "0000" and accesses during TAG initialization cycle are all treated as access to non-cache area. Cache access judgment is resumed after TAG initialization cycle has completed.

5.4.3 Invalidating Data Cache

The FLSH bit of DCHCR register is used to invalidate (flush) the content of this data cache.

The data cache must be flushed in the following cases:

- After reset, before starting the use of data cache
- After data cache operation is stopped, before resuming use of data cache
- When content of data cache memory is rewritten with content other than instruction (such as DMA transfer)

At the first flush operation after reset and every 15 flush operations thereafter, the TAG of all lines are initialized during the same cycles (256 cycles if 8 Kbytes) as the number of lines. Access during this period is treated as access to non-cache area or in the same manner as access while cache operation is disabled and the cache is not accessed even if it is an access to a cache enabled area. Normal cache operation is resumed after TAG initialization has completed.

For all other flush operations, all entries are invalidated in one cycle and cache operation is not suppressed.

5.4.4 Enabling Data Cache Operation

Write "1" in the ENAB bit of DCHCR register to enable the operation of this data cache.

If instruction cache operation is enabled during TAG initialization, no cache operation is performed until initialization completes.

5.4.5 Cache Content in each Cache Operation State

The following table shows the cache hardware content during each state.

	After Reset	When Cache is Disabled	When Flushed
Cache Memory	Content undefined	Retains previous state	Retains previous state
Address TAG	Content undefined	Retains previous state	Retains previous state
TAGV bit	Content undefined	Retains previous state Able to update with FLSH	All entries become [1111] only when TAG reference becomes [0000] Otherwise, retains previous state
TAG reference	1111	Retains previous state Able to update with FLSH	Reset to [0000] when [1110][1111] Otherwise, previous value is incremented by 1
LRU bit	Content undefined	Retains previous state	Retains previous state
FLSH bit	0=Not erased	Retains previous state Flushed when 1 is written	1=Erased Returns to 0 at next cycle
ENAB bit	0=Cache disabled	0=Cache disabled	Retains previous state

5.4.6 RAM Operation Mode when Cache is Off

This data cache operates as a data RAM with high-speed access when caching is disabled (ENAB=0 in DCHR register).

The memory map during data RAM operation is shown below (the following RAM area segmentation depends on the cache size).

- 0000 8000_H to 0000 8FFF_H : RAM area (1) (4 Kbytes)
- 0000 9000_H to 0000 9FFF_H : RAM area (2) (4 Kbytes)
- 0000 A000_H to 0000 AFFF_H : RAM area (1) mirror area
- 0000 B000_H to 0000 BFFF_H : RAM area (2) mirror area
- 0000 C000_H to 0000 CFFF_H : RAM area (1) mirror area
- 0000 D000_H to 0000 DFFF_H : RAM area (2) mirror area
- 0000 E000_H to 0000 EFFF_H : RAM area (1) mirror area
- 0000 F000_H to 0000 FFFF_H : RAM area (2) mirror area

The content of the data cache is returned when any of the above areas is read while data cache operation is enabled (ENAB=1 in DCHCR register). However, the data is read regardless of its validity. Write to the above areas is ignored while data cache operation is enabled.

This data RAM cannot be used for DMA transfer. Only program access from CPU is allowed.

CHAPTER 6 Operation Mode

This chapter describes the functions and behavior of operation modes.

- 6.1 Overview
- 6.2 Configuration
- 6.3 Registers
- 6.4 Explanation of Operations

6.1 Overview

This series has the following operation modes that can be selected during activation.

- User mode, external ROM external bus
 - Serial writer mode
-

■ Overview

The relationship between MD pins and operation modes is shown below.

Table 6.1-1 Relationship between MD pins and Operation modes

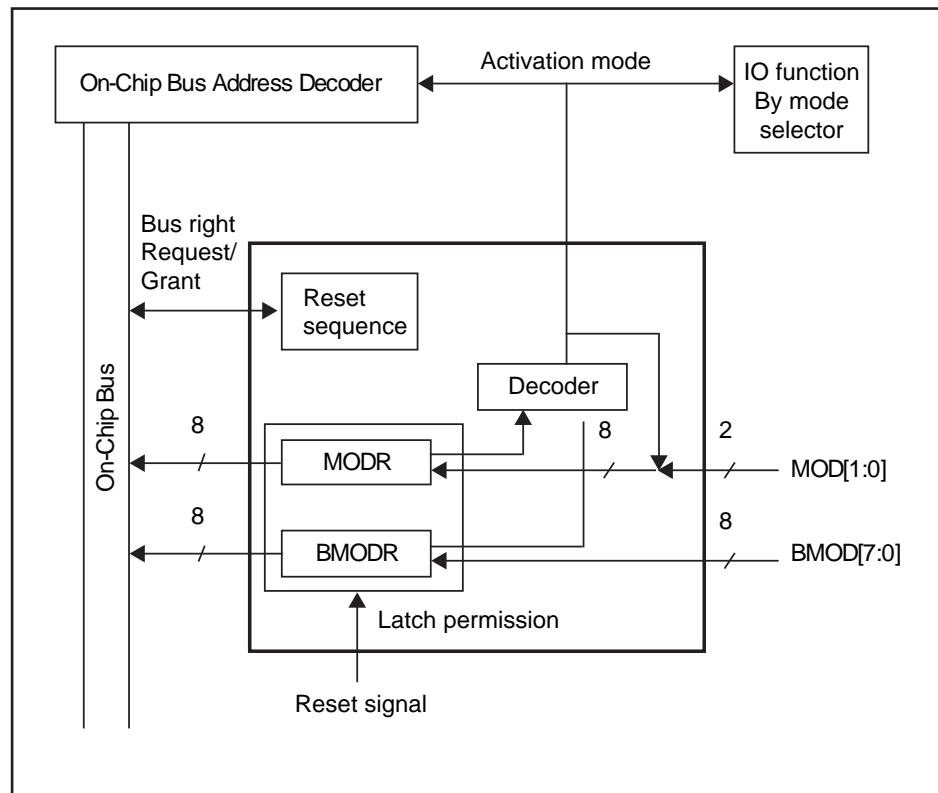
MD Pin		Operation Mode
MD1	MD0	
0	0	User mode, external ROM external bus
	1	Serial writer mode

6.2 Configuration

Figure 6.2-1 shows the block diagram of operation mode.

■ Block diagram

Figure 6.2-1 Block Diagram of Operation Mode



6.3 Registers

This section describes the configurations and functions of registers used in each operation mode.

■ List of registers for operation mode

Table 6.3-1 shows a list of registers for operation mode.

Table 6.3-1 Operation Mode Registers

Register Abbreviation	Register Name	See
BMODR	Bus mode data register	6.3.1
MODR	Mode register	6.3.2

6.3.1 Bus Mode Data Register (BMODR)

This is an 8-bit register indicating the bus mode data set during activation.

This register is read only. Writing to this register has no affect.

Figure 6.3-1 shows the bit configuration of the bus mode register (BMODR).

Figure 6.3-1 Bit Configuration of Bus Mode Register (BMODR)

bit	7	0
	BMOD[7:0]	
Attribute	R	
Initial Value	Depends on mode setting	
R: Read only		

[bit 7 to bit 0]: BMOD[7:0] (Bus mode bits)

These bits indicate the bus mode data. They indicate the following content according to the value set in MD1 and MD0 pin.

MD Pin		Mode Data BMOD[7:0]	Operation Mode
MD1	MD0		
0	0	XXXXXXXX	User mode, external ROM external bus
	1	0111XXXX	Serial writer mode

6.3.2 Mode Register (MODR)

This is an 8-bit register indicating the mode set during activation.

This register is read only. Writing to this register has no effect.

Figure 6.3-2 shows the bit configuration of the mode register (MODR).

Figure 6.3-2 Bit Configuration of Mode Register (MODR)

bit	7	5	4	3	2	1	0
	Reserved			MOD[1:0]		ROMA	WTH[1:0]
Attribute	R						
Initial Value	Depends on mode setting						
R: Read only							

[bit 7 to bit 5]: Reserved bits

Fixed to "000".

[bit 4, bit 3]: MOD[1:0] (Mode value bits)

These bits indicate the following content according to each mode.

Read Value	Active Mode
00	User mode, external ROM external bus
11	Serial writer mode

[bit 2]: ROMA (ROM area bit)

This bit indicates the following content according to the ROM area mapping during activation including reset vector.

This bit is "0" (external bus area) in user mode/external ROM external bus and "1" (internal ROM area) in serial writer mode.

Read Value	ROM Area
0	External bus area
1	Internal ROM area

[bit 1, bit 0]: WTH[1:0] (External bus width initial value bits)

These bits indicate the following content according to the initial external bus width setting during activation.

These bits are fixed to "11" in user mode/external ROM external bus and serial writer mode.

6.4 Explanation of Operations

This section describes the operation modes of this series.

6.4.1 How to Set the Operation Mode

The operation mode is set with the MD pin (MD1, MD0) input.

6.4.2 Details of Each Mode

The detailed behavior of each mode is described below.

■ User mode/external ROM external bus

User mode/external ROM external bus is a mode in which the external bus expansion function is enabled. Initially, the CS4 as general purpose CS area functions as the external bus area. In this case, the initial external bus width is 16 bits.

In this mode, the external bus area is mapped to the boot area. Therefore, the CPU obtains the reset vector from the CS4 as general purpose CS area after activation and starts the operation.

In user mode/external ROM external bus, the operation mode is unaffected when the MD pins (MD1, MD0) change after RST is canceled.

■ Serial writer mode

In the serial writer mode, the FLASH memory is written with the serial writer.

Fix always the MD pins (MD1, MD0) in serial writer mode. If MD pins are changed, the operation mode is affected.

See "CHAPTER 28 Serial Programming Connection Example" for details.

CHAPTER 7 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.

- 7.1 Overview
- 7.2 Configuration
- 7.3 Pins
- 7.4 Registers
- 7.5 Explanation of Operations

7.1 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 2 clocks can be selected for the source clock (SRCCLK):

- Main clock (MAINCLK)
- PLL clock (PLLCLK)

The clock generating parts control the following:

- Main clock (MAINCLK) generation
 - Controls the oscillation of the main clock (MAINCLK).
 - Sets the oscillation stabilization wait time of the main clock (MAINCLK).
 - Controls the main timer or generation of main 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 (MAINCLK) multiple rate for generating the PLL clock (PLLCLK)).
The multiple rate can be set only for the main clock (MAINCLK).
- Source clock (SRCCLK) selection
Selects one of 2 clocks for use as the source clock (SRCCLK).

7.2 Configuration

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

7.2.1 Clock Generating Parts

There are 2 clock generating parts. Any of the clocks generated by the clock generating parts can be selected for the source clock (SRCCLK).

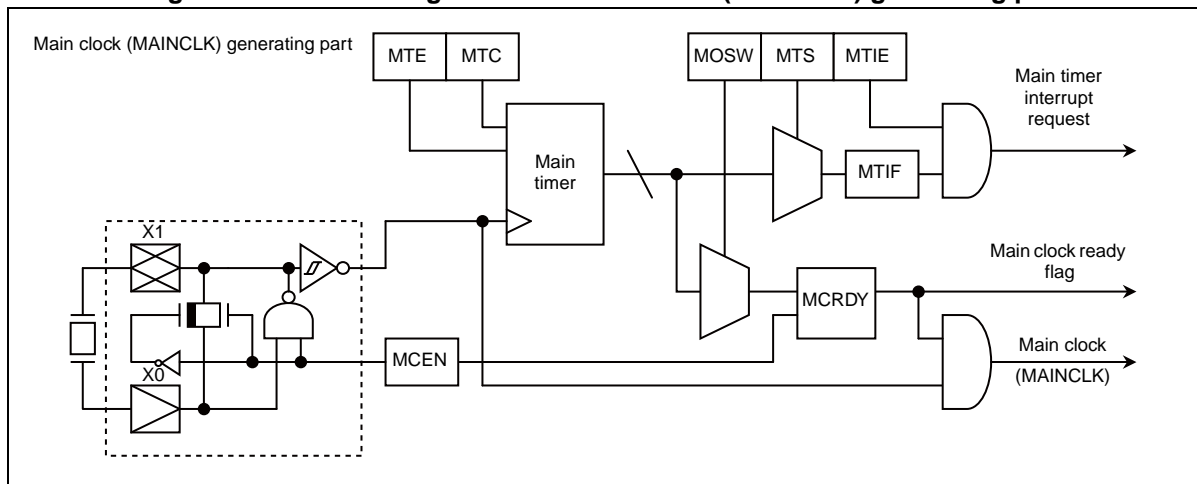
■ Main clock (MAINCLK) generating part

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

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

Figure 7.2-1 shows a block diagram of the main clock (MAINCLK) generating part.

Figure 7.2-1 Block diagram of the main clock (MAINCLK) generating part



- Main timer

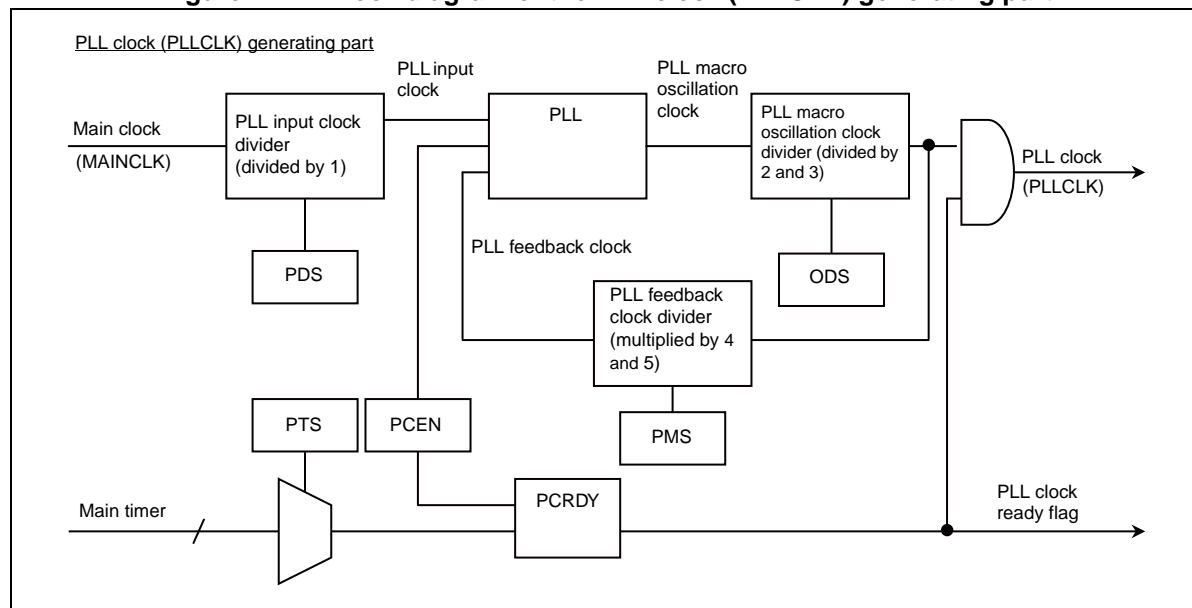
The main timer operates with the main clock (MAINCLK). For details, see "CHAPTER 9 Main Timer".

■ PLL clock (PLLCLK) generating part

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

Figure 7.2-2 shows a block diagram of the PLL clock (PLLCLK) generating part.

Figure 7.2-2 Block diagram of the PLL clock (PLLCLK) generating part



- PLL
Clock multiplication circuit
- PLL input clock divider
This divider divides the main clock (MAINCLK) 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).

<Note>

Use this part with setting the main clock to 10MHz or higher and PLL macro oscillation clock frequency to 150MHz to 200MHz.

7.2.2 Source Clock (SRCCLK) Selection Block

This section explains selection of the source clock (SRCCLK). The source clock (SRCCLK) is selected from the following 2 clock sources:

- Main clock (MAINCLK) divided by 2
- PLL clock (PLLCLK)

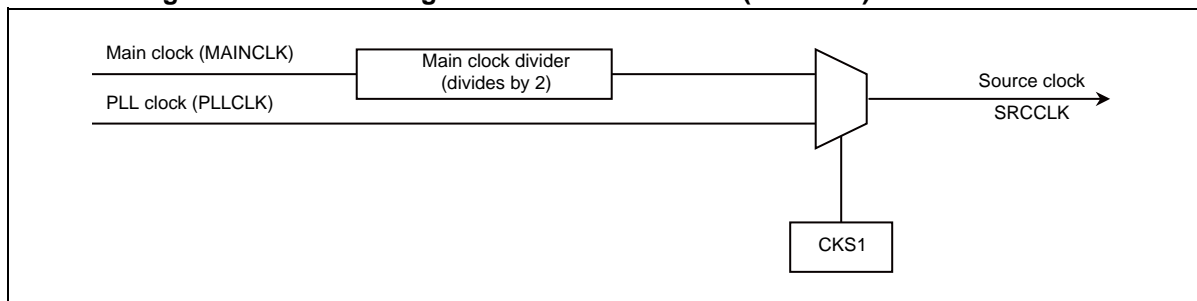
When an initialization reset (INIT) is generated, the settings of the source clock (SRCCLK) are initialized, and the main clock (MAINCLK) divided by 2 is set for the source clock (SRCCLK).

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 7.2-3 shows a block diagram of the source clock (SRCCLK) selection block.

Figure 7.2-3 Block diagram of the source clock (SRCCLK) selection block



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

7.3 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 (MAINCLK).

7.4 Registers

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

■ Registers of the clock generating parts

Table 7.4-1 lists the registers of the clock generating parts.

Table 7.4-1 Registers of the clock generating parts

Abbreviated Register Name	Register Name	Reference
CSELR	Clock source select register	7.4.1
CMONR	Clock source monitor register	7.4.2
CSTBR	Clock stabilization time select register	7.4.3
PLLCR	PLL configuration register	7.4.4

7.4.1 Clock Source Select Register (CSELR)

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

Figure 7.4-1 shows the bit configuration of the clock source select register (CSELR).

Figure 7.4-1 Bit configuration of the clock source select register (CSELR)

bit	7	6	5	4	3	2	1	0
	Reserved	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	1	-	-	-	0	0
Initial value (at RST)	-	*	*	-	-	-	*	*
R/W: Read/Write								
*: Uninitialized bit								
-: Undefined								

<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).

[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 (MAINCLK) 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 (MAINCLK).
- 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 (MAINCLK).

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

<Notes>

- This bit cannot be changed with this device.
- In stop mode, the oscillation of the main clock (MAINCLK) is stopped regardless of the value of the bit.

[bit7, 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 (MAINCLK) divided by 2
0	1	
1	0	PLL clock (PLLCLK)
1	1	Setting prohibited

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

Table 7.4-2 lists the conditions for changes of this bit.

Table 7.4-2 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	-
		10	PCRDY = 1	
0	1	00, 01	MCRDY = 1	10
1	0	00	MCRDY = 1	01
		10	PCRDY = 1	

Do not write the unchangeable values listed in Table 7.4-2. For the procedures for switching the source clock (SRCCLK), see "7.5.2 Switching the Source Clock (SRCCLK)".

7.4.2 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 7.4-2 shows the bit configuration of the clock source monitor register (CMONR).

Figure 7.4-2 Bit configuration of the clock source monitor register (CMONR)

	bit	7	6	5	4	3	2	1	0
		Reserved	PCRDY	MCRDY	Reserved	Reserved	Reserved	CKM1	CKM0
Attribute		R	R	R	R	R	R	R	R
Initial value (at INIT)		-	0	1	-	-	-	0	0
Initial value (at RST)		-	*	*	-	-	-	*	*
R: Read only									
*: Uninitialized bit									
-: Undefined									

<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.

[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 (MAINCLK) 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 (MAINCLK) nor the PLL clock (PLLCLK) can be selected as the source clock (SRCCLK).
- 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".

[bit7, 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 (MAINCLK) divided by 2 is selected.
0	1	
1	0	The PLL clock (PLLCLK) is selected.
1	1	-

7.4.3 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
- When the main oscillation is stopped and an initialize reset (INIT) is generated

The ready bits are as follows:

- PLL clock: PCRDY bit
- Main clock: MCRDY bit

Figure 7.4-3 shows the bit configuration of the clock stabilization select register (CSTBR).

Figure 7.4-3 Bit configuration of the clock stabilization time select register (CSTBR)

	bit 7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	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
Initial value (at INIT)	-	-	-	-	*	*	*	*
Initial value (at RST)	0	*	*	*	*	*	*	*
R/W: Read/Write								
*: Uninitialized bit								
-: Undefined								

<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 to 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]: MOSW3 to MOSW0 (Main clock oscillation stabilization select bits)

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

MOSW3	MOSW2	MOSW1	MOSW0	Main Clock (MAINCLK) Oscillation Stabilization Wait Time	16MHz
0	0	0	0	$2^1 \times$ main clock (MAINCLK) cycle	500ns
0	0	0	1	$2^5 \times$ main clock (MAINCLK) cycle	2ns
0	0	1	0	$2^6 \times$ main clock (MAINCLK) cycle	4μs
0	0	1	1	$2^7 \times$ main clock (MAINCLK) cycle	8μs
0	1	0	0	$2^8 \times$ main clock (MAINCLK) cycle	16μs
0	1	0	1	$2^9 \times$ main clock (MAINCLK) cycle	32μs
0	1	1	0	$2^{10} \times$ main clock (MAINCLK) cycle	64μs
0	1	1	1	$2^{11} \times$ main clock (MAINCLK) cycle	128μs
1	0	0	0	$2^{12} \times$ main clock (MAINCLK) cycle	256μs
1	0	0	1	$2^{13} \times$ main clock (MAINCLK) cycle	512μs
1	0	1	0	$2^{14} \times$ main clock (MAINCLK) cycle	1024μs
1	0	1	1	$2^{15} \times$ main clock (MAINCLK) cycle	2048μs
1	1	0	0	$2^{17} \times$ main clock (MAINCLK) cycle	8196μs
1	1	0	1	$2^{19} \times$ main clock (MAINCLK) cycle	32.8ms
1	1	1	0	$2^{21} \times$ main clock (MAINCLK) cycle	131.1ms
1	1	1	1	$2^{23} \times$ main clock (MAINCLK) cycle	524.5ms

<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.

7.4.4 PLL Configuration Register (PLLCR)

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

For the calculation of the clock frequency and the multiple rate related to generating the PLL clock (PLLCLK), see "7.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK)".

Figure 7.4-4 shows the bit configuration of the PLL configuration register (PLLCR).

Figure 7.4-4 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
Initial value (at RST)	-	-	*	*	*	*	*	*

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)	*	*	*	*	*	*	*	*

R/W: Read/Write
 *: Uninitialized bit
 -: Undefined

<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

<Note>

Be sure to change these bits to "01" or "10" in advance to allow the PLL clock (PLLCLK) oscillation. See "7.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK)".

[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

<Note>

Be sure to change these bits to "0100" or "0011" in advance to allow the PLL clock (PLLCLK) oscillation. See "7.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK)".

[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 16 MHz
1	0	0	0	$2^9 \times$ Main clock (MAINCLK) period	32.0 μ s
1	0	0	1	$2^{10} \times$ Main clock (MAINCLK) period	64.0 μ s
1	0	1	0	$2^{11} \times$ Main clock (MAINCLK) period	128.0 μ s
1	0	1	1	$2^{12} \times$ Main clock (MAINCLK) period	256.0 μ s
1	1	0	0	$2^{13} \times$ Main clock (MAINCLK) period	512.0 μ s
1	1	0	1	$2^{14} \times$ Main clock (MAINCLK) period	1024 μ s
1	1	1	0	$2^{15} \times$ Main clock (MAINCLK) period	2048.0 μ s
1	1	1	1	$2^{16} \times$ Main clock (MAINCLK) period	4096.0 μ 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.
- 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 (MAINCLK) division rate for generating the PLL input clock.

These bits must always be set to "0000" (PLL input clock = Main clock (MAINCLK)/1).

7.5 Explanation of Operations

This section explains the operations of the clock generating parts.

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

7.5.1 Explanation of Clock Source Operations

This section explains mainly oscillation control of the clock sources.

■ Main clock (MAINCLK)

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 "CHAPTER 9 Main Timer".)

● Conditions for stopping oscillation

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

- When stop mode is in effect

Supplying of the main clock (MAINCLK) 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 (MAINCLK) 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 (MAINCLK).

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 (MAINCLK) period.

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

● End of the oscillation stabilization wait time

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

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

● 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 (MAINCLK) 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):
 - CKS1 or CKS0 bit in the clock source select register (CSELR) = a value other than 10
 - 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 $2^{16} \times \text{Main clock (MAINCLK) period}$.

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)

7.5.2 Switching the Source Clock (SRCCLK)

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

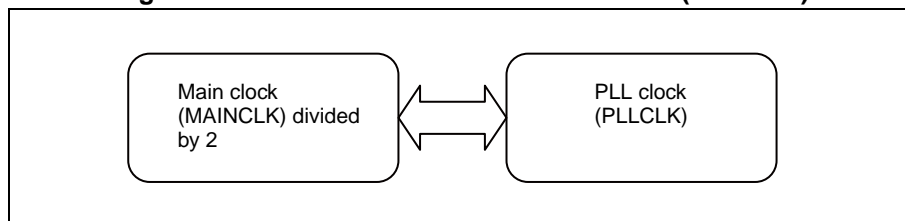
■ Overview

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

Figure 7.5-1 shows how to switch the source clock (SRCCLK).

Figure 7.5-1 How to switch the source clock (SRCCLK)



<Note>

Even if the source clock (SRCCLK) is switched, the oscillation enable settings (the values of the 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 (MAINCLK) divided by 2 to the PLL clock (PLLCLK)

To switch the source clock (SRCCLK) from the main clock (MAINCLK) 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 (MAINCLK) 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, CKM0 = 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 (MAINCLK) divided by 2

To switch the source clock (SRCCLK) from the PLL clock (PLLCLK) to the main clock (MAINCLK) 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 (MAINCLK) 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 (MAINCLK) divided by 2.

7.5.3 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 7.5-1 lists sample settings of the PLL clock (PLLCLK).

Table 7.5-1 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 to ODS0	PMS3 to PMS0				
16MHz	0000	01	0100	16MHz	Multiplied by 10	160MHz	80MHz
16MHz	0000	10	0011	16MHz	Multiplied by 12	192MHz	64MHz

<Notes>

- Use this part with setting the main clock to 10MHz or higher and PLL macro oscillation clock frequency to 150MHz to 200MHz.
- In case of using FUJITSU FLASH MCU Programmer, be sure to set the PLL input clock, PLL multiple rate, and PLL macro oscillation clock (if allowing PLL oscillation) to either of the following beforehand because the main oscillation frequency is fixed to 16MHz.

PLL input clock frequency	16MHz	16MHz
PLL multiple rate	Multiplied by 10	Multiplied by 12
PLL macro oscillation clock frequency	160MHz	192MHz
PLL clock frequency	80MHz	64MHz

*: When DIVB (base clock division configuration bits) is set as divided by 1.

CHAPTER 8 Clock Division Control Part

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

- 8.1 Overview
- 8.2 Internal Clocks
- 8.3 Configuration
- 8.4 Registers
- 8.5 Division Rate

8.1 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 8.1-1 lists the internal clocks that are generated. These clocks are collectively called internal clocks.

Table 8.1-1 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)
External output clock (TCLK)	Base clock (BCLK) divided by a value from 1 to 8
Peripheral clock (PCLK)	Base clock (BCLK) divided by a value from 1 to 16

For details of the source clock (SRCCLK), see "CHAPTER 7 Clock Generating Parts".

8.2 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
- 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
- Stop mode

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

■ 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
- Stop mode

Clock Name	Typical Supply Destination
On-chip bus clock (HCLK)	DMA controller (DMAC) External bus interface Built-in RAM

■ External output clock (TCLK)

This clock is supplied to an external SYSCLK pin

The DIVT2 to DIVT0 bits of divide clock configuration register 1 (DIVR1) are used when this clock is generated by dividing the base clock (BCLK) by a value ranging from 1 to 8.

If there is no on-chip bus access in bus sleep mode, the clock can be stopped by specifying the TSTP bit in divide clock configuration register 1 (DIVR1).

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

- Main timer mode
- Stop mode

Clock Name	Typical Supply Destination
External output clock (TCLK)	SYSCLK pin

<Notes>

- The same frequency as that for the external output clock (TCLK) is output for the bus clock (SYSCLK) from the SYSCLK pin.
- If an odd number is specified for the division rate of the base clock (BCLK) (DIVT2 to DIVT0 bits in divide clock configuration register 1 (DIVR1)), the duty ratio of the bus clock (SYSCLK) output from the SYSCLK pin cannot be 50%. The "H" level output period becomes 50% or less of the output period.

■ 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
- Stop mode

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

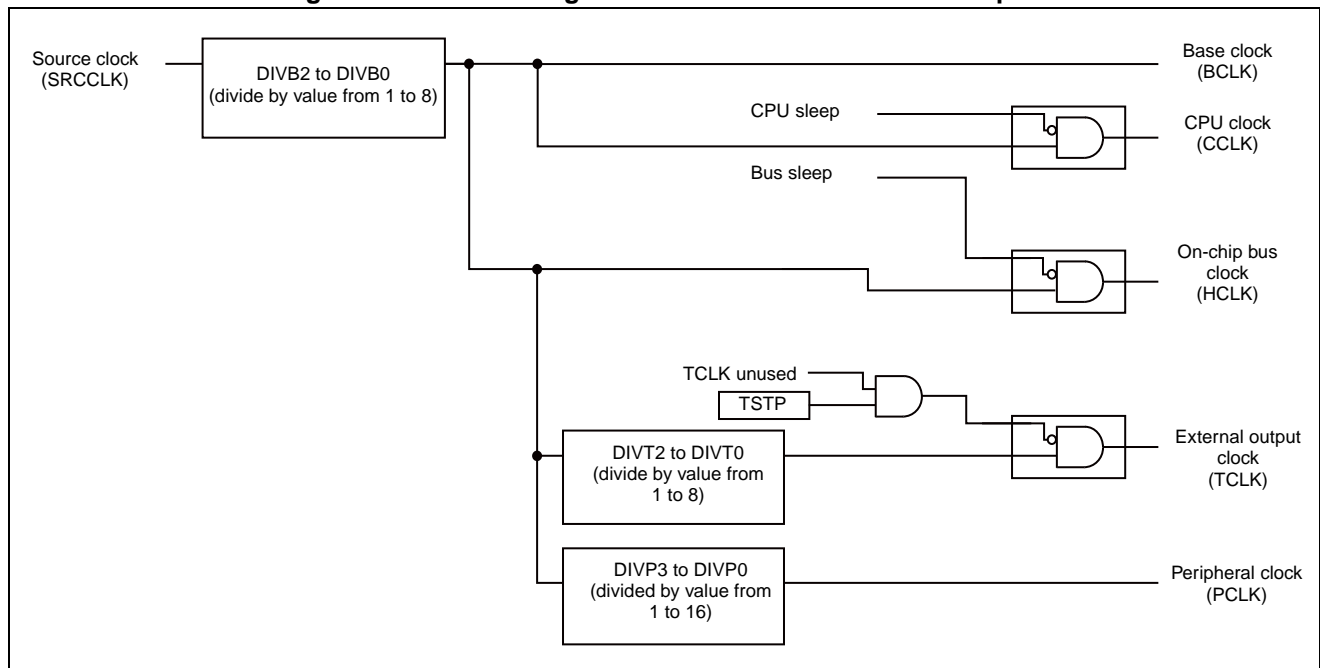
8.3 Configuration

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 8.3-1 is a block diagram of the clock division control part.

Figure 8.3-1 Block diagram of the clock division control part



8.4 Registers

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

■ Registers of the clock division control part

Table 8.4-1 lists the registers of the clock division control part.

Table 8.4-1 Registers of the clock division control part

Abbreviated Register Name	Register Name	Reference
DIVR0	Divide clock configuration register 0	8.4.1
DIVR1	Divide clock configuration register 1	8.4.2
DIVR2	Divide clock configuration register 2	8.4.3

8.4.1 Divide Clock Configuration Register 0 (DIVR0)

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

Figure 8.4-1 shows the bit configuration of divide clock configuration register 0 (DIVR0).

Figure 8.4-1 Bit configuration of divide clock configuration register 0 (DIVR0)

bit	7	6	5	4	3	2	1	0
	DIVB2	DIVB1	DIVB0	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								

[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 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]: Reserved bits

In case of writing	Always write "1" to this (these) bit (bits)
In case of reading	"1" is read.

8.4.2 Divide Clock Configuration Register 1 (DIVR1)

This register sets the base clock (BCLK) division rate for generating the external output clock (TCLK). It also controls the stopping of the external output clock (TCLK).

Figure 8.4-2 shows the bit configuration of divide clock configuration register 1 (DIVR1).

Figure 8.4-2 Bit configuration of divide clock configuration register 1 (DIVR1)

bit	7	6	5	4	3	2	1	0
	TSTP	DIVT2	DIVT1	DIVT0	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	1	0	0	0	0
R/W: Read/Write								

[bit7]: TSTP (External output clock stop enable bit)

This bit specifies whether to stop the external output clock (TCLK) when the clock output is stopped in sleep mode.

If such stopping is enabled, the external output clock (TCLK) is not supplied.

Written Value	Explanation
0	Do not stop the external output clock (TCLK).
1	Stop the external output clock (TCLK).

[bit6 to bit4]: DIVT2 to DIVT0 (External output clock division configuration bits)

These bits set the division rate for generating the external output clock (TCLK) from the base clock (BCLK).

DIVT2	DIVT1	DIVT0	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

[bit3 to bit0]: Reserved bits

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

8.4.3 Divide Clock Configuration Register 2 (DIVR2)

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

Figure 8.4-3 shows the bit configuration of divide clock configuration register 2 (DIVR2).

Figure 8.4-3 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

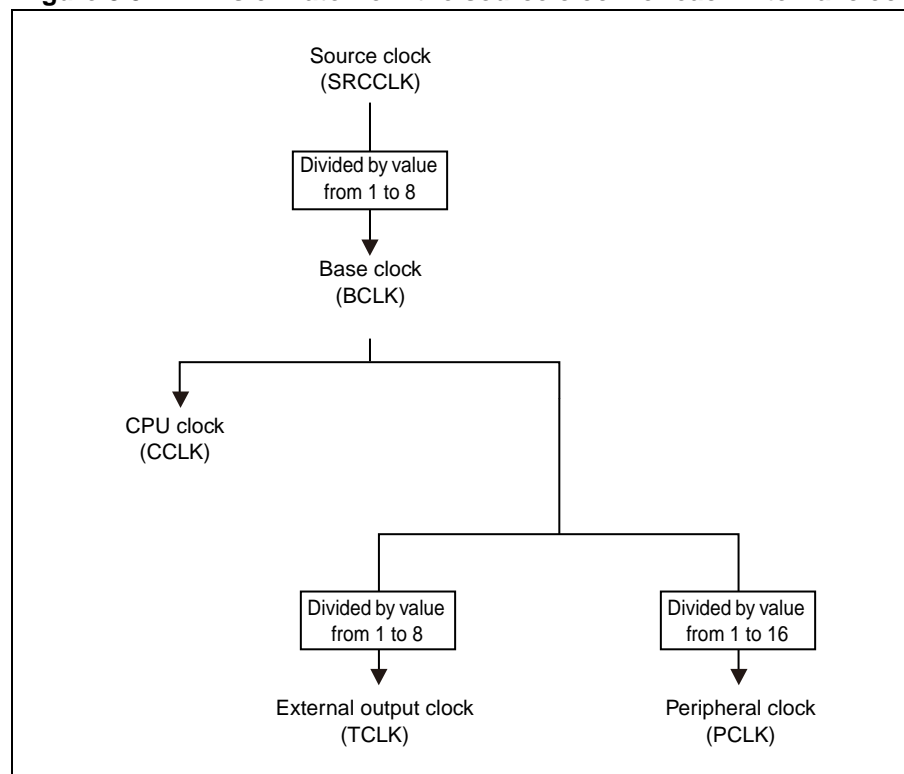
In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

8.5 Division Rate

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

Figure 8.5-1 shows the division rate from the source clock for each internal clock.

Figure 8.5-1 Division rate from the source clock for each internal clock



■ Division rates after initialization

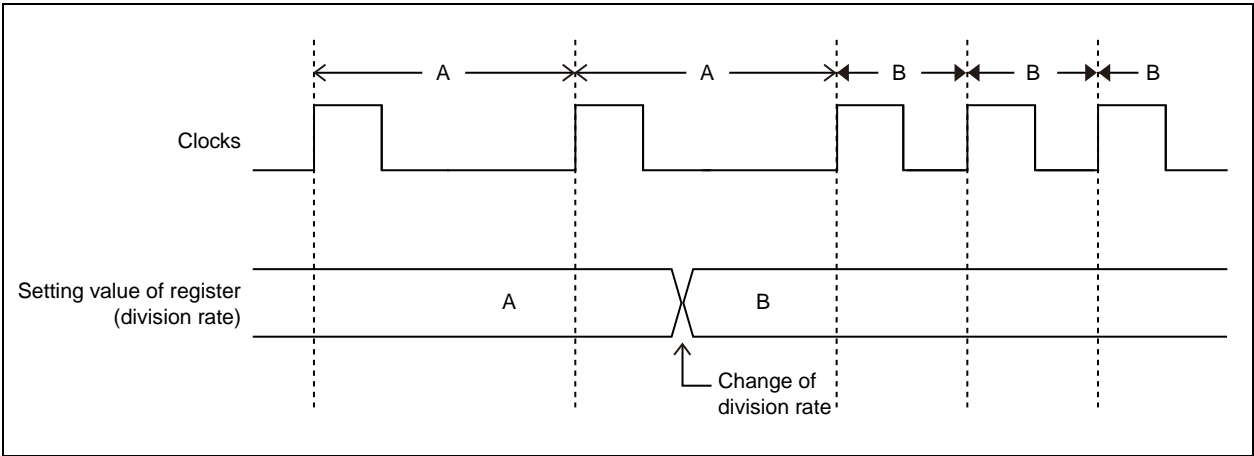
Table 8.5-1 shows the division of internal clocks after a reset.

Table 8.5-1 Division rates after a reset

Clock Name	Division Rate after Initialization
Base clock (BCLK)	Source clock (SRCCLK) divided by 1 (undivided)
CPU clock (CCLK)	Base clock (BCLK) divided by 1 (undivided)
On-chip bus clock (HCLK)	Base clock (BCLK) divided by 1 (undivided)
External output clock (TCLK)	Base clock (BCLK) divided by 2
Peripheral clock (PCLK)	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.



CHAPTER 9 Main Timer

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

- 9.1 Overview
- 9.2 Configuration
- 9.3 Registers
- 9.4 Interrupts
- 9.5 An Explanation of Operations and Setting Procedure Examples

9.1 Overview

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

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

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

When main clock (MAINCLK) 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 (MAINCLK) 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 (MAINCLK) and PLL clock (PLLCLK).

9.2 Configuration

This section explains the main timer configuration.

■ Main timer block diagram

For the main timer block diagram, see "■ Main clock (MAINCLK) generating part" in "CHAPTER 7 Clock Generating Parts".

■ Clocks

Table 9.2-1 shows the clocks used by the main timer.

Table 9.2-1 Clocks used by the main timer

Clock Name	Description
Operation clock	Main clock (MAINCLK)

9.3 Registers

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

■ Registers of main timer

Table 9.3-1 shows the registers used by the main timer.

Table 9.3-1 Main timer registers

Abbreviated Register Name	Register Name	Reference
MTMCR	Main timer control register	9.3.1

9.3.1 Main Timer Control Register (MTMCR)

This register controls the main timer.

Figure 9.3-1 shows the bit configuration of the main timer control register (MTMCR).

Figure 9.3-1 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 (MAINCLK) 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 "CHAPTER 11 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 (MAINCLK) 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 (MAINCLK) 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 (MAINCLK) 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.
- When the following operation is executed, the main timer will run until the MTC bit becomes 0.
 - MTE = 1 → 0 (rewriting)
 - MTC = 1 (writing)

During the above operation, the bit may become "1".

[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 (MAINCLK) 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 (MAINCLK) is oscillating (the MCEN bit of the clock source select register (CSELR) is 1).
 - The main clock (MAINCLK) 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 (MAINCLK) 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	16 MHz
1	0	0	0	$2^9 \times$ Main clock cycle	32 μ s
1	0	0	1	$2^{10} \times$ Main clock cycle	64.0 μ s
1	0	1	0	$2^{11} \times$ Main clock cycle	128.0 μ s
1	0	1	1	$2^{12} \times$ Main clock cycle	256.0 μ s
1	1	0	0	$2^{13} \times$ Main clock cycle	512.0 μ s
1	1	0	1	$2^{14} \times$ Main clock cycle	1024 μ s
1	1	1	0	$2^{15} \times$ Main clock cycle	2048 μ s
1	1	1	1	$2^{16} \times$ Main clock cycle	4096 μ s

Always write "1" to the MTS3 bit.

<Note>

Change the values of these bits after stopping the main timer using the MTE bit (MTE = 0).

9.4 Interrupts

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

Table 9.4-1 outlines the interrupts that can be used with the main timer.

Table 9.4-1 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 "APPENDIX C 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 "CHAPTER 12 Interrupt Controller".

9.5 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.

9.5.1 Main Timer Operation

■ Overview

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

When main clock (MAINCLK) 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 (MAINCLK) 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 (MAINCLK).
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).

The MTIF bit of the main timer control register (MTMCR) changes to "1".

If the MTIE bit of the main timer control register (MTMCR) is "1" at this time, a main timer interrupt request is generated.

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 (MAINCLK) 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 (MAINCLK) 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 (MAINCLK) oscillation is oscillating (the MCEN bit of the clock source select register (CSELR) is 1).
- The main clock (MAINCLK) 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.

9.5.2 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 (MAINCLK) 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.

CHAPTER 10 Low-power Dissipation Mode

This chapter explains the functions and operations of low-power dissipation mode.

- 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

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 two modes can be selected.
 - Main timer mode
This mode stops all the operations other than the main clock oscillation.
 - Stop mode
This mode stops all operations including the oscillation of all clocks.

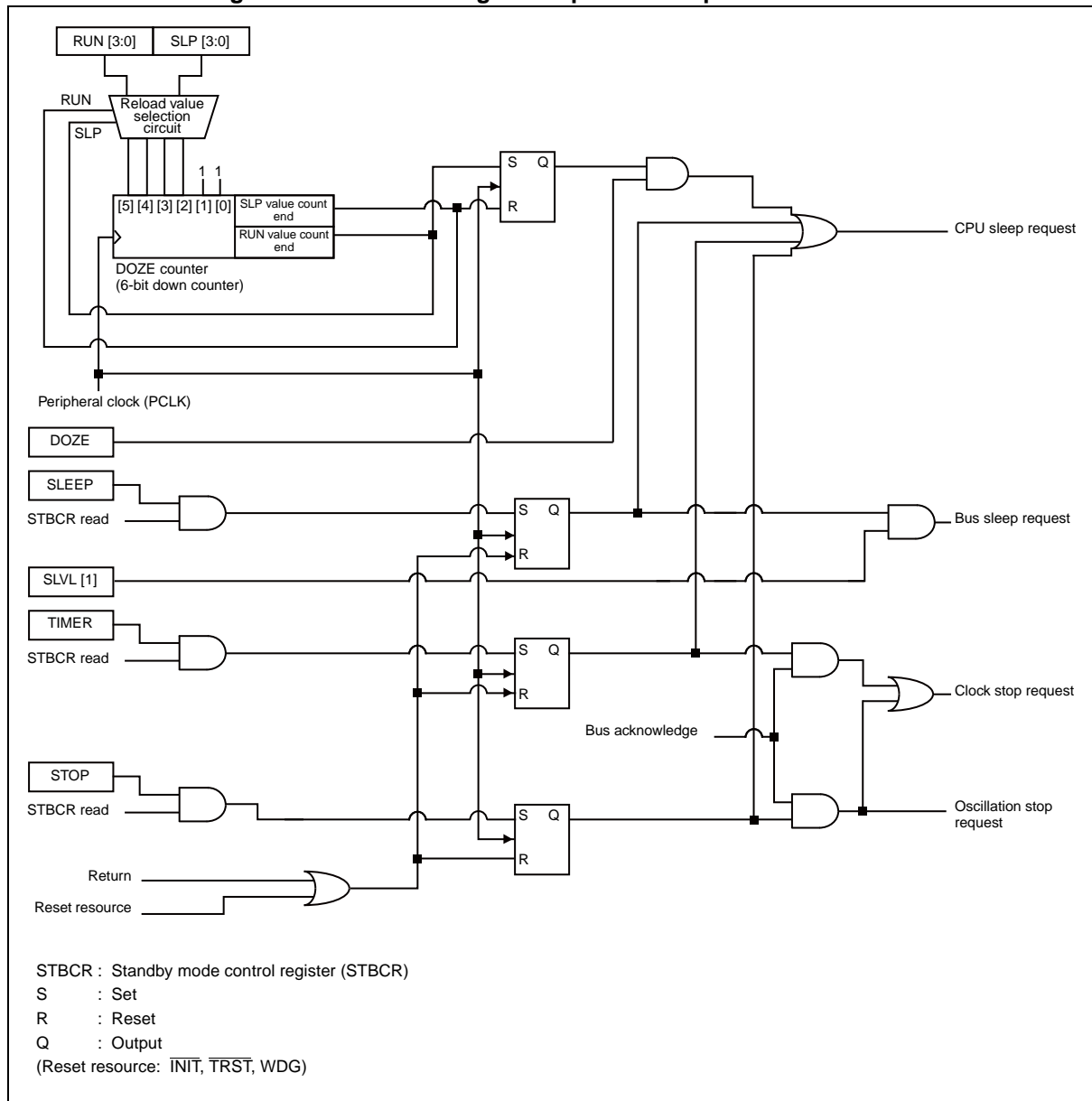
10.2 Configuration

The configuration of the power dissipation controller is shown below.

■ Block diagram of power dissipation controller

Figure 10.2-1 is a block diagram of the power dissipation controller.

Figure 10.2-1 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 10.2-1 shows the clock used in the power dissipation controller.

Table 10.2-1 Clock used in power dissipation controller

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-

10.3 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 10.3-1 is a list of registers that control power dissipation.

Table 10.3-1 List of registers that control power dissipation

Abbreviated Register Name	Register Name	Reference
STBCR	Standby mode control register	10.3.1
SLPRR	Sleep rate configuration register	10.3.2

10.3.1 Standby Mode Control Register (STBCR)

This register controls low-power dissipation mode.

Figure 10.3-1 shows the bit configuration of the standby mode control register (STBCR).

Figure 10.3-1 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 "10.4.5 Operation in Stop Mode".

[bit6]: TIMER (Main timer mode enable bit)

This bit enables transition to main timer mode.

Written Value	Explanation
0	Does not transit to main timer mode.
1	Transits to main timer mode.

If this register is read after this bit enables transition to main timer mode, power dissipation mode moves to main timer 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 is enabled by writing "1" to this bit.

If the return resource from main timer 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 "10.4.4 Operation in Main Timer 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 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 "10.4.3 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 "10.4.2 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 varies depending on the low-power dissipation mode.

Low-power Dissipation Mode	SLVL1	SLVL0	Explanation
Stop mode/ Main timer mode	0	0	Does not place the output from each pin in Hi-Z in stop mode/main timer mode.
	0	1	
	1	0	Places the output from each pin in Hi-Z in stop mode/main timer 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	
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	

* 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, see "APPENDIX D Pin State in Each CPU State".
- The setting value of SLVL0 bit has no effect on the operation.

10.3.2 Sleep Rate Configuration Register (SLPRR)

This register configures the operation state (RUN state) rate and sleep state rate (sleep rate) in doze mode.

Figure 10.3-2 shows the bit configuration of the sleep rate configuration register (SLPRR).

Figure 10.3-2 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.

$$(\text{Value of this bit} + 1) \times 4 \times t_{\text{CYCP}}$$

t_{CYCP} : Period of the peripheral clock (PCLK)

For details of operation period, see "10.4.2 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.

$$(\text{Value of this bit} + 1) \times 4 \times t_{\text{CYCP}}$$

t_{CYCP} : Period of the peripheral clock (PCLK)

For details of the sleep state period, see "10.4.2 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.
-

10.4 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 two modes can be selected.

- Main timer mode

- Stop mode

10.4.1 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 10.4-1 shows each operation clock and settable division ratio.

Table 10.4-1 Operation clock and settable division ratio

Operation Clock	Division Ratio
Base clock (BCLK)	Source clock (SRCCLK) divided by 1 to 8.
External output clock (TCLK)	Base clock (BCLK) divided by 1 to 8.
Peripheral clock (PCLK)	Base clock (BCLK) divided by 1 to 16.

<Note>

The division method or condition differs depending on the operation clock. For information on the division of the operation clock, see "CHAPTER 8 Clock Division Control Part".

■ Stopping the clock

You can reduce power dissipation by stopping the unused operation clock.

Table 10.4-2 shows the relationship between the operation clock that can be stopped and the deliver/stop timing.

Table 10.4-2 Relationship between the operation clock that can be stopped and the deliver/stop timing

Operation Clock	Deliver/Stop Timing
External output clock (TCLK)	Bus in sleep mode

Enabling the stop of the external output clock (TCLK) automatically disables the external output clock (TCLK) delivery during the period in which there is no access by using the external bus.

If an access is attempted, clock delivery is resumed automatically and delivery is disabled again after access is completed. For information on conditions for disabling external output clock (TCLK), see "CHAPTER 8 Clock Division Control Part".

10.4.2 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.

$$(\text{RUN} + 1) \times 4 \times t_{\text{CYCP}} + (\text{SLP} + 1) \times 4 \times t_{\text{CYCP}}$$

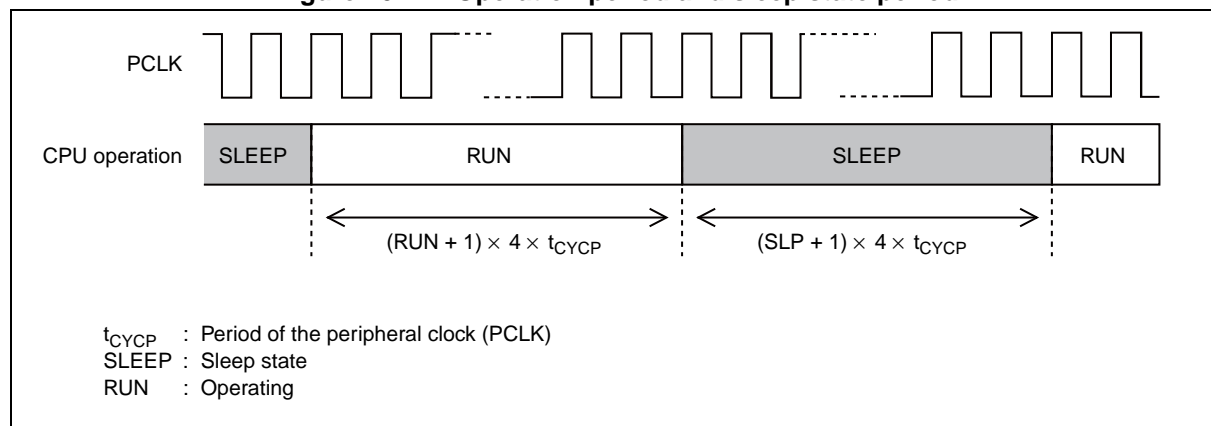
RUN: Value for the RUN3 to RUN0 bits

SLP: Value for the SLP3 to SLP0 bits

t_{CYCP} : Period of the peripheral clock (PCLK)

Figure 10.4-1 shows each cycle.

Figure 10.4-1 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, or stop mode.

10.4.3 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 "CHAPTER 8 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.

You can set whether to disable external output clock (TCLK) delivery in bus sleep mode, using the TSTP bit in the divide clock configuration register 1 (DIVR1).

For information on the divide clock configuration register 1 (DIVR1), see "8.4.2 Divide Clock Configuration Register 1 (DIVR1)".

While this mode is slower in responding to the DMA transfer request than in CPU sleep mode, it can reduce power dissipation.

■ Setting

Table 10.4-3 shows the settings required before changing to sleep mode.

Table 10.4-3 Setting register

Registers	Bit	Explanation
Divide clock configuration register 1 (DIVR1)	TSTP	Sets whether to enable the external output clock (TCLK) delivery 0 = Enabling 1 = Disabling
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

<Note>

If the external output clock (TCLK) delivery is disabled by setting the TSTP bit (TSTP =1) in divide clock configuration register 1 (DIVR1), DMA transfer cannot be activated by the external DMA transfer request.

■ 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.
- A NMI request is generated.
- An interrupt request is generated (whose interrupt level is other than "31").

For information on the interrupt level, see "CHAPTER 12 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.

10.4.4 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 (MAINCLK) 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 (MAINCLK) oscillation is permitted as a source clock (SRCCLK) for the CPU, the count operation of the main timer is executed.

■ Setting

Table 10.4-4 shows the settings required before changing to main timer mode.

Table 10.4-4 Setting register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects main clock (MAINCLK) for the CPU source clock (SRCCLK) (CKS1, CKS0=00 or 01)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
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
 - External interrupt

For the interrupt level, see "CHAPTER 12 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.

10.4.5 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

Table 10.4-5 shows the settings required before changing to stop mode.

Table 10.4-5 Setting register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects the main clock (MAINCLK) 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.
- A NMI request is generated.
- Below interrupt requests are generated (whose interrupt level is other than "31").

External interrupt

For information on the interrupt level, see "CHAPTER 12 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.

10.5 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
 - 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.

CHAPTER 11 Reset

This chapter explains the functions and operations of reset.

- 11.1 Overview
- 11.2 Configuration
- 11.3 Pins
- 11.4 Registers
- 11.5 Explanation of Operations
- 11.6 Operating State and Transition

11.1 Overview

This section explains "reset" to initialize the internal circuit.

■ Overview

This device has the following four types of reset resource.

- $\overline{\text{TRST}}$ pin input
- $\overline{\text{INIT}}$ pin input
- Watchdog reset 0
- 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.

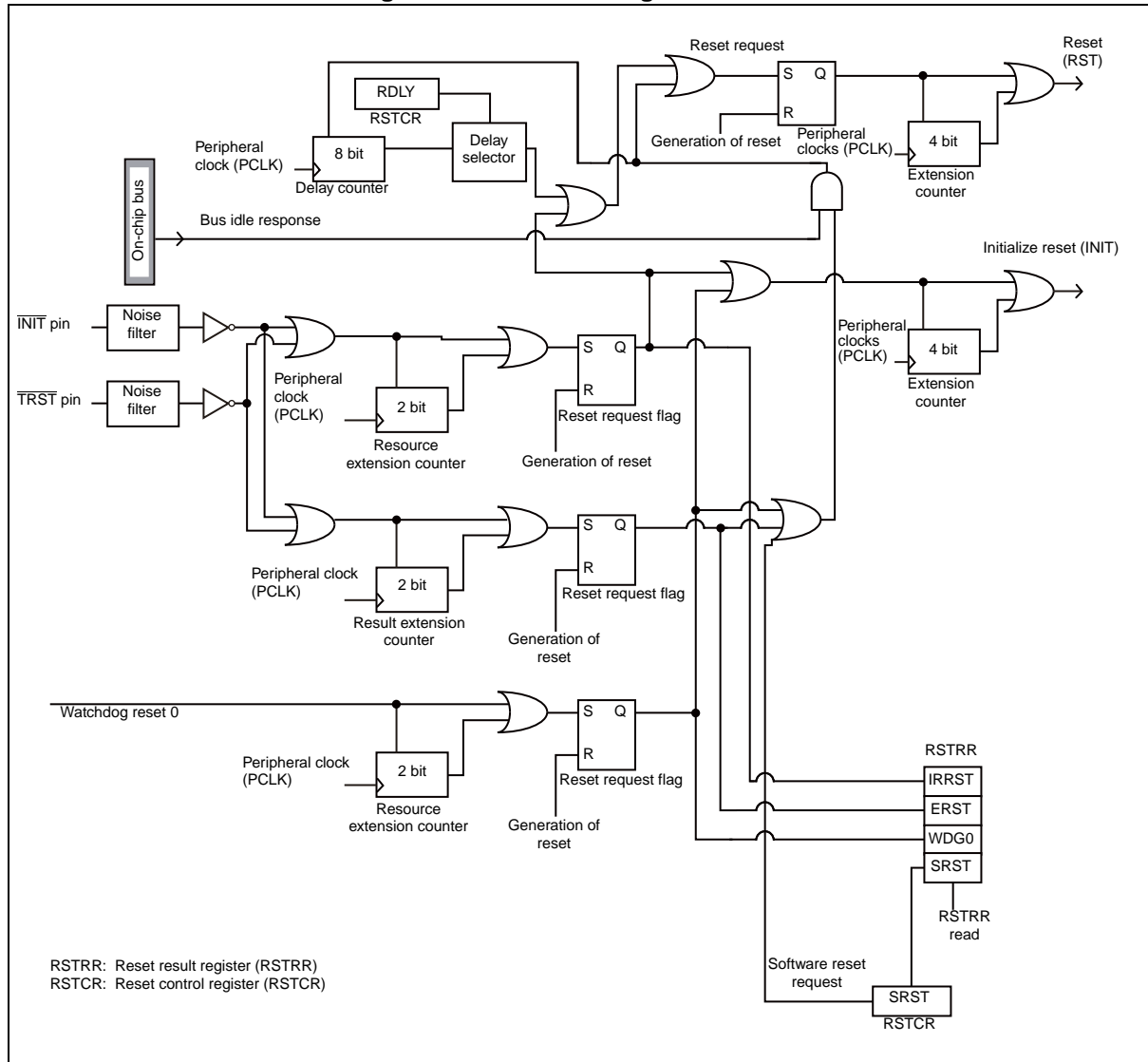
11.2 Configuration

The configuration of reset is shown.

■ Block diagram of reset

Figure 11.2-1 is a block diagram of reset.

Figure 11.2-1 Block diagram of reset



- 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 11.2-1 shows clocks to be used for reset.

Table 11.2-1 Clocks used for reset

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

11.3 Pins

This section explains the pins that are used for reset.

■ Overview

The following pins are used for reset.

- $\overline{\text{INIT}}$ pin, $\overline{\text{TRST}}$ pin

The external input pins are used to input the reset request.

11.4 Registers

This section explains the configuration and functions of registers used for reset.

■ List of registers used for reset

Table 11.4-1 shows the list of registers used for reset.

Table 11.4-1 List of registers used for reset

Abbreviated Register Name	Register Name	Reference
RSTRR	Reset result register	11.4.1
RSTCR	Reset control register	11.4.2

11.4.1 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 11.4-1 shows the bit configuration of the reset result register (RSTRR).

Figure 11.4-1 Bit configuration of the reset result register (RSTRR)

bit	7	6	5	4	3	2	1	0
	IRRST	ERST	Undefined	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
Software reset	XXXXXXX1
Timeout for software reset	1XXXXXX1
Register reading	00000000

X: Not initialized.

<Note>

If this register is read, all the bits are cleared.

[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 "■ Irregular reset" in "11.5.3 Operation of 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]: Undefined bit

In case of reading	A value is undefined.
--------------------	-----------------------

[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.

11.4.2 Reset Control Register (RSTCR)

This register controls issuing of reset.

Figure 11.4-2 shows the bit configuration of the reset control register (RSTCR).

Figure 11.4-2 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 "■ Irregular reset" in "11.5.3 Operation of 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 (MAINCLK) divided by 2.
-

11.5 Explanation of Operations

This section explains the operation of reset.

11.5.1 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 (MAINCLK)
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
Target of initialization	- Source clock = Main clock (MAINCLK) divided by 2 - Clock oscillation = Main clock oscillates, 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
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 - 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

11.5.2 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.

- $\overline{\text{TRST}}$ pin input (TRST)

This pin input is hardware reset from the ICE while connecting with the ICE.

This pin input is identified as the $\overline{\text{INIT}}$ pin input (INIT) inside the device.

- $\overline{\text{INIT}}$ 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 elapses. If the watchdog reset 0 request is generated, the initialize reset (INIT) is issued.

Generation source	The period set for the watchdog timer elapses
Cancellation source	Automatically cancelled 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	<ul style="list-style-type: none"> - 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).

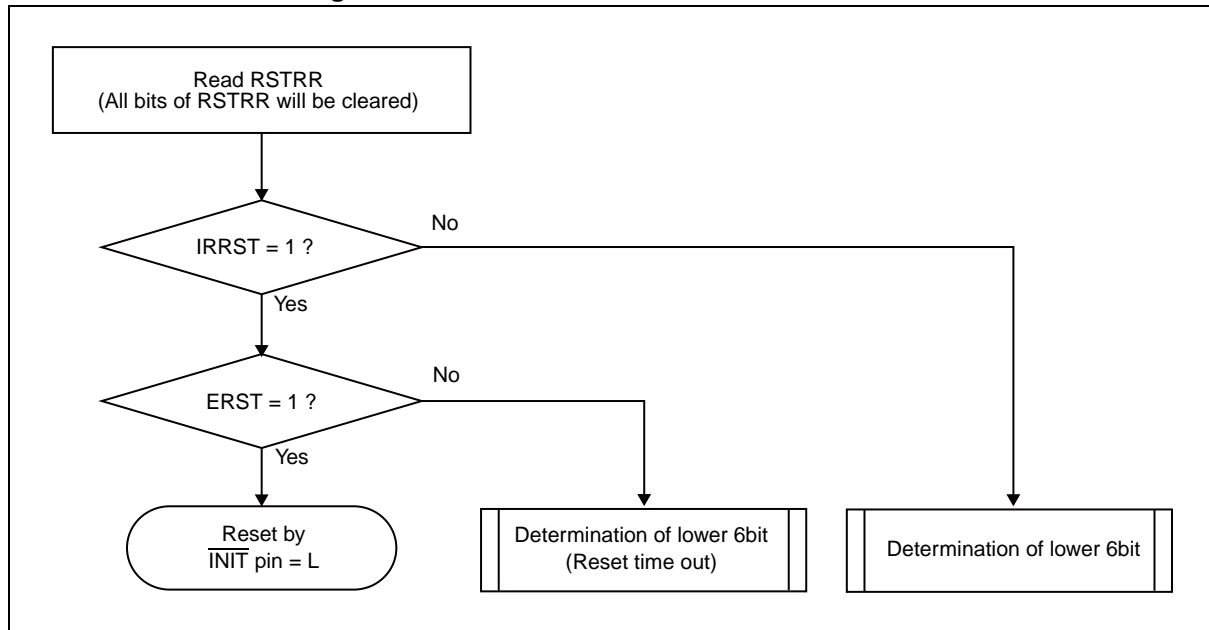
- Software reset (RSTCR:SRST)

If the reset control register (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).
Cancellation source	Automatically cancelled after the reset (RST) is issued.
Reset level	Issues only reset (RST)
Corresponding flag	SRST bit of the reset result register (RSTRR) = 1
Operation	<ul style="list-style-type: none"> - 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 11.5-1 Flow of reset result determination



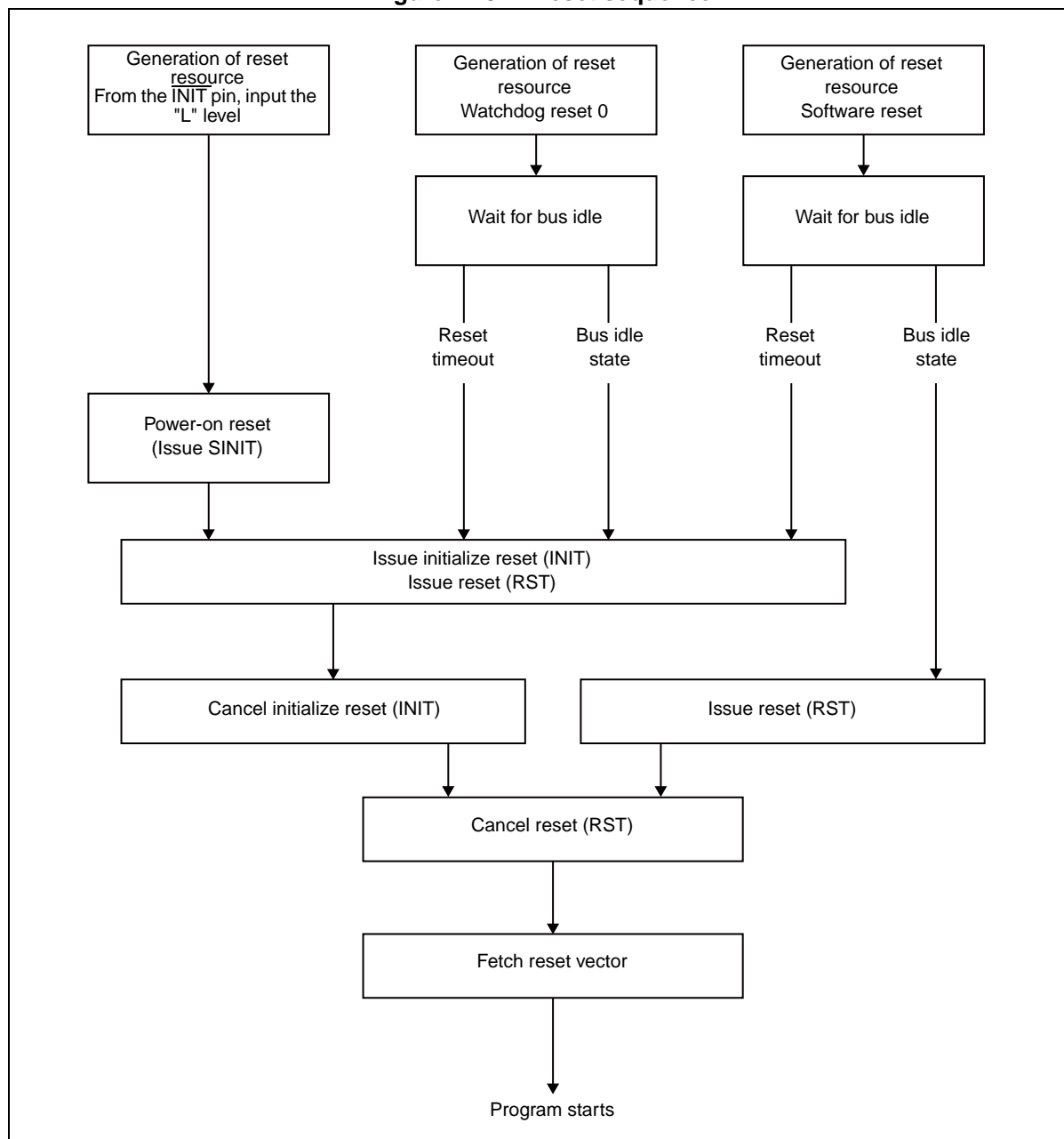
11.5.3 Operation of Reset

■ 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 11.5-2 shows the reset sequence.

Figure 11.5-2 Reset sequence

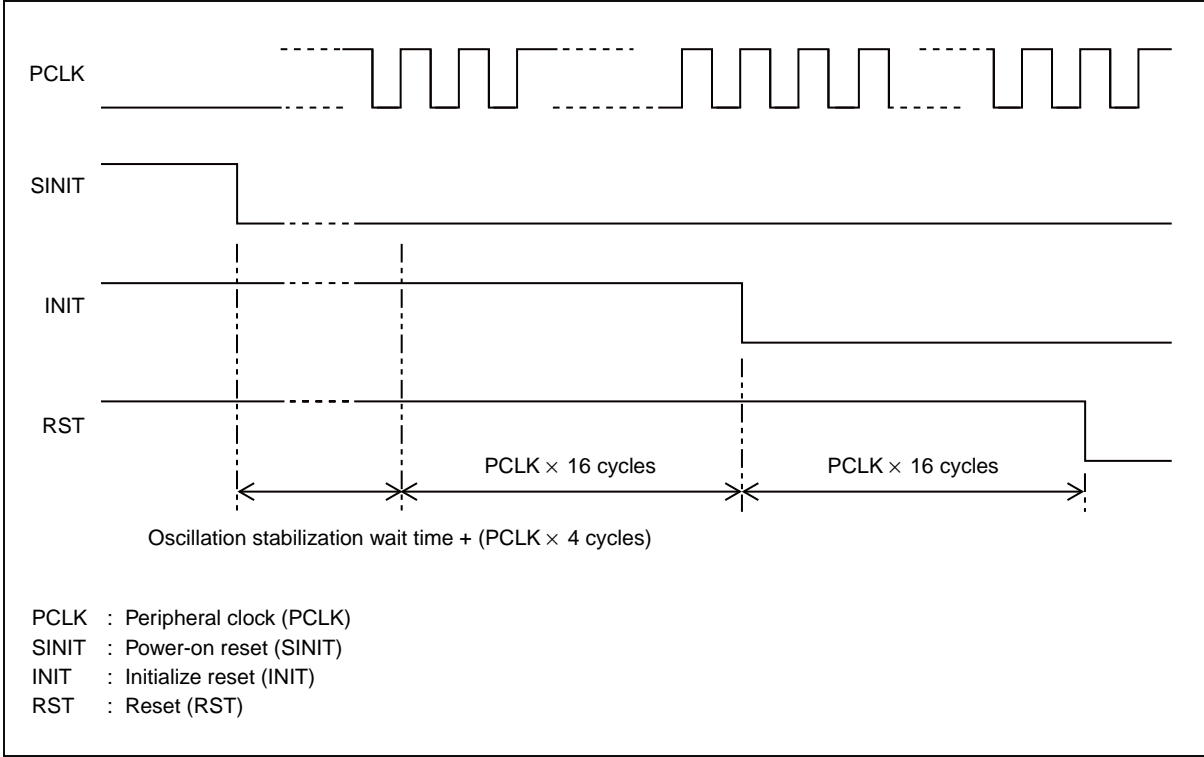


1. Retrieval and extension of reset resource
The generated reset resource is asynchronously retrieved and retained until reset is issued.
2 bits of resource extension counter retains the reset resource for at least 4Ts (T: Peripheral clock (PCLK) period).
2. Generation of the reset request
Reports the generated reset request to the internal bus controller to perform the following processing.
 - Stops the program operation of the CPU (same as for sleep mode).
 - 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
 - Input "L" level to $\overline{\text{INIT}}$ pin
Issues a power-on reset (SINIT), initialize reset (INIT), and reset (RST).
 - Watchdog reset 0
Issues initialize reset (INIT) and reset (RST).
 - Reset time out
Issues initialize reset (INIT) and reset (RST).
 - Software reset (RSTCR:SRST)
Issues reset (RST).
5. Cancellation of reset resource
If the reset resource is cancelled, 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 cancelled 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 11.5-3 shows the respective reset issue sequence after the reset resource of the power-on reset (SINIT) is cancelled.

Figure 11.5-3 Each reset issue sequence after the reset resource of the power-on reset (SINIT) is cancelled

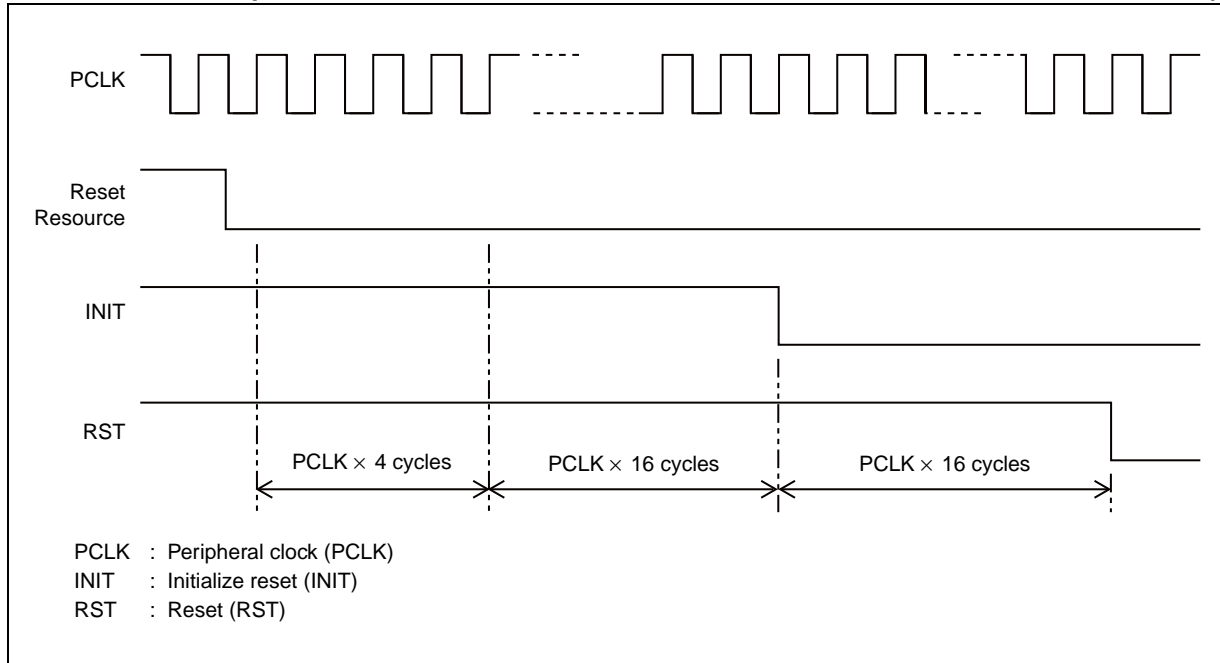


■ Initialize reset (INIT)

When initialize reset (INIT) is issued, reset (RST) is also issued at the same time.

Figure 11.5-4 shows the issue sequence of the respective resets after the reset resource of initialize reset (INIT) is cancelled.

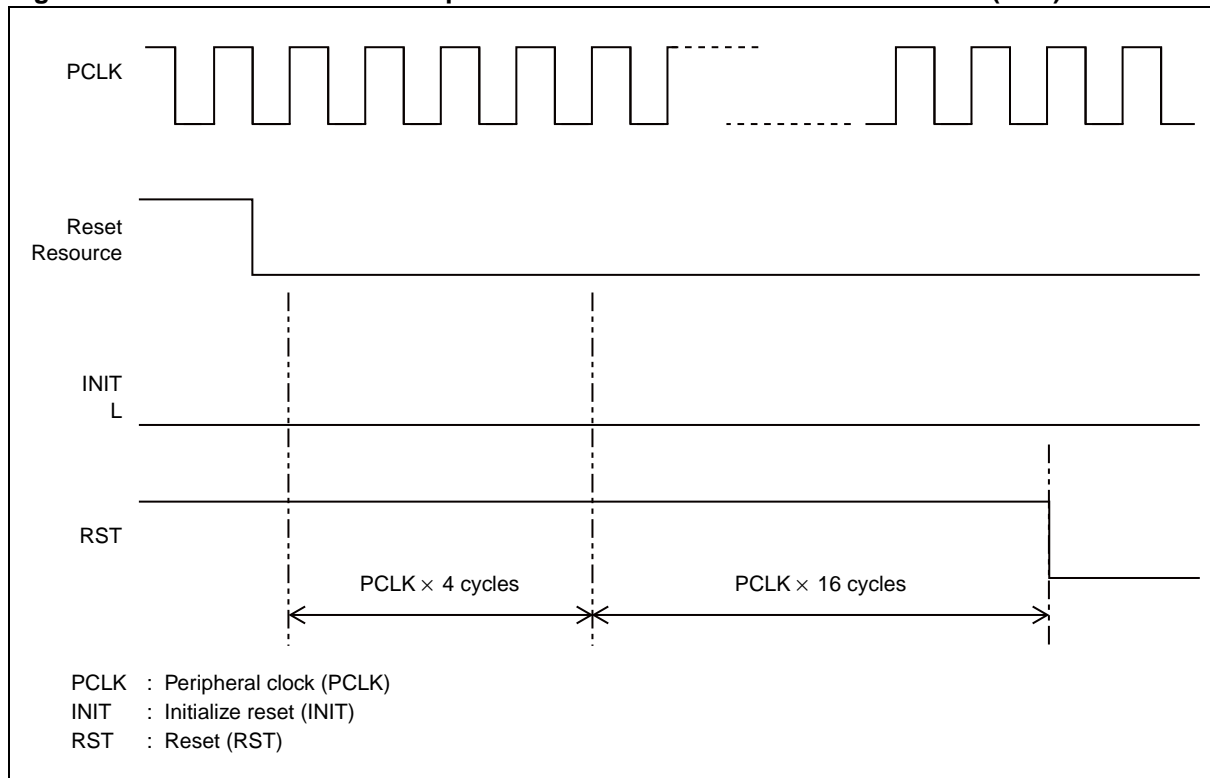
Figure 11.5-4 Issue sequence of each reset after cancellation of the reset resource of initialize reset (INIT)



■ Reset (RST)

Figure 11.5-5 shows the respective reset issue sequence after the reset resource of reset (RST) is cancelled.

Figure 11.5-5 Each reset issue sequence after the reset resource of the reset (RST) is cancelled



■ 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 / 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.

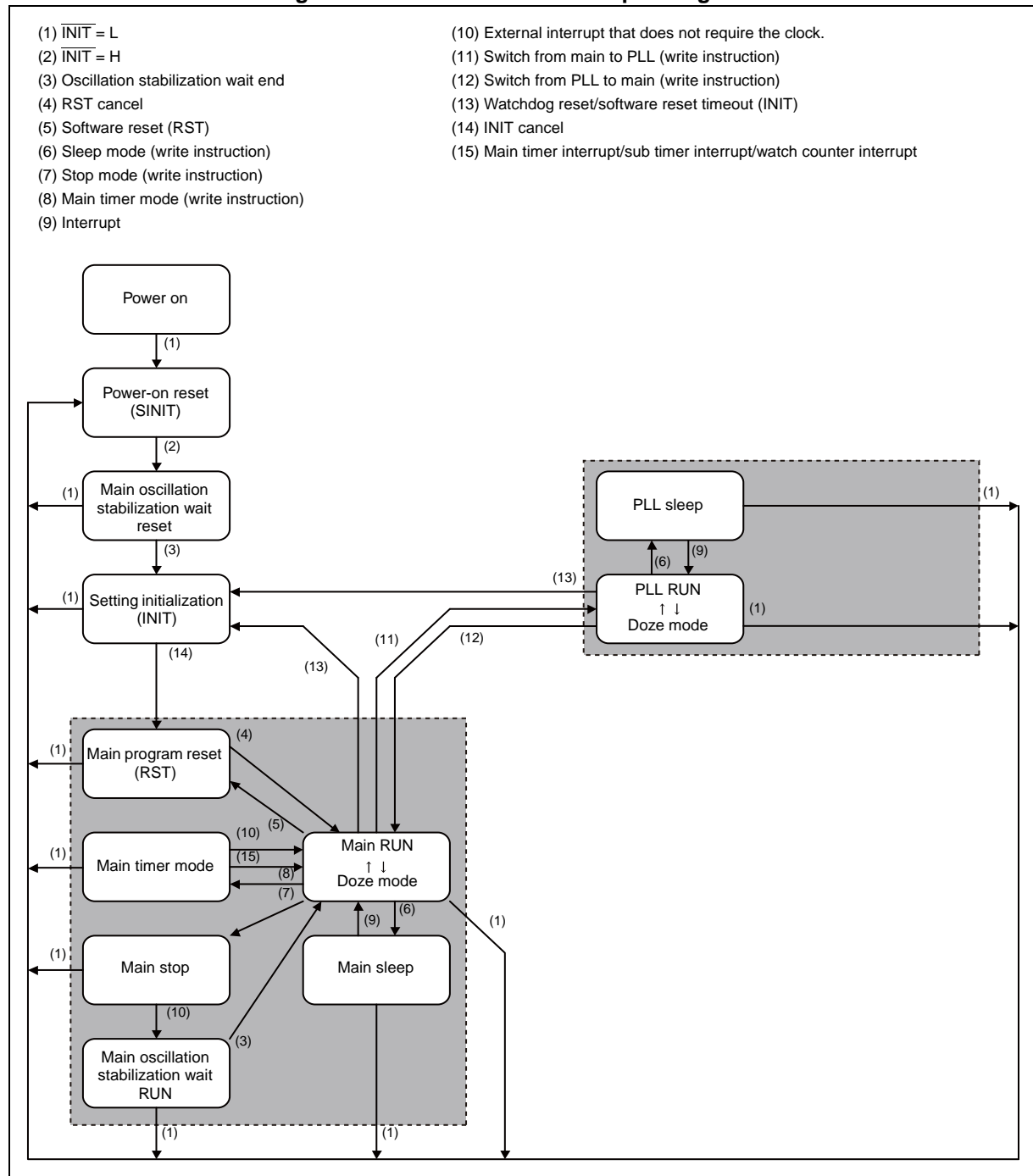
11.6 Operating State and Transition

This section explains each operating state and how to control it.

■ Operating state

Figure 11.6-1 shows transition of the operating state.

Figure 11.6-1 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 and main timer mode state is cancelled.

● 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.

● 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 (MAINCLK)) 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 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 (MAINCLK) oscillation circuit operates (while the 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 cancelled 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.

<div>Highest priority</div> <div><div></div></div> <div>Lowest priority</div>	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 main timer mode state
	Stop mode request (register write)	Occurs only in the RUN state
	Main timer mode request (register write)	Occurs only in the RUN state
	Sleep mode request (register write)	Occurs only in the RUN state

CHAPTER 12 Interrupt Controller

This chapter explains the functions and operations of the interrupt controller.

- 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 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 a NMI request/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 a NMI/interrupt level other than "1111".

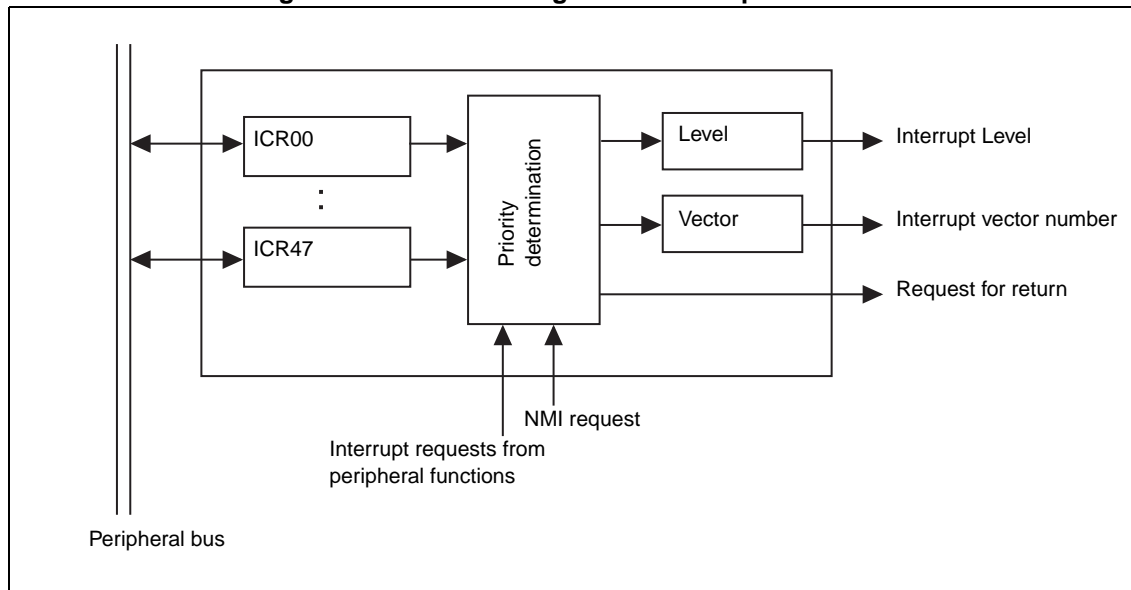
12.2 Configuration

This section explains the interrupt controller configuration.

■ Block diagram of interrupt controller

Figure 12.2-1 shows a block diagram of the interrupt controller.

Figure 12.2-1 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)

12.3 Registers

This section explains the configurations and functions of the registers used by the interrupt controller.

■ Interrupt controller registers

Table 12.3-1 lists the interrupt controller registers.

Table 12.3-1 Interrupt controller registers

Abbreviated Register Name	Register Name	Reference
ICR00 to ICR47	Interrupt control registers 00 to 47	12.3.1

12.3.1 Interrupt Control Register (ICR00 to ICR47)

These registers are used to set interrupt levels. This register is provided for input of each interrupt.

Figure 12.3-1 shows the bit configuration of the interrupt control registers (ICR00 to ICR47).

Figure 12.3-1 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.
- When a NMI request occurs, the interrupt level is 15 ("01111_B").
- 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 "APPENDIX C Interrupt Vectors".
- IL4 bit is fixed to "1" and IL3 to IL0 can be set.

12.4 An Explanation of Operations and Setting Procedure Examples

This section explains the operations of the interrupt controller.

12.4.1 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 "APPENDIX C 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 request a NMI?
2. Is the interrupt level of the interrupt request "30" or lower (Level 31 is "Interrupt Disabled").
3. Is the value of the interrupt level of the interrupt request the smallest.
4. 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 a NMI or external 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 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).

12.5 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.

CHAPTER 13 NMI Input

This chapter describes the operation of the NMI input function.

13.1 Overview

13.2 Explanation of Operations

13.1 Overview

NMI (Non Maskable Interrupt) is an interrupt input from the $\overline{\text{NMI}}$ pin that cannot be masked.

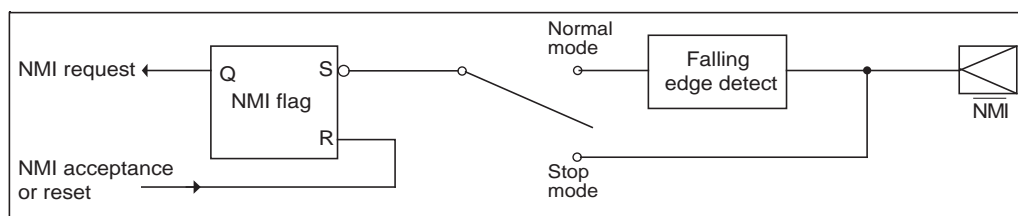
■ Overview

NMI can be used as a trigger to return from stop mode.

■ Block Diagram

Figure 13.1-1 shows the block diagram of NMI.

Figure 13.1-1 NMI Block Diagram



13.2 Explanation of Operations

This section describes the operation of NMI.

■ NMI

NMI is the highest level of user interrupt and cannot be masked. As an exception, NMI is masked after a reset until the CPU sets the interrupt level mask register (ILM).

■ NMI Request Acceptance Level

NMI acceptance level is when "L level is detected" during stop mode and when "falling edge is detected" during any other mode.

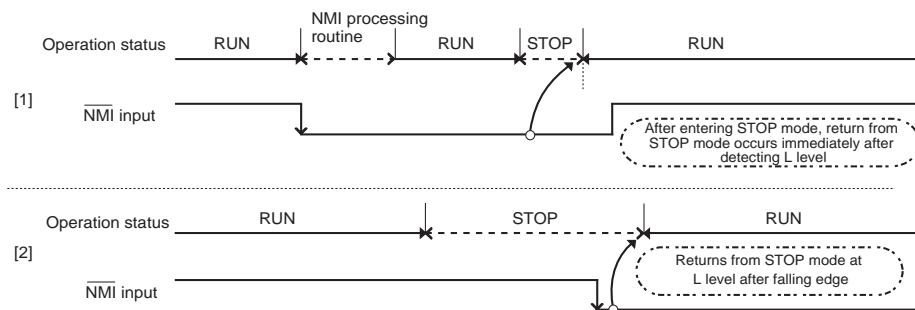
■ Interrupt Request Output

The NMI request detection unit has a NMI flag which is reset with a NMI request and is cleared only when it receives a NMI interrupt itself or during reset. The NMI flag is read/write disabled.

■ Return from Stop Mode

Return from stop mode occurs when the $\overline{\text{NMI}}$ pin input is at "L" level and a NMI request is output to the interrupt controller during stop mode. After NMI processing routine completes in normal state (not in stop mode), if stop mode is entered without returning the $\overline{\text{NMI}}$ pin to "H" level, return from stop mode occurs immediately ([1] in Figure 13.2-1). Return the $\overline{\text{NMI}}$ pin to "H" level before entering stop mode and set the $\overline{\text{NMI}}$ pin to "L" level while in stop mode ([2] in Figure 13.2-1).

Figure 13.2-1 Return from Stop Mode



CHAPTER 14 Interrupt Request Batch-Read Function

This section explains the interrupt request batch-read function.

- 14.1 Overview
- 14.2 Configuration
- 14.3 Registers
- 14.4 Notes on Use

14.1 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 that this function cannot clear the interrupt request flag. Use the register of each peripheral function to clear the interrupt request flag.

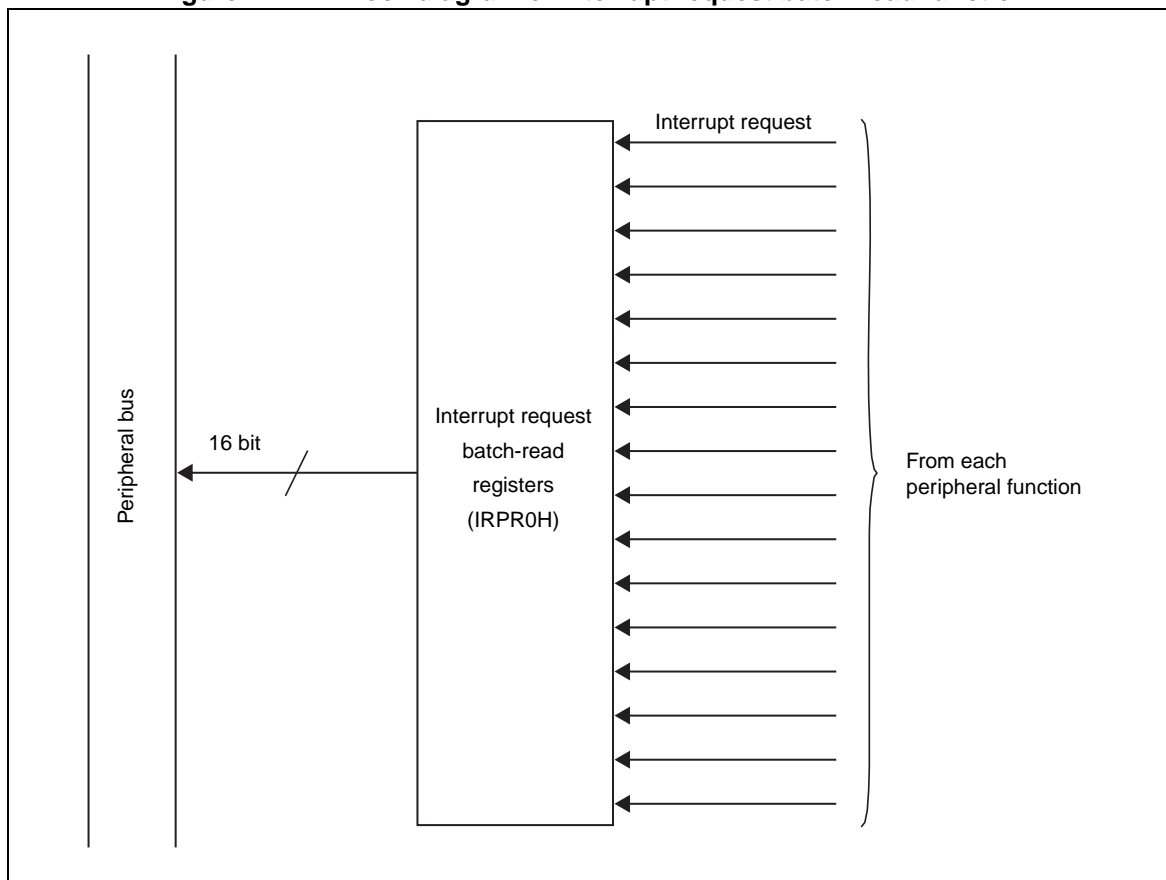
14.2 Configuration

This section shows the configuration of the interrupt request batch-read function.

■ Block diagram of interrupt request batch-read function

Figure 14.2-1 is a block diagram of the interrupt request batch-read function.

Figure 14.2-1 Block diagram of interrupt request batch-read function



■ Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

14.3 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 14.3-1 lists the registers for the interrupt request batch-read function.

Table 14.3-1 Registers for the interrupt request batch-read function

Abbreviated Register Name	Register Name	Reference
IRPR0H	Interrupt request batch-read register 0 upper	14.3.1

14.3.1 Interrupt Request Batch-Read Register 0 Upper (IRPR0H)

Interrupt vector number 57 (decimal) is used for base timer channels ch.8/ch.9/ch.10/ch.11. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 14.3-1 shows the bit configuration of interrupt request batch-read register 0 upper (IRPR0H).

Figure 14.3-1 Bit configuration of Interrupt request batch-read register 0 upper (IRPR0H)

bit	15	14	13	12	11	10	9	8
	BT0IR8	BT1IR8	BT0IR9	BT1IR9	BT0IR10	BT1IR10	BT0IR11	BT1IR11
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	BT0IR8	0	No interrupt request 0 generated on base timer ch.8
		1	Interrupt request 0 generated on base timer ch.8
bit14	BT1IR8	0	No interrupt request 1 generated on base timer ch.8
		1	Interrupt request 1 generated on base timer ch.8
bit13	BT0IR9	0	No interrupt request 0 generated on base timer ch.9
		1	Interrupt request 0 generated on base timer ch.9
bit12	BT1IR9	0	No interrupt request 1 generated on base timer ch.9
		1	Interrupt request 1 generated on base timer ch.9
bit11	BT0IR10	0	No interrupt request 0 generated on base timer ch.10
		1	Interrupt request 0 generated on base timer ch.10
bit10	BT1IR10	0	No interrupt request 1 generated on base timer ch.10
		1	Interrupt request 1 generated on base timer ch.10
bit9	BT0IR11	0	No interrupt request 0 generated on base timer ch.11
		1	Interrupt request 0 generated on base timer ch.11
bit8	BT1IR11	0	No interrupt request 1 generated on base timer ch.11
		1	Interrupt request 1 generated on base timer ch.11

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

14.4 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) is disabled. To cancel an interrupt request, clear the interrupt request flag bit of the corresponding function register.

CHAPTER 15 Delay Interrupt

This chapter explains the functions and operations of the delay interrupt function.

- 15.1 Overview
- 15.2 Configuration
- 15.3 Registers
- 15.4 An Explanation of Operations and Setting Procedure
Examples
- 15.5 Notes on Use

15.1 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.

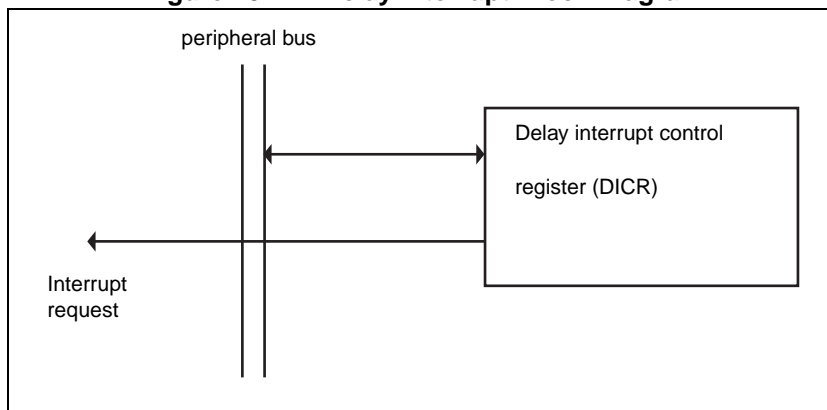
15.2 Configuration

This section explains the configuration of delay interrupts.

■ Delay Interrupt Block Diagram

Figure 15.2-1 shows a delay interrupt block diagram.

Figure 15.2-1 Delay Interrupt Block Diagram



- Delayed interrupt control register (DICR)
This register controls delay interrupts.

■ Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

15.3 Registers

This section explains the configuration and functions of the register used for delay interrupts.

■ Delay interrupt register

Table 15.3-1 shows the delay interrupt register.

Table 15.3-1 Delay interrupt register

Abbreviated Register Name	Register Name	Reference
DICR	Delayed interrupt control register	15.3.1

15.3.1 Delayed Interrupt Control Register (DICR)

This register controls delay interrupts.

Figure 15.3-1 shows the bit configuration of the delayed interrupt control register (DICR).

Figure 15.3-1 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.

15.4 An Explanation of Operations and Setting Procedure Examples

This section explains delay interrupt operations and the setting procedure for delay interrupts.

15.4.1 Explanation of Delay Interrupt Operations

Software can use delay interrupts to generate interrupt requests to the CPU or cancel them.

Table 15.4-1 lists the conditions for generating delay interrupts.

Table 15.4-1 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 "APPENDIX C 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 "CHAPTER 12 Interrupt Controller".
-

15.5 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.

CHAPTER 16 External Bus Interface

This chapter describes the functions and operations of the external bus interface.

- 16.1 Overview
- 16.2 Configuration
- 16.3 Registers
- 16.4 Register Setting Examples
- 16.5 Notes on Use

16.1 Overview

The external bus interface is a bus interface to perform data In/Out that connects with external device (such as memory or I/O device).

■ Overview

The external bus interface has the following features:

Eight CS areas

$\overline{CS0}$ to $\overline{CS6}$: SRAM/FLASH area

- Support for 8-bit/16-bit bus width.
- 256 Mbytes address space.
- 256 Mbytes mirror space that can be set non-cacheable.
- Separate bus/multiplex bus support.
- Up to 128 Mbytes per area.
- RDY input is able to be set for each CS area.

$\overline{CS8}$: SDRAM dedicated area

- 64 Mbytes address space.
- 64 Mbytes mirror space that can be set non-cacheable.
- Support only 16-bit bus width.
- CAS latency: Selectable from 1 to 3.
- Page size: Up to 1 Kbytes.

Clock division function

1:1 to 1:4 clock settable for on-chip bus.

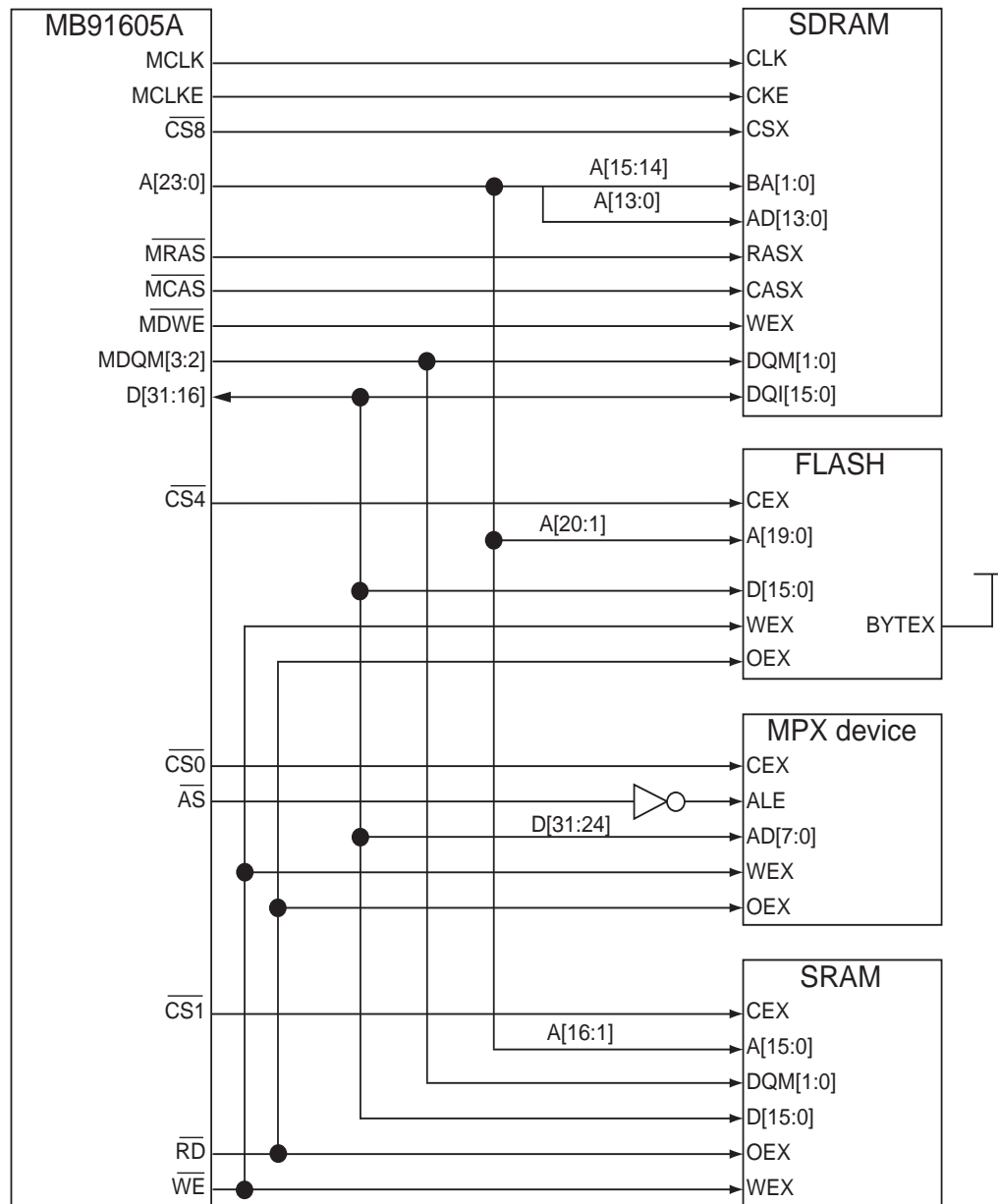
16.2 Configuration

This section describes the configuration of the external bus interface.

■ Connection example

Figure 16.2-1 shows a connection example of the external bus interface.

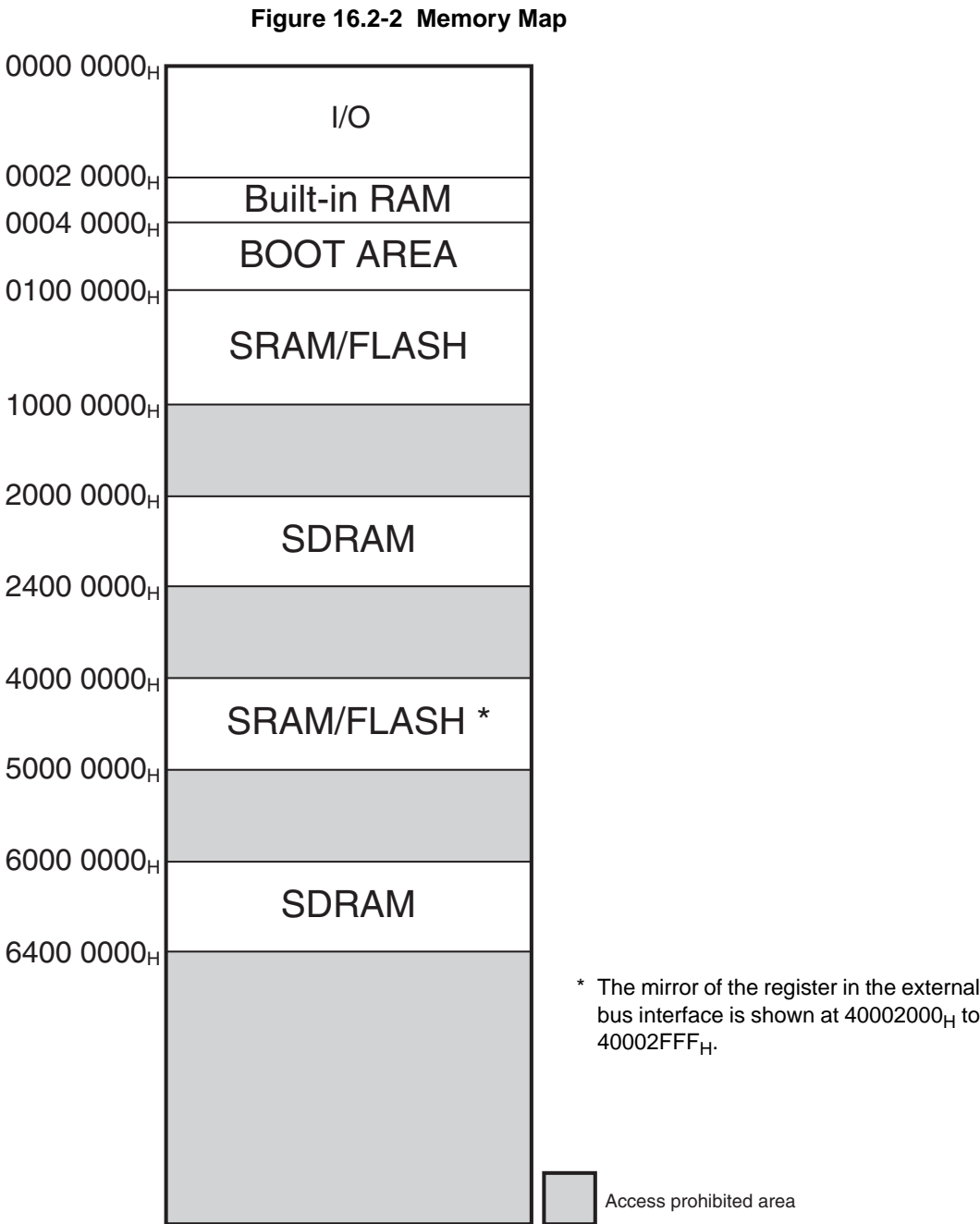
Figure 16.2-1 Connection Example of External Bus Interface



- *: • The BOOT area is assigned to $\overline{CS4}$ just after reset.
 • MDQM3 is a byte enable output pin for D[31:24], and MDQM2 is that for D[23:16].

■ Memory map

Figure 16.2-2 shows the memory map.



* The BOOT area is assigned to $\overline{CS4}$ just after reset.

16.3 Registers

This section describes the configurations and functions of registers used by the external bus interface.

■ List of registers for external bus interface

Table 16.3-1 shows a list of registers for external bus interface.

Table 16.3-1 List of Registers for External Bus Interface

SRAM/FLASH control register

Register Abbreviation	Register Name	See
MCMR0	SRAM/FLASH mode register 0	16.3.1
MCMR1	SRAM/FLASH mode register 1	16.3.1
MCMR2	SRAM/FLASH mode register 2	16.3.1
MCMR3	SRAM/FLASH mode register 3	16.3.1
MCMR4	SRAM/FLASH mode register 4	16.3.1
MCMR5	SRAM/FLASH mode register 5	16.3.1
MCMR6	SRAM/FLASH mode register 6	16.3.1
MCMR7	SRAM/FLASH mode register 7 (reserved)	16.3.1
MCTR0	SRAM/FLASH timing register 0	16.3.2
MCTR1	SRAM/FLASH timing register 1	16.3.2
MCTR2	SRAM/FLASH timing register 2	16.3.2
MCTR3	SRAM/FLASH timing register 3	16.3.2
MCTR4	SRAM/FLASH timing register 4	16.3.2
MCTR5	SRAM/FLASH timing register 5	16.3.2
MCTR6	SRAM/FLASH timing register 6	16.3.2
MCTR7	SRAM/FLASH timing register 7 (reserved)	16.3.2
MCAR0	SRAM/FLASH area register 0	16.3.3
MCAR1	SRAM/FLASH area register 1	16.3.3
MCAR2	SRAM/FLASH area register 2	16.3.3
MCAR3	SRAM/FLASH area register 3	16.3.3
MCAR4	SRAM/FLASH area register 4	16.3.3
MCAR5	SRAM/FLASH area register 5	16.3.3
MCAR6	SRAM/FLASH area register 6	16.3.3
MCAR7	SRAM/FLASH area register 7 (reserved)	16.3.3

SDRAM control register

Register Abbreviation	Register Name	See
SDMR	SDRAM mode register	16.3.4
SDRTR	SDRAM refresh timer register	16.3.5
SDPDR	SDRAM power down counter register	16.3.6
SDTR	SDRAM timing register	16.3.7
SDCMR	SDRAM command register	16.3.8

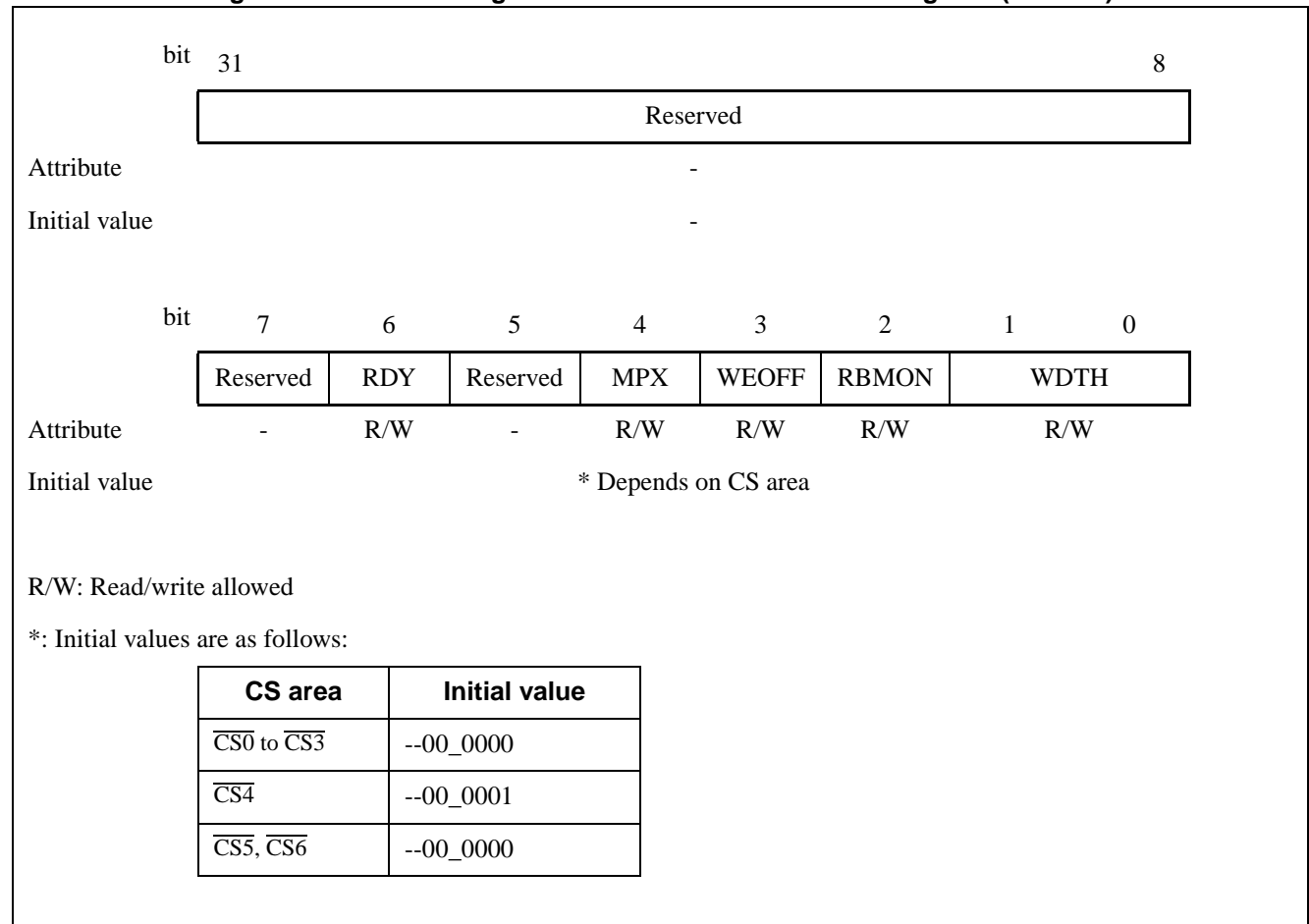
Clock control register

Register Abbreviation	Register Name	See
CLKCTL	Clock control register	16.3.9

16.3.1 SRAM/FLASH Mode Register (MCMRx)

Figure 16.3-1 shows the bit configuration of the SRAM/FLASH mode register (MCMRx).

Figure 16.3-1 Bit Configuration of SRAM/FLASH Mode Register (MCMRx)



<Note>

This register allows only 32-bit access.

[bit 31 to bit 7]: Reserved bits

[bit 6]: RDY

This bit enables external RDY.

Written Value	Description
0	RDY disabled
1	RDY enabled

<Note>

When enabling RDY, set RADC/WADC to 3 cycles or greater.

[bit 5]: Reserved bit

Write "0".

[bit 4]: MPX

This bit selects the multiplex bus mode.

Written Value	Description
0	Separate bus mode
1	Address/data multiplex mode

<Note>

When using asynchronous multiplex mode, set RADC/WADC to 2 cycles or greater in order to provide address output cycle.

[bit 3]: WEOFF

This bit controls \overline{WE} output.

Written Value	Description
0	\overline{WE} output enabled
1	\overline{WE} output disabled

<Note>

When \overline{WE} output is disabled with this bit, no coherency with cache is guaranteed during write to the corresponding CS area. Be careful when disabling output in cache area.

[bit 2]: RBMON

This bit enables byte mask signal MDQM output during read access.

Written Value	Description
0	MDQM output disabled
1	MDQM output enabled

- In case of writing
Byte enable is always enabled regardless of RBMON.
- In case of reading
Only when RBMON=1, byte enable is controlled by each byte.
When RBMON=0, all bits are enabled (all bits of the byte width are read).

[bit 1, bit 0]: WIDTH

These bits set the bus width of the corresponding CS area.

Written Value	Description
00	8 bits
01	16 bits
10	32 bits (forbidden to set)
11	Reserved (forbidden to set)

16.3.2 SRAM/FLASH Timing Register (MCTRx)

Figure 16.3-2 shows the bit configuration of the SRAM/FLASH timing register (MCTRx).

Figure 16.3-2 Bit Configuration of SRAM/FLASH Timing Register (MCTRx)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WIDLC				WVEC				WADC				WACC			
Attribute	R/W															
Initial value	0	0	0	0	0	1	0	1	0	1	0	1	1	1	1	1
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RIDLC				Reserved				RADC				RACC			
Attribute	R/W															
Initial value	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
R/W: Read/write allowed																

<Notes>

- This register allows only 32-bit access.
- The setting is valid from the next access after the access that MCTR was changed.
To access after the setting is surely valid, dummy read the written external bus register.

[bit 31 to bit 28]: WIDLC

These bits set the number of idle cycles after write.

Written Value	Description
0	1 cycle
1	2 cycles
and so on	and so on
15	16 cycles

[bit 27 to bit 24]: WVEC

These bits set the assert width of the write strobe signal.

Written Value	Description
0	1 cycle
and so on	and so on
5	6 cycles
and so on	and so on
14	15 cycles
15	Reserved (forbidden to set)

[bit 23 to bit 20]: WADC

These bits set the number of address setup cycles during write.

Written Value	Description
0	1 cycle
and so on	and so on
5	6 cycles
and so on	and so on
14	15 cycles
15	Reserved (forbidden to set)

[bit 19 to bit 16]: WACC

These bits set the number of access cycles during write.

Written Value	Description
0	1 cycle (forbidden to set)
1	2 cycles
and so on	and so on
15	16 cycles

<Note>

The number of cycles to be set in WACC must be equal to or greater than the sum of cycles set in WVEC and WADC.

[bit 15 to bit 12]: RIDLC

These bits set the number of idle cycles after read.

Written Value	Description
0	1 cycle
1	2 cycles
and so on	and so on
15	16 cycles

[bit 11 to bit 8]: Reserved bits

Write "0".

[bit 7 to bit 4]: RADC

These bits set the number of address setup cycles during read.

Written Value	Description
0	0 cycle
and so on	and so on
5	5 cycles
and so on	and so on
15	15 cycles

<Note>

When using \overline{AS} , set to 1 cycle or greater.

[bit 3 to bit 0]: RACC

These bits set the number of access cycles during read.

Written Value	Description
0	1 cycle
1	2 cycles
and so on	and so on
15	16 cycles

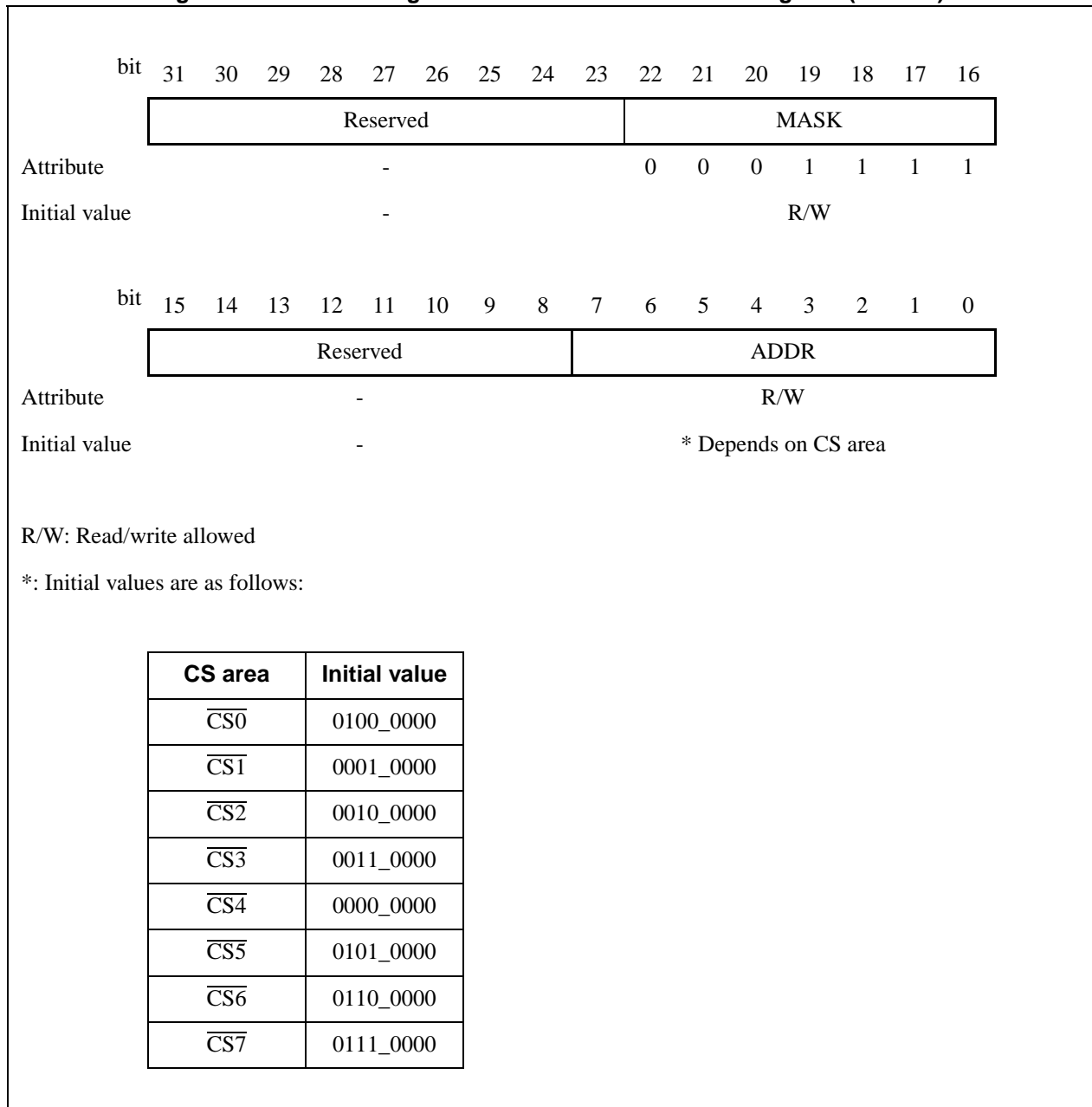
<Note>

The number of cycles to be set in RACC must be greater than the number of cycles set in RADC.

16.3.3 SRAM/FLASH Area Register (MCARx)

Figure 16.3-3 shows the bit configuration of the SRAM/FLASH area register (MCARx).

Figure 16.3-3 Bit Configuration of SRAM/FLASH Area Register (MCARx)



ADDR is the start address of the area and MASK corresponds to the area size.

<Note>

This register allows only 32-bit access.

[bit 31 to bit 23]: Reserved bits

[bit 22 to bit 16]: MASK

These bits set the CS area address mask.

Set bit at "0" to compare the bit 26 to bit 20 of access address with the setting of the following bit of ADDR. Set bit (mask bit) at "1" when bit is not compared. If all "0", all bits are compared. If all "1", only bit 27 is compared.

Setting Example	
000_0000	1 Mbyte
000_0001	2 Mbytes
000_0011	4 Mbytes
000_0111	8 Mbytes
000_1111	16 Mbytes
001_1111	32 Mbytes
011_1111	64 Mbytes
111_1111	128 Mbytes

[bit 15 to bit 8]: Reserved bits

[bit 7 to bit 0]: ADDR

These bits set the comparison address bit 27 to bit 20 of high-order address in the CS area.

Initial Value of each CS Area		Address Range for Initial Value
$\overline{CS0}$	0100_0000	0x0400_0000 to 0x04FF_FFFF 0x4400_0000 to 0x44FF_FFFF
$\overline{CS1}$	0001_0000	0x0100_0000 to 0x01FF_FFFF 0x4100_0000 to 0x41FF_FFFF
$\overline{CS2}$	0010_0000	0x0200_0000 to 0x02FF_FFFF 0x4200_0000 to 0x42FF_FFFF
$\overline{CS3}$	0011_0000	0x0300_0000 to 0x03FF_FFFF 0x4300_0000 to 0x43FF_FFFF
$\overline{CS4}$	0000_0000	0x0004_0000 to 0x00FF_FFFF(*1) 0x4000_0000 to 0x40FF_FFFF
$\overline{CS5}$	0101_0000	0x0500_0000 to 0x05FF_FFFF 0x4500_0000 to 0x45FF_FFFF
$\overline{CS6}$	0110_0000	0x0600_0000 to 0x06FF_FFFF 0x4600_0000 to 0x46FF_FFFF
$\overline{CS7}(*2)$	0111_0000	0x0700_0000 to 0x07FF_FFFF 0x4700_0000 to 0x47FF_FFFF

*1 BOOT area is assigned to $\overline{CS4}$ area by default.

*2 There is no $\overline{CS7}$ pin, but area set must be performed.

<Note>
Make sure the set areas do not overlap.

16.3.4 SDRAM Mode Register (SDMRx)

Figure 16.3-4 shows the bit configuration of the SDRAM mode register (SDMRx).

Figure 16.3-4 Bit Configuration of SDRAM Mode Register (SDMRx)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	Reserved																
Attribute	-																
Initial value	-																
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	BASEL				RASEL				Reserved		CASEL		Reserved		ROFF	PDON	SDON
Attribute	R/W								-				R/W				
Initial value	0	0	0	1	0	0	1	1	-	-	0	0	-	0	0	0	
R/W: Read/write allowed																	

<Note>
This register allows only 32-bit access.

[bit 31 to bit 16]: Reserved bits

[bit 15 to bit 12]: BASEL

These bits select the bit position of BANK address output to A[15:14] pin.

Written Value	Description
0000	bit20, bit19
0001	bit21, bit20
0010	bit22, bit21
0011	bit23, bit22
0100	bit24, bit23
0101	bit25, bit24
0110	bit26, bit25
0111 to 1111	forbidden to set

[bit 15 to bit 8]: RASEL

These bits select the bit position of ROW address output to A[13:0] pin.

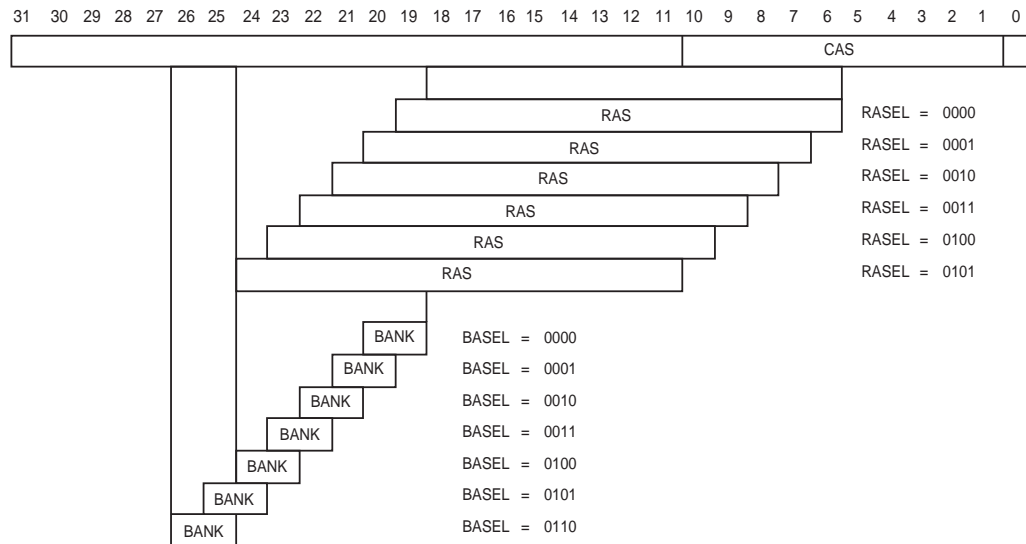
Written Value	Description
0000	bit19 to bit6
0001	bit20 to bit7
0010	bit21 to bit8
0011	bit22 to bit9
0100	bit23 to bit10
0101	bit24 to bit11
0110 to 1111	forbidden to set

[bit 7, bit 6]: Reserved bits

[bit 5, bit 4]: CASEL

These bits select the bit position of COLUMN address output to A[9:0] pin.

Written Value	Description
00	bit10 to bit1
01 to 11	forbidden to set

**[bit 3]: Reserved bit****[bit 2]: ROFF**

This bit stops refreshing while SDRAM command register (SDCMR) is accessed.

Written Value	Description
0	Refresh enabled
1	Refresh disabled

[bit 1]: PDON

This bit sets SDRAM to power down mode when the time set in power down counter has elapsed.

Written Value	Description
0	Power down disabled
1	Power down enabled

[bit0]: SDON

This bit enables SDRAM access.

Written Value	Description
0	Access disabled
1	Access enabled

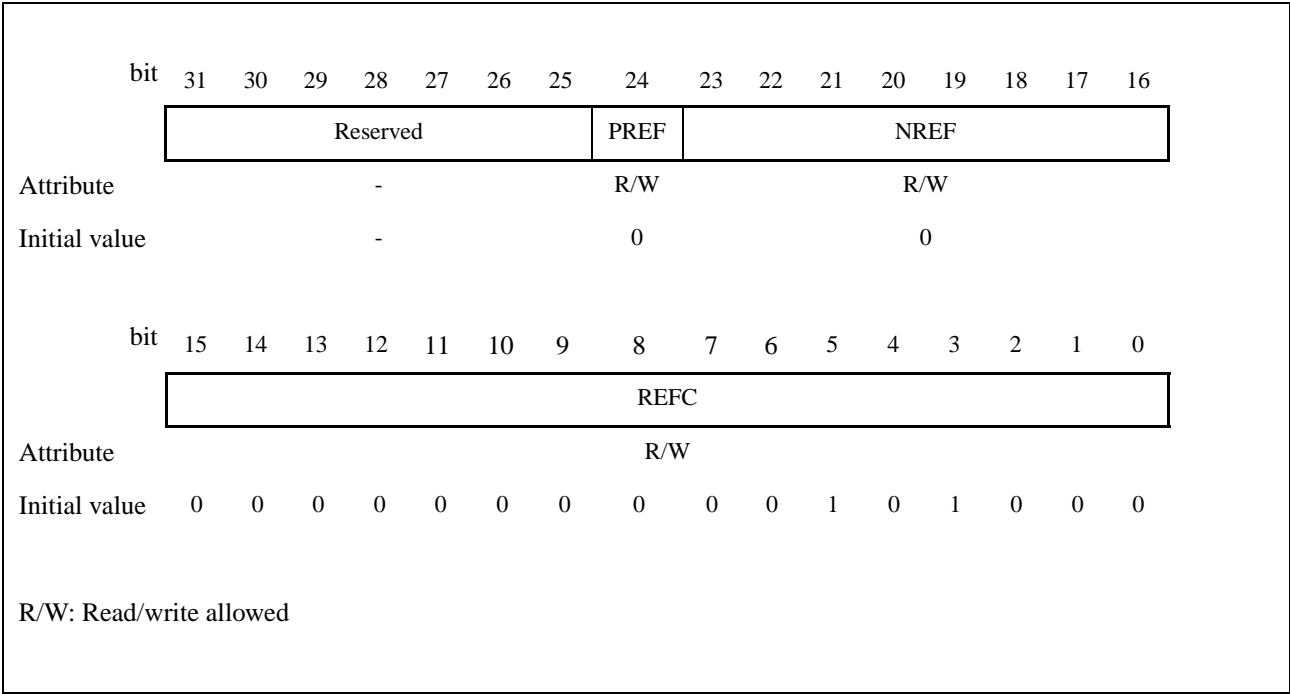
<Note>

When this bit is set to "1", mode data is transferred to SDRAM.
Set other SDRAM control registers before setting this register.

16.3.5 SDRAM Refresh Timer Register (SDRTRx)

Figure 16.3-5 shows the bit configuration of the SDRAM refresh timer register (SDRTRx).

Figure 16.3-5 Bit Configuration of SDRAM Refresh Timer Register (SDRTRx)



<Note>

This register allows only 32-bit access.

[bit 31 to bit 25]: Reserved bits

[bit 24]: PREF

This bit enables pre-refresh.

Written Value	Description
0	Disabled
1	Enabled

[bit 23 to bit 16]: NREF

These bits set the number of refresh commands to be issued per refresh interval.

Written Value	Description
0000_0000	1
and so on	and so on
1111_1111	256

[bit 15 to bit 0]: REFC

These bits set the refresh interval.

Written Value	Description
0000 _H to 0009 _H	Reserved (forbidden to set)
0028 _H	40 cycles
FFFF _H	65535 cycles

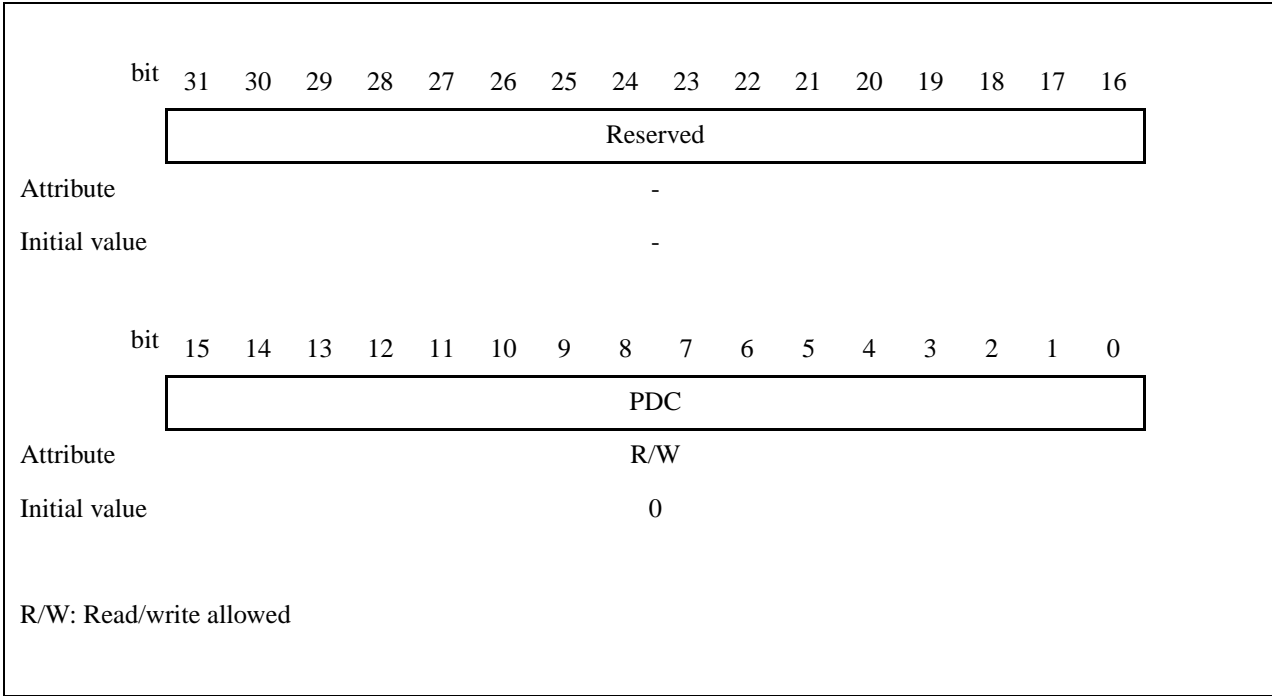
On the data sheet for SDRAM, if it is mentioned that the refresh interval is 4096cycles/64ms, set REFC and NREF to have the condition as follows.

$$64 \times 10^{-3} [\text{s}] \times \text{MCLK} [\text{Hz}] / 4096 [\text{cyc}] \geq (\text{REFC}) / (\text{NREF} + 1)$$

16.3.6 SDRAM Power Down Count Register (SDPDRx)

Figure 16.3-6 shows the bit configuration of the SDRAM power down count register (SDPDRx).

Figure 16.3-6 Bit Configuration of SDRAM Power Down Count Register (SDPDRx)



<Note>
This register allows only 32-bit access.

[bit 31 to bit 16]: Reserved bits

[bit 15 to bit 0]: PDC

These bits set the count of the power down counter.

Written Value	Description
0000 _H	0
and so on	and so on
FFFF _H	65535

16.3.7 SDRAM Timing Register (SDTRx)

Figure 16.3-7 shows the bit configuration of the SDRAM timing register (SDTRx).

Figure 16.3-7 Bit Configuration of SDRAM Timing Register (SDTRx)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved						TDPL		TREFC				TRAS			
Attribute	-						R/W									
Initial value	-	-	-	-	-	-	0	0	0	1	0	0	0	0	1	0

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRCD				TRP			TRC				Reserved		CL		
Attribute	R/W						-						R/W			
Initial value	0	0	0	1	0	0	0	1	0	1	0	0	-	-	0	1

R/W: Read/write allowed

<Note>

This register allows only 32-bit access.

[bit 31 to bit 26]: Reserved bits

[bit 25, bit 24]: TDPL

These bits set the latency from write to pre-charge.

Written Value	Description
00	1 cycle
01	2 cycles
10	3 cycles
11	4 cycles

[bit 23 to bit 20]: TREFC

These bits set the latency from command to refresh.

Written Value	Description
0000	1 cycle
and so on	and so on
0100	5 cycles
and so on	and so on
0111	8 cycles
1000 to 1111	Reserved (forbidden to set)

[bit 19 to bit 16]: TRAS

These bits set the RAS active time.

Written Value	Description
0000	1 cycle
0001	2 cycles
0010	3 cycles
and so on	and so on
0111	8 cycles
1000 to 1111	Reserved (forbidden to set)

[bit 15 to bit 12]: TRCD

These bits set the RAS-CAS delay.

Written Value	Description
0000	1 cycle
0001	2 cycles
0010 to 1111	Reserved (forbidden to set)

[bit 11 to bit 8]: TRP

These bits set the RAS pre-charge time.

Written Value	Description
0000	1 cycle
0001	2 cycles
0010	3 cycles
0011	4 cycles
0100 to 1111	Reserved (forbidden to set)

[bit 7 to bit 4]: TRC

These bits set the RAS cycle time.

Written Value	Description
0000	1 cycle
and so on	and so on
0100	5 cycles
and so on	and so on
0111	8 cycles
1000 to 1111	Reserved (forbidden to set)

[bit 3, bit 2]: Reserved bits

[bit 1, bit 0]: CL

These bits sets the CAS latency.

Written Value	Description
00	1 cycle
01	2 cycles
10	3 cycles
11	Reserved (forbidden to set)

16.3.8 SDRAM Command Register (SDCMRx)

Figure 16.3-8 shows the bit configuration of the SDRAM command register (SDCMRx).

Figure 16.3-8 Bit Configuration of SDRAM Command Register (SDCMRx)

bit	31	30 to 21	20	19	18	17	16
	PEND	Reserved	SDCKE	SDCS	SDRAS	SDCAS	SDWE
Attribute	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial value	0	-	0	0	0	0	0
bit	15	0					
	SDAD						
Attribute	R/W						
Initial value	0						
R/W: Read/write allowed							
R: Read only							

<Notes>

- This register allows only 32-bit access.
- Command is issued to SDRAM when write is performed to this register.

[bit 31]: PEND

This bit indicates that access request is pending.

Read Value	Description
0	Access enabled
1	Access suppressed

When an SDRAM command is issued by writing to this register, this bit indicates whether the next data access request can be accepted.

[bit 30 to bit 21]: Reserved bits

[bit 20]: SDCKE

This bit sets the value output to MCLKE pin.

[bit 19]: SDCS

This bit sets the value output to $\overline{\text{CS8}}$ pin.

[bit 18]: SDRAS

This bit sets the value output to $\overline{\text{MRAS}}$ pin.

[bit 17]: SDCAS

This bit sets the value output to $\overline{\text{MCAS}}$ pin.

[bit 16]: SDWE

This bit sets the value output to $\overline{\text{MDWE}}$ pin.

[bit 15 to bit 0]: SDAD

These bits set the value output to address pins.

Each bit setting example

When issuing a self-refresh command:

SDCKE=0, SDCS=0, SDRAS=0, SDCAS=0, SDWE=1

16.3.9 Clock Control Register (CLKCTL)

Figure 16.3-9 shows the bit configuration of the clock control register (CLKCTL).

Figure 16.3-9 Bit Configuration of Clock Control Register (CLKCTL)

bit	31	30	29	28	27	26	25	24
	Reserved						CLKSEL	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	-	-	0	0
R/W: Read/write allowed								

<Note>

This register allows only byte access.

[bit 31 to bit 26]: Reserved bits

Write "0".

[bit 25, bit 24]: CLKSEL

These bits select the external bus clock.

Clocks dividing the base clock can be selected.

Written Value	Description
00	Divide by 1 (Undivided)
01	Divide by 2
10	Divide by 3
11	Divide by 4

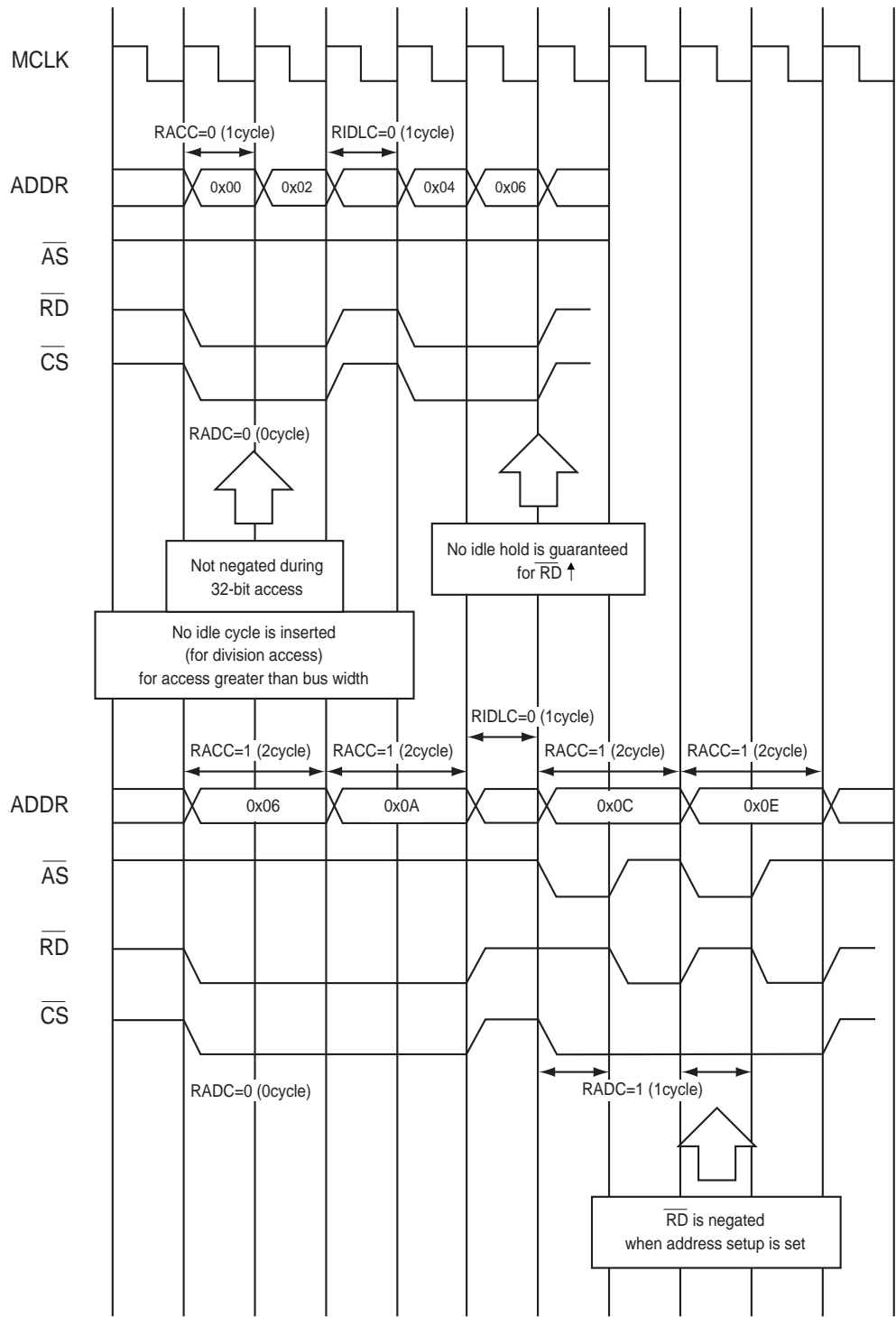
16.4 Register Setting Examples

This section shows examples of external bus interface register settings.

16.4.1 Read Access Examples of SRAM/FLASH Area

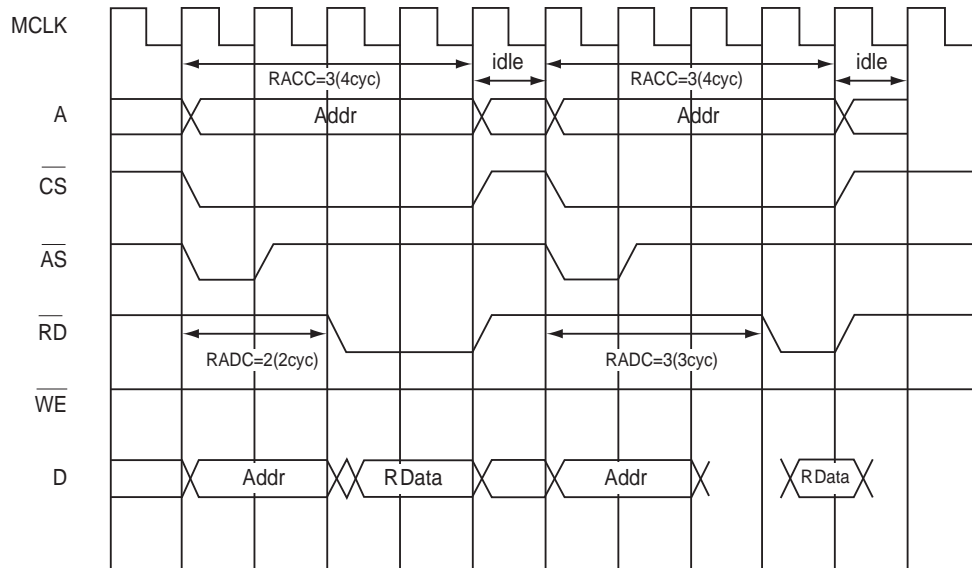
■ Example for separate bus

Figure 16.4-1 Example for Separate Bus (read access)



■ Example for multiplex bus

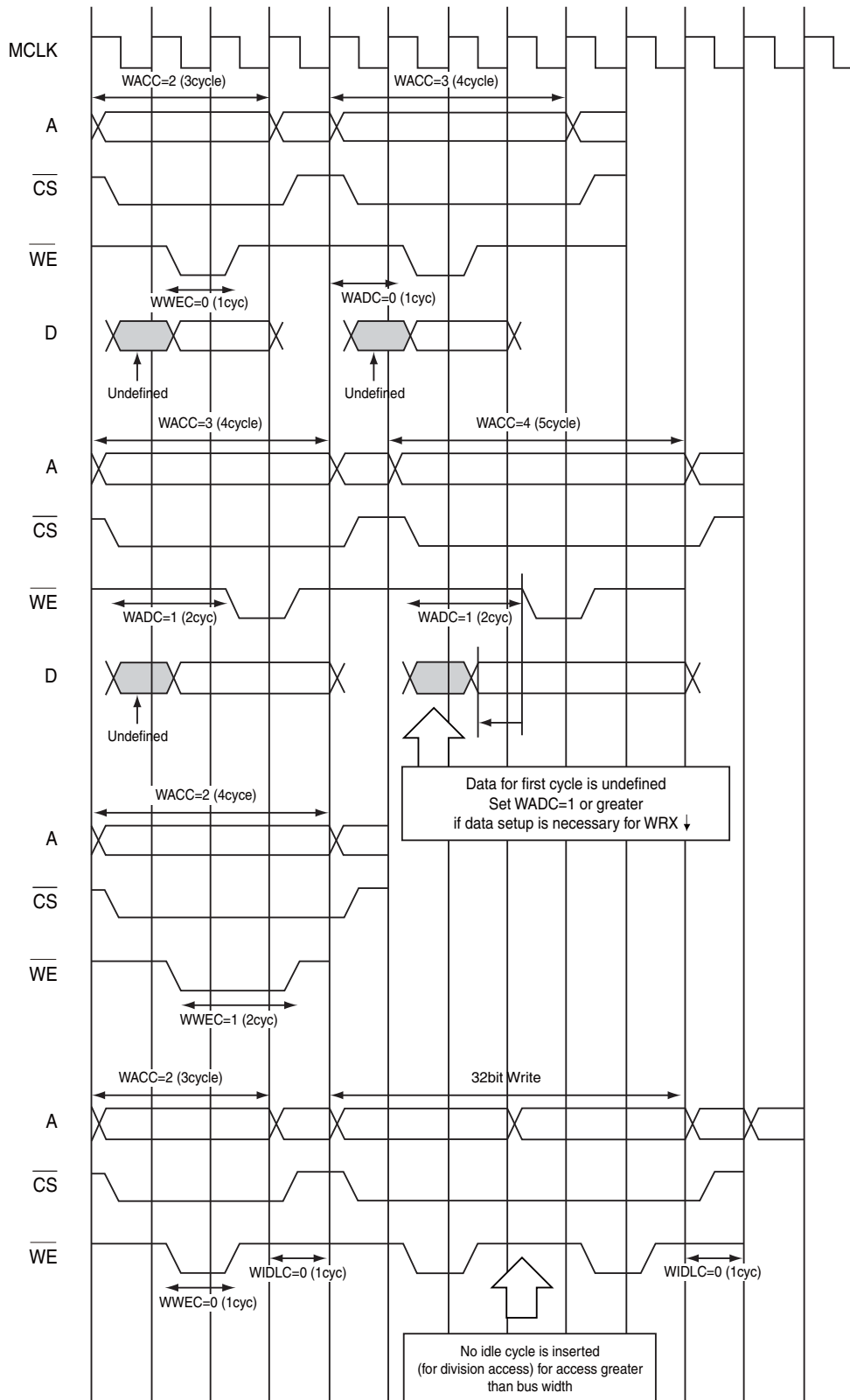
Figure 16.4-2 Example for Multiplex Bus (read access)



16.4.2 Write Access Examples of SRAM/FLASH Area

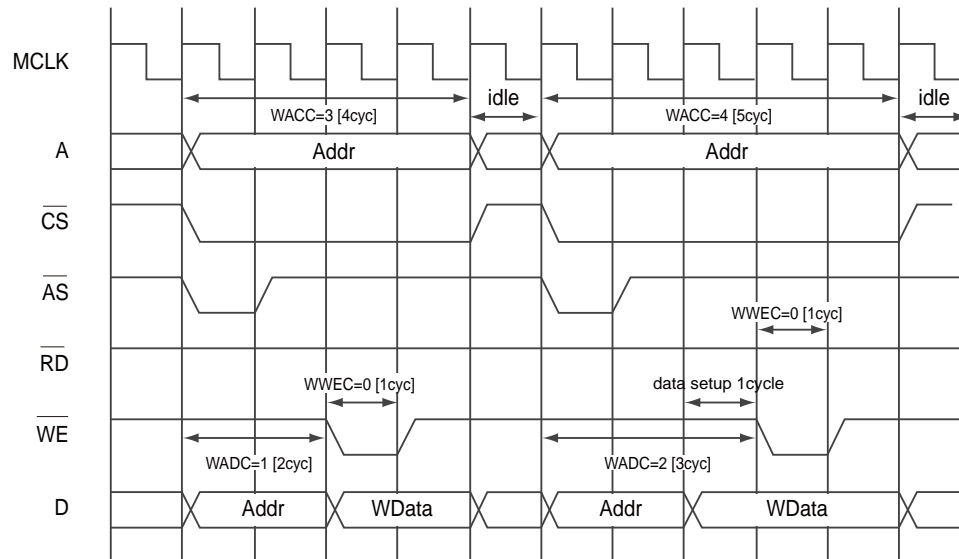
■ Example for separate bus

Figure 16.4-3 Example for Separate Bus (write access)

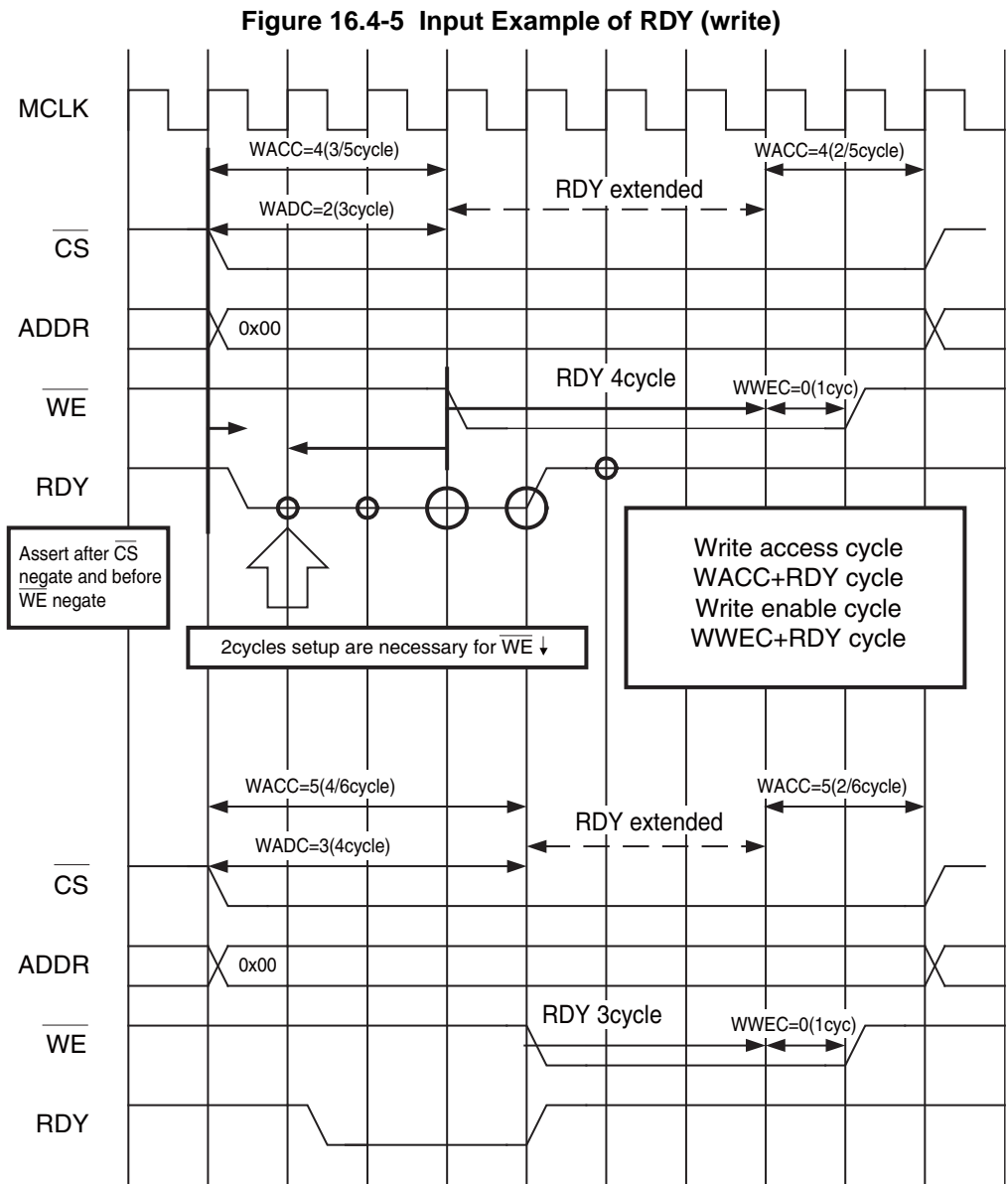


■ Example for multiplex bus

Figure 16.4-4 Example for Multiplex Bus (write access)

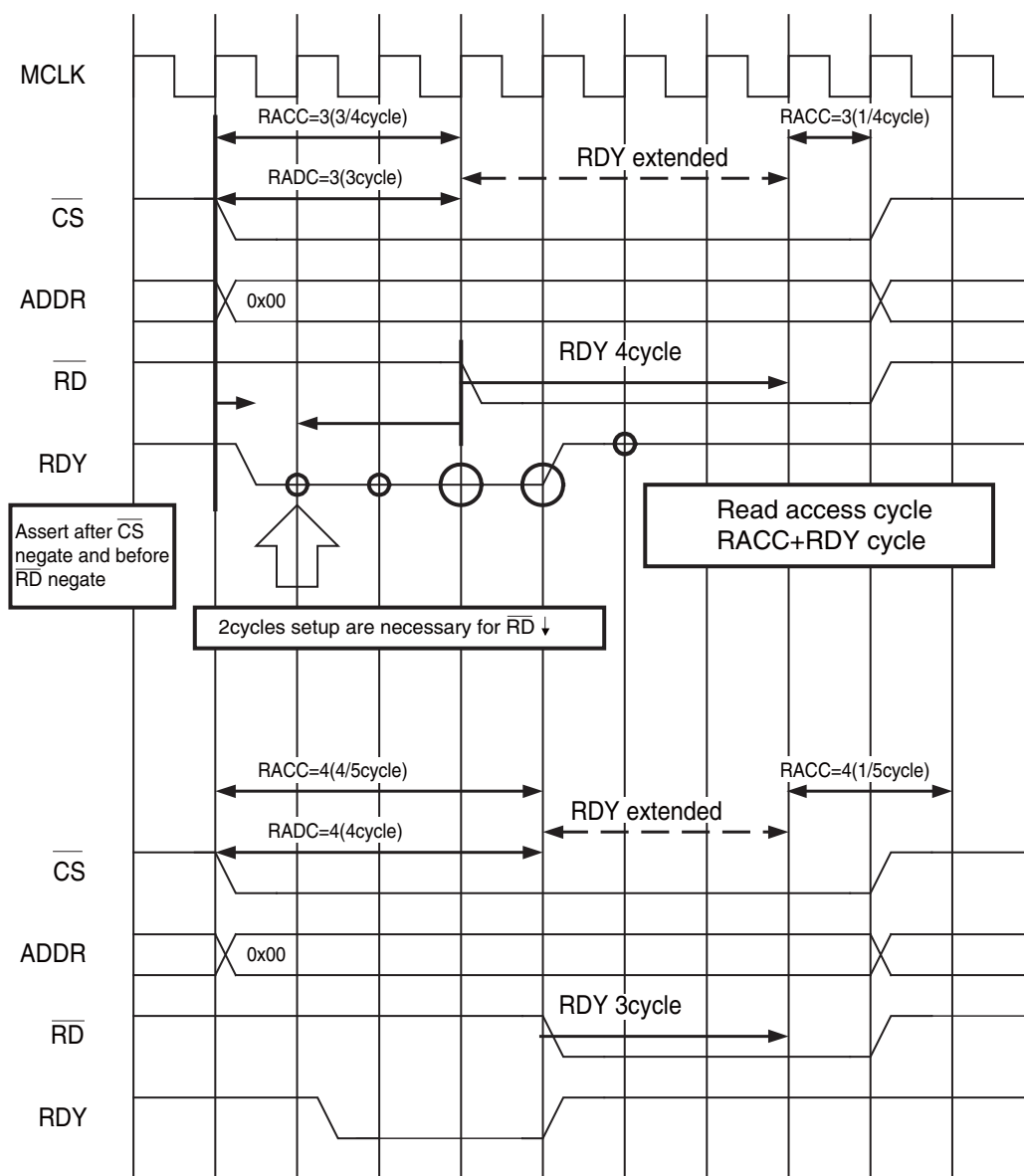


■ Input Example of RDY (write)



■ Input Example of RDY (read)

Figure 16.4-6 Input Example of RDY (read)



16.4.3 Address Area Setting Examples of SRAM/FLASH Area

Table 16.4-1 shows the address area setting examples of SRAM/FLASH area.

Table 16.4-1 Address Area Setting Examples of SRAM/FLASH Area

Setting Example of CS Area		Address Range	Size	Remark
$\overline{\text{CS0}}$	ADDR:0100_0000 MASK:-000_1111	0x0400_0000 to 0x04FF_FFFF 0x4400_0000 to 0x44FF_FFFF	16 Mbytes	Initial value
$\overline{\text{CS1}}$	ADDR:0001_0000 MASK:-000_1111	0x0100_0000 to 0x01FF_FFFF 0x4100_0000 to 0x41FF_FFFF	16 Mbytes	Initial value
$\overline{\text{CS2}}$	ADDR:0010_0000 MASK:-000_1111	0x0200_0000 to 0x02FF_FFFF 0x4200_0000 to 0x42FF_FFFF	16 Mbytes	Initial value
$\overline{\text{CS3}}$	ADDR:0011_0000 MASK:-000_1111	0x0300_0000 to 0x03FF_FFFF 0x4300_0000 to 0x43FF_FFFF	16 Mbytes	Initial value
$\overline{\text{CS4}}$	ADDR:1000_0000 MASK:-111_1111	0x0800_0000 to 0x0FFF_FFFF 0x4800_0000 to 0x4FFF_FFFF	128 Mbytes	Maximum size
$\overline{\text{CS5}}$	ADDR:0101_0000 MASK:-000_1111	0x0500_0000 to 0x05FF_FFFF 0x4500_0000 to 0x45FF_FFFF	16 Mbytes	Initial value
$\overline{\text{CS6}}$	ADDR:0110_0000 MASK:-000_1111	0x0600_0000 to 0x06FF_FFFF 0x4600_0000 to 0x46FF_FFFF	16 Mbytes	Initial value
$\overline{\text{CS7}}(*)$	ADDR:0111_0000 MASK:-000_1111	0x0700_0000 to 0x07FF_FFFF 0x4700_0000 to 0x47FF_FFFF	16 Mbytes	Initial value

* There is no pin, but the area setting must not be overlapped.

16.5 Notes on Use

Note the following points when using the external bus interface.

- \overline{AS} can only be output to SRAM/FLASH area. When using \overline{AS} output, set the address setup cycle (RADC, WADC) of the corresponding CS area to 1 cycle or greater. The \overline{AS} assert width is fixed to 1 cycle.
- Address/data multiplex can only be set to SRAM/FLASH area. When using multiplex, set the address setup cycle (RADC, WADC) of the corresponding CS area to 2 cycle or greater. The address cycle is fixed to 2 cycles.
- If read exceeding the bus width (such as 32-bit read to 16-bit bus width) is performed when set to multiplex, the next address cycle occurs immediately after the first read cycle (after $\overline{RD}\uparrow$). Therefore, momentary bus collision may occur between the data read from memory and address output.
Do not perform read access of data exceeding the bus width in order to prevent bus collision. If the access is within bus width, no bus collision occurs because an idle cycle is inserted after read.
- The clock is switched during memory access because the switching of clock division setting is not synchronized with the external bus operation. Make sure the access speed can keep up with the access data when switching from low frequency to high frequency.
- Make sure the setting of each CS area do not overlap in the SRAM/FLASH area register (MCAR). CS in multiple areas are asserted simultaneously because there is no priority judgment. There is no pin for $\overline{CS7}$ area, but be sure to set the register so that the areas do not overlap.
- Set the access cycles in RACC and WACC in the SRAM/FLASH timing register (MCTR) as follows:
 - Set RACC to number of cycles greater than the number of address cycles set in RADC.
 - Set WACC to number of cycles equal to or greater than the sum of cycles set in WADC and WVEC.

CHAPTER 17 I/O Ports

This chapter describes the functions and operations of I/O ports.

- 17.1 Overview
- 17.2 Configuration
- 17.3 Pins
- 17.4 Registers

17.1 Overview

Pins of this product that are not used for external bus interface or peripheral function can be used as I/O ports.

This product is equipped with 92 I/O ports.

■ Overview

I/O ports have the following features:

- A port data register (PDR) is available for each port to store output data. The contents of the PDR registers are not initialized upon reset.
- A data direction register (DDR) of each port is available to switch the input/output direction of port. All ports are input ports (DDR=00_H) upon reset.
- The following input/output modes can be selected with the setting of each register.

Table 17.1-1 shows the I/O modes.

Table 17.1-1 I/O Modes

I/O Mode	PFR	DDR	PDR Access	
Port input mode	0	0	Read (non-RMW instructions)	The level of the corresponding external pin is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR.
Port output mode	0	1	Read (non-RMW instructions)	The value in PDR is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR and output to corresponding external pin.
Peripheral function output mode	1	0	Read (non-RMW instructions)	Output value from peripheral function is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR.
	1	1	Read (non-RMW instructions)	The value in PDR is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR.

PFR: Port function registers (PFR2 to PFRE)

DDR: Data direction registers (DDR2 to DDRE)

PDR: Port data registers (PDR2 to PDRE)

RMW type instructions: Read-modify-write type instructions

- Except for special cases, inputs to peripheral function are always connected to pins. Normally, perform input to peripheral function in port input mode.
- Ports B, C, and E have a pull-up control register (PCR) that enables setting of 33k Ω pull-up resistor for each pin.
- Each port has a port function register (PFR) which mainly controls peripheral function output.
- In external bus mode, the DDR register and PFR register settings are ignored for pins assigned to external bus interface and the bus interface function has priority. When using these pins as general purpose port/peripheral function output in external bus mode, set the PFR register and disable the bus interface function.
- During STOP mode, input is fixed to "0". However, external interrupt input is not fixed when the corresponding interrupt is enabled (set ENIR bit and select input pin with PFR) and the input to the pin is used as interrupt.
- The bi-direction signal (I2C function of multi-function serial interface as SOUT/SDA, SCK/SCL) of the peripheral function is enabled with the PFR register. See the chapter on the corresponding peripheral function for information on how to switch I/O.

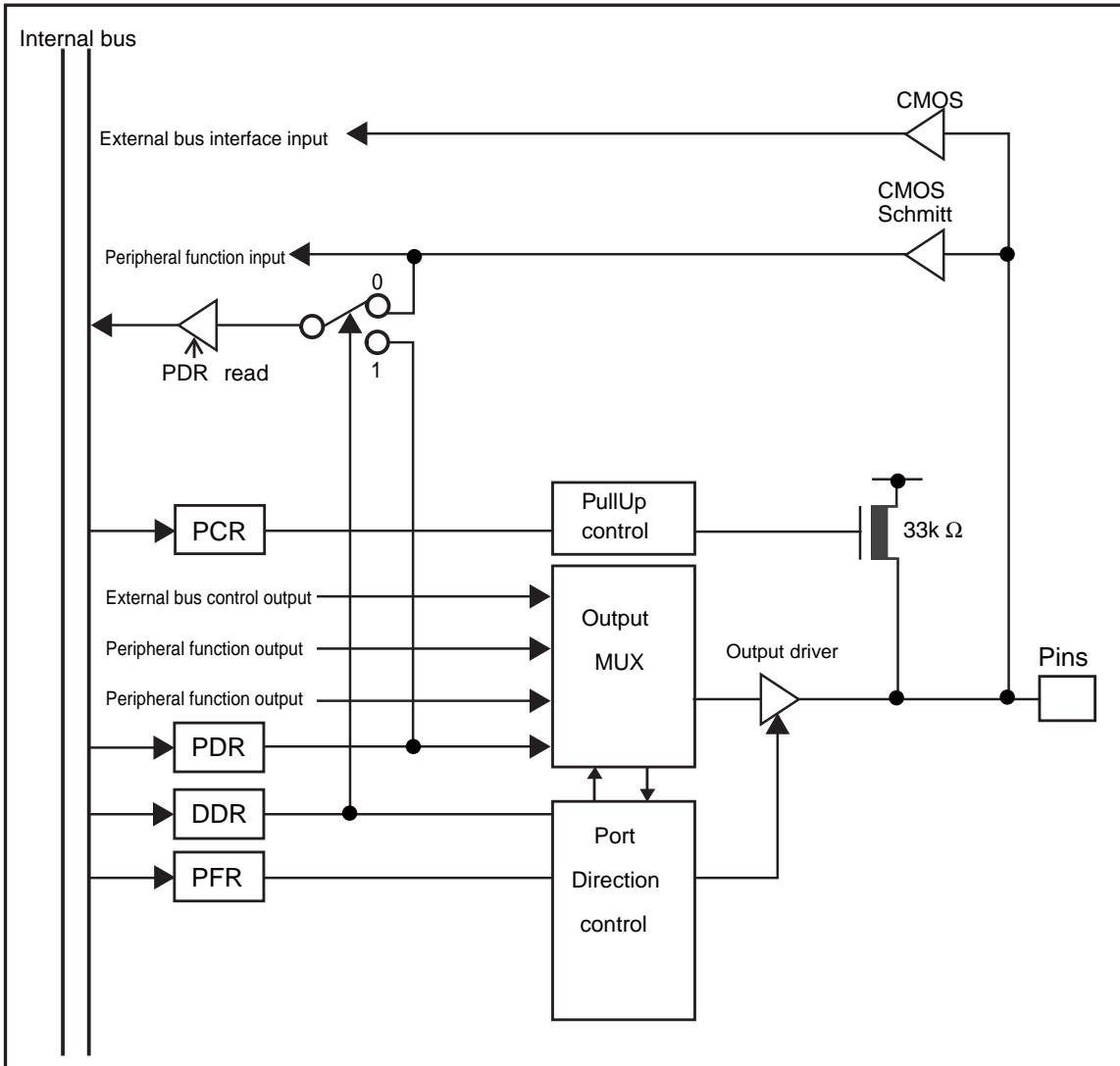
17.2 Configuration

This section describes the configuration of I/O ports.

■ Block Diagram of Port Basic

Figure 17.2-1 shows the basic configuration of port.

Figure 17.2-1 Block Diagram of Port Basic



17.3 Pins

This section describes the pins used by I/O ports.

■ Overview

Up to 92 I/O ports, categorized into port 2 to port E are available.

Read/write to ports categorized with the same number can be performed at the same time.

- P20 to P23 (Port 2)
- P30 to P37 (Port 3)
- P40 to P45 (Port 4)
- P50 to P57 (Port 5)
- P60 to P67 (Port 6)
- P70 to P77 (Port 7)
- P80 to P85 (Port 8)
- P90 to P95 (Port 9)
- PA0 to PA5 (Port A)
- PB0 to PB7 (Port B)
- PC0 to PC7 (Port C)
- PD0 to PD7 (Port D)
- PE0 to PE7 (Port E)

17.4 Registers

This section describes the configurations and functions of registers used by I/O ports.

■ List of registers for I/O port

Table 17.4-1 shows a list of registers for I/O port.

Table 17.4-1 List of Registers for I/O Port (1 / 2)

Port	Register Abbreviation	Register Name	See
Common	ADER	ADER control register	17.4.5
2	PDR2	Port data register 2	17.4.1
	DDR2	Data direction register 2	17.4.2
	PFR2	Port function register 2	17.4.3
3	PDR3	Port data register 3	17.4.1
	DDR3	Data direction register 3	17.4.2
	PFR3	Port function register 3	17.4.3
4	PDR4	Port data register 4	17.4.1
	DDR4	Data direction register 4	17.4.2
	PFR4	Port function register 4	17.4.3
5	PDR5	Port data register 5	17.4.1
	DDR5	Data direction register 5	17.4.2
	PFR5	Port function register 5	17.4.3
6	PDR6	Port data register 6	17.4.1
	DDR6	Data direction register 6	17.4.2
	PFR6	Port function register 6	17.4.3
7	PDR7	Port data register 7	17.4.1
	DDR7	Data direction register 7	17.4.2
	PFR7	Port function register 7	17.4.3
8	PDR8	Port data register 8	17.4.1
	DDR8	Data direction register 8	17.4.2
	PFR8	Port function register 8	17.4.3
9	PDR9	Port data register 9	17.4.1
	DDR9	Data direction register 9	17.4.2
	PFR9	Port function register 9	17.4.3

Table 17.4-1 List of Registers for I/O Port (2 / 2)

Port	Register Abbreviation	Register Name	See
A	PDRA	Port data register A	17.4.1
	DDRA	Data direction register A	17.4.2
	PFRA	Port function register A	17.4.3
B	PDRB	Port data register B	17.4.1
	DDRB	Data direction register B	17.4.2
	PFRB	Port function register B	17.4.3
	PCRB	Pull-up control register	17.4.4
C	PDRC	Port data register C	17.4.1
	DDRC	Data direction register C	17.4.2
	PFRC	Port function register C	17.4.3
	PCRC	Pull-up control register C	17.4.4
D	PDRD	Port data register D	17.4.1
	DDRD	Data direction register D	17.4.2
	PFRD	Port function register D	17.4.3
E	PDRE	Port data register E	17.4.1
	DDRE	Data direction register E	17.4.2
	PFRE	Port function register E	17.4.3
	PCRE	Pull-up control register E	17.4.4

17.4.1 Port Data Registers (PDR2 to PDRE)

A port data register (PDR2 to PDRE) is available for each port to store output data.

The content of these registers are not initialized upon reset.

Figure 17.4-1 shows the bit configuration of the port data registers (PDR2 to PDRE).

Figure 17.4-1 Bit Configuration of Port Data Register (PDR2 to PDRE)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PDR2	-	-	-	-	PDR23	PDR22	PDR21	PDR20	----XXXX	R/W
PDR3	PDR37	PDR36	PDR35	PDR34	PDR33	PDR32	PDR31	PDR30	XXXXXXXX	R/W
PDR4	-	-	PDR45	PDR44	PDR43	PDR42	PDR41	PDR40	--XXXXXX	R/W
PDR5	PDR57	PDR56	PDR55	PDR54	PDR53	PDR52	PDR51	PDR50	XXXXXXXX	R/W
PDR6	PDR67	PDR66	PDR65	PDR64	PDR63	PDR62	PDR61	PDR60	XXXXXXXX	R/W
PDR7	PDR77	PDR76	PDR75	PDR74	PDR73	PDR72	PDR71	PDR70	XXXXXXXX	R/W
PDR8	-	-	PDR85	PDR84	PDR83	PDR82	PDR81	PDR80	--XXXXXX	R/W
PDR9	-	-	PDR95	PDR94	PDR93	PDR92	PDR91	PDR90	--XXXXXX	R/W
PDRA	-	-	PDRA5	PDRA4	PDRA3	PDRA2	PDRA1	PDRA0	--XXXXXX	R/W
PDRB	PDRB7	PDRB6	PDRB5	PDRB4	PDRB3	PDRB2	PDRB1	PDRB0	XXXXXXXX	R/W
PDRC	PDRC7	PDRC6	PDRC5	PDRC4	PDRC3	PDRC2	PDRC1	PDRC0	XXXXXXXX	R/W
PDRD	PDRD7	PDRD6	PDRD5	PDRD4	PDRD3	PDRD2	PDRD1	PDRD0	XXXXXXXX	R/W
PDRE	PDRE7	PDRE6	PDRE5	PDRE4	PDRE3	PDRE2	PDRE1	PDRE0	XXXXXXXX	R/W

R/W: Read/write allowed
-: Undefined
X: Undefined

PDR2 to PDRE are input/output data registers of I/O port.

Input/output is controlled by the corresponding DDR2 to DDRE.

Regardless of the port status, the register setting is read with a read-modify-write instruction to the port data register.

17.4.2 Data Direction Registers (DDR2 to DDRE)

A data direction register (DDR2 to DDRE) of each port is available to switch the input/output direction of port.

All ports are input ports (DDR=00_H) upon reset.

Figure 17.4-2 shows the bit configuration of the data direction registers (DDR2 to DDRE).

Figure 17.4-2 Bit Configuration of Data Direction Register (DDR2 to DDRE)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
DDR2	-	-	-	-	DDR23	DDR22	DDR21	DDR20	----0000	R/W
DDR3	DDR37	DDR36	DDR35	DDR34	DDR33	DDR32	DDR31	DDR30	00000000	R/W
DDR4	-	-	DDR45	DDR44	DDR43	DDR42	DDR41	DDR40	--000000	R/W
DDR5	DDR57	DDR56	DDR55	DDR54	DDR53	DDR52	DDR51	DDR50	00000000	R/W
DDR6	DDR67	DDR66	DDR65	DDR64	DDR63	DDR62	DDR61	DDR60	00000000	R/W
DDR7	DDR77	DDR76	DDR75	DDR74	DDR73	DDR72	DDR71	DDR70	00000000	R/W
DDR8	-	-	DDR85	DDR84	DDR83	DDR82	DDR81	DDR80	--000000	R/W
DDR9	-	-	DDR95	DDR94	DDR93	DDR92	DDR91	DDR90	--000000	R/W
DDRA	-	-	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	--000000	R/W
DDRB	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	00000000	R/W
DDRC	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	00000000	R/W
DDRD	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	00000000	R/W
DDRE	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0	00000000	R/W

R/W: Read/write allowed
-: Undefined
X: Undefined

The meaning of read/write value of port data register (PDR2 to PDRE) depends on the setting of these bits and the setting of port function register (PFR2 to PFRE).

Table 17.4-2 shows the relationship of read/write value between the register settings and port data register (PDR2 to PDRE).

Table 17.4-2 Relationship of Read/Write Value between Register Settings and Port Data Register (PDR2 to PDRE)

I/O Mode	PFR	DDR	PDR Access	
Port input mode	0	0	Read (non-RMW instructions)	The level of the corresponding external pin is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR.
Port output mode	0	1	Read (non-RMW instructions)	The value in PDR is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR and output to corresponding external pin.
Peripheral function output mode	1	0	Read (non-RMW instructions)	Output value from peripheral function is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR.
	1	1	Read (non-RMW instructions)	The value in PDR is read.
			Read (RMW instructions)	The value in PDR is read.
			Write	A value is written in PDR.

PFR: Port function registers (PFR2 to PFRE)

DDR: Data direction registers (DDR2 to DDRE)

PDR: Port data registers (PDR2 to PDRE)

RMW type instructions : Read-modify-write type instructions

Except for special cases, inputs to peripheral function are always connected to pins. Normally, perform input to peripheral function in port input mode.

17.4.3 Port Function Registers (PFR2 to PFRE)

This section describes this register for each port.

■ Port 2

Port 2 is controlled by port function register 2 (PFR2).

In external bus mode, it functions as bus interface control pins (SYSCLK, $\overline{\text{WE}}$, MDQM3, MDQM2). For the selectable input signal, select the input pin with each resource.

Figure 17.4-3 shows the bit configuration of the port function register 2 (PFR2).

Figure 17.4-3 Bit Configuration of Port Function Register 2 (PFR2)

bit	7	6	5	4	3	2	1	0
	-	-	-	-	PFR23	PFR22	PFR21	PFR20
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	-	-	-	1	1	1	0

R/W: Read/write allowed
-: Undefined

Each bit is described below.

Bit	PFR2	Function
bit7	0	Ignored
	1	Ignored
bit6	0	Ignored
	1	Ignored
bit5	0	Ignored
	1	Ignored
bit4	0	Ignored
	1	Ignored
bit3	0	General purpose port
	1	MDQM2 output
bit2	0	General purpose port
	1	MDQM3 output
bit1	0	General purpose port
	1	$\overline{\text{WE}}$ output
bit0	0	General purpose port
	1	SYSCLK output

■ Port 3

Port 3 is controlled by port function register 3 (PFR3).

In external bus mode, it functions as bus interface control pins (\overline{CSn} , \overline{AS}). For the selectable input signal, select the input pin with each resource.

Figure 17.4-4 shows the bit configuration of the port function register 3 (PFR3).

Figure 17.4-4 Bit Configuration of Port Function Register 3 (PFR3)

bit	7	6	5	4	3	2	1	0
	PFR37	PFR36	PFR35	PFR34	PFR33	PFR32	PFR31	PFR30
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	1	1	1	1

R/W: Read/write allowed

Each bit is described below.

Bit	PFR3	Function
bit7	0	General purpose port
	1	Chip select 8 ($\overline{CS8}$) output
bit6	0	General purpose port
	1	Chip select 6 ($\overline{CS6}$) output
bit5	0	General purpose port
	1	Chip select 5 ($\overline{CS5}$) output
bit4	0	General purpose port
	1	Address Strobe output (\overline{AS}) output
bit3	0	General purpose port
	1	Chip select 3 ($\overline{CS3}$) output
bit2	0	General purpose port
	1	Chip select 2 ($\overline{CS2}$) output
bit1	0	General purpose port
	1	Chip select 1 ($\overline{CS1}$) output
bit0	0	General purpose port
	1	Chip select 0 ($\overline{CS0}$) output

■ Port 4

Port 4 is controlled by port function register 4 (PFR4).

In external bus mode, port 4 becomes the bus interface control pins (RDY, $\overline{\text{MRAS}}$, $\overline{\text{MCAS}}$, $\overline{\text{MDWE}}$, MCLKE, MCLK). For the selectable input signal, select the input pin with each resource.

Figure 17.4-5 shows the bit configuration of the port function register 4 (PFR4).

Figure 17.4-5 Bit Configuration of Port Function Register 4 (PFR4)

bit	7	6	5	4	3	2	1	0
	-	-	PFR45	PFR44	PFR43	PFR42	PFR41	PFR40
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-	-	1	1	0	0	0	0
R/W: Read/write allowed								
-: Undefined								

Each bit is described below.

Bit	PFR4	Function
bit7	0	Ignored
	1	Ignored
bit6	0	Ignored
	1	Ignored
bit5	0	General purpose port
	1	MCLK output
bit4	0	General purpose port
	1	MCLKE output
bit3	0	General purpose port
	1	$\overline{\text{MDWE}}$ output
bit2	0	General purpose port
	1	$\overline{\text{MCAS}}$ output
bit1	0	General purpose port
	1	$\overline{\text{MRAS}}$ output
bit0	0	General purpose port
	1	Forbidden to set

■ Port 5

Port 5 is controlled by port function register 5 (PFR5).
In external bus mode, port 5 becomes bus interface pins A23 to A16. For the selectable input signal, select the input pin with each resource.
Figure 17.4-6 shows the bit configuration of the port function register 5 (PFR5).

Figure 17.4-6 Bit Configuration of Port Function Register 5 (PFR5)

bit	7	6	5	4	3	2	1	0
	PFR57	PFR56	PFR55	PFR54	PFR53	PFR52	PFR51	PFR50
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	1
R/W: Read/write allowed								

Each bit is described below.

Bit	PFR5	Function
bit7	0	General purpose port
	1	External address output A23
bit6	0	General purpose port
	1	External address output A22
bit5	0	General purpose port
	1	External address output A21
bit4	0	General purpose port
	1	External address output A20
bit3	0	General purpose port
	1	External address output A19
bit2	0	General purpose port
	1	External address output A18
bit1	0	General purpose port
	1	External address output A17
bit0	0	General purpose port
	1	External address output A16

■ Port 6

Port 6 is controlled by port function register 6 (PFR6).

Port 6 is multiplexed with the analog input of the AD converter. When the corresponding bit in the ADER register is set, Port 6 setting is ignored and functions as an analog input pin.

In other modes, it is assigned to multi-function serial interface ch.10/ch.11. For the selectable input signal, select the input pin with each resource.

Figure 17.4-7 shows the bit configuration of the port function register 6 (PFR6).

Figure 17.4-7 Bit Configuration of Port Function Register 6 (PFR6)

bit	7	6	5	4	3	2	1	0
	PFR67	PFR66	PFR65	PFR64	PFR63	PFR62	PFR61	PFR60
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 allowed								

Each bit is described below.

Bit	PFR6	Function
bit7	0	General purpose port
	1	Multi-function serial interface ch.11 SCK11/SCL11 output
bit6	0	General purpose port
	1	Multi-function serial interface ch.11 SOUT11/SDA11 output
bit5	0	General purpose port
	1	Forbidden to set
bit4	0	General purpose port
	1	Multi-function serial interface ch.10 SCK10/SCL10 output
bit3	0	General purpose port
	1	Multi-function serial interface ch.10 SOUT10/SDA10 output
bit2	0	General purpose port
	1	Forbidden to set
bit1	0	General purpose port
	1	Forbidden to set
bit0	0	General purpose port
	1	Forbidden to set

<Note>

The ADER register is set to A/D analog input with default setting. Clear the ADER register setting to use functions other than analog input.

■ Port 7

Port 7 is controlled by port function register 7 (PFR7).
Bit 0 to bit 3 of port 7 are multiplexed with the analog input of the AD converter. When the corresponding bit in the ADER register is set, Port 6 setting is ignored and function as analog input pins.
In other modes, they are assigned to base timer ch.10/ch.11, multi-function serial interface ch.0, and HDMI-CEC I/O. For the selectable input signal, select the input pin with each resource.
Figure 17.4-8 shows the bit configuration of the port function register 7 (PFR7).

Figure 17.4-8 Bit Configuration of Port Function Register 7 (PFR7)

bit	7	6	5	4	3	2	1	0
	PFR77	PFR76	PFR75	PFR74	PFR73	PFR72	PFR71	PFR70
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 allowed								

Each bit is described below.

Bit	PFR7	Function
bit7	0	General purpose port
	1	Multi-function serial interface ch.0 SCK0/SCL0 output
bit6	0	General purpose port
	1	Multi-function serial interface ch.0 SOUT0/SDA0 output
bit5	0	General purpose port
	1	Forbidden to set
bit4	0	General purpose port
	1	HDMI-CEC I/O (open drain)
bit3,bit2	00	General purpose port
	01	Base timer ch.11 TIOA11 output (normal output)
	10	Forbidden to set
	11	Base timer ch.11 TIOA11 output (open drain output)

Bit	PFR7	Function
bit1,bit0	00	General purpose port
	01	Base timer ch.10 TIOA10 output (normal output)
	10	Forbidden to set
	11	Base timer ch.10 TIOA10 output (open drain output)

<Notes>

- The ADER register is set to A/D analog input with default setting. Clear the ADER register setting to use functions other than analog input.
 - Set to HDMI-CEC I/O mode only when using HDMI-CEC mode. In other remote control reception modes, select the general purpose port and set to input with DDR7 register.
-

■ Port 8

Port 8 is controlled by port function register 8 (PFR8). It is multiplexed with ch.1/ch.2 of multi-function serial interface. For the selectable input signal, select the input pin with each resource.

Figure 17.4-9 shows the bit configuration of the port function register 8 (PFR8).

Figure 17.4-9 Bit Configuration of Port Function Register 8 (PFR8)

bit	7	6	5	4	3	2	1	0
	-	-	PFR85	PFR84	PFR83	PFR82	PFR81	PFR80
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

R/W: Read/write allowed
-: Undefined

Each bit is described below.

Bit	PFR8	Function
bit7	0	Ignored
	1	Ignored
bit6	0	Ignored
	1	Ignored
bit5	0	General purpose port
	1	Multi-function serial interface ch.2 SCK2/SCL2 output
bit4	0	General purpose port
	1	Multi-function serial interface ch.2 SOUT2/SDA2 output
bit3	0	General purpose port
	1	Forbidden to set
bit2	0	General purpose port
	1	Multi-function serial interface ch.1 SCK1/SCL1 output
bit1	0	General purpose port
	1	Multi-function serial interface ch.1 SOUT1/SDA1 output
bit0	0	General purpose port
	1	Forbidden to set

■ Port 9

Port 9 is controlled by port function register 9 (PFR9). It is multiplexed with ch.3/ch.4 of multi-function serial interface. For the selectable input signal, select the input pin with each resource.

Figure 17.4-10 shows the bit configuration of the port function register 9 (PFR9).

Figure 17.4-10 Bit Configuration of Port Function Register 9 (PFR9)

bit	7	6	5	4	3	2	1	0
	-	-	PFR95	PFR94	PFR93	PFR92	PFR91	PFR90
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
R/W: Read/write allowed								
-: Undefined								

Each bit is described below.

Bit	PFR9	Function
bit7	0	Ignored
	1	Ignored
bit6	0	Ignored
	1	Ignored
bit5	0	General purpose port
	1	Multi-function serial interface ch.4 SCK4/SCL4 output
bit4	0	General purpose port
	1	Multi-function serial interface ch.4 SOUT4/SDA4 output
bit3	0	General purpose port
	1	Forbidden to set
bit2	0	General purpose port
	1	Multi-function serial interface ch.3 SCK3/SCL3 output
bit1	0	General purpose port
	1	Multi-function serial interface ch.3 SOUT3/SDA3 output
bit0	0	General purpose port
	1	Forbidden to set

■ Port A

Port A is controlled by port function register A (PFRA). It is multiplexed with ch.5/ch.6 of multi-function serial interface. For the selectable input signal, select the input pin with each resource.

Figure 17.4-11 shows the bit configuration of the port function register A (PFRA).

Figure 17.4-11 Bit Configuration of Port Function Register A (PFRA)

bit	7	6	5	4	3	2	1	0
	-	-	PFRA5	PFRA4	PFRA3	PFRA2	PFRA1	PFRA0
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

R/W: Read/write allowed
-: Undefined

Each bit is described below.

Bit	PFRA	Function
bit7	0	Ignored
	1	Ignored
bit6	0	Ignored
	1	Ignored
bit5	0	General purpose port
	1	Multi-function serial interface ch.6 SCK6/SCL6 output
bit4	0	General purpose port
	1	Multi-function serial interface ch.6 SOUT6/SDA6 output
bit3	0	General purpose port
	1	Forbidden to set
bit2	0	General purpose port
	1	Multi-function serial interface ch.5 SCK5/SCL5 output
bit1	0	General purpose port
	1	Multi-function serial interface ch.5 SOUT5/SDA5 output
bit0	0	General purpose port
	1	Forbidden to set

■ Port B

Port B is controlled by port function register B (PFRB). It is multiplexed with ch.7 of multi-function serial interface, ch.0 to ch.2 of reload timer output, ch.8 of base timer, and external interrupts 0 to 7. For the selectable input signal, select the input pin with each resource.

Figure 17.4-12 shows the bit configuration of the port function register B (PFRB).

Figure 17.4-12 Bit Configuration of Port Function Register B (PFRB)

bit	7	6	5	4	3	2	1	0
	PFRB7	PFRB6	PFRB5	PFRB4	PFRB3	PFRB2	PFRB1	PFRB0
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 allowed

Each bit is described below.

Bit	PFRB	Function
bit7	0	General purpose port
	1	Forbidden to set
bit6	0	General purpose port
	1	Ch.8 of base timer TIOA8 output
bit5	0	General purpose port
	1	Ch.2 of reload timer TOUT2 output
bit4	0	General purpose port
	1	Ch.1 of reload timer TOUT1 output
bit3	0	General purpose port
	1	Ch.0 of reload timer TOUT0 output
bit2	0	General purpose port
	1	Ch.7 of multi-function serial interface SCK7/SCL7 output
bit1	0	General purpose port
	1	Ch.7 of multi-function serial interface SOUT7/SDA7 output
bit0	0	General purpose port
	1	Forbidden to set

■ Port C

Port C is controlled by port function register C (PFRC). It is multiplexed with ch.8/ch.9 of multi-function serial interface, ch.9 of base timer, and external interrupts 8 to 15. For the selectable input signal, select the input pin with each resource.

Figure 17.4-13 shows the bit configuration of the port function register C (PFRC).

Figure 17.4-13 Bit Configuration of Port Function Register C (PFRC)

bit	7	6	5	4	3	2	1	0
	PFRC7	PFRC6	PFRC5	PFRC4	PFRC3	PFRC2	PFRC1	PFRC0
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 allowed

Each bit is described below.

Bit	PFRC	Function
bit7	0	General purpose port
	1	Multi-function serial interface ch.9 SCK9/SCL9 output
bit6	0	General purpose port
	1	Multi-function serial interface ch.9 SOUT9/SDA9 output
bit5	0	General purpose port
	1	Forbidden to set
bit4	0	General purpose port
	1	Multi-function serial interface ch.8 SCK8/SCL8 output
bit3	0	General purpose port
	1	Multi-function serial interface ch.8 SOUT8/SDA8 output
bit2	0	General purpose port
	1	Forbidden to set
bit1	0	General purpose port
	1	Forbidden to set
bit0	0	General purpose port
	1	Ch.9 of base timer TIOA9 output

■ Port D

Port D is controlled by port function register D (PFRD). It is multiplexed with ch.0 to ch.3 of base timer. For the selectable input signal, select the input pin with each resource.

Figure 17.4-14 shows the bit configuration of the port function register D (PFRD).

Figure 17.4-14 Bit Configuration of Port Function Register D (PFRD)

bit	7	6	5	4	3	2	1	0
	PFRD7	PFRD6	PFRD5	PFRD4	PFRD3	PFRD2	PFRD1	PFRD0
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 allowed

Each bit is described below.

Bit	PFRD	Function
bit7,bit6	00	General purpose port
	01	Ch.3 of base timer TIOA3 output (normal output)
	10	Forbidden to set
	11	Ch.3 of base timer TIOA3 output (open drain output)
Bit5,bit4	00	General purpose port
	01	Ch.2 of base timer TIOA2 output (normal output)
	10	Forbidden to set
	11	Ch.2 of base timer TIOA2 output (open drain output)
Bit3,bit2	00	General purpose port
	01	Ch.1 of base timer TIOA1 output (normal output)
	10	Forbidden to set
	11	Ch.1 of base timer TIOA1 output (open drain output)
Bit1,bit0	00	General purpose port
	01	Ch.0 of base timer TIOA0 output (normal output)
	10	Forbidden to set
	11	Ch.0 of base timer TIOA0 output (open drain output)

■ Port E

Port E is controlled by port function register E (PFRE). It is multiplexed with ch.4 to ch.7 of base timer and external interrupts 16 to 23. For the selectable input signal, select the input pin with each resource. Figure 17.4-15 shows the bit configuration of the port function register E (PFRE).

Figure 17.4-15 Bit Configuration of Port Function Register E (PFRE)

bit	7	6	5	4	3	2	1	0
	PFRE7	PFRE6	PFRE5	PFRE4	PFRE3	PFRE2	PFRE1	PFRE0
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 allowed

Each bit is described below.

Bit	PFRE	Function
bit7	0	General purpose port
	1	Forbidden to set
bit6	0	General purpose port
	1	Ch.7 of base timer TIOA7 output
bit5	0	General purpose port
	1	Forbidden to set
bit4	0	General purpose port
	1	Ch.6 of base timer TIOA6 output
bit3	0	General purpose port
	1	Forbidden to set
bit2	0	General purpose port
	1	Ch.5 of base timer TIOA5 output
bit1	0	General purpose port
	1	Forbidden to set
bit0	0	General purpose port
	1	Ch.4 of base timer TIOA4 output

17.4.4 Pull-up Control Register (PCR)

Pins have function to add a 33kΩ pull-up resistor. This function can be controlled in bit units by software. This register is used for this control.

Figure 17.4-16 shows the bit configuration of the Pull-up Control Register (PCR).

Figure 17.4-16 Bit Configuration of Pull-up Control Register (PCR)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PCRB	PCRB7	PCRB6	PCRB5	PCRB4	PCRB3	PCRB2	PCRB1	PCRB0	00000000	R/W
PCRC	PCRC7	PCRC6	PCRC5	PCRC4	PCRC3	PCRC2	PCRC1	PCRC0	00000000	R/W
PCRE	PCRE7	PCRE6	PCRE5	PCRE4	PCRE3	PCRE2	PCRE1	PCRE0	00000000	R/W

R/W: Read/write allowed

The settings of this register is shown below.

Bit	Pull-up Control Register (PCR)	
	0	1
PCRxy	No pull-up	Pull-up

The setting of each bit is valid only when the corresponding PCR is set.

Pull-up control is available for ports PB7 to PB0, PC7 to PC0, and PE7 to PE0. Corresponding bits are provided for these ports.

<Notes>

Pull-up of pin is automatically disabled in the following cases:

- Port is in output mode
- STOP mode output Hi-Z is selected

17.4.5 ADER Control Register (ADER)

This register enables A/D input. This function can be controlled in bit units by software.

Figure 17.4-17 shows the bit configuration of the ADER Control Register (ADER).

Figure 17.4-17 Bit Configuration of ADER Control Register (ADER)

bit	15	14	13	12	11	10	9	8
	-	-	-	-	ADER11	ADER10	ADER9	ADER8
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
bit	7	6	5	4	3	2	1	0
	ADER7	ADER6	ADER5	ADER4	ADER3	ADER2	ADER1	ADER0
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	1
R/W: Read/write allowed								

[bit15 to bit12]: Reserved bits

Always write "0".

[bit11 to bit0]: ADER11 to ADER0 (A/D input enable bits 11 to 0)

Pin of P60 to P67 and P70 to P73 are multiplexed with A/D converter input. Set the corresponding bit to use as A/D converter analog input.

The internal input gate on the chip is fixed to "0" when this bit is set (=1). Clear this bit (=0) to use functions other than analog input.

CHAPTER 18 External Interrupt Controllers

This chapter explains the functions and operations of external interrupt controllers.

- 18.1 Overview
- 18.2 Configuration
- 18.3 Pins
- 18.4 Registers
- 18.5 Explanation of Operations and Setting Procedure Examples

18.1 Overview

The external interrupt controllers detect edges/levels in external interrupt signals, and they control external interrupt requests.

This series has 24 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 (main timer mode or stop mode).

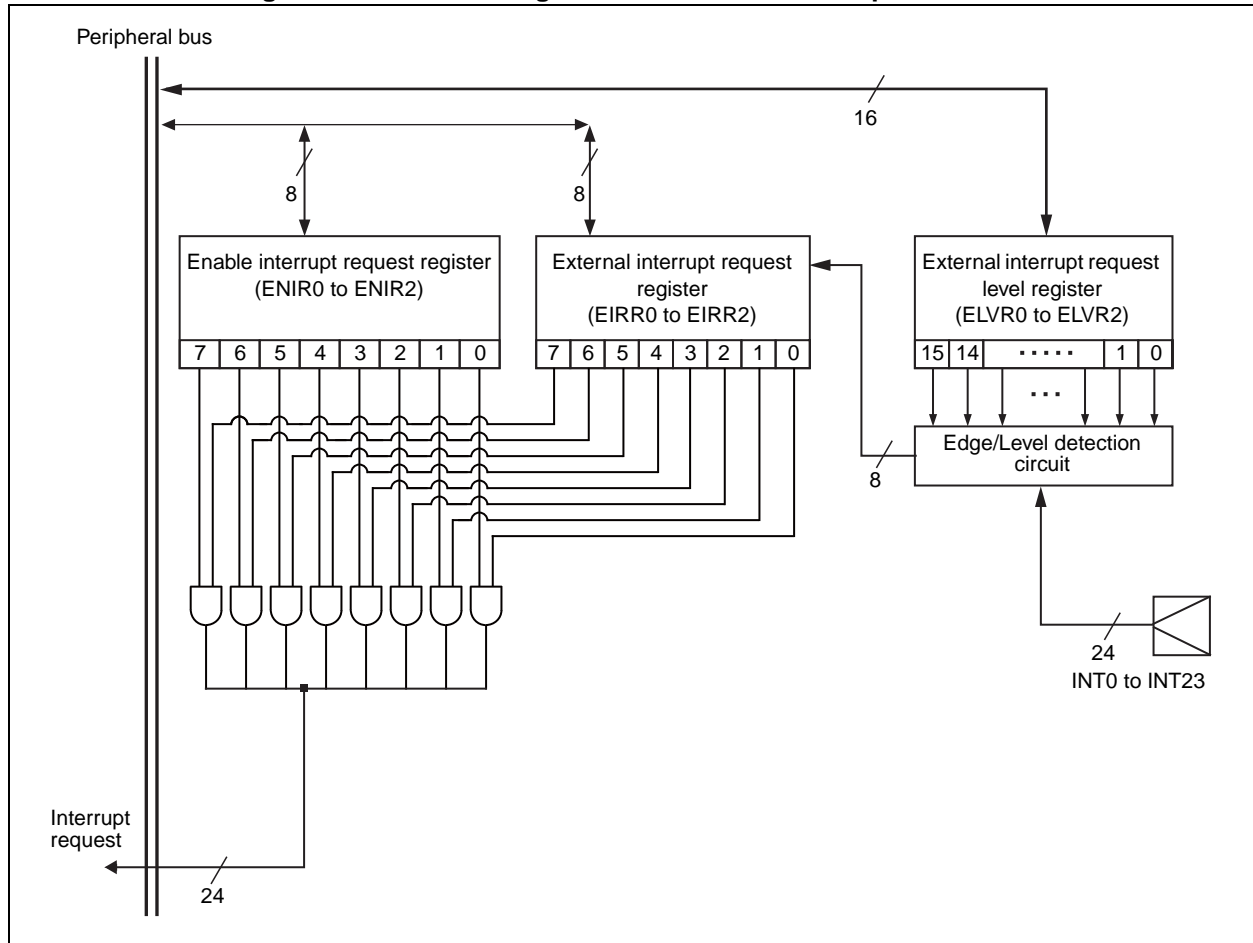
18.2 Configuration

This section shows the configuration of an external interrupt controller.

■ Block diagram of an external interrupt controller

Figure 18.2-1 is a block diagram of an external interrupt controller.

Figure 18.2-1 Block diagram of an external interrupt controller



- **External interrupt request level register (ELVR0 to ELVR2)**
This register sets the edge/level used to determine whether a signal input to the INT0 to INT23 pins is for an external interrupt request.
- **External interrupt request register (EIRR0 to EIRR2)**
This register maintains the states of interrupt sources (indicating which pins have generated external interrupt requests).
- **Enable interrupt request register (ENIR0 to ENIR2)**
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 INT23 pins.

■ Clocks

Table 18.2-1 lists the clock used by the external interrupt controllers.

Table 18.2-1 Clock used by the external interrupt controllers

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

18.3 Pins

This section explains the pins of the external interrupt controllers.

■ Overview

The external interrupt controllers have the following pins:

- INT0 to INT23 pins

These are external interrupt signal input pins. These pins are multiplexed pins.

18.4 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 18.4-1 lists the registers for the external interrupt controllers.

Table 18.4-1 Registers for the external interrupt controllers

Channel	Abbreviated Register Name	Register Name	Reference
Common	ELVR0	External interrupt request level register 0	18.4.1
	EIRR0	External interrupt request register 0	18.4.2
	ENIR0	Enable interrupt request register 0	18.4.3
	ELVR1	External interrupt request level register 1	18.4.1
	EIRR1	External interrupt request register 1	18.4.2
	ENIR1	Enable interrupt request register 1	18.4.3
	ELVR2	External interrupt request level register 2	18.4.1
	EIRR2	External interrupt request register 2	18.4.2
	ENIR2	Enable interrupt request register 2	18.4.3

18.4.1 External Interrupt Request Level Registers (ELVR0 to ELVR2)

These registers set the edges/levels to be detected for external interrupt requests.

Figure 18.4-1 shows the bit configuration of the external interrupt request level registers (ELVR0 to ELVR2).

Figure 18.4-1 Bit configuration of the external interrupt request level registers (ELVR0 to ELVR2)

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

(Continued)

(Continued)

External interrupt request level register 2 (ELVR2)								
bit	15	14	13	12	11	10	9	8
	LB23	LA23	LB22	LA22	LB21	LA21	LB20	LA20
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
	LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
R/W: Read/Write								

LB23 to LB0, LA23 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 LB23 bits correspond to the INT0 to INT23 bits, and the LA0 to LA23 bits similarly correspond to the INT0 to INT23 bits. For example, the INT0 pin is set with the LB0 and LA0 bits.

LB23 to LB0	LA23 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 "18.5.2 Return from Standby Mode".

<Notes>

- For return from STOP, only level can be set for ch.0 to ch.7 and all request levels can be set for ch.8 to ch.23.
- For detection of an edge/level specified by these bits, the pulse width of the signal must be 4T 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 to EIRR2) 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 to EIRR2).

However, even when the external interrupt request register (EIRR0 to EIRR2) is cleared, the external interrupt request remains as is while any signals at the effective level are input from the INT0 to INT23 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 "18.5 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 to ELVR2).
 2. Write "0" in the external interrupt request register (EIRR0 to EIRR2) to clear the interrupt source.

18.4.2 External Interrupt Request Registers (EIRR0 to EIRR2)

These registers maintain the states of interrupt sources of external interrupt requests (indicating which pins have generated the external interrupt requests).

Figure 18.4-2 shows the bit configuration of the external interrupt request registers (EIRR0 to EIRR2).

Figure 18.4-2 Bit configuration of the external interrupt request registers (EIRR0 to EIRR2)

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
External interrupt request register 2 (EIRR2)								
bit	7	6	5	4	3	2	1	0
	ER23	ER22	ER21	ER20	ER19	ER18	ER17	ER16
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								

ER23 to ER0 (External interrupt request flag bits)

These bits indicate that external interrupt requests have been detected.

The ER0 to ER23 bits correspond to the INT0 to INT23 pins. For example, the ER0 bit is used to detect external interrupt requests from the INT0 pin, and the ER23 bit is used to detect external interrupt requests from the INT23 pin.

An external interrupt request is generated when "1" is set in any of the EN0 to EN23 bits of an enable interrupt request register (ENIR0 to ENIR2) and the corresponding bit among the ER0 to ER23 bits becomes "1".

ER23 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 INT23 pins when "L" level detection/"H" level detection has been set as the detection condition by an external interrupt request level register (ELVR0 to ELVR2), "1" is set in the corresponding bit among the ER23 to ER0 bits even after the bit is cleared.
-

18.4.3 Enable Interrupt Request Registers (ENIR0 to ENIR2)

These registers enable/disable external interrupt requests.

Figure 18.4-3 shows the bit configuration of the enable interrupt request registers (ENIR0 to ENIR2).

Figure 18.4-3 Bit configuration of the enable interrupt request registers (ENIR0 to ENIR2)

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
Enable interrupt request register 2 (ENIR2)								
bit	7	6	5	4	3	2	1	0
	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
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								

EN23 to EN0 (Interrupt enable bits)

These bits enable/disable external interrupts.

Each of the EN0 to EN23 bits corresponds to the respective bits of the external interrupt request registers (EIRR0 to EIRR2).

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.

18.5 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the external interrupt controllers and provides examples of setting procedures.

18.5.1 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.

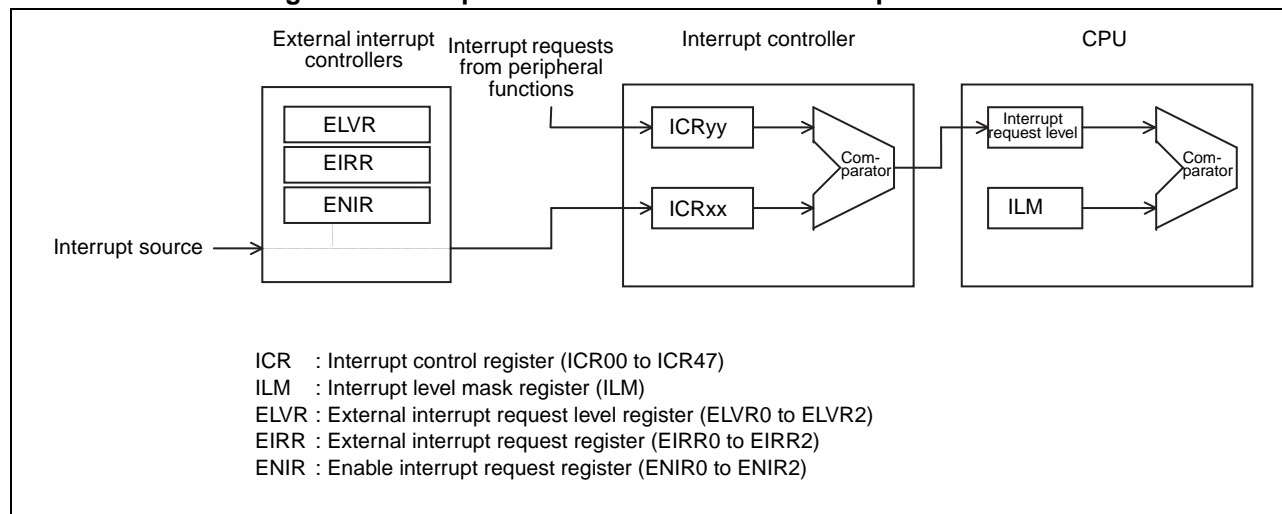
The edge/level to be detected can be selected from the following 4 types:

- "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 INT23 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 INT23 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 18.5-1 shows operation with the external interrupt controllers.

Figure 18.5-1 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 to ENIR2).
2. Change the detection condition (effective edge /level) by using an external interrupt request level register (ELVR0 to ELVR2).
3. Read the external interrupt request level register (ELVR0 to ELVR2).
4. Clear interrupt sources by using an external interrupt request register (EIRR0 to EIRR2).
5. Enable external interrupts by using the enable interrupt request register (ENIR0 to ENIR2).
(16-bit data can be written at the same time to 3. and 4.)

<Notes>

- Before making settings for the external interrupt controller, disable external interrupts by using an enable interrupt request register (ENIR0 to ENIR2).
 - Before enabling output of external interrupt requests, clear interrupt sources by using an external interrupt request register (EIRR0 to EIRR2).
-

■ 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 to ELVR2) in a signal input to any of the INT0 to INT23 pins.
2. The external interrupt controller determines whether external interrupts are enabled by checking the enable interrupt request registers (ENIR0 to ENIR2).
3. If external interrupts are enabled, the external interrupt controller outputs an external interrupt request to the interrupt controller.

When the request level is edge request, a minimum pulse width of 3T (T: Peripheral clock (PCLK) cycle) is necessary in order to detect the occurrence of edge.

■ 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 to EIRR2) 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 to EIRR2). This operation clears the interrupt source, and the external interrupt request is canceled.

However, even when the external interrupt request register (EIRR0 to EIRR2) is cleared, the external interrupt remains at the interrupt controller, to which a request it has been output, while any signals at the effective level are input from the INT0 to INT23 pins.

Figure 18.5-2 shows the state of an interrupt source being maintained, and Figure 18.5-3 shows the clearing of an interrupt source.

Figure 18.5-2 Maintaining the state of an interrupt source

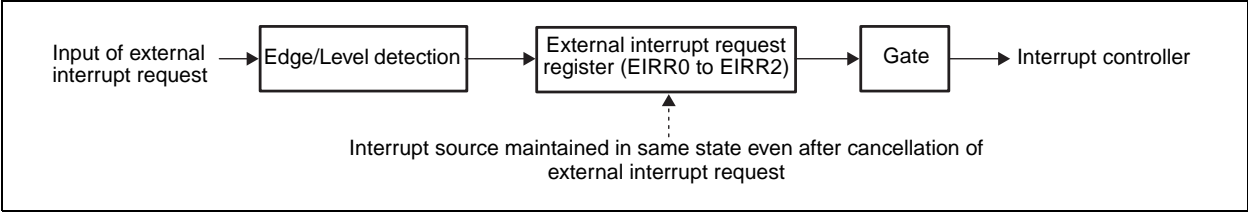
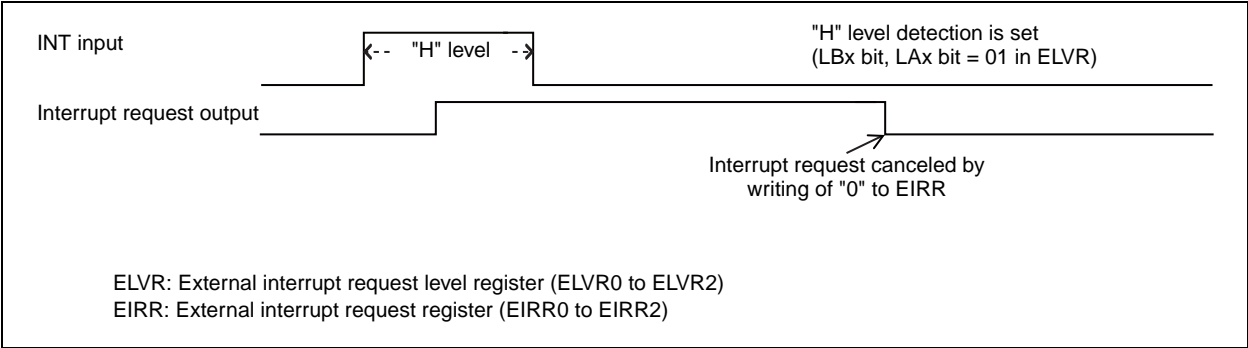


Figure 18.5-3 Clearing of an interrupt source



18.5.2 Return from Standby Mode

■ Overview

External interrupt requests can be used for a return from standby mode (main timer mode or stop mode). A signal already input to any of the INT0 to INT23 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 INT23 pins must be made with the enable interrupt request registers (ENIR0 to ENIR2):

- 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 INT23 pins in standby mode.

Table 18.5-1 shows the relationship between external interrupt request detection conditions and the levels for returning from standby mode.

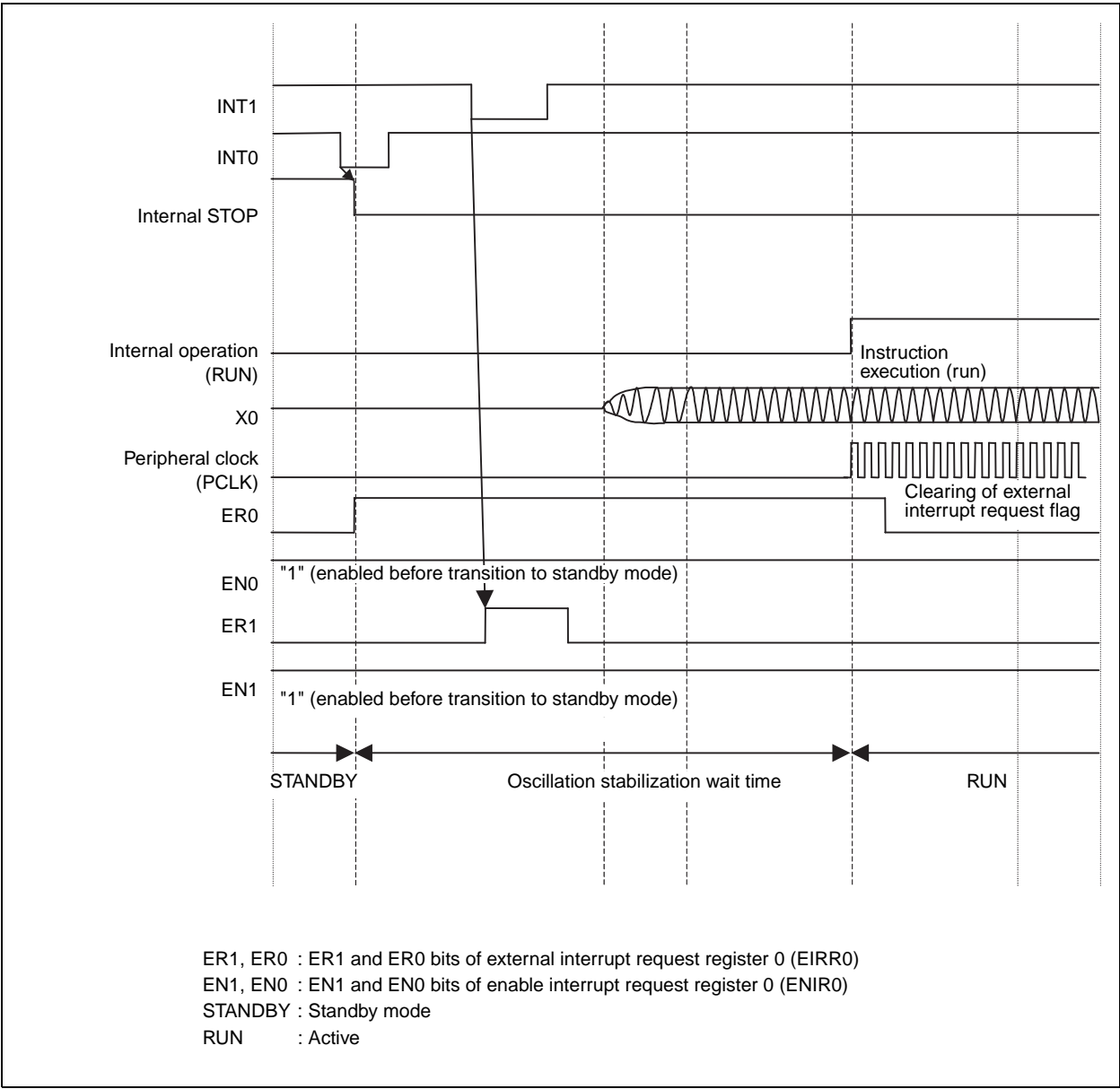
Table 18.5-1 Relationship between external interrupt request detection conditions and the levels for returning from standby mode

Detection Condition	LB23 to LB0	LA23 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 INT23 pins
Falling edge detection	1	1	"H" level detection at the INT0 to INT7 pins, and falling edge detection at the INT8 to INT23 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 18.5-4 shows an example of operation at the time of return from standby mode, where the INT0 and INT1 pins are used.

Figure 18.5-4 Operation when returning from standby mode



18.5.3 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 INT23 pins must be made with the enable interrupt request registers (ENIR0 to ENIR2):

- 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 INT23 pins in sleep mode.

CHAPTER 19 Watchdog Timer

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

- 19.1 Overview
- 19.2 Configuration
- 19.3 Registers
- 19.4 Explanation of Operations and Setting Procedure Examples

19.1 Overview

The watchdog timer is a monitoring timer used to determine whether software hangs up or performs other abnormal operations.

■ 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.

- 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 main timer mode).
- The watchdog timer can detect a transition to standby mode (main timer 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 period: peripheral clock (PCLK) \times (2^9 to 2^{24})

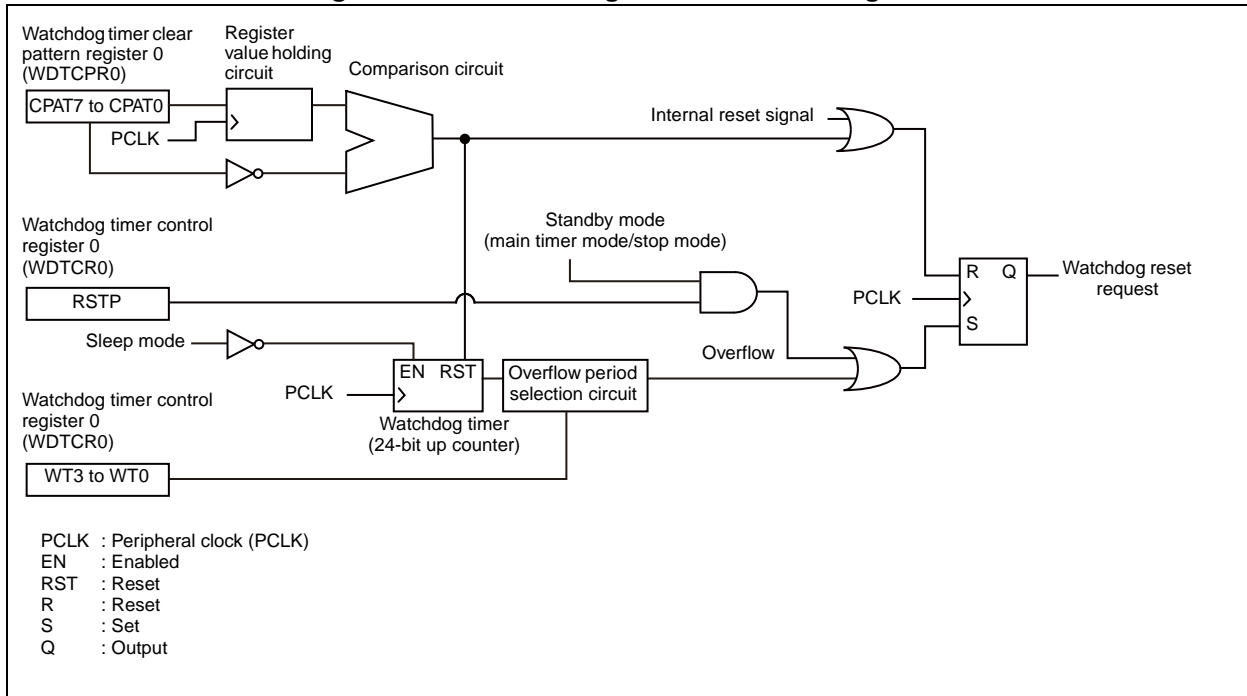
19.2 Configuration

This section shows the configuration of the watchdog timer.

■ Block diagram of the watchdog timer

Figure 19.2-1 is a block diagram of the watchdog timer.

Figure 19.2-1 Block diagram of the watchdog timer



- **Watchdog timer control register 0 (WDTCR0)**
This register controls the operation of the watchdog timer.
- **Watchdog timer clear pattern register 0 (WDTCPR0)**
This register activates and clears the watchdog timer.
- **Watchdog timer**
This is a 24-bit up counter.
- **Register value holding circuit**
This circuit retains the value written in watchdog timer clear pattern register 0 (WDTCPR0).
- **Comparison circuit**
This circuit compares the value written in watchdog timer clear pattern register 0 (WDTCPR0) with the previous value that was written.
- **Overflow period selection circuit**
This circuit selects the overflow period of the watchdog timer.

■ Clocks

Table 19.2-1 lists the clock used by the watchdog timer.

Table 19.2-1 Clock used by the watchdog timer

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

19.3 Registers

This section explains the configuration and functions of registers for the watchdog timer.

■ List of registers for the watchdog timer

Table 19.3-1 lists the registers for the watchdog timer.

Table 19.3-1 Registers for the watchdog timer

Abbreviated Register Name	Register Name	Reference
WDTCR0	Watchdog timer control register 0	19.3.1
WDTCPRO	Watchdog timer clear pattern register 0	19.3.2

19.3.1 Watchdog Timer Control Register 0 (WDTCR0)

This register controls the operation of the watchdog timer.

Figure 19.3-1 shows the bit configuration of watchdog timer control register 0 (WDTCR0).

Figure 19.3-1 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.

[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 (main timer mode/stop mode) while the watchdog timer 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 (main timer 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 (main timer mode/stop mode) is detected.

<Notes>

- To use standby mode (main timer mode/stop mode), set "0" in this bit.
- This register can be written only before the watchdog timer is activated. If "1" is set in this bit after the watchdog timer is activated, standby mode (main timer 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 clearing to watchdog reset request output.

WT3 to WT0	Watchdog Timer Period
0000	$PCLK \times 2^9$
0001	$PCLK \times 2^{10}$
0010	$PCLK \times 2^{11}$
0011	$PCLK \times 2^{12}$
0100	$PCLK \times 2^{13}$
0101	$PCLK \times 2^{14}$
0110	$PCLK \times 2^{15}$
0111	$PCLK \times 2^{16}$
1000	$PCLK \times 2^{17}$
1001	$PCLK \times 2^{18}$
1010	$PCLK \times 2^{19}$
1011	$PCLK \times 2^{20}$
1100	$PCLK \times 2^{21}$
1101	$PCLK \times 2^{22}$
1110	$PCLK \times 2^{23}$
1111	$PCLK \times 2^{24}$

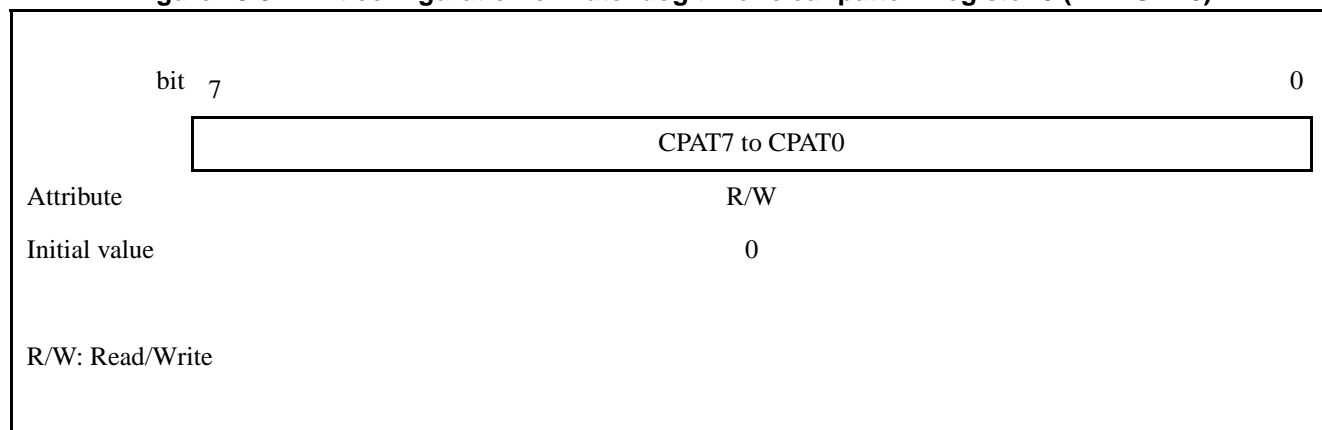
PCLK : Period of Peripheral clock (PCLK)

19.3.2 Watchdog Timer Clear Pattern Register 0 (WDT CPR0)

This register activates and clears the watchdog timer.

Figure 19.3-2 shows the bit configuration of watchdog timer clear pattern register 0 (WDT CPR0).

Figure 19.3-2 Bit configuration of watchdog timer clear pattern register 0 (WDT CPR0)



[bit7 to bit0]: CPAT7 to CPAT0 bits

The watchdog timer 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 is activated, the timer must be cleared before the timer period has elapsed.

To clear the watchdog timer, invert the bit pattern written in these bits and write the inverted value to the bits.

For details of clearing the watchdog timer, see "■ Clearing the watchdog timer" in "19.4.1 Operations of the Watchdog Timer".

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.	

19.4 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.

19.4.1 Operations of the Watchdog Timer

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.

■ Overview

While the watchdog timer is operating, if it is not cleared before the specified period has elapsed, it judges that software has hung up and outputs a watchdog reset request to the CPU.

A watchdog reset request is also output if an incorrect value is written to watchdog timer clear pattern register 0 (WDTCPR0) or at the transition time of the CPU to standby mode (main timer mode/stop mode).

Also, the watchdog timer stops the counting operation when the CPU is stopped.

■ 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 (main timer mode/stop mode) (RSTP)

<Notes>

- 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 (main timer 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 (main timer 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.

<Notes>

- 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 (main timer 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 "11.5 Explanation of Operations" of "CHAPTER 11 Reset".

CHAPTER 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.

This section explains the 16-bit reload timer configuration.

Figure 20.2-1 is a block diagram of the 16-bit reload timer.

The diagram illustrates the internal architecture of the 16-bit timer module. Key components and their connections include:

- Peripheral bus:** Connected to TMRLRA (Read/Write), TMR (Read only), and the Count control block (Read/Write).
- Counting Path:** The peripheral clock (PCLK) passes through a prescaler and a clock select circuit to the Count control block. The Count control block also receives a trigger signal from the TRG register and a gate signal from the Gate Control block. Its output is the timer count, which is also fed back to the TMR register.
- Reload Path:** The TMRLRA register provides a reload value to the TMR register.
- Control and Status Registers:** The TMCSR register contains control bits (GATE, TRGM1, TRGM0, CSL2, CSL1, CSL0, TRG, CNTE) and status bits (OUTL, UF, INTE, RELD). The Count control block has an "End of one shot" output that triggers the RELD register.
- Interrupt and Output:** The UF (Underflow) status bit generates an interrupt request. The OUTL (Output Latch) register is connected to the TMR register and the Count control block. The TOUT0 to TOUT2 pins are connected to the OUTL register.
- Edge and Gate Control:** The Edge Control block receives the peripheral clock (PCLK) and the timer count. The Gate Control block receives the peripheral clock (PCLK) and the timer count. Both blocks have select inputs connected to the TMCSR register.

Legend:

- TMRLRA : 16-bit timer reload register A (TMRLRA0 to TMRLRA2)
- TMR : 16-bit timer register (TMR0 to TMR2)
- TMCSR : Timer control status register (TMCSR0 to TMCSR2)

The bits are in random order.

- 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 TIN0 to TIN2 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 TIN0 to TIN2 pins are used as gate input pins.
- Count controller
The count controller controls the counts of the 16-bit reload timer.

■ Clocks

Table 20.2-1 shows the clock used for the 16-bit reload timer.

Table 20.2-1 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 TIN0 to TIN2 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.

- TOUT0 to TOUT2 pins
16-bit reload timer wave form output pin These pins are multiplexed pins.
- TIN0 to TIN2 pins
16-bit reload timer input pin This inputs count clock, clock, trigger, or gate depending on its setting.
These pins are multiplexed pins.

■ Relationship between pins and channels

Table 20.3-1 outlines the relationship between channels and pins.

Table 20.3-1 Relationship between channels and pins

Channel	Wave Form Output Pin	Input Pin
0	TOUT0	TIN0
1	TOUT1	TIN1
2	TOUT2	TIN2

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 20.4-1 lists the registers of the 16-bit reload timer.

Table 20.4-1 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 20.4-1 shows the bit configuration of the timer control status registers (TMCSR0 to TMCSR2).

Figure 20.4-1 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 TIN0 to TIN2 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 TIN0 to TIN2 pins (GATE = 0).
 Select an effective edge.
 When the edge set with this bit is detected in the signal input from the TIN0 to TIN2 pins, the down counter starts counting down.
 - Select the gate function with TIN0 to TIN2 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 TIN0 to TIN2 pins.

TRGM1	TRGM0	When the Trigger Input Is Selected * (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

* 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 TIN0 to TIN2 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		External clock

* 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 TIN0 to TIN2 pins.

- Trigger input function: When an effective edge is input from TIN0 to TIN2 pins, a countdown starts.
- Gate function: A countdown is performed only while the effective level signal is input from TIN0 to TIN2 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 TOUT0 to TOUT2 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>

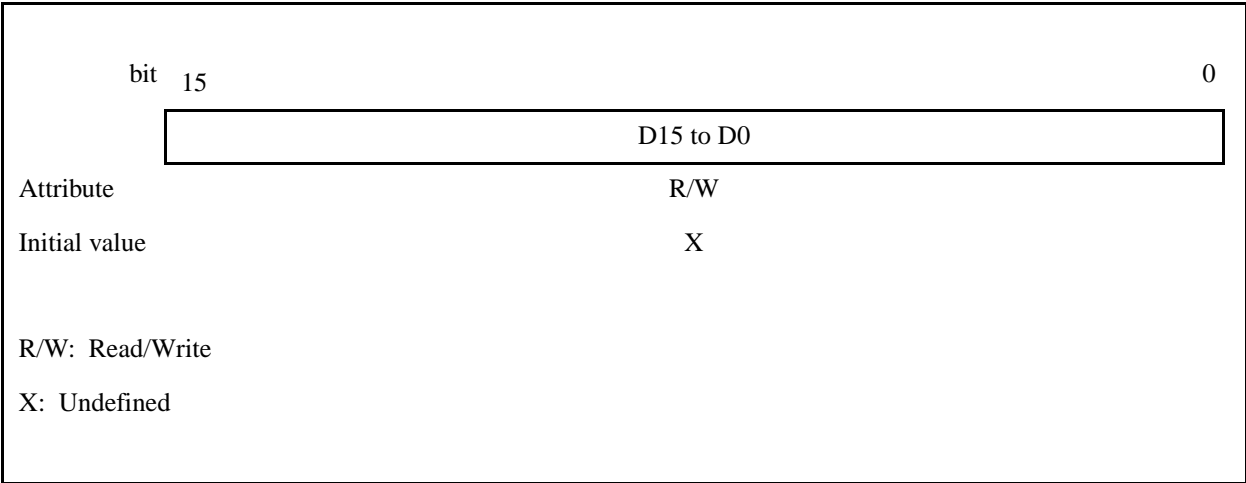
- The down counter does not operate while the CNTE bit is "0" even if "1" is written to this bit.
 - When the 16-bit reload timer operation is enabled (CNTE=1), if "1" is written to this bit, the down counter starts regardless of the setting of TRGM1 or TRGM0 bit.
-

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 20.4-2 shows the bit configuration of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2).

Figure 20.4-2 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 TOUT0 to TOUT2 pins is inverted.

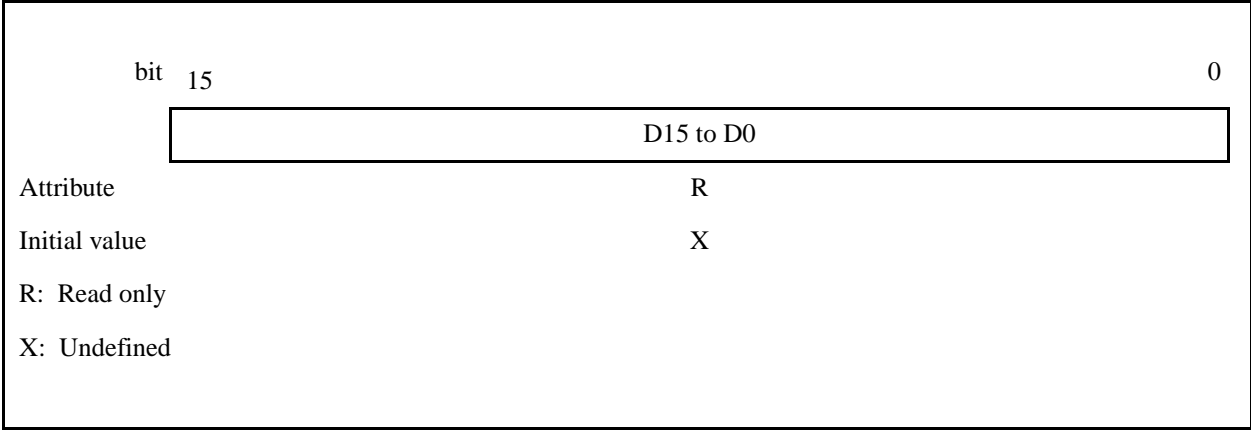
<Note>
Be sure to access this register in units of half words.

20.4.3 16-bit Timer Register (TMR0 to TMR2)

When this register is read, the down counter value can be read.

Figure 20.4-3 shows the bit configuration of the 16-bit timer registers (TMR0 to TMR2).

Figure 20.4-3 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 20.5-1 outlines the interrupts that can be used with the 16-bit reload timer

Table 20.5-1 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 "APPENDIX C 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 "CHAPTER 12 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 TIN0 to TIN2 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 TOUT0 to TOUT2 pins.

The signal level of the signals output from TOUT0 to TOUT2 pins varies with the settings of OUTL bit of the timer control status register (TMCSR0 to TMCSR2).

● In reload mode

Table 20.6-1 shows the signal level of the signals output from TOUT0 to TOUT2 pins in reload mode.

Table 20.6-1 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 20.6-2 shows the signal level of the signals output from TOUT0 to TOUT2 pins in one shot mode.

Table 20.6-2 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 20.6-1 shows the OUTL bits of the timer control status register (TMCSR0 to TMCSR2) and their output wave forms.

Figure 20.6-1 OUTL bits of the timer control status registers (TMCSR0 to TMCSR2) and their output wave forms

Mode	OUTL	Initial value	Activation trigger	Counting	Underflow	Underflow	Underflow
Reload	0						
	1						
One shot	0				trigger wait state		
	1						

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 ($= 2^6$)

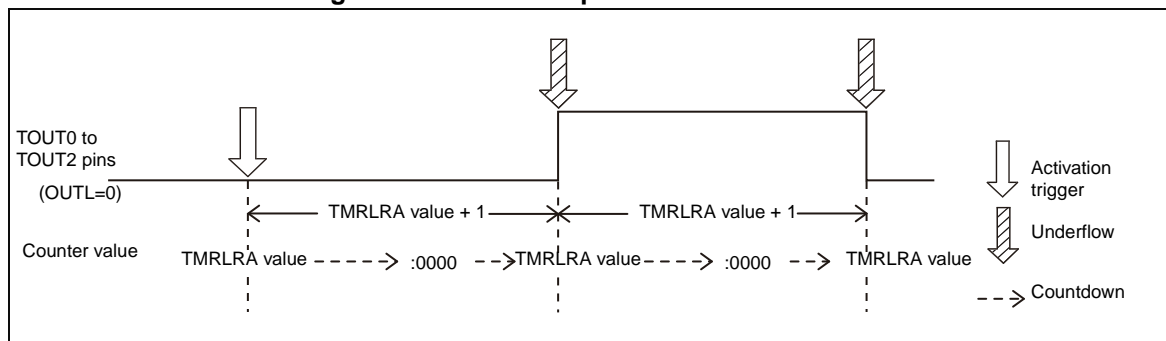
● Operation mode settings

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 20.6-2 shows the basic operation in reload mode.

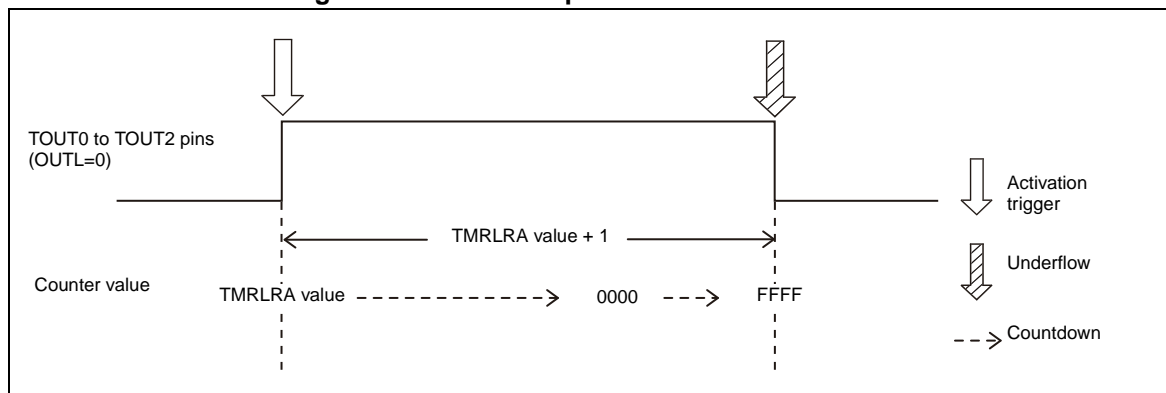
Figure 20.6-2 Basic operation in reload mode



- One shot mode (RELD = 0)

In this mode, counting stops when the down counter enters an underflow condition. Figure 20.6-3 shows the basic operation in one shot mode.

Figure 20.6-3 Basic operation in one shot mode



● TIN0 to TIN2 pin function settings

Using TRGM1 and TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) and the GATE bit, the function of TIN0 to TIN2 pins can be selected from the following list.

Table 20.6-3 shows the combination of bits.

Table 20.6-3 Combination of bits

TRGM1, TRGM0	GATE	Pin Function
00	0	TIN0 to TIN2 pins do not work.
01	0	TIN0 to TIN2 pins operate as the trigger input function. The effective edge is a rising edge.
10	0	TIN0 to TIN2 pins operate as the trigger input function. The effective edge is a falling edge.
11	0	TIN0 to TIN2 pins operate as the trigger input function. The effective edge is both edges.
00/10	1	TIN0 to TIN2 pins operate as the gate input function. The effective level is "L".
01/11	1	TIN0 to TIN2 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 TOUT0 to TOUT2 pins in interval timer mode is explained below.

$$\text{Pulse width} = T \times (L + 1)$$

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.

$$T \times (L + 1)$$

T Cycles of the count clock

L Value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

■ Operations in reload mode (TIN0 to TIN2 pins = trigger input)

In this mode, TIN0 to TIN2 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:

- One of the TRGM1, TRGM0 bits = 01 to 11
- GATE bit = 0
- 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.

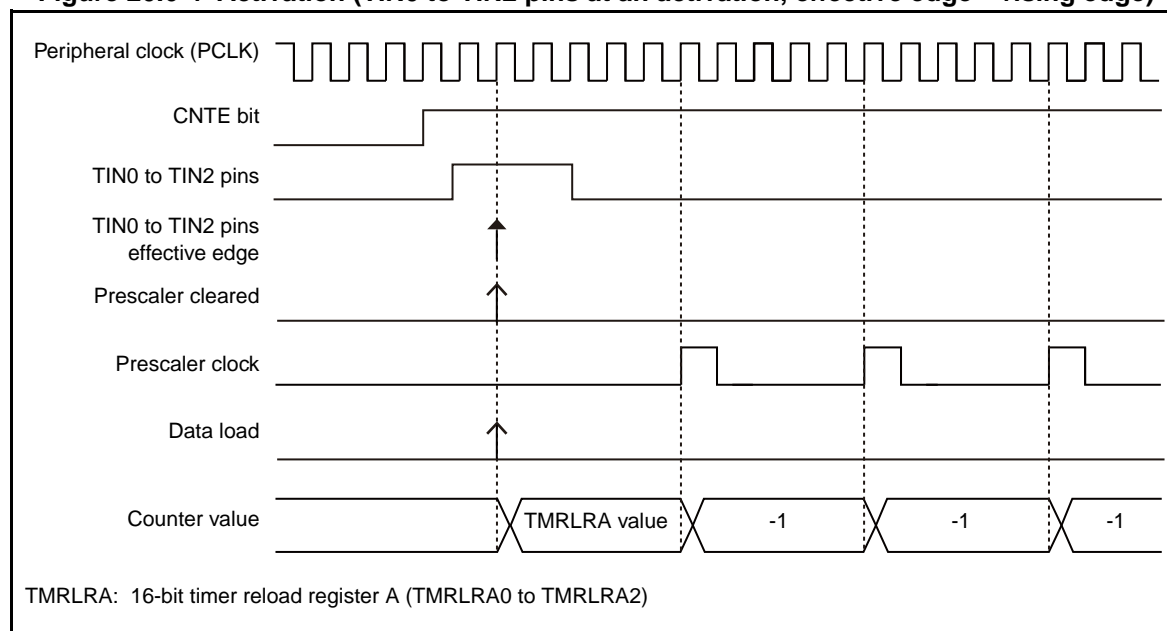
2. The activation trigger is input in either of the following ways:

- Input the edge set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TIN0 to TIN2 pins.
- 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 20.6-4 shows an activation.

Figure 20.6-4 Activation (TIN0 to TIN2 pins at an activation, effective edge = rising edge)



<Note>

Be sure that the pulse width of the activation trigger input from TIN0 to TIN2 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).

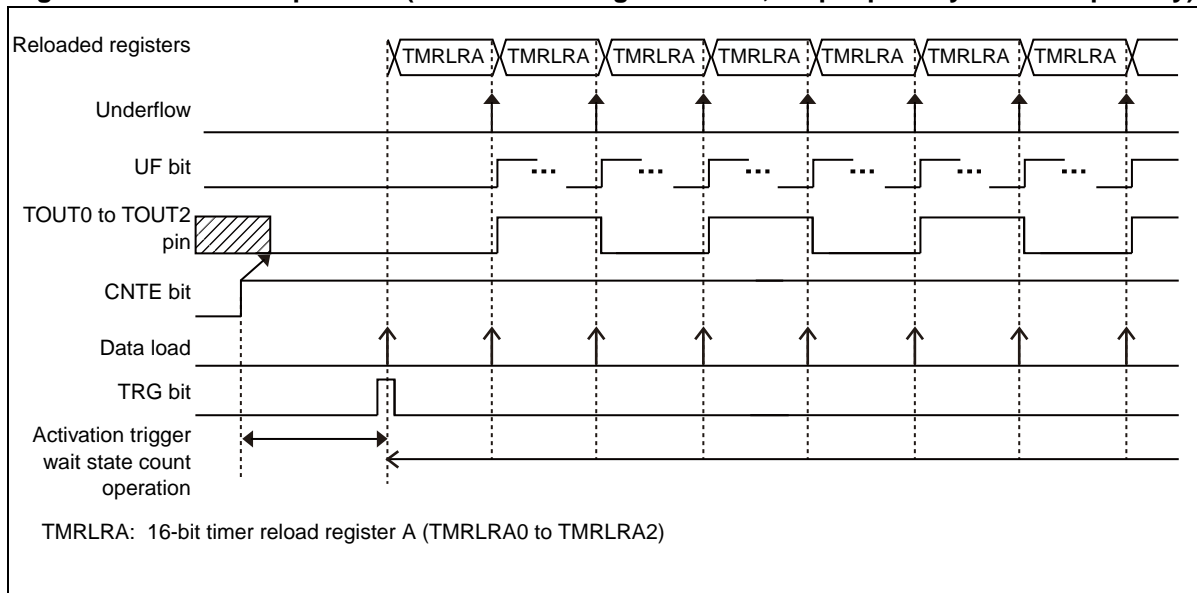
If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TOUT0 to TOUT2 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 20.6-5 shows the count operation.

Figure 20.6-5 Count operation (activation through software, output polarity = normal polarity)



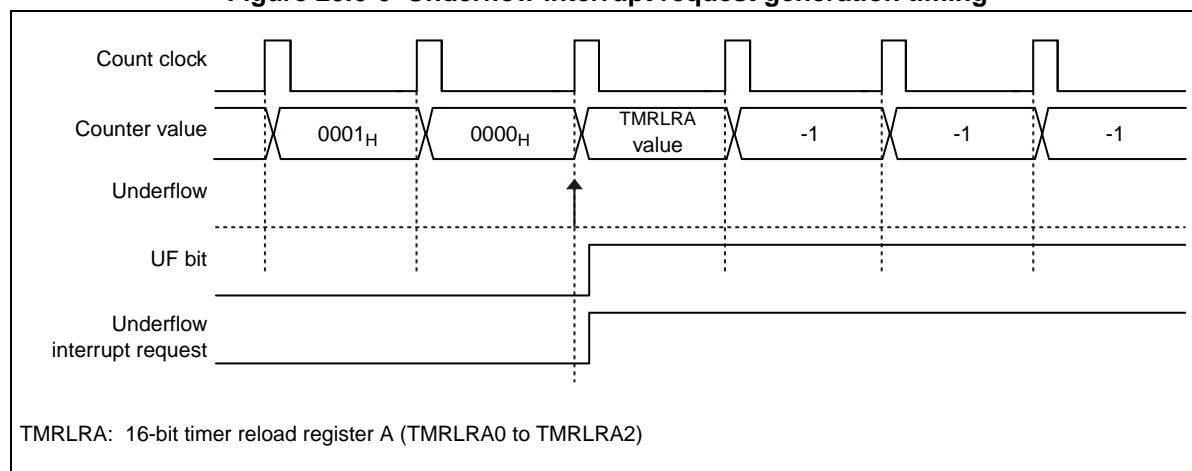
● Operation of interrupt processing

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 20.6-6 shows the underflow interrupt request generation timing.

Figure 20.6-6 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.

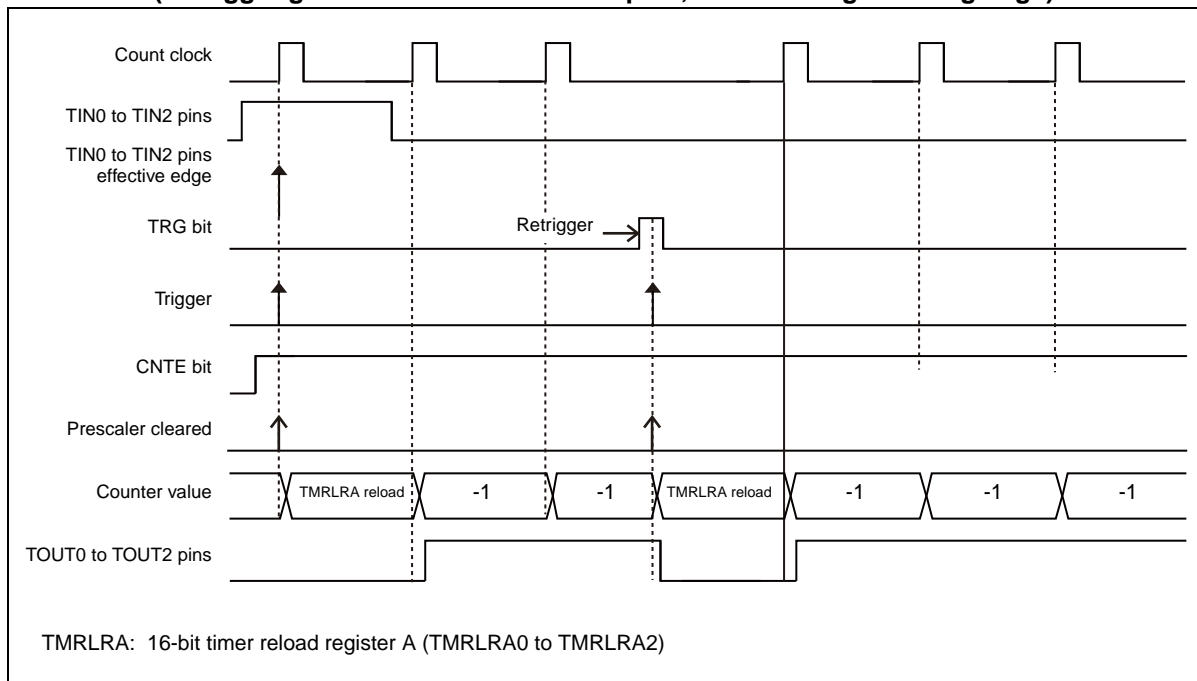
● Retrigger operation

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 TIN0 to TIN2 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 20.6-7 shows the operation when a retrigger is generated.

Figure 20.6-7 Operation when a retrigger is generated.
(Retrigger generated on TIN0 to TIN2 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 (TIN0 to TIN2 pins = at a gate input)

In this mode, TIN0 to TIN2 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:

- TRGM0 bit = 0/1
- GATE bit = 1
- 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)

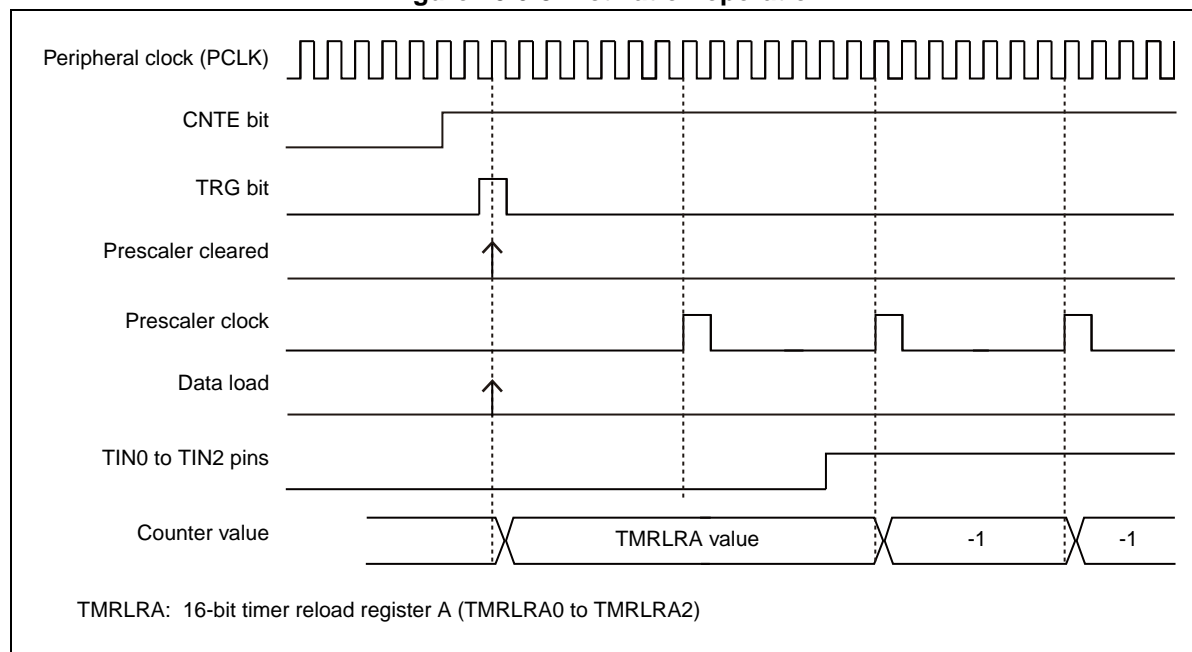
The prescaler is cleared. The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is loaded to the down counter, and the 16-bit reload timer enters the effective input polarity (from TIN0 to TIN2 pins) wait state.

3. Input the signal with the level set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TIN0 to TIN2 pins.

The counter starts counting.

Figure 20.6-8 shows an activation operation.

Figure 20.6-8 Activation operation



<Note>

Be sure that the effective level input from TIN0 to TIN2 pins never falls below 2T (T: cycle of the peripheral clock (PCLK)).

● Count operation

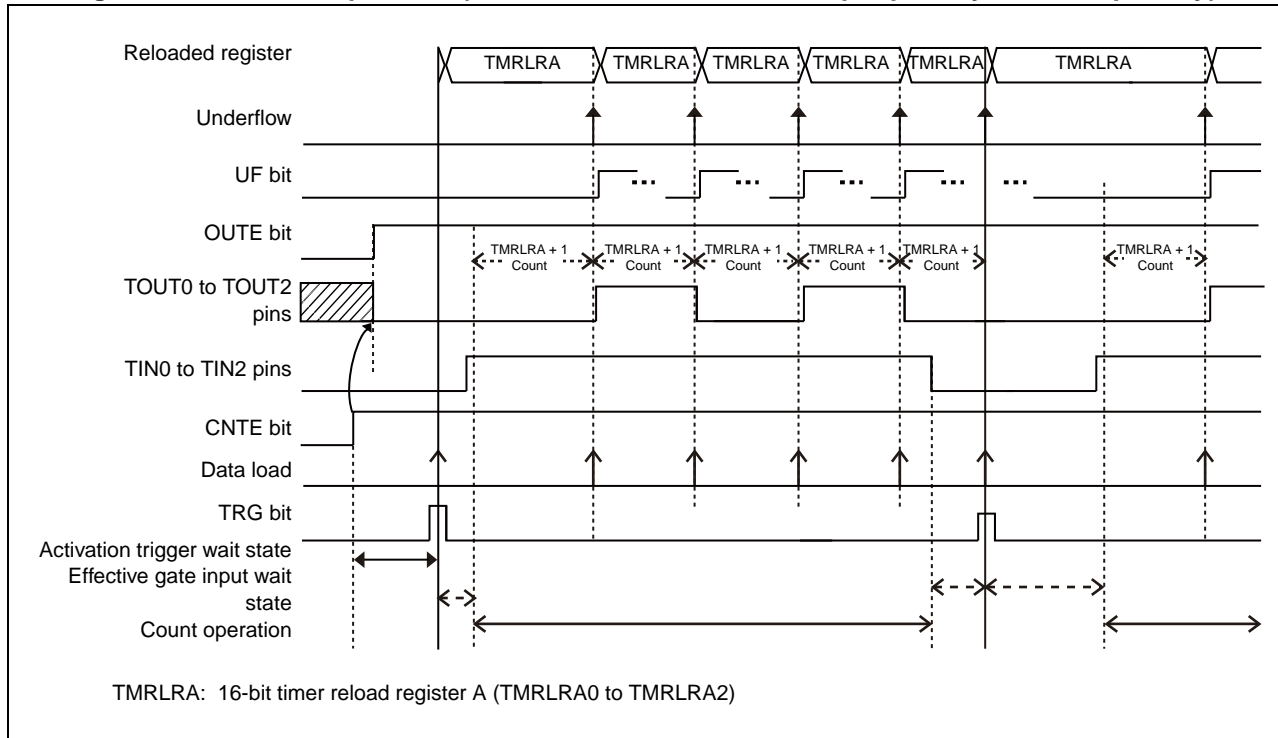
Only while the effective level signal is input from TIN0 to TIN2 pins does the down counter perform a countdown from the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) synchronously with the count clock.

If the effective level signal is not input from TIN0 to TIN2 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 TIN0 to TIN2 pins = trigger input function is set. See "■ Operations in reload mode (TIN0 to TIN2 pins = trigger input)".

Figure 20.6-9 shows the count operation.

Figure 20.6-9 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 (TIN0 to TIN2 pins = trigger input)".

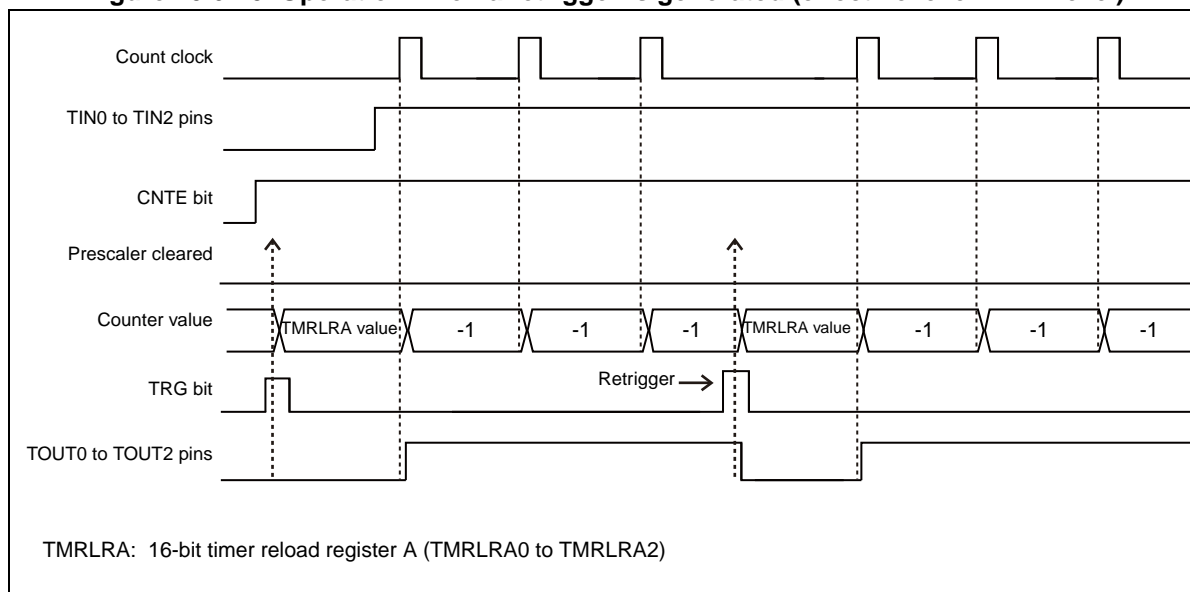
● Retrigger operation

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 TIN0 to TIN2 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 TIN0 to TIN2 pin in such condition, counting starts. Figure 20.6-10 shows the operation when a retrigger is generated.

Figure 20.6-10 Operation when a retrigger is generated (effective level = "H" level)



<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 one shot mode (TIN0 to TIN2 pins = trigger input)

When TIN0 to TIN2 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:

- One of the TRGM1, TRGM0 bits = 01 to 11
- GATE bit = 0
- RELD bit = 0

● Activate

The operation is the same as in reload mode. See "■ Operations in reload mode (TIN0 to TIN2 pins = trigger input)".

However, if an activation trigger is detected in one shot mode, the signal level of the signals output from TOUT0 to TOUT2 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 TOUT0 to TOUT2 pins is initialized.
- Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "FFFF_H").

Figure 20.6-11 shows the count operation when TIN0 to TIN2 pins are used for activation.

Figure 20.6-11 Count operation (effective edge = rising edge, output polarity = normal polarity)

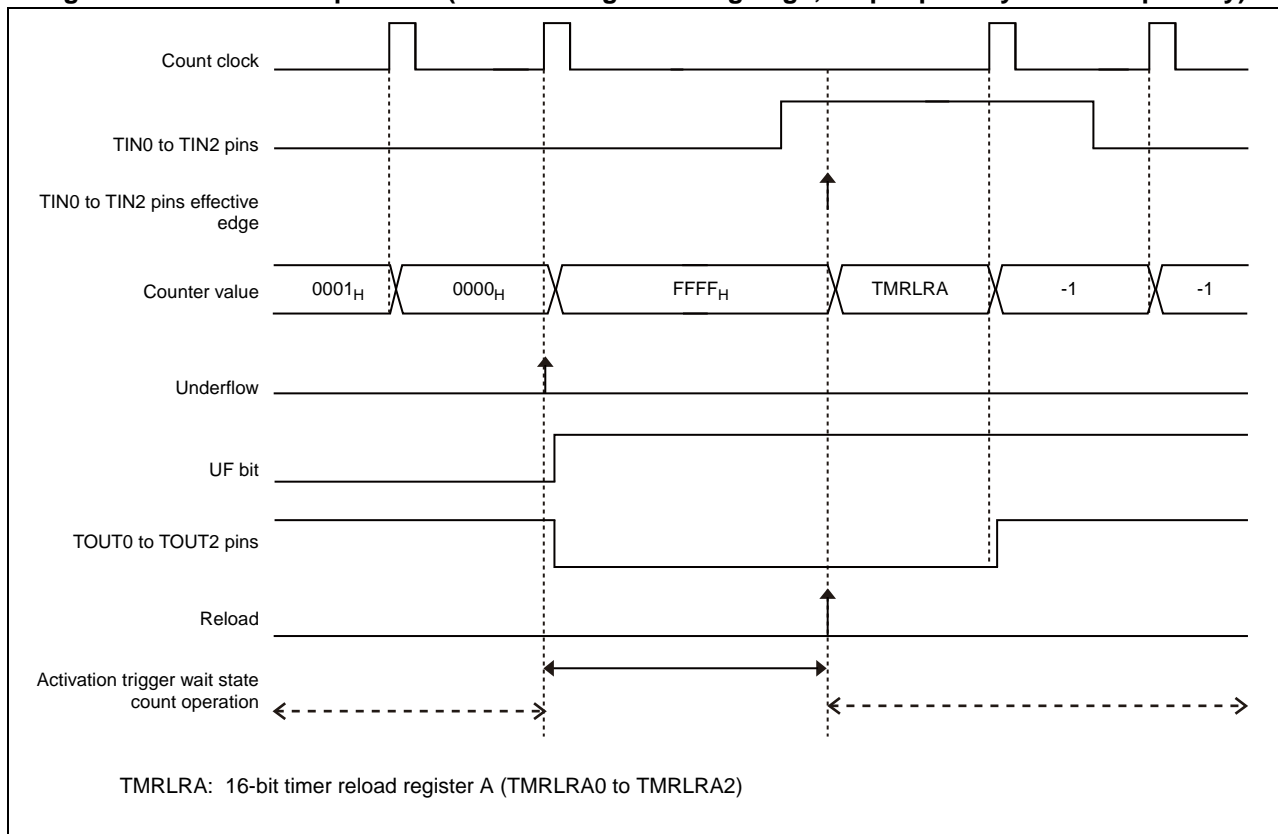
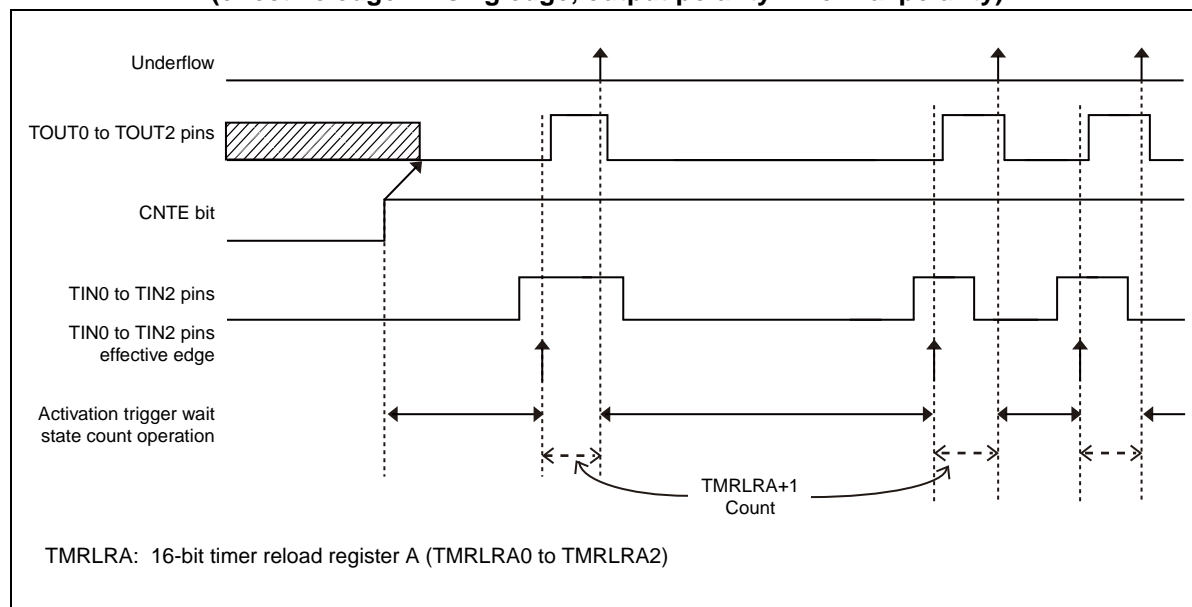


Figure 20.6-12 shows the detailed operation when an underflow occurs.

Figure 20.6-12 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 (TIN0 to TIN2 pins = trigger input)".

● Retrigger operation

The operation is the same as in reload mode. See "■ Operations in reload mode (TIN0 to TIN2 pins = trigger input)".

However, if a retrigger is detected in one shot mode, the signal level of the signals output from TOUT0 to TOUT2 pins is inverted.

■ Operations in one shot mode (TIN0 to TIN2 pins = gate input)

When TIN0 to TIN2 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:

- TRGM0 bit = 0/1
- GATE bit = 1
- RELD bit = 0

● Activate

The operation is the same as in reload mode. See "■ Operations in reload mode (TIN0 to TIN2 pins = at a gate input)".

However, if an activation trigger is detected in one shot mode, the signal level of the signals output from TOUT0 to TOUT2 pins is inverted.

● Count operation

Only while the effective level signal is input from TIN0 to TIN2 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.

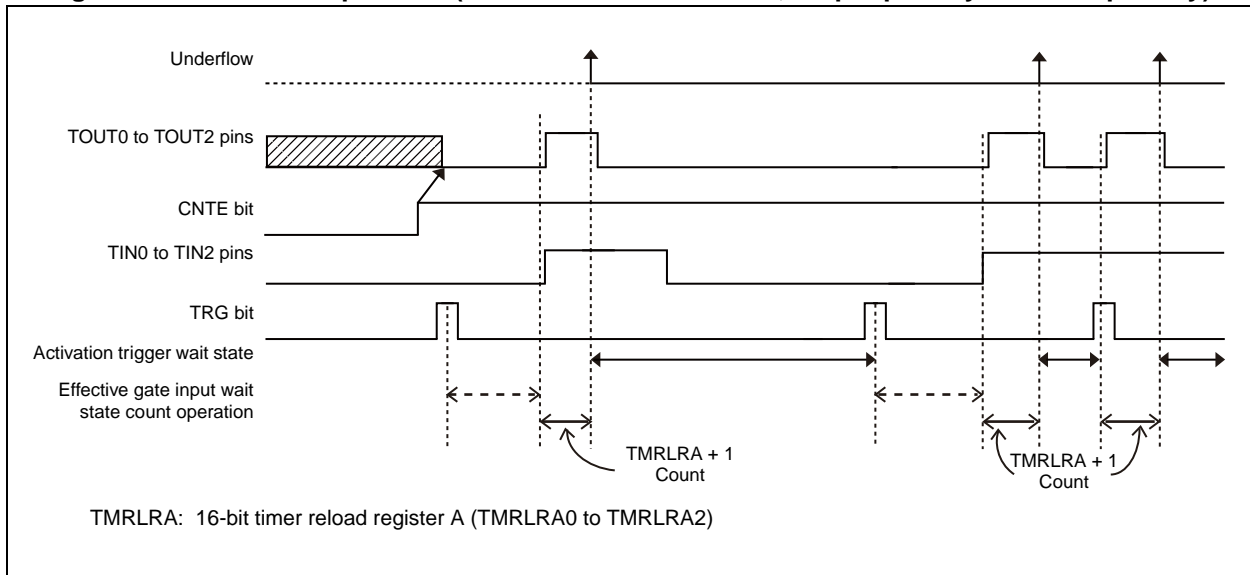
If the effective level signal is not input from TIN0 to TIN2 pins, the down counter stops counting. If the effective level signal is input while the down counter is stopped, the counter starts counting from the value where it stopped.

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 TOUT0 to TOUT2 pins is initialized.
- Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "FFFF_H").

Figure 20.6-13 shows the count operation.

Figure 20.6-13 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 (TIN0 to TIN2 pins = trigger input)".

● Retrigger operation

The operation is the same as in reload mode. See "■ Operations in reload mode (TIN0 to TIN2 pins = at a gate input)".

However, if a retrigger is detected in one shot mode, the signal level of the signals output from TOUT0 to TOUT2 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 TIN0 to TIN2 pins are counted. It performs a countdown every time an effective edge is input from TIN0 to TIN2 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 TIN0 to TIN2 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	TIN0 to TIN2 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:

- One of the TRGM1, TRGM0 bits = 01 to 11
- 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 input from TIN0 to TIN2 pins) wait state.

3. Input the effective edge set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TIN0 to TIN2 pins.

The counter starts counting.

● Count operation

Every time an effective edge is detected in the input signal from TIN0 to TIN2 pins, it performs a countdown.

Figure 20.6-14 to Figure 20.6-16 show the count timing.

Figure 20.6-14 Count timing (effective edge = rising edge)

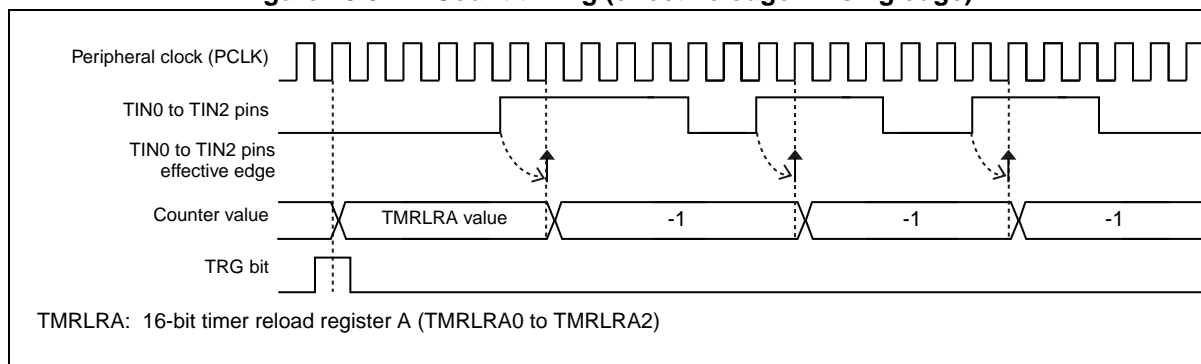


Figure 20.6-15 Count timing (effective edge = falling edge)

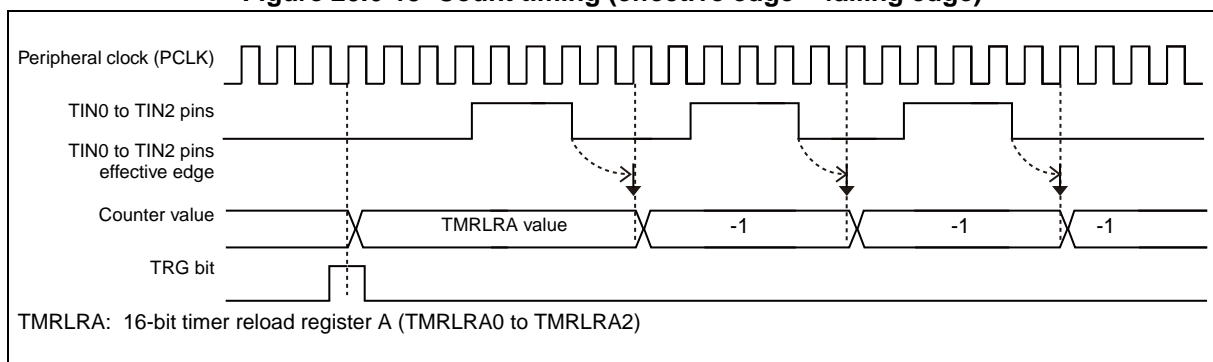
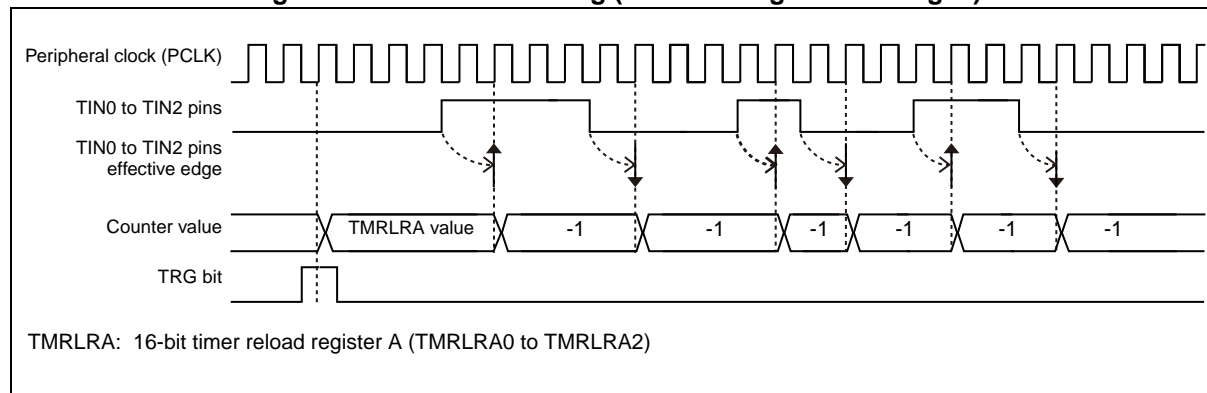


Figure 20.6-16 Count timing (effective edge = both edges)



If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

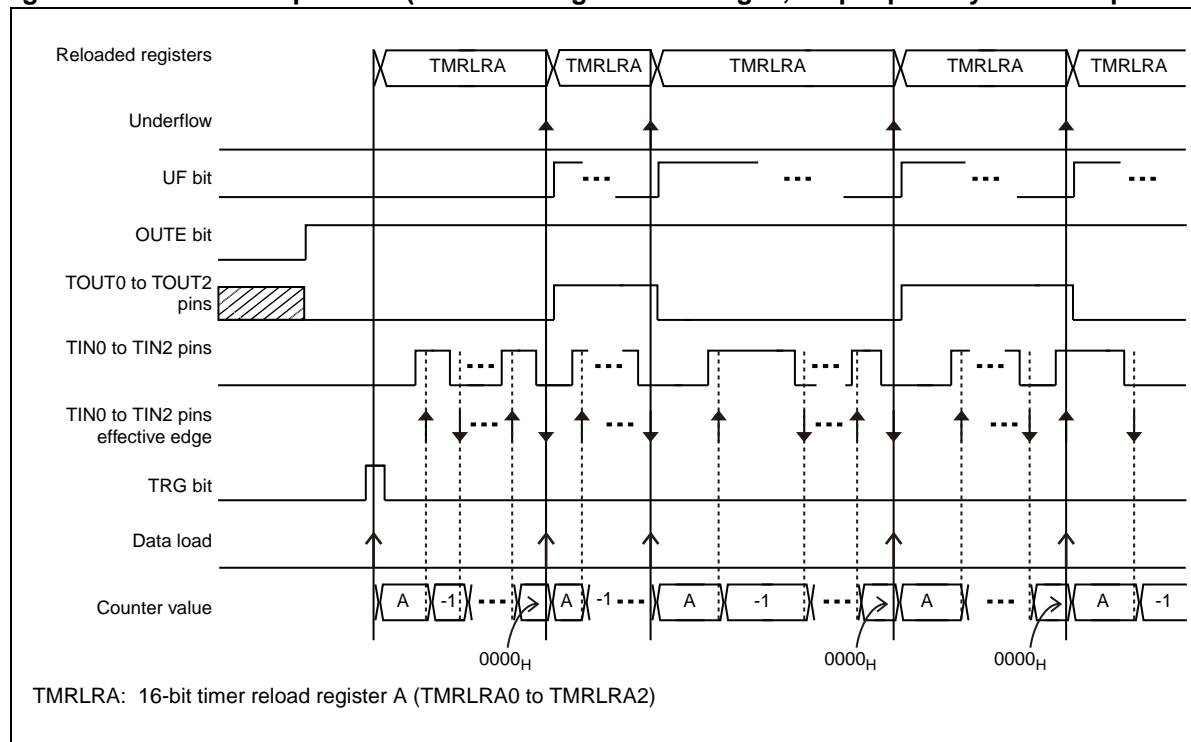
- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TOUT0 to TOUT2 pins is inverted.
- The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.
- The counter continues counting when an effective level signal is input from TIN0 to TIN2 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 TIN0 to TIN2 pins is detected.

Figure 20.6-17 shows the count operation.

Figure 20.6-17 Count operation (detection edge = both edges, output polarity = normal polarity)



● Operation of interrupt processing

The operation is the same as in interval timer mode. See "■ Operations in reload mode (TIN0 to TIN2 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 TOUT0 to TOUT2 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 TIN0 to TIN2 pin in such condition, counting starts.

<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.

■ 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:

- One of the TRGM1, TRGM0 bits = 01 to 11
- 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 TIN0 to TIN2 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 TOUT0 to TOUT2 pins is initialized.
- Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "FFFF_H").

Figure 20.6-18 and Figure 20.6-19 show the count operations.

Figure 20.6-18 Count operation (detection edge = both edges)

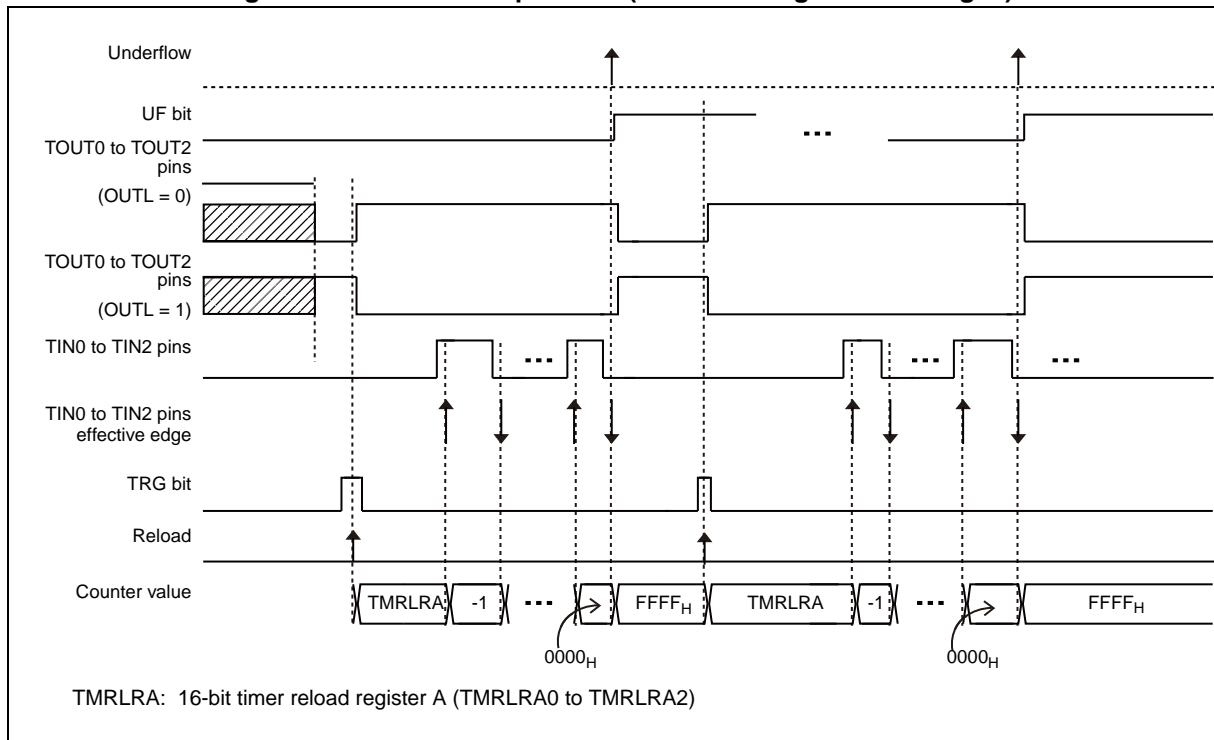
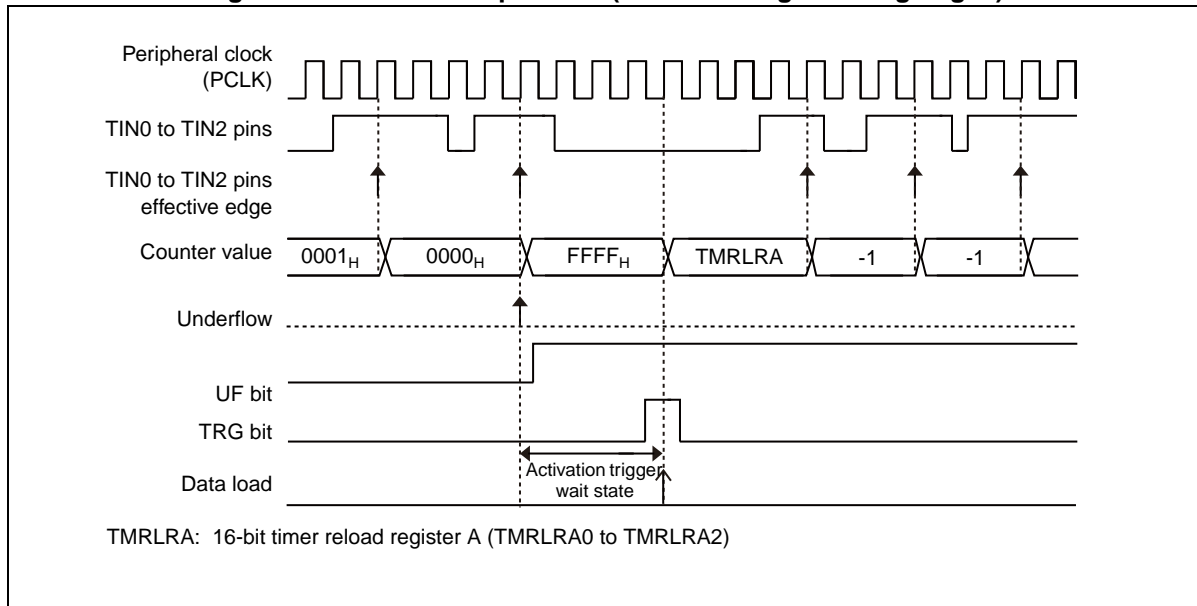


Figure 20.6-19 Count operation (detection edge = rising edges)



● 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.

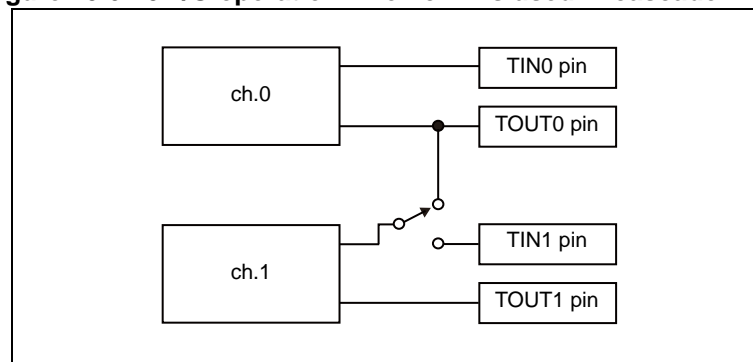
■ Operation

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 20.6-20 shows the I/O operation when ch.1 is used in cascade mode.

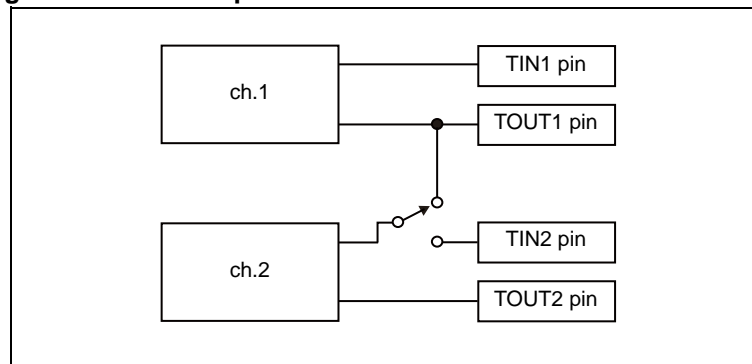
Figure 20.6-20 I/O operation when ch.1 is used in cascade mode



- When ch.2 is connected in cascade mode

It counts the outputs from ch.1. Figure 20.6-21 shows the I/O operation when ch.2 is used in cascade mode.

Figure 20.6-21 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

$$T \times (\text{TMRLRA0 value} + 1) \times (\text{TMRLRA1 value} + 1)$$

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

$$T \times (\text{TMRLRA1 value} + 1) \times (\text{TMRLRA2 value} + 1)$$

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

CHAPTER 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.

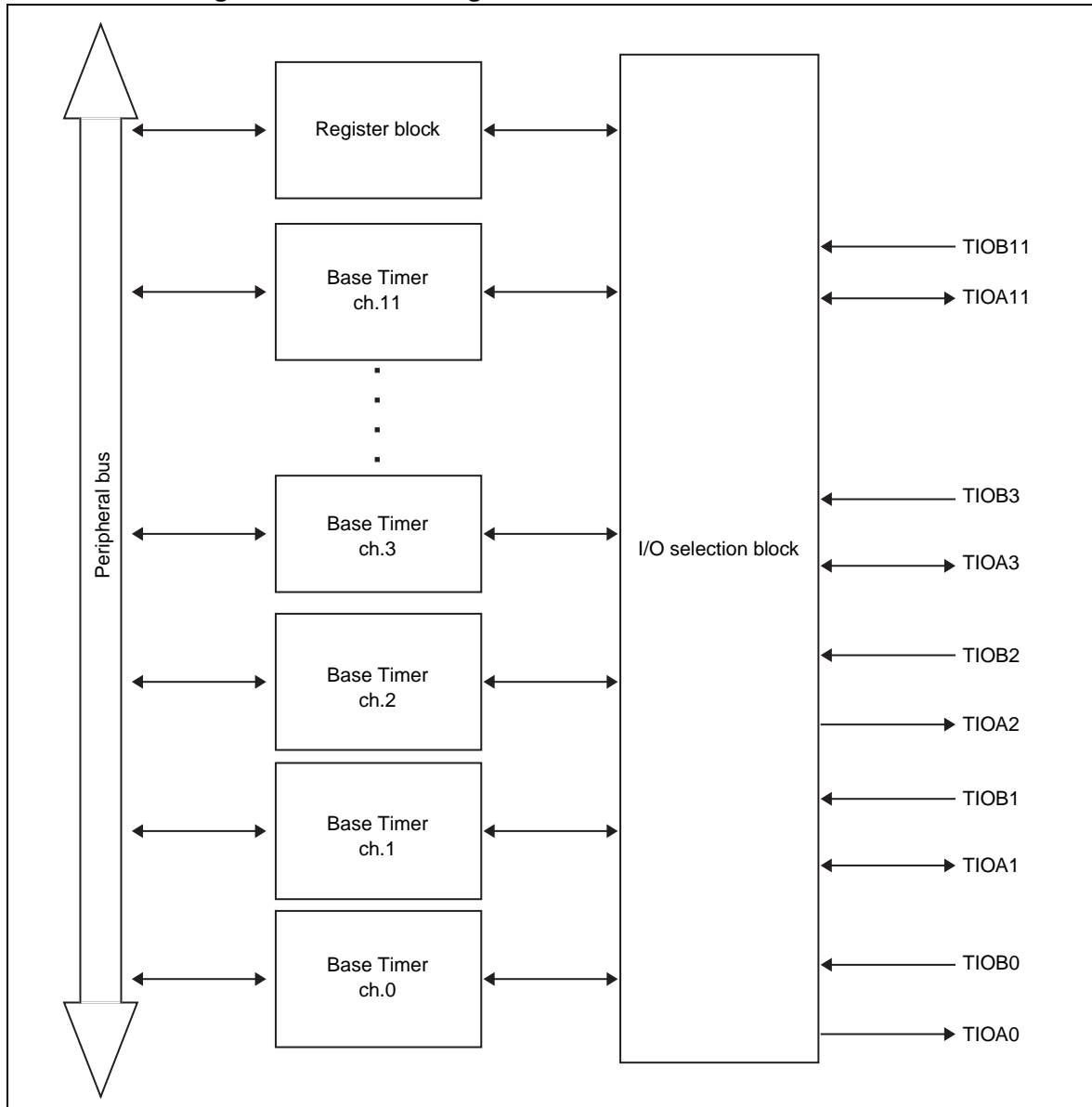
21.2 Configuration

The base timer I/O select function consists of the following blocks:

■ Block diagram of the base timer I/O select function

Figure 21.2-1 is a block diagram of the base timer I/O select function.

Figure 21.2-1 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.11)
ch.0 to ch.11 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 7 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 TIOA11 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.

- TIOB0 to TIOB11 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.

● 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 21.3-1 outlines the relationship between the I/O mode and pin connections.

Table 21.3-1 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, 8, 10

*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 21.4-1 lists registers of the base timer I/O select function.

Table 21.4-1 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	21.4.4
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
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	21.4.3

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 21.4-1 shows the bit configuration of the base timer io select register for ch.0/1/2/3 (BTSEL0123).

Figure 21.4-1 Bit configuration of base timer io 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 21.4-2 shows the bit configuration of the base timer io select register for ch.4/5/6/7 (BTSEL4567).

Figure 21.4-2 Bit configuration of base timer io 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 IO Select Register for Ch.8/9/A/B (BTSEL89AB)

This register sets the I/O mode of ch.8 to ch.11 of the base timer.

Figure 21.4-3 shows the bit configuration of the base timer io select register for ch.8/9/A/B (BTSEL89AB).

Figure 21.4-3 Bit configuration of base timer io select register for ch.8/9/A/B (BTSEL89AB)

bit	7	6	5	4	3	2	1	0
	SELAB_3	SELAB_2	SELAB_1	SELAB_0	SEL89_3	SEL89_2	SEL89_1	SEL89_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]: SELAB_3 to SELAB_0 (I/O select bit for ch.10/ch.11)

These bits set the I/O mode of ch.10 and ch.11 of the base timer.

SELAB_3	SELAB_2	SELAB_1	SELAB_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]: SEL89_3 to SEL89_0 (I/O select bit for ch.8/ch.9)

These bits set the I/O mode of ch.8 and ch.9 of the base timer.

SEL89_3	SEL89_2	SEL89_1	SEL89_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.

21.4.4 Base Timer Same Time Soft Start Register (BTSSSR)

This register simultaneously activates the base timers using the software.
Up to 12 channels corresponding to the bits in which "1" is written can be simultaneously activated.

Figure 21.4-4 shows the bit configuration of the base timer same time soft start register (BTSSSR).

Figure 21.4-4 Bit configuration of base timer same time soft start register (BTSSSR)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	Reserved	Reserved	SSSR11	SSSR10	SSSR9	SSSR8
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 bit12]: Reserved bits

Written Value	Explanation
0	Ignored
1	Writing prohibited

[bit11]: SSSR11 (Same time software start bit for ch.11)

This bit activates the ch.11 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.11 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SELAB_3 to SELAB_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB) (SELAB_3 to SELAB_0 = 0101)

[bit10]: SSSR10 (Same time software start bit for ch.10)

This bit activates the ch.10 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.10 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SELAB_3 to SELAB_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB)

- "5" (Same time software activation mode) (SELAB_3 to SELAB_0 = 0101)
- "6" (Software activation timer activation/stop mode) (SELAB_3 to SELAB_0 = 0110)

[bit9]: SSSR9 (Same time software start bit for ch.9)

This bit activates the ch.9 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.9 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SEL89_3 to SEL89_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB) (SEL89_3 to SEL89_0 = 0101)

[bit8]: SSSR8 (Same time software start bit for ch.8)

This bit activates the ch.8 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.8 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SEL89_3 to SEL89_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB)

- "5" (Same time software activation mode) (SEL89_3 to SEL89_0 = 0101)
- "6" (Software activation timer activation/stop mode) (SEL89_3 to SEL89_0 = 0110)

[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.*

* 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 io 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.*

* 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 io 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.*

* 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 io 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.*

* 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 io 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.*

* 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 io 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.*

* 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 io 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.*

* 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 io 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.*

* 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 io select register for ch.0/1/2/3 (BTSEL0123)

- "5" (Same time software activation mode) SEL01_3 to SEL01_0=0101)
- "6" (Software activation timer activation/stop mode) (SEL01_3 to SEL01_0=0110)

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 io select register (BTSEL0123 to BTSEL89AB).

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 21.5-1 lists the external pins used when this mode is set.

Table 21.5-1 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	1
Output pin	1	1

Table 21.5-2 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-2 Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOA0 to TIOA11	Output	TOUT	Output wave form the base timer
TIOB0 to TIOB11	Input	ECK/TGIN/TIN*	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)

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21.5-1 is a block diagram of I/O mode 0 (16-bit timer standard mode).

Figure 21.5-1 Block Diagram of I/O Mode 0 (16-bit Timer Standard Mode)

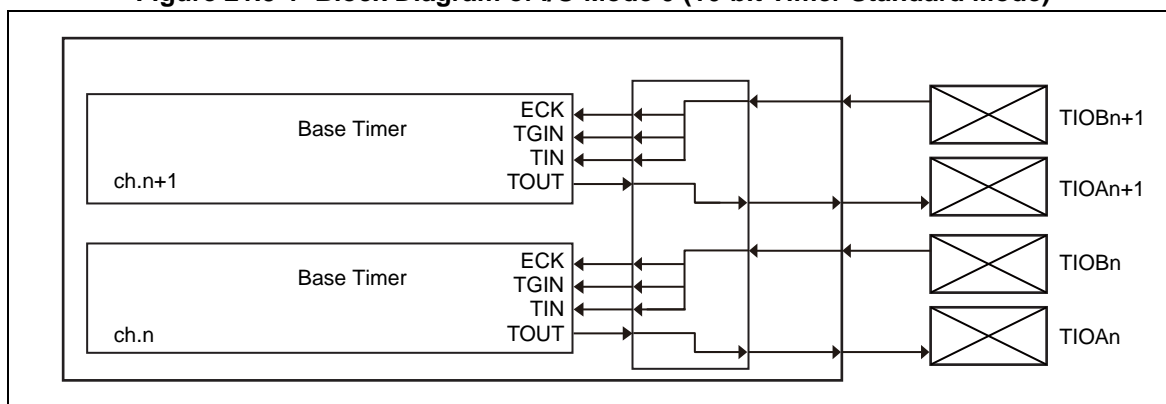


Table 21.5-3 lists the connections for I/O mode 0.

Table 21.5-3 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, 8, 10

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 21.5-4 lists the external pins used when this mode is set.

Table 21.5-4 External Pins Used

Input pin	3
Output pin	1

Table 21.5-5 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-5 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, 8, 10

Figure 21.5-2 is a block diagram of I/O mode 1 (timer full mode).

Figure 21.5-2 Example of Block Diagram of I/O Mode 1 (Timer Full Mode)

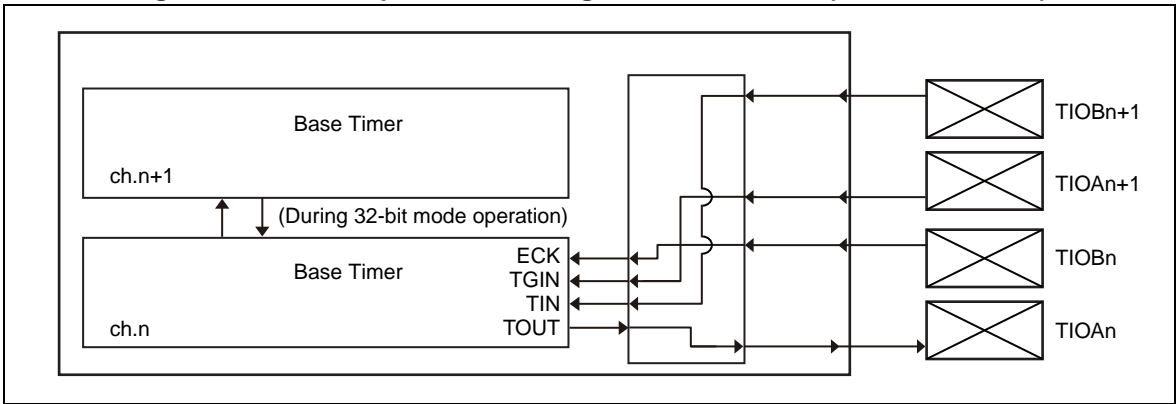


Table 21.5-6 lists the connections for I/O mode 1.

Table 21.5-6 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 TIN signal
TOUT signal of ch.n+1	Input to ch.n as a TGIN signal
TIOBn+1 pin	Input to ch.n as an ECK signal

n=0, 2, 4, 6, 8, 10

<Note>

If this mode is set, set the TIOAn pins (TIOA1, TIOA3, TIOA5, ... TIOA11) corresponding to the odd-numbered channel to the port input mode in the port function register (PFR).

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 21.5-7 lists the external pins used when this mode is set.

Table 21.5-7 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1 (shared by 2 channels)	
Output pin	1	1

Table 21.5-8 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-8 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*	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, 8, 10

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21.5-3 is a block diagram of I/O mode 2 (external trigger shared mode).

Figure 21.5-3 Block Diagram of I/O Mode 2 (External Trigger Shared Mode)

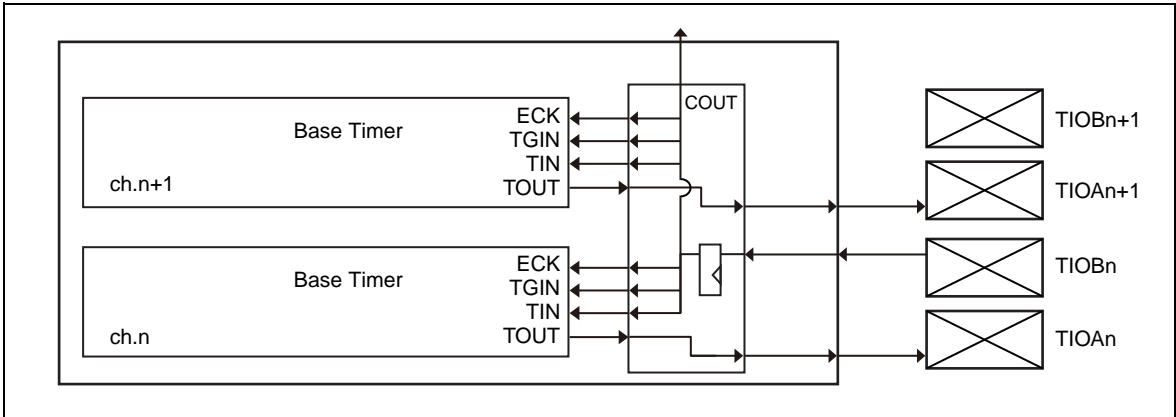


Table 21.5-9 lists the connections for I/O mode 2.

Table 21.5-9 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	<ul style="list-style-type: none">- 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, 8, 10

<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 21.5-10 lists the external pins used when this mode is set.

Table 21.5-10 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21.5-11 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-11 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, 8, 10

Figure 21.5-4 is a block diagram of I/O mode 3 (other channel trigger shared mode).

Figure 21.5-4 Block Diagram of I/O Mode 3 (Other Channel Trigger Shared Mode)

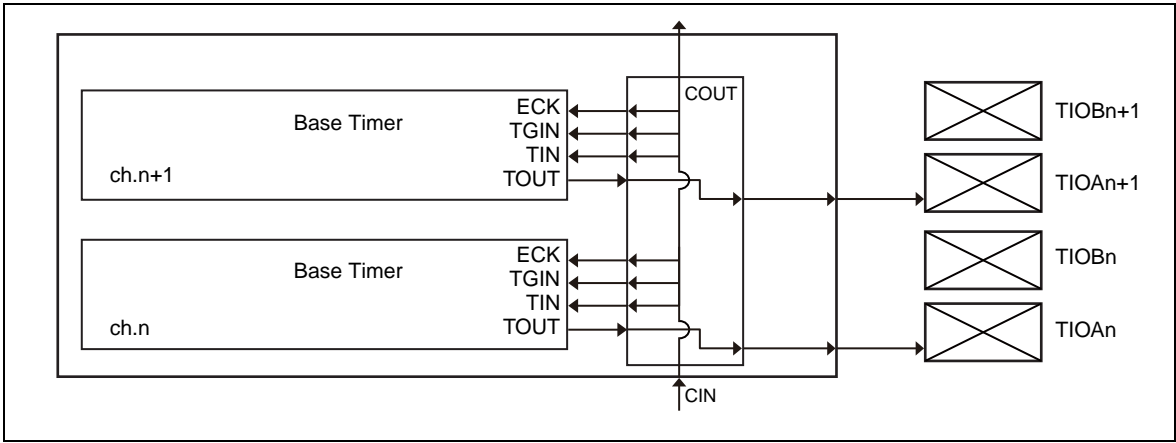


Table 21.5-12 lists the connections for I/O mode 3.

Table 21.5-12 Connections for I/O Mode 3

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
CIN signal*	- 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, 8, 10

* 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 21.5-13 lists the external pins used when this mode is set.

Table 21.5-13 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	Not used
Output pin	1	1

Table 21.5-14 lists the functions of pins.

Table 21.5-14 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*	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, 8, 10

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21.5-5 is a block diagram of I/O mode 4 (timer activation/stop mode).

Figure 21.5-5 Block Diagram of I/O Mode 4 (Timer Activation/Stop Mode)

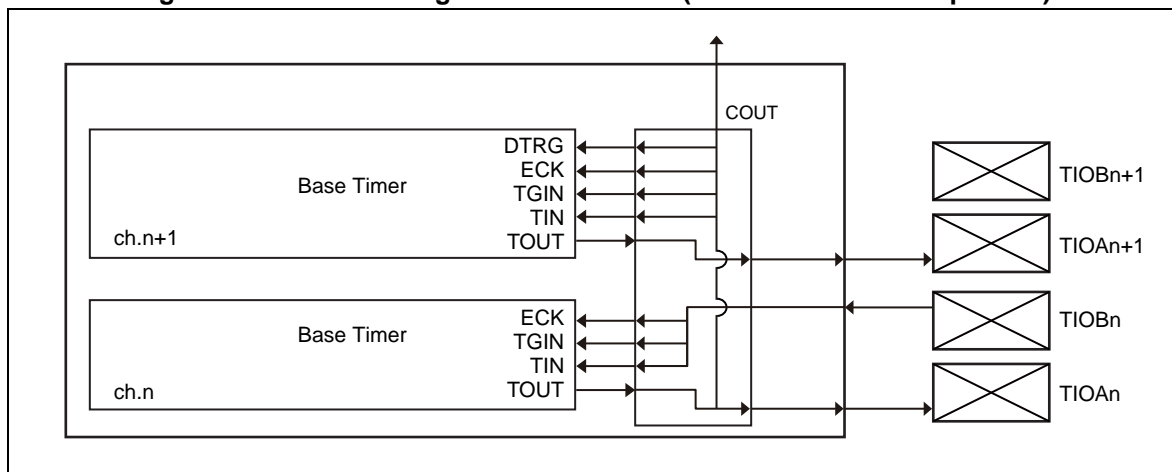


Table 21.5-15 lists the connections for I/O mode 4.

Table 21.5-15 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, 8, 10

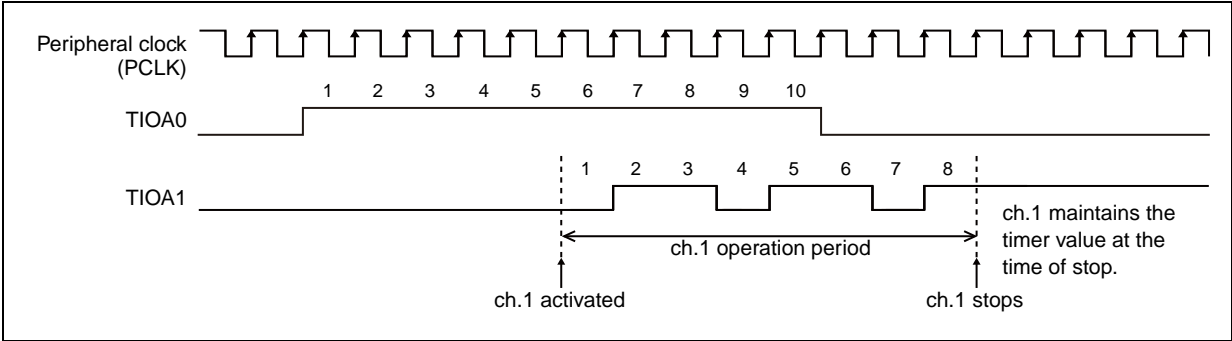
<Notes>

- 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 21.5-6 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 21.5-6 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 21.5-16 lists the external pins used when this mode is set.

Table 21.5-16 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21.5-17 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-17 Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=0, 2, 4, 6, 8, 10

Figure 21.5-7 is a block diagram of I/O mode 5 (same time software activation mode).

Figure 21.5-7 Block Diagram of I/O Mode 5 (Same Time Software Activation Mode)

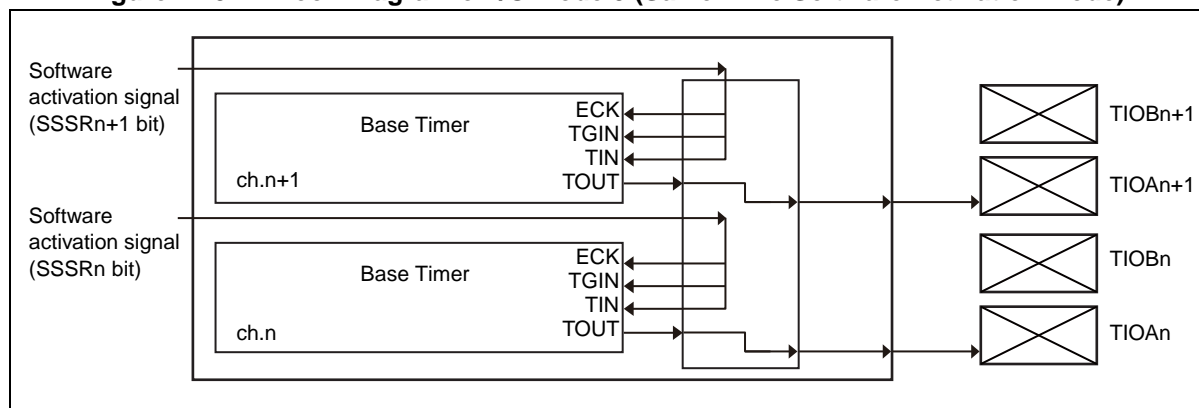


Table 21.5-18 lists the connections for I/O mode 5.

Table 21.5-18 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, 8, 10

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 21.5-19 lists the external pins used when this mode is set.

Table 21.5-19 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21.5-20 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-20 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, 8, 10

Figure 21.5-8 is a block diagram of I/O mode 6 (software activation timer activation/stop mode).

Figure 21.5-8 Block Diagram of I/O Mode 6 (Software Activation Timer Activation/Stop Mode)

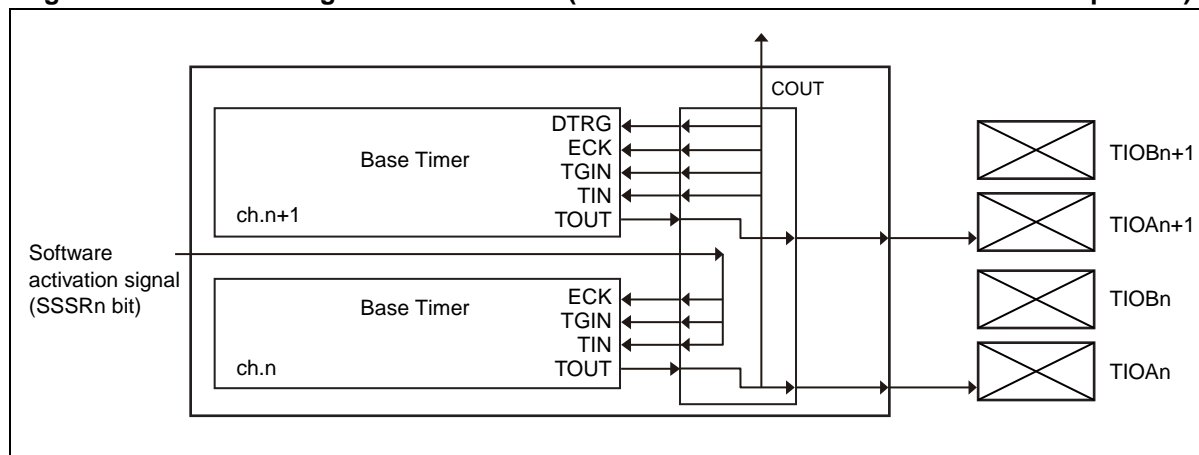


Table 21.5-21 lists the connections for I/O mode 6.

Table 21.5-21 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, 8, 10

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 21.5-22 lists the external pins used when this mode is set.

Table 21.5-22 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	Not used
Output pin	1	1

Table 21.5-23 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-23 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*	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, 8, 10

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 21.5-9 is a block diagram of I/O mode 7 (timer activation mode).

Figure 21.5-9 Block Diagram of I/O Mode 7 (Timer Activation Mode)

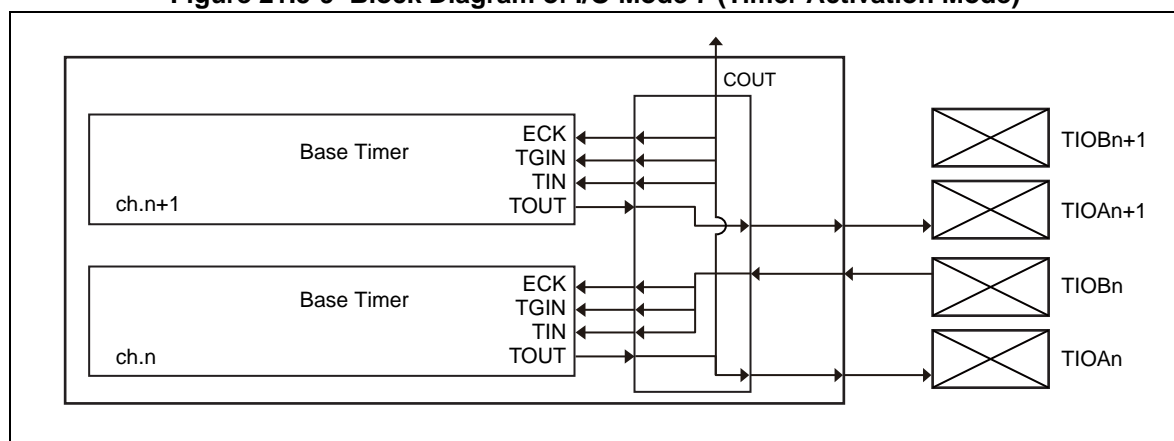


Table 21.5-24 lists the connection for I/O mode 7.

Table 21.5-24 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/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, 8, 10

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 21.5-25 lists the external pins used when this mode is set.

Table 21.5-25 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 21.5-26 lists the connection destinations of the external pins used and I/O signals.

Table 21.5-26 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, 8, 10

Figure 21.5-10 is a block diagram of I/O mode 8 (other channel trigger shared timer activation/stop mode).

Figure 21.5-10 Block Diagram of I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode)

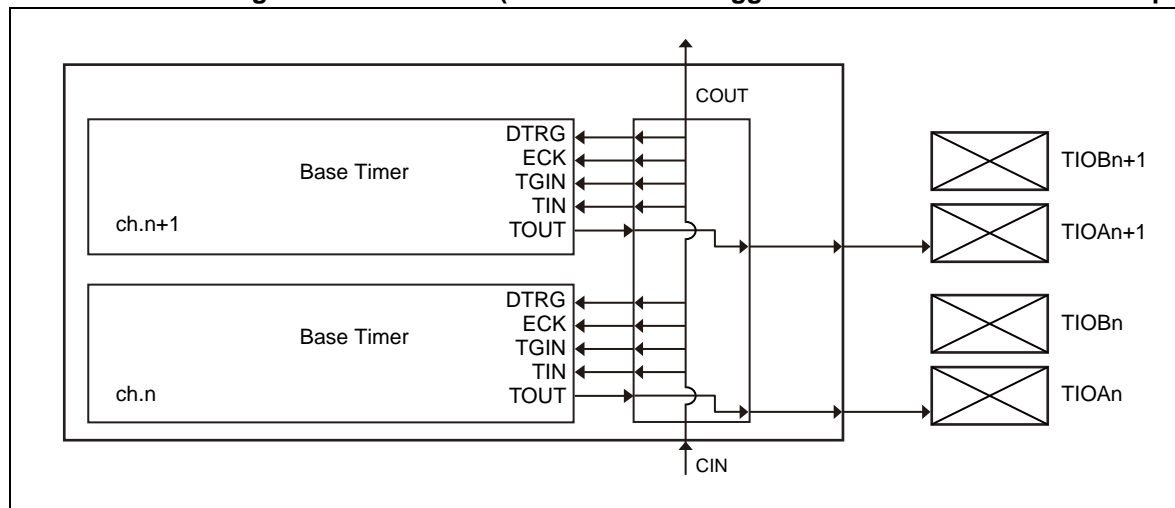


Table 21.5-27 lists the connections for I/O mode 8.

Table 21.5-27 Connections for I/O Mode 8

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
CIN signal*	- 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, 8, 10

* 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.
-

CHAPTER 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 12 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 (\uparrow to \downarrow) / "L" pulse width (\downarrow to \uparrow)
 Rising period (\uparrow to \uparrow) / Falling period (\downarrow to \downarrow)
 Inter-edge measurement (\uparrow or \downarrow to \downarrow or \uparrow)

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 22.2-1 Block Diagram of 16-bit PWM Timer

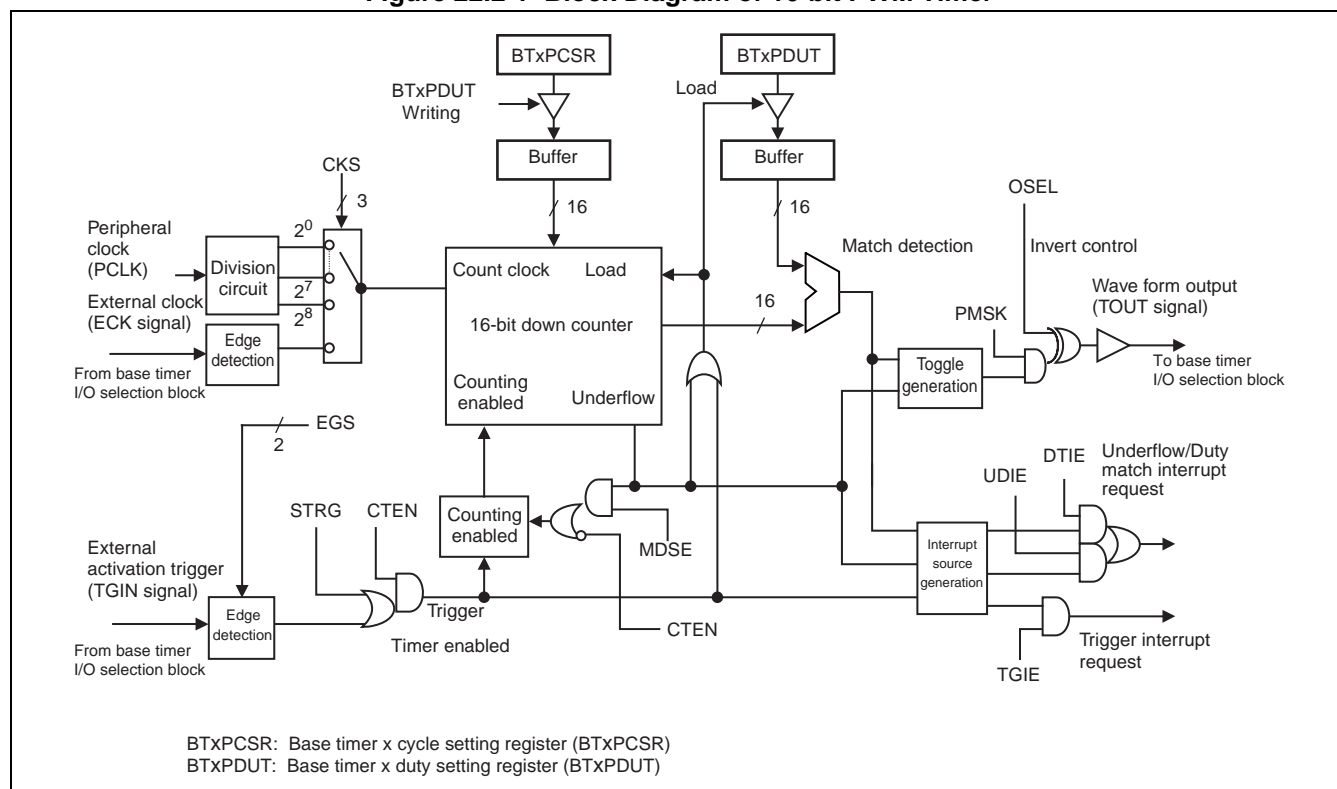


Figure 22.2-2 Block Diagram of 16-bit PPG Timer



■ Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)

Figure 22.2-3 Block Diagram of 16-bit Reload Timer (ch.1, ch.0)

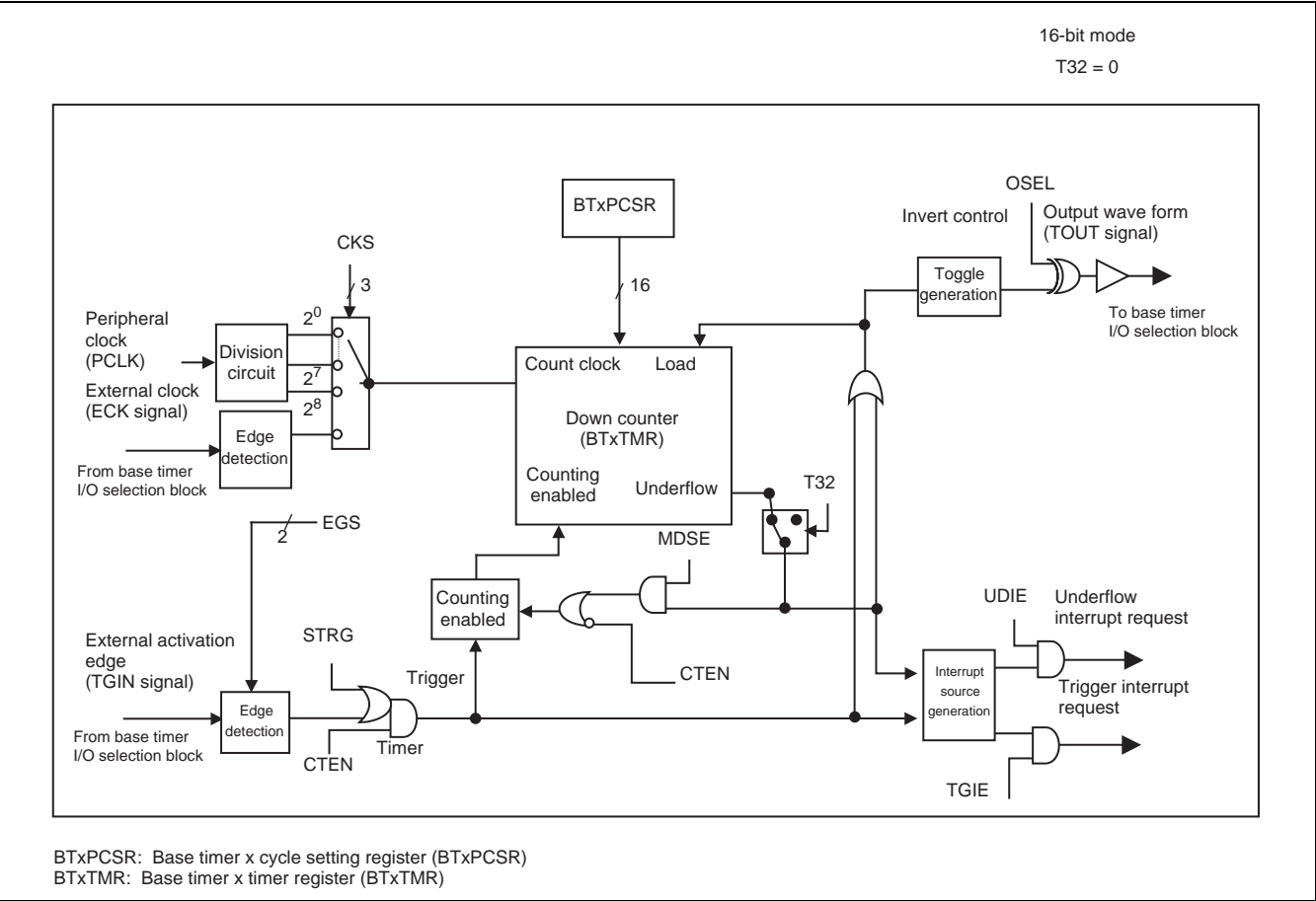
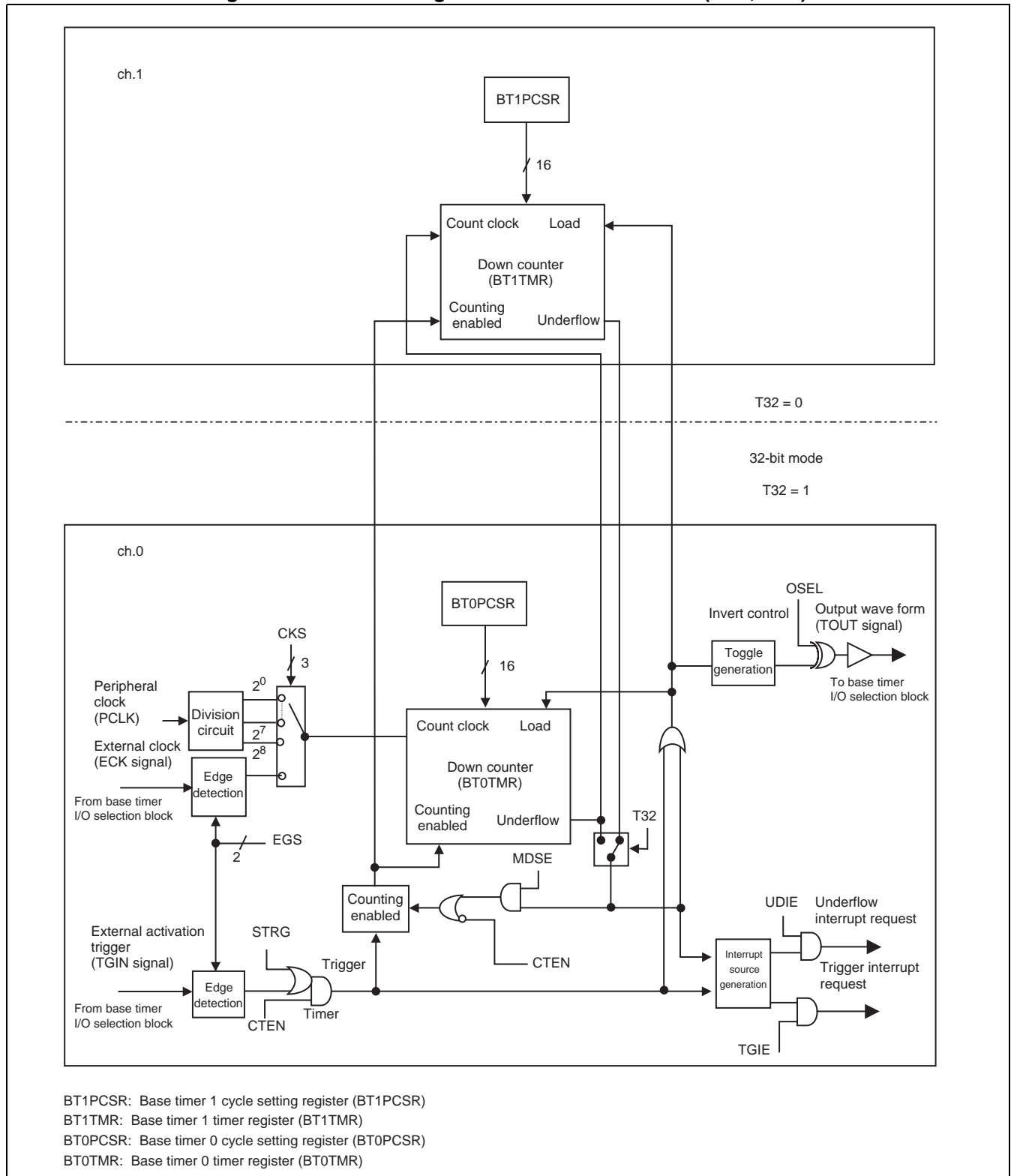


Figure 22.2-4 Block Diagram of 32-bit Reload Timer (ch.1, ch.0)



<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, between ch.6 and ch.7, between ch.8 and ch.9, and between ch.10 and ch.11. No 32-bit operation is applicable to any other combination of channels.
 - This function supports simultaneous activation. For details, see "CHAPTER 21 Base Timer I/O Select Function".
-

■ Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)

Figure 22.2-5 Block Diagram of 16-bit PWC (ch.1, ch.0)

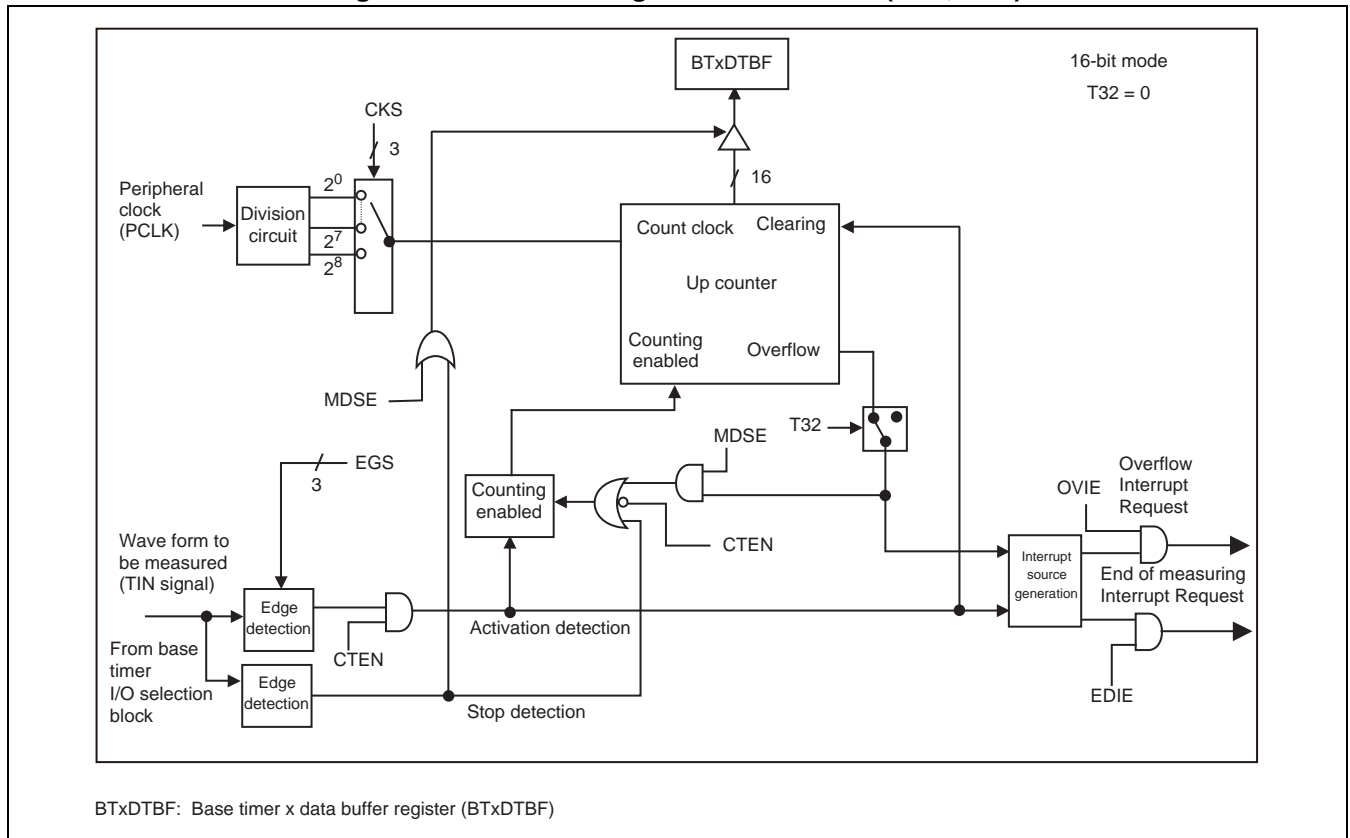
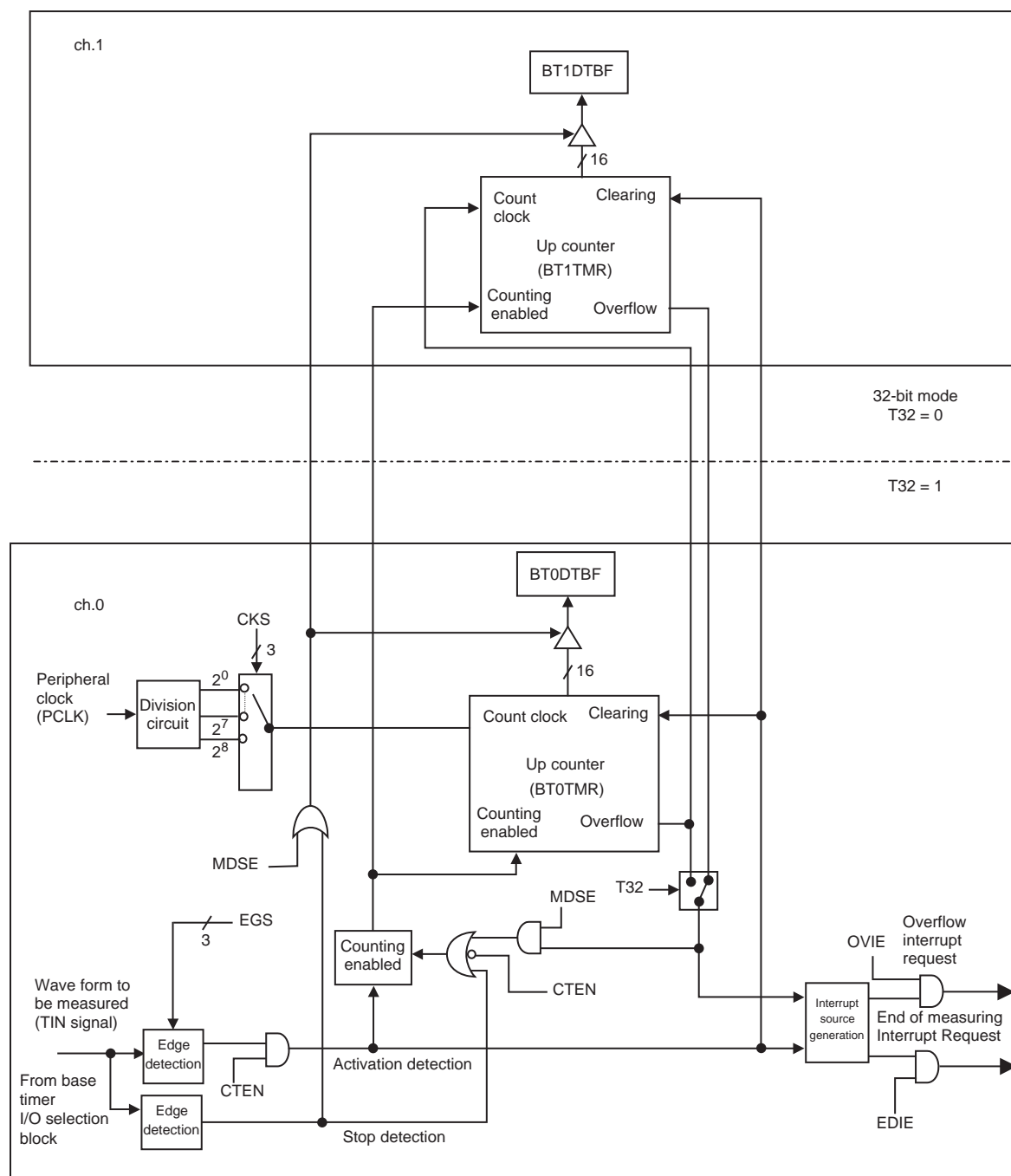


Figure 22.2-6 Block Diagram of 32-bit PWC (ch.1, ch.0)



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, between ch.6 and ch.7, between ch.8 and ch.9, and between ch.10 and ch.11. No 32-bit operation is applicable to any other combination of channels.
 - This function supports simultaneous activation. For details, see "CHAPTER 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 22.3-1 Registers used for 16-bit PWM timer (1 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.4
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
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	21.4.3
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

Table 22.3-1 Registers used for 16-bit PWM timer (2 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
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
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
8	BT8TMCR	Base timer 8 timer control register	22.8.1.1
	BT8STC	Base timer 8 status control register	22.8.1.1
	BT8PCSR	Base timer 8 cycle setting register	22.8.1.2
	BT8PDUT	Base timer 8 duty setting register	22.8.1.3
	BT8TMR	Base timer 8 timer register	22.8.1.4
9	BT9TMCR	Base timer 9 timer control register	22.8.1.1
	BT9STC	Base timer 9 status control register	22.8.1.1
	BT9PCSR	Base timer 9 cycle setting register	22.8.1.2
	BT9PDUT	Base timer 9 duty setting register	22.8.1.3
	BT9TMR	Base timer 9 timer register	22.8.1.4

Table 22.3-1 Registers used for 16-bit PWM timer (3 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
10	BTATMCR	Base timer 10 timer control register	22.8.1.1
	BTASTC	Base timer 10 status control register	22.8.1.1
	BTAPCSR	Base timer 10 cycle setting register	22.8.1.2
	BTAPDUT	Base timer 10 duty setting register	22.8.1.3
	BTATMR	Base timer 10 timer register	22.8.1.4
11	BTBTMCR	Base timer 11 timer control register	22.8.1.1
	BTBSTC	Base timer 11 status control register	22.8.1.1
	BTBPCSR	Base timer 11 cycle setting register	22.8.1.2
	BTBPDUT	Base timer 11 duty setting register	22.8.1.3
	BTBTMR	Base timer 11 timer register	22.8.1.4

Table 22.3-2 Registers for the 16-bit PPG timer (1 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.4
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
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	21.4.3
0	BT0TMCR	Base timer 0 timer control register	22.8.2.1
	BT0STC	Base timer 0 status control register	22.8.2.1
	BT0PRL	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
	BT1PRL	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

Table 22.3-2 Registers for the 16-bit PPG timer (2 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
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 22.3-2 Registers for the 16-bit PPG timer (3 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
8	BT8TMCR	Base timer 8 timer control register	22.8.2.1
	BT8STC	Base timer 8 status control register	22.8.2.1
	BT8PRLH	Base timer 8 L width setting register	22.8.2.2
	BT8PRLH	Base timer 8 H width setting register	22.8.2.3
	BT8TMR	Base timer 8 timer register	22.8.2.4
9	BT9TMCR	Base timer 9 timer control register	22.8.2.1
	BT9STC	Base timer 9 status control register	22.8.2.1
	BT9PRLH	Base timer 9 L width setting register	22.8.2.2
	BT9PRLH	Base timer 9 H width setting register	22.8.2.3
	BT9TMR	Base timer 9 timer register	22.8.2.4
10	BTATMCR	Base timer 10 timer control register	22.8.2.1
	BTASTC	Base timer 10 status control register	22.8.2.1
	BTAPRLH	Base timer 10 L width setting register	22.8.2.2
	BTAPRLH	Base timer 10 H width setting register	22.8.2.3
	BTATMR	Base timer 10 timer register	22.8.2.4
11	BTBTMCR	Base timer 11 timer control register	22.8.2.1
	BTBSTC	Base timer 11 status control register	22.8.2.1
	BTBPRLH	Base timer 11 L width setting register	22.8.2.2
	BTBPRLH	Base timer 11 H width setting register	22.8.2.3
	BTBTMR	Base timer 11 timer register	22.8.2.4

Table 22.3-3 Registers for the 16/32-bit reload timer (1 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.4
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
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	21.4.3
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

Table 22.3-3 Registers for the 16/32-bit reload timer (2 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
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
8	BT8TMCR	Base timer 8 timer control register	22.8.3.1
	BT8STC	Base timer 8 status control register	22.8.3.1
	BT8PCSR	Base timer 8 cycle setting register	22.8.3.2
	BT8TMR	Base timer 8 timer register	22.8.3.3
9	BT9TMCR	Base timer 9 timer control register	22.8.3.1
	BT9STC	Base timer 9 status control register	22.8.3.1
	BT9PCSR	Base timer 9 cycle setting register	22.8.3.2
	BT9TMR	Base timer 9 timer register	22.8.3.3
10	BTATMCR	Base timer 10 timer control register	22.8.3.1
	BTASTC	Base timer 10 status control register	22.8.3.1
	BTAPCSR	Base timer 10 cycle setting register	22.8.3.2
	BTATMR	Base timer 10 timer register	22.8.3.3
11	BTBTMCR	Base timer 11 timer control register	22.8.3.1
	BTBSTC	Base timer 11 status control register	22.8.3.1
	BTBPCSR	Base timer 11 cycle setting register	22.8.3.2
	BTBTMR	Base timer 11 timer register	22.8.3.3

Table 22.3-4 List of registers used for 16/32-bit PWC timer (1 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	21.4.4
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
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	21.4.3
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

Table 22.3-4 List of registers used for 16/32-bit PWC timer (2 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
8	BT8TMCR	Base timer 8 timer control register	22.8.4.1
	BT8STC	Base timer 8 status control register	22.8.4.1
	BT8DTBF	Base timer 8 data buffer register	22.8.4.2
9	BT9TMCR	Base timer 9 timer control register	22.8.4.1
	BT9STC	Base timer 9 status control register	22.8.4.1
	BT9DTBF	Base timer 9 data buffer register	22.8.4.2
10	BTATMCR	Base timer 10 timer control register	22.8.4.1
	BTASTC	Base timer 10 status control register	22.8.4.1
	BTADTBF	Base timer 10 data buffer register	22.8.4.2
11	BTBTMCR	Base timer 11 timer control register	22.8.4.1
	BTBSTC	Base timer 11 status control register	22.8.4.1
	BTBDTBF	Base timer 11 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, between ch.6 and ch.7, between ch.8 and ch.9, and between ch.10 and ch.11. No 32-bit operation is applicable to any other combination of channels.
 - This function supports simultaneous activation. For details, see "CHAPTER 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, between ch.6 and ch.7, between ch.8 and ch.9, and between ch.10 and ch.11. No 32-bit operation is applicable to any other combination of channels.
 - This function supports simultaneous activation. For details, see "CHAPTER 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.

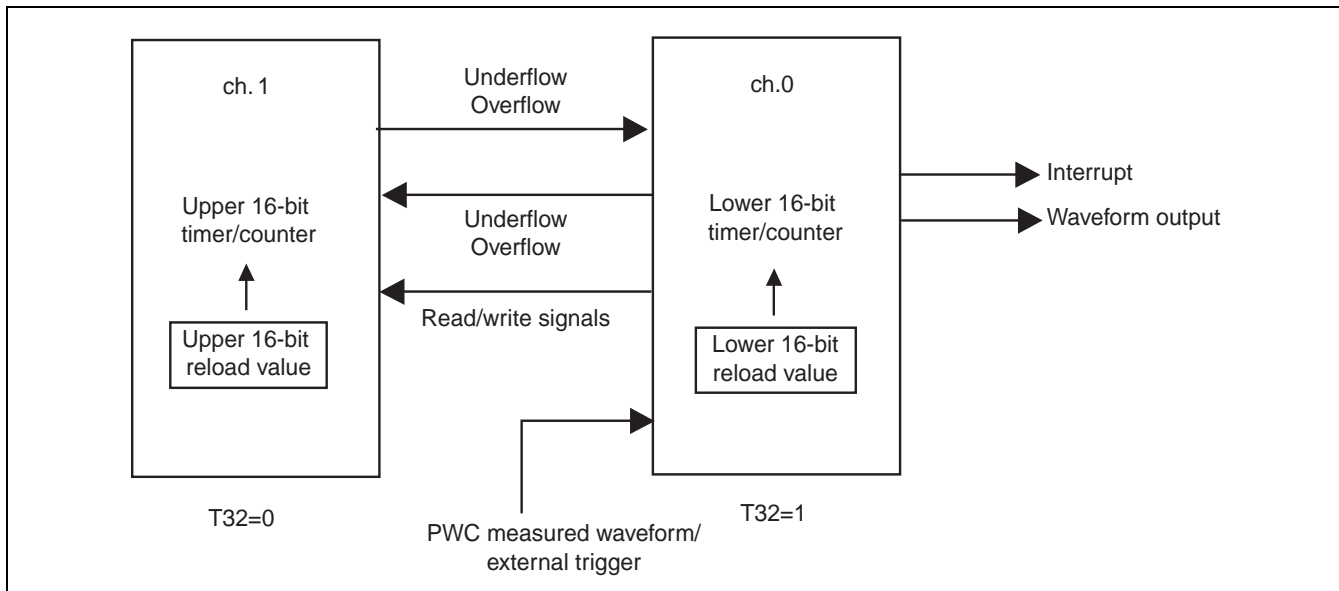
To switch from 32-bit mode to 16-bit mode, set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register for the even-numbered channel to "000_B" to reset the states of both of the even-numbered and odd-numbered channels in reset mode. Then set each channel for operation in 16-bit mode.

■ 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, between ch.6 and ch.7, between ch.8 and ch.9, and between ch.10 and ch.11. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "CHAPTER 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 BTxDTBFB 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 22.7-1 lists the interrupt control bits and interrupt factors for the base timer in each timer function mode.

Table 22.7-1 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 22.8-1 Timer Control Register (BTxTMCR Upper Byte)

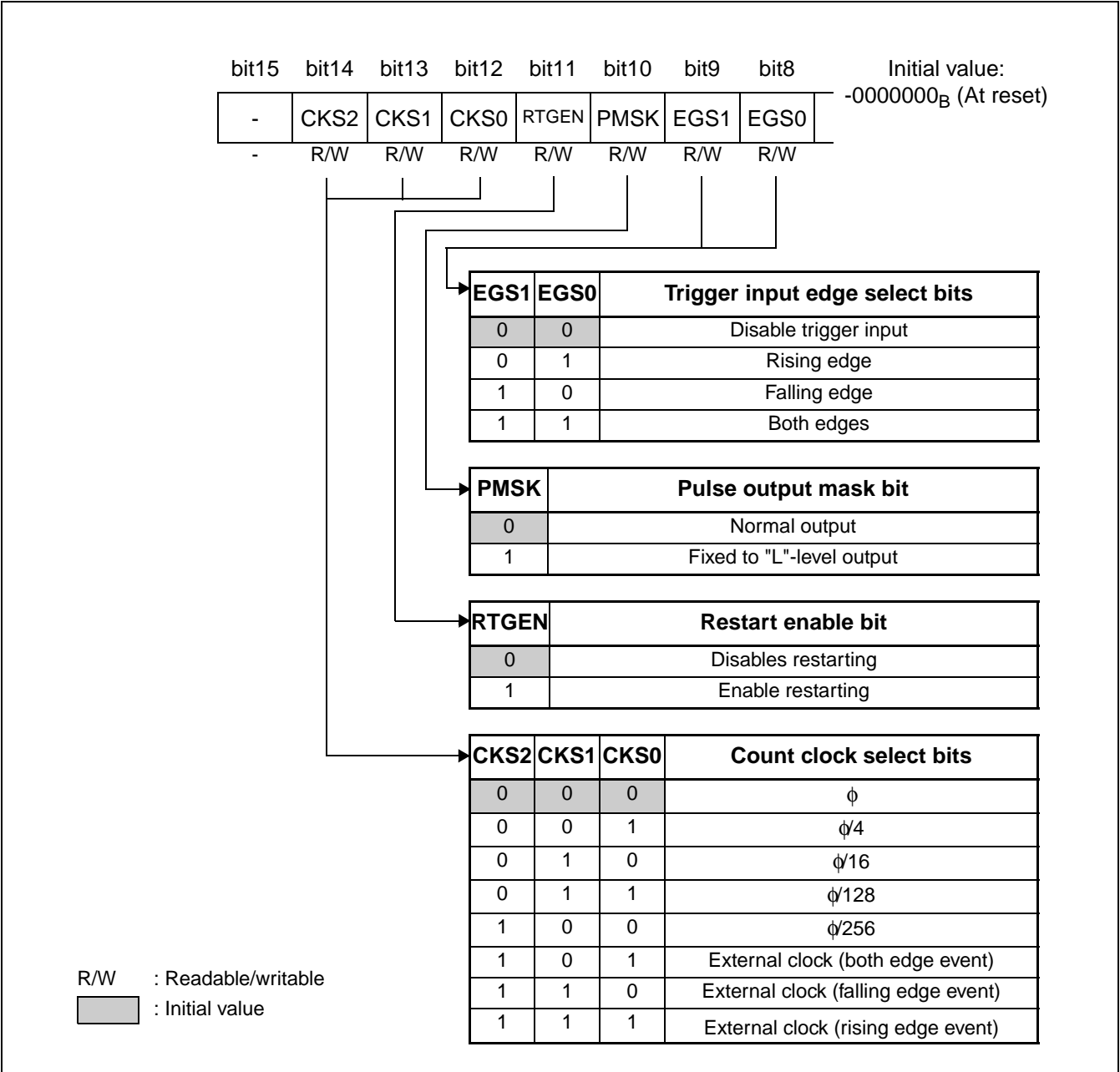


Table 22.8-1 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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. <p>Note: Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PWM output to the "H" level.</p>
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	<ul style="list-style-type: none"> 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. <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0.</p> <ul style="list-style-type: none"> 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 22.8-2 Timer Control Register (BTxTMCR Lower Byte)

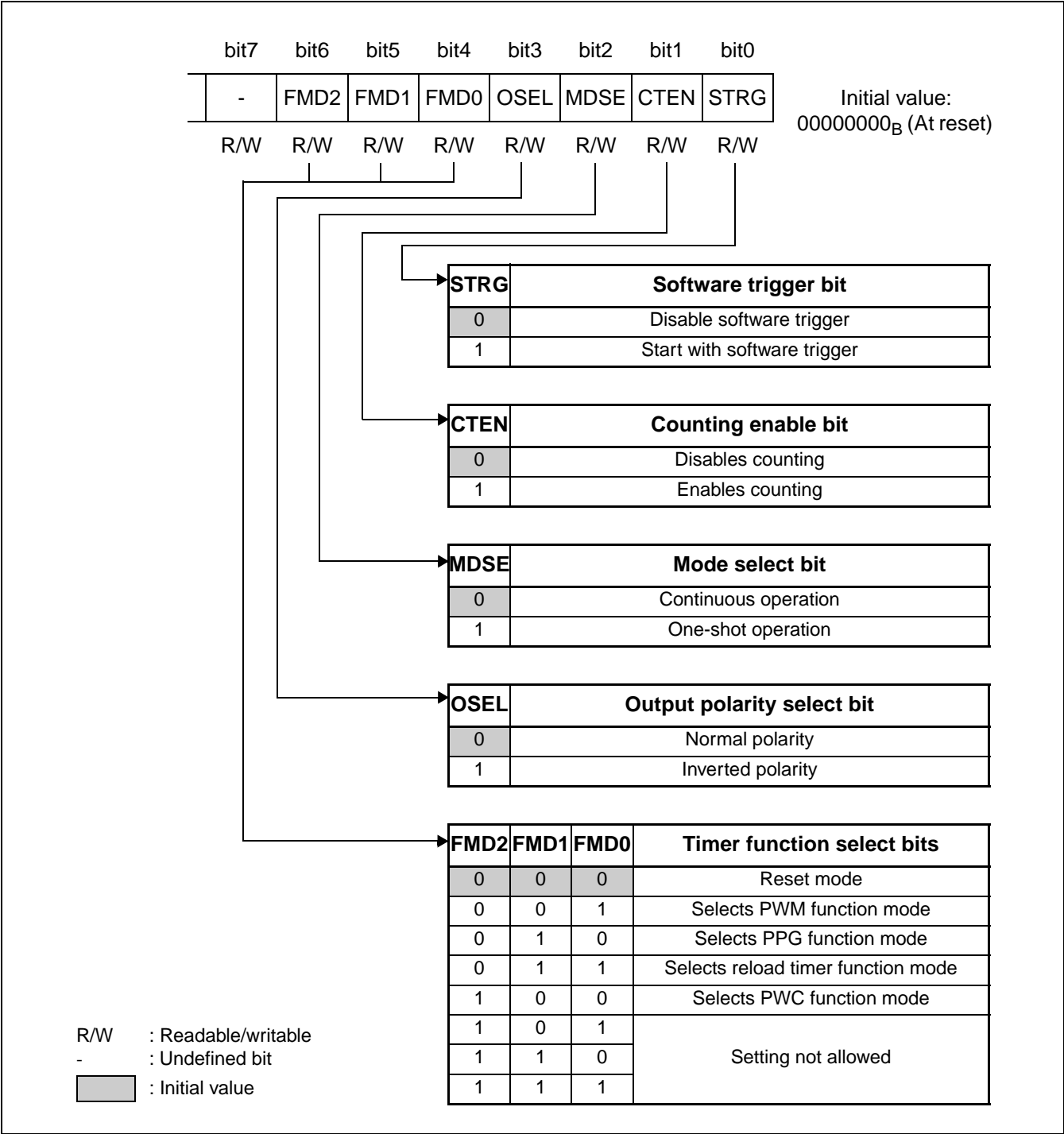














Table 22.8-2 Timer Control Register (BTxTMCR Lower Byte)

Bit name		Function												
bit7	Undefined bit	<ul style="list-style-type: none">The value read is "0"When writing to this bit, write "0".												
bit6 to bit4	FMD2, FMD1, FMD0: Timer function select bits	<ul style="list-style-type: none">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.												
bit3	OSEL: Output polarity select bit	<div>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></div>	Polarity	After reset	Duty match	Underflow	Normal	"L" output			Inverted	"H" output		
Polarity	After reset	Duty match	Underflow											
Normal	"L" output													
Inverted	"H" output													
bit2	MDSE: Mode select bit	<ul style="list-style-type: none">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.												
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none">This bit enables the down counter.Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.												
bit0	STRG: Software trigger bit	<ul style="list-style-type: none">Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger. <div>Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.</div> <ul style="list-style-type: none">The value read from the STRG bit is always "0". <div>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</div>												

■ Status Control Register (BTxSTC)

Figure 22.8-3 Status Control Register (BTxSTC)

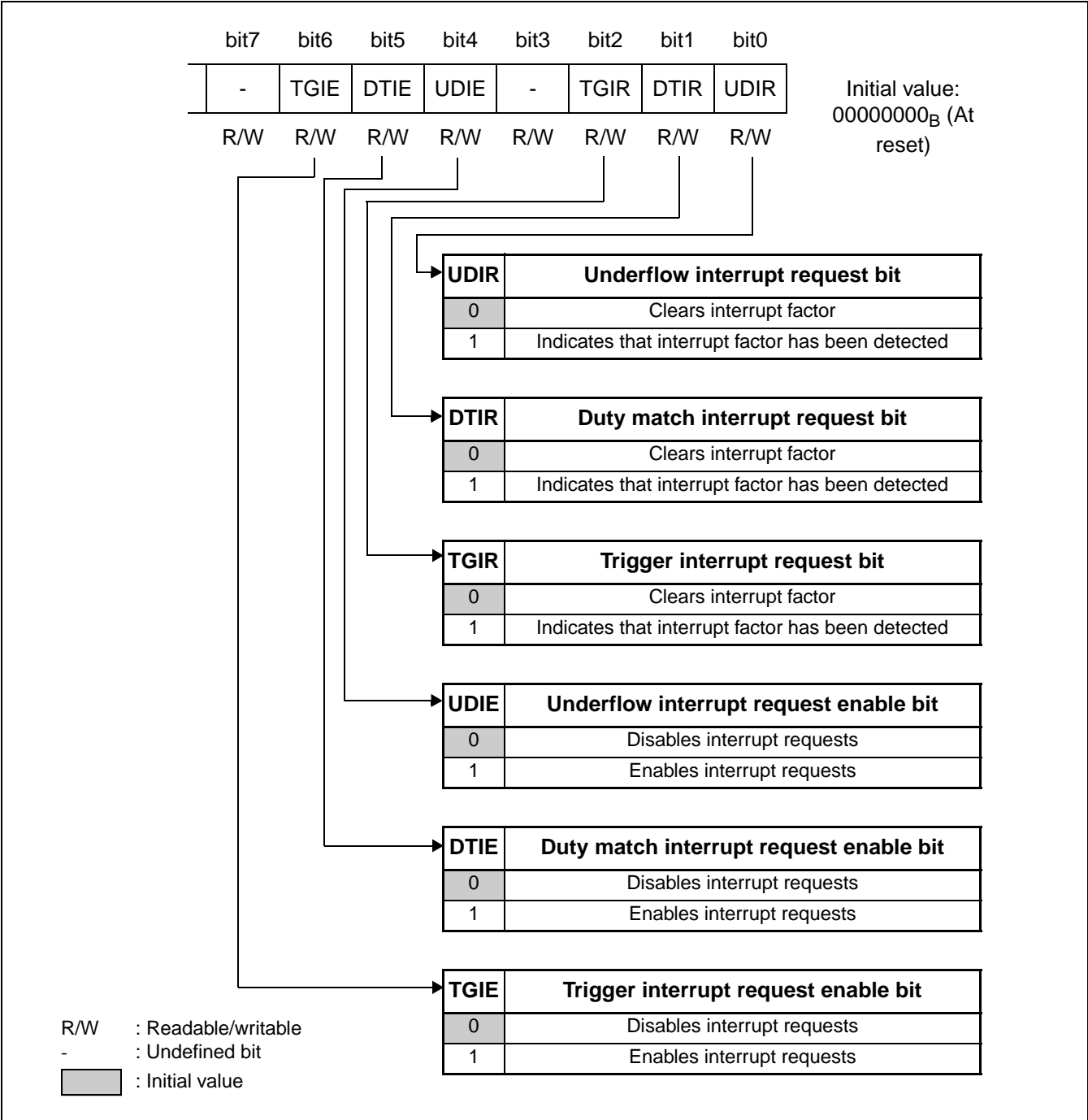


Table 22.8-3 Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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.

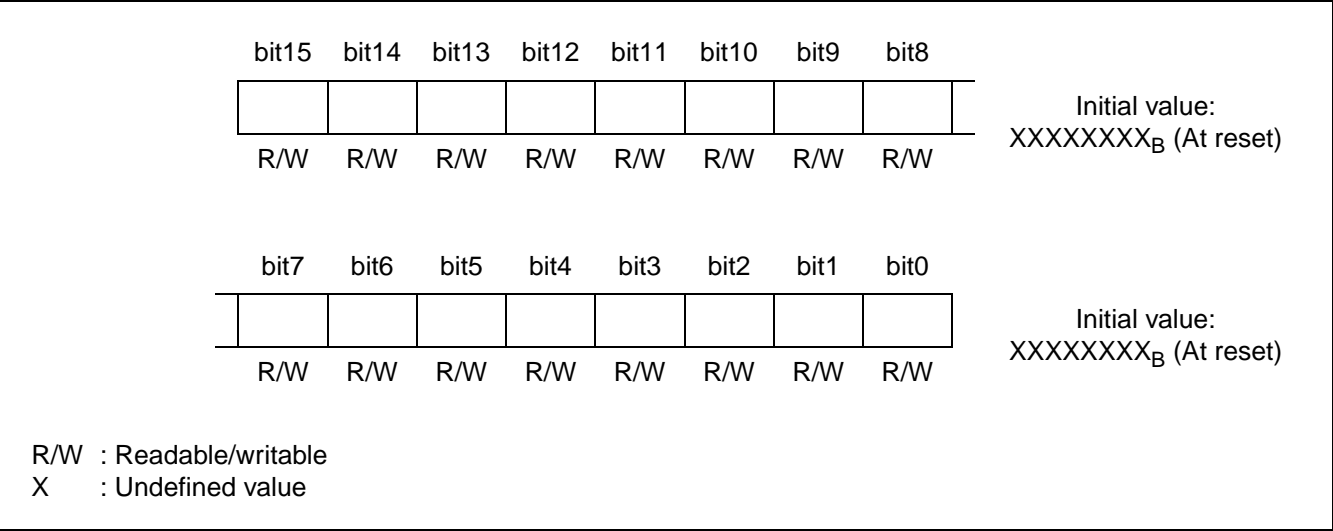
22.8.1.2 PWM Period Setting Register (BTxPCSR)

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 22.8-4 shows the bit configuration of the PWM period setting register (BTxPCSR).

Figure 22.8-4 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.

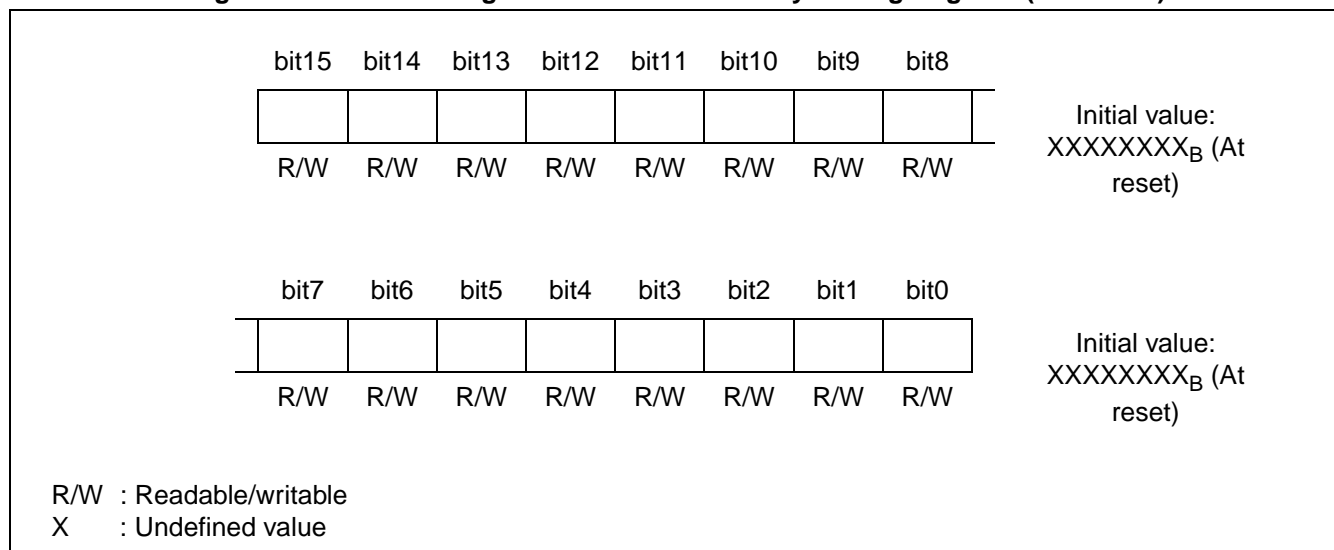
22.8.1.3 PWM Duty Setting Register (BTxPDUT)

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 22.8-5 shows the bit configuration of the PWM duty setting register (BTxPDUT).

Figure 22.8-5 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 PSCR 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.

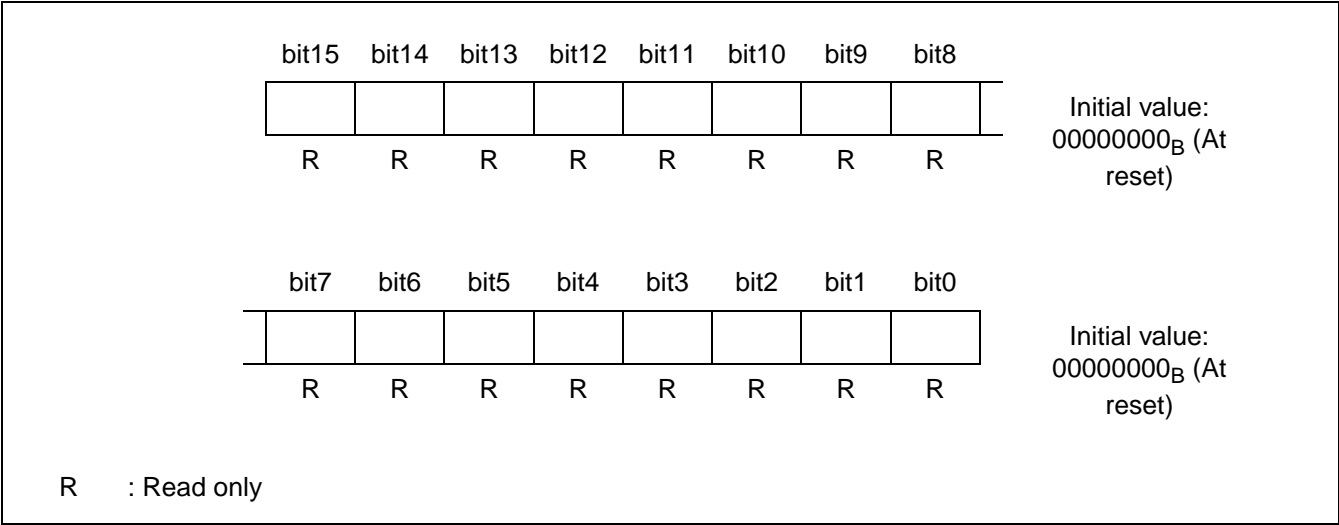
22.8.1.4 Timer Register (BTxTMR)

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 22.8-6 shows the bit configuration of the PWM timer register (BTxTMR).

Figure 22.8-6 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.

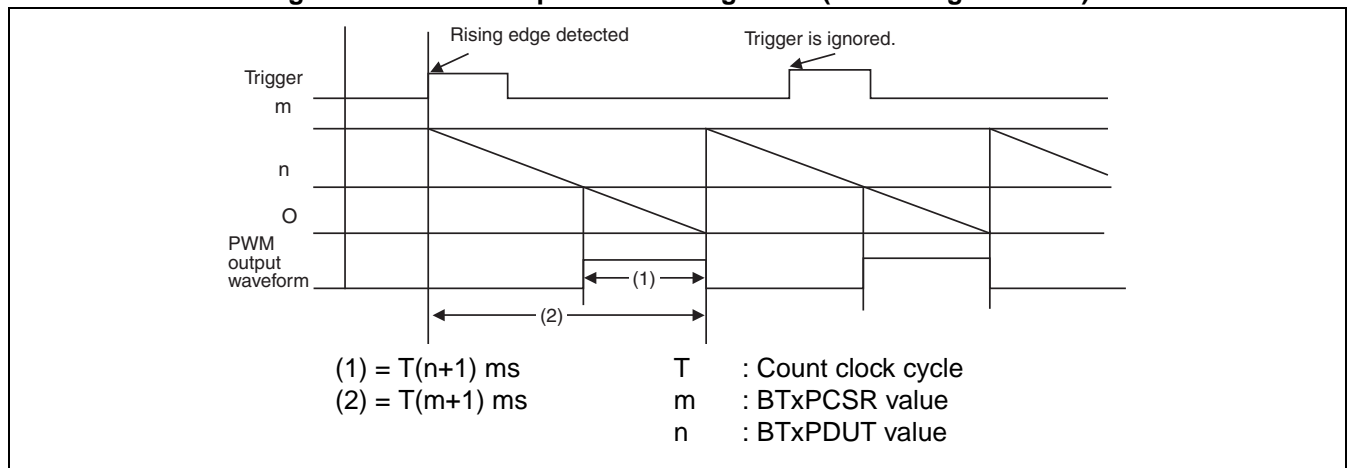
The period of output pulses can be controlled by changing the BTxPCSR value.

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

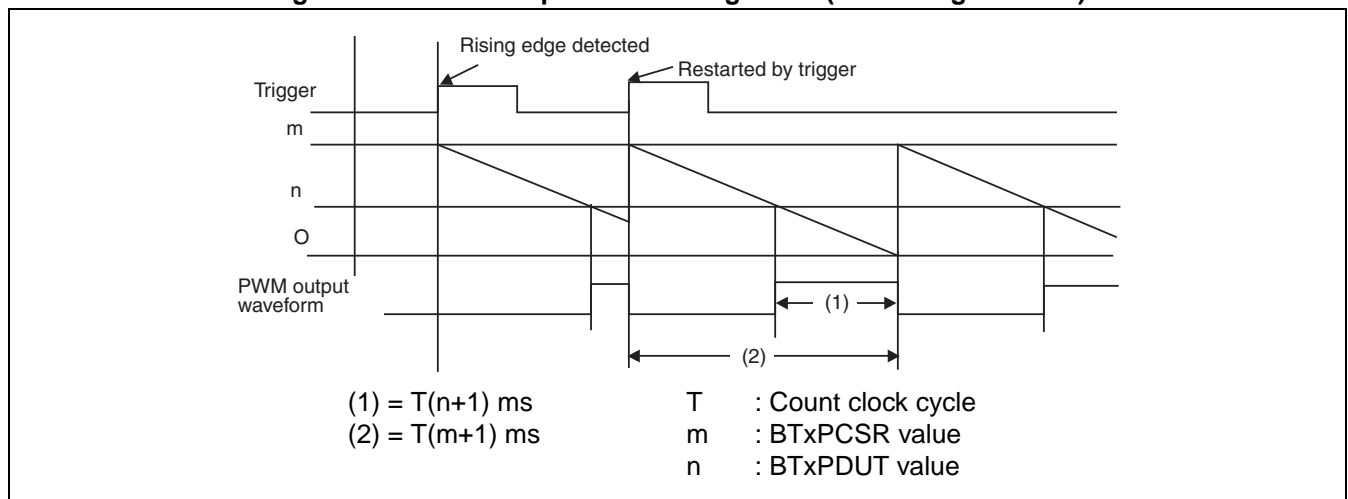
● When restarting is disabled (RTGEN = 0)

Figure 22.8-7 PWM Operation Timing Chart (Restarting Disabled)



● When restarting is enabled (RTGEN = 1)

Figure 22.8-8 PWM Operation Timing Chart (Restarting Enabled)



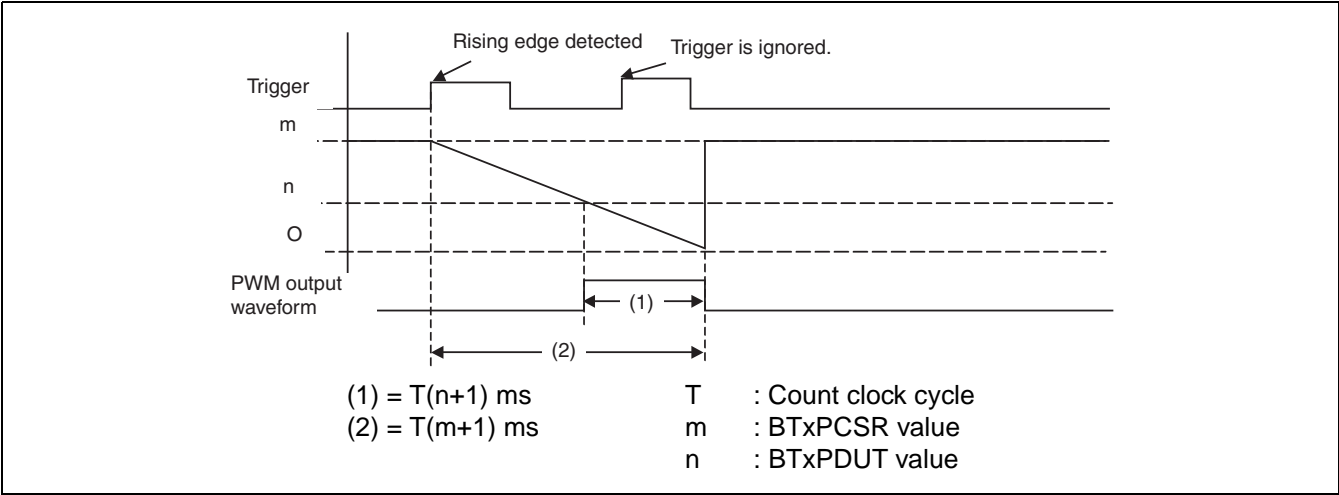
22.8.1.6 One-shot Operation

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

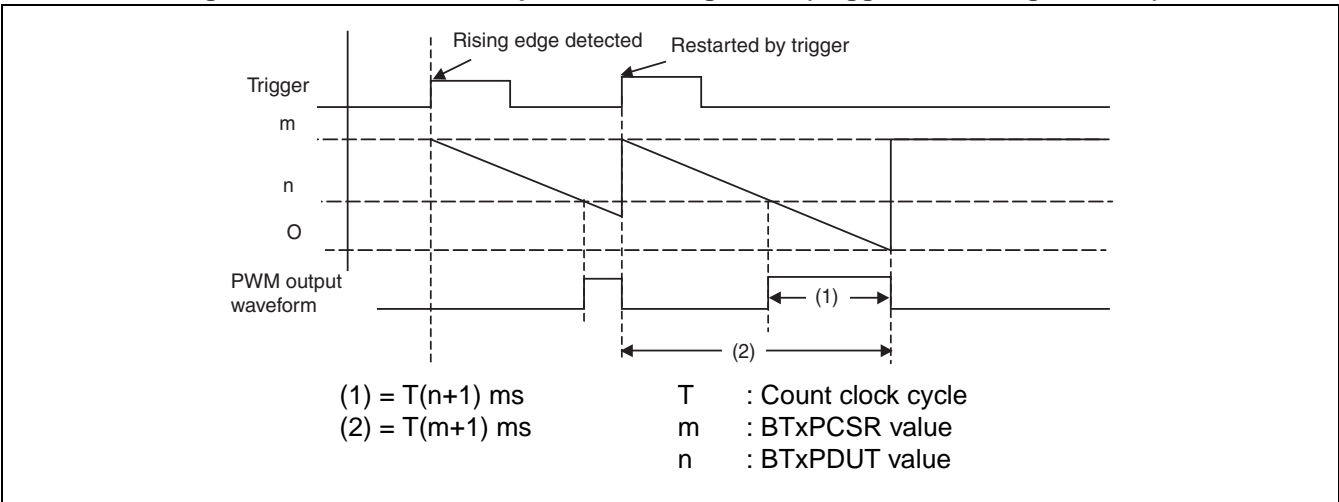
● When restarting is disabled (RTGEN = 0)

Figure 22.8-9 One-shot Operation Timing Chart (Trigger Restarting Disabled)



● When restarting is enabled (RTGEN = 1)

Figure 22.8-10 One-shot Operation Timing Chart (Trigger Restarting Enabled)



22.8.1.7 Interrupt Factors and Timing Chart

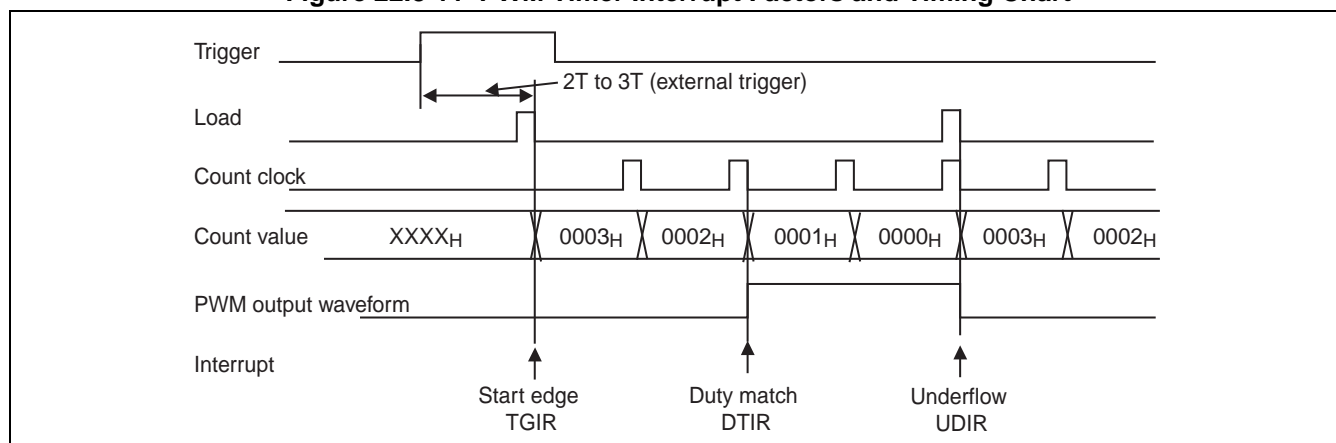
This section provides the interrupt factors and timing chart.

■ 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 22.8-11 shows the interrupt factors and timing chart, assuming "period setting" = 3 and "duty value" = 1.

Figure 22.8-11 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 22.8-12 and Figure 22.8-13 illustrate how to provide PWM output at all "L" and all "H" levels, respectively.

Figure 22.8-12 Example of PWM Output at All "L" Level

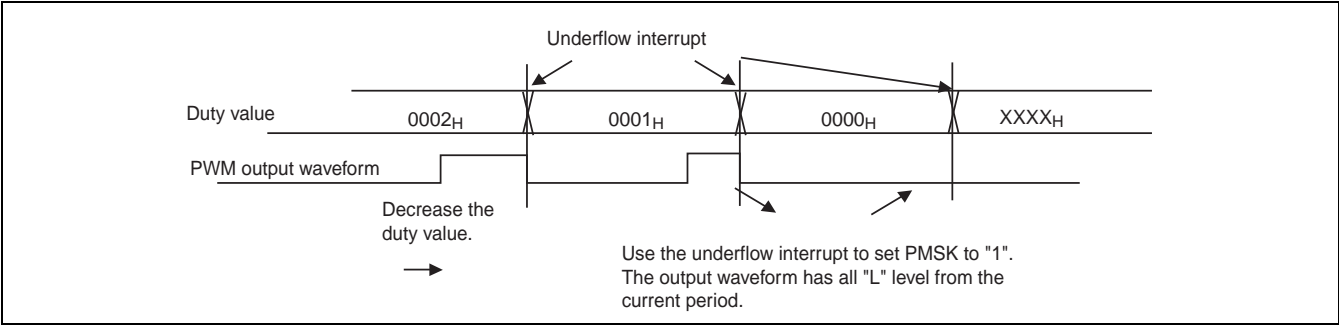
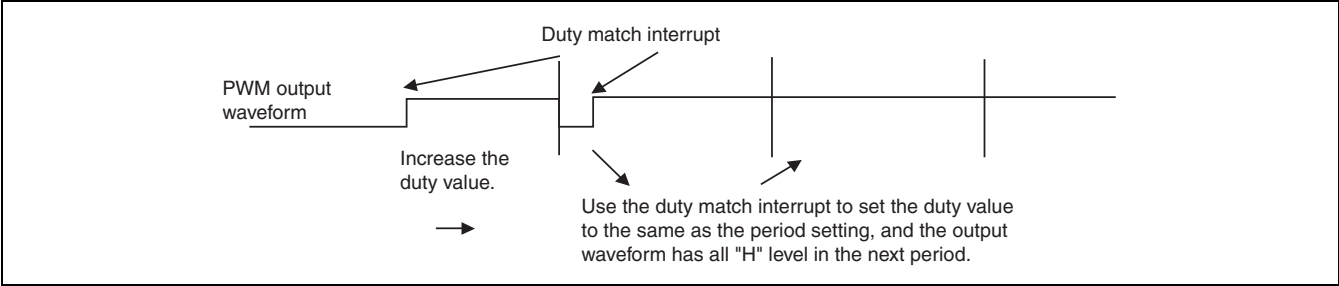


Figure 22.8-13 Example of PWM Output at All "H" Level



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 22.8-14 Timer Control Register (BTxTMCR Upper Byte)

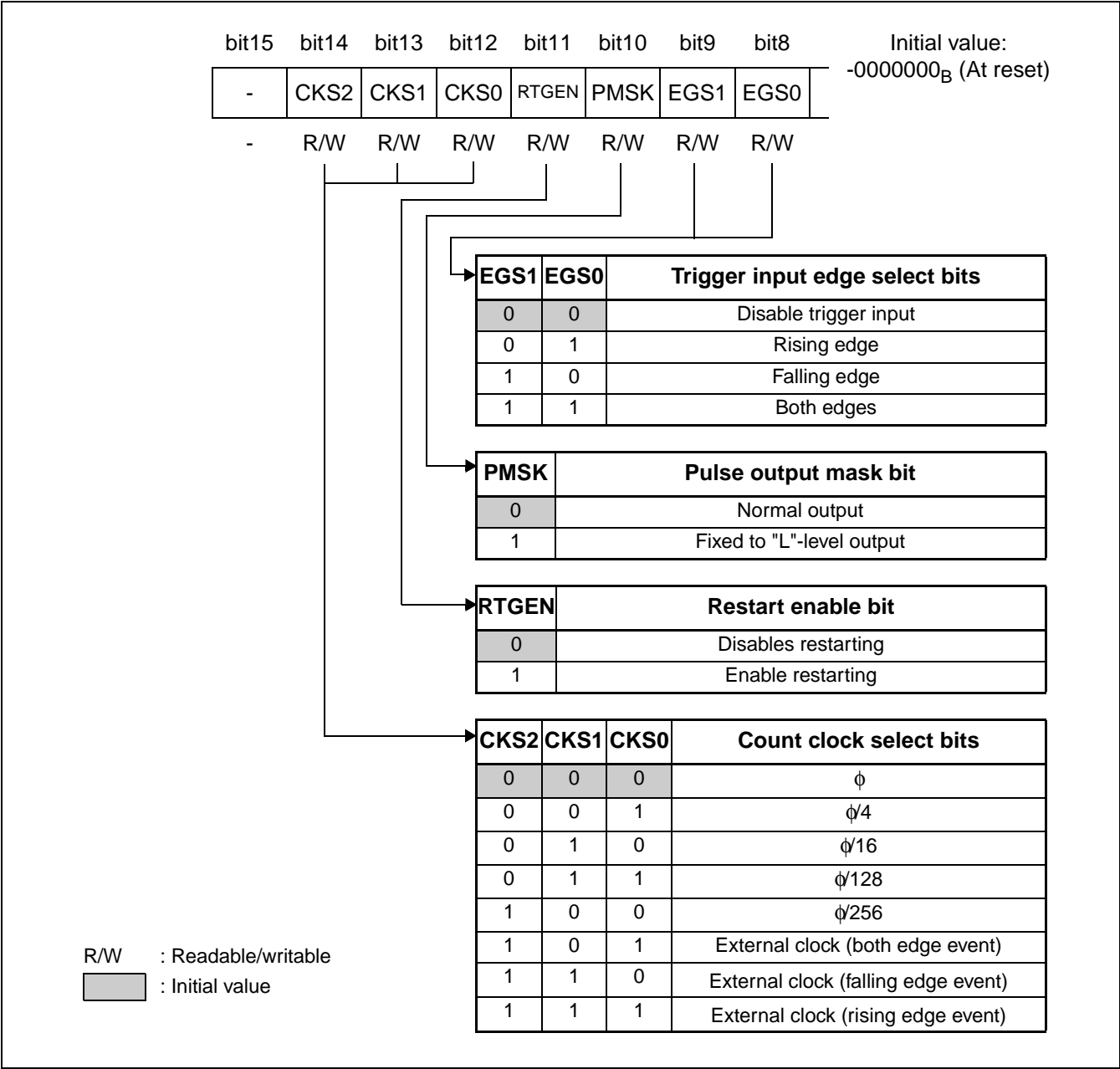


Table 22.8-4 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Select the count clock for the 16-bit down counter. The count clock promptly reflects any changes made to its setting. <p>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.</p>
bit11	RTGEN: Restart enable bit	This bit enables restarting with a software trigger or trigger input.
bit10	PMSK: Pulse output mask bit	<ul style="list-style-type: none"> 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. <p>Note: Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PPG output to the "H" level.</p>
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	<ul style="list-style-type: none"> 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. <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0.</p> <ul style="list-style-type: none"> 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 22.8-15 Timer Control Register (BTxTMCR Lower Byte)

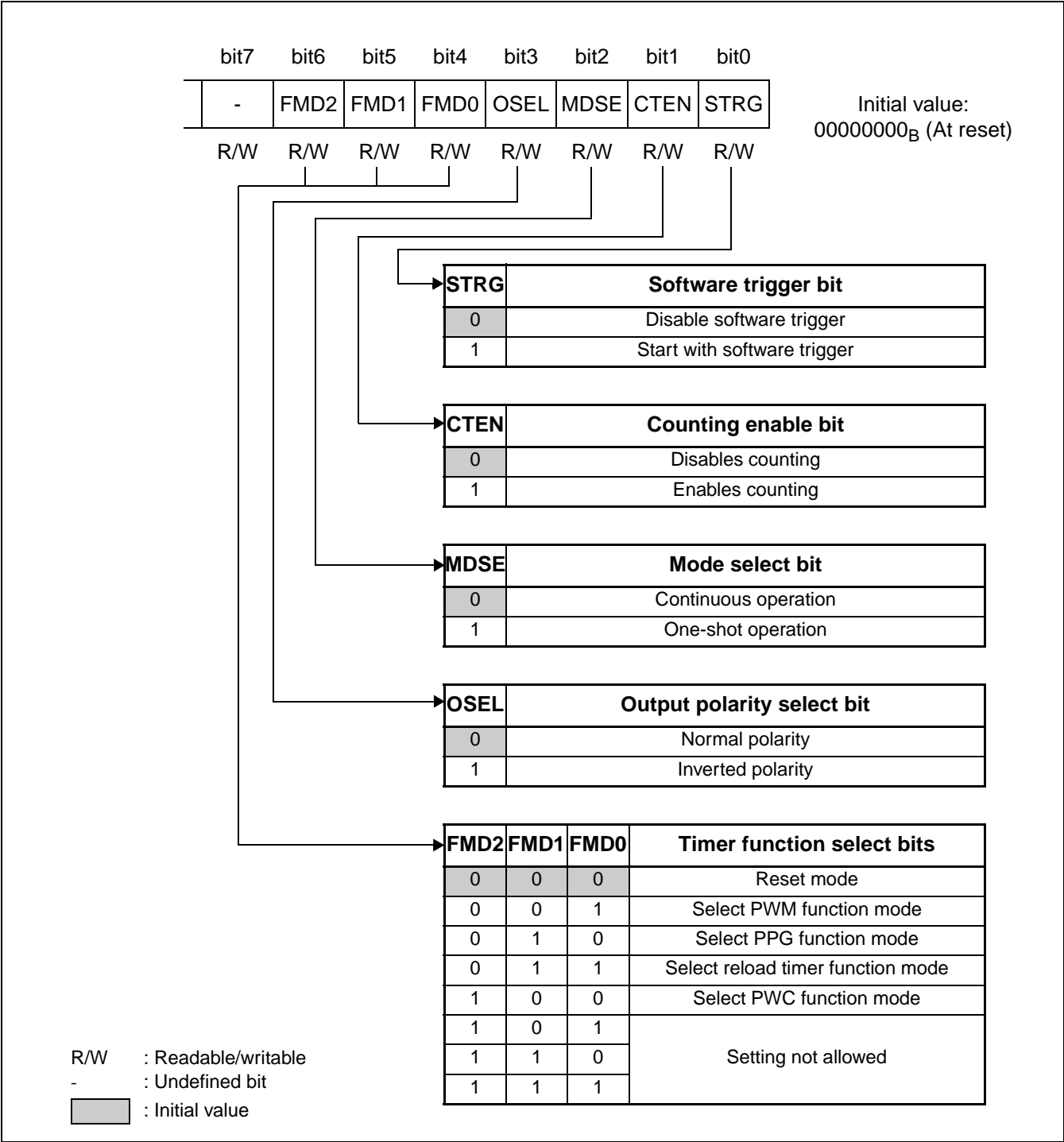














Table 22.8-5 Timer Control Register (BTxTMCR Lower Byte)

Bit name		Function												
bit7	Undefined bit	<ul style="list-style-type: none">The value read is "0"When writing to this bit, write "0".												
bit6 to bit4	FMD2, FMD1, FMD0: Timer function select bits	<ul style="list-style-type: none">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.												
bit3	OSEL: Output polarity select bit	<ul style="list-style-type: none">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>	Polarity	After reset	End of "L"-width counting	End of "H"-width counting	Normal	"L" output			Inverted	"H" output		
Polarity	After reset	End of "L"-width counting	End of "H"-width counting											
Normal	"L" output													
Inverted	"H" output													
bit2	MDSE: Mode select bit	<ul style="list-style-type: none">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.												
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none">This bit enables the down counter.Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.												
bit0	STRG: Software trigger bit	<ul style="list-style-type: none">Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger. <p>Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.</p> <ul style="list-style-type: none">The value read from the STRG bit is always "0". <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</p>												

■ Status Control Register (BTxSTC)

Figure 22.8-16 Status Control Register (BTxSTC)

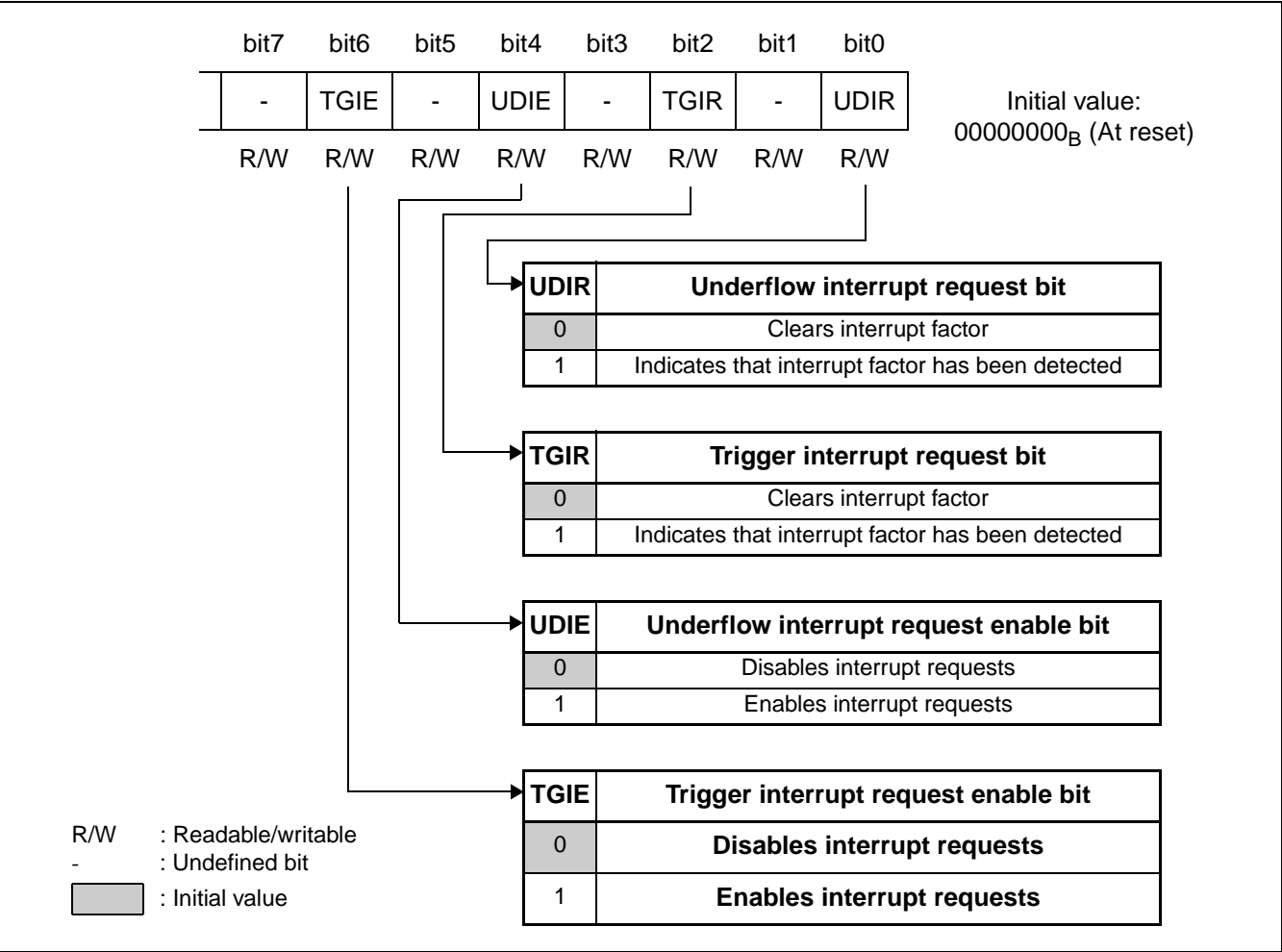


Table 22.8-6 Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	<ul style="list-style-type: none"> 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.

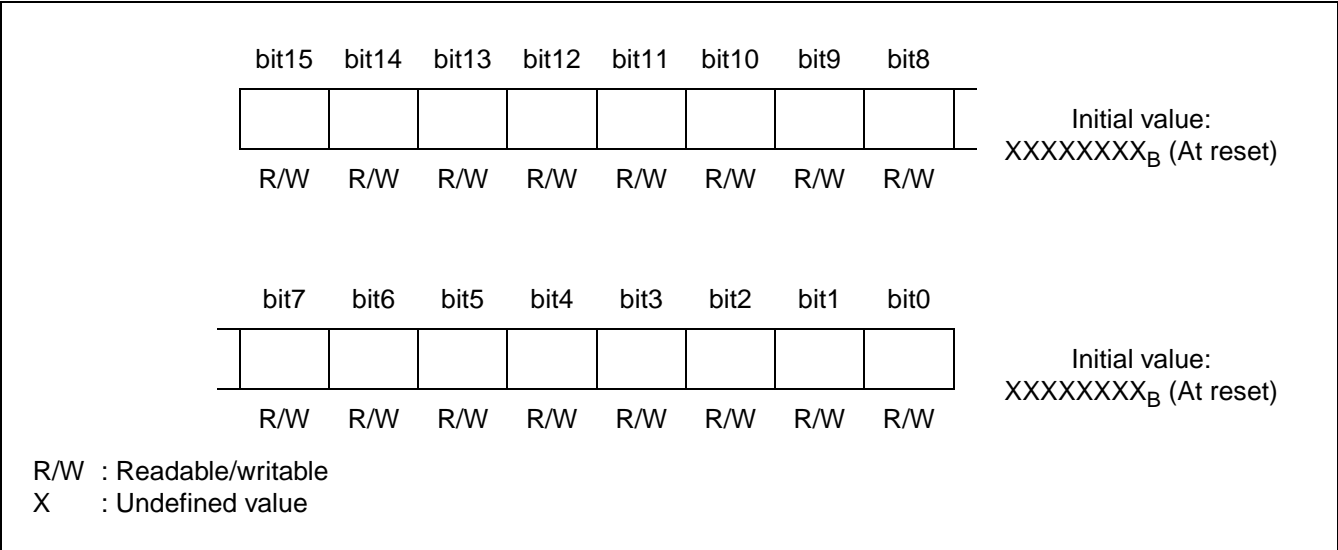
22.8.2.2 "L"-width Setting Reload Register (BTxPRL)

The "L"-width setting reload register (BTxPRL) 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 (BTxPRL)

Figure 22.8-17 shows the bit configuration of the "L"-width setting reload register (BTxPRL).

Figure 22.8-17 Bit Configuration of the "L"-width Setting Reload Register (BTxPRL)



- The BTxPRL 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 BTxPRL register using 16-bit data.
 - Set the "L" width using the BTxPRL register after selecting the PPG function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

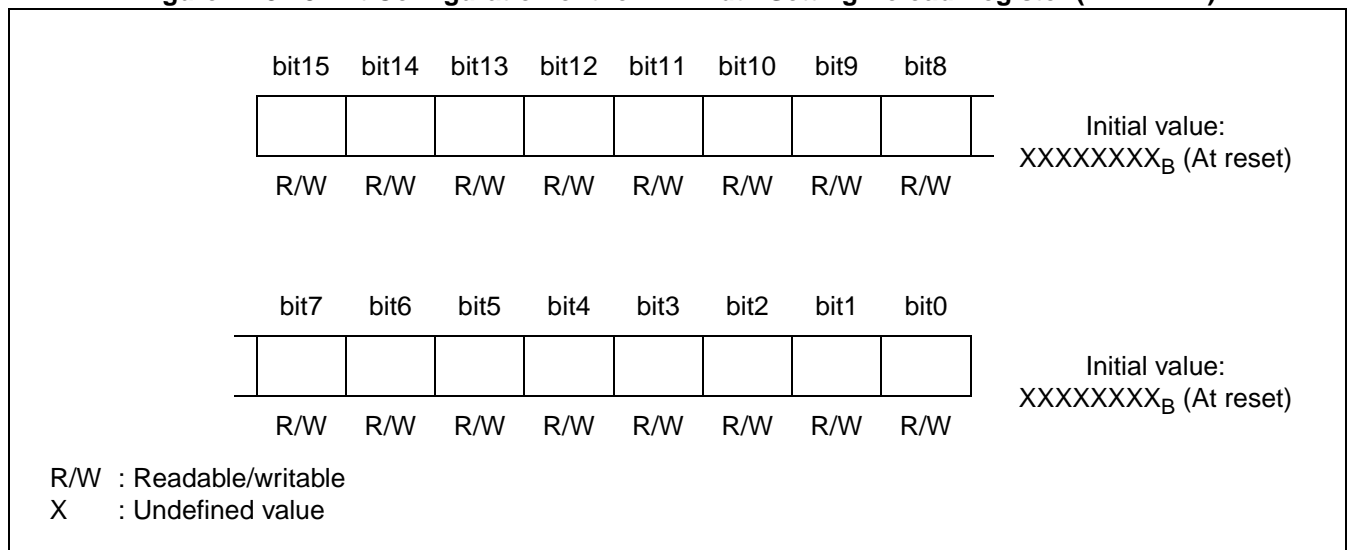
22.8.2.3 "H"-width Setting Reload Register (BTxPRLH)

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 22.8-18 shows the bit configuration of the "H"-width setting reload register (BTxPRLH).

Figure 22.8-18 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.

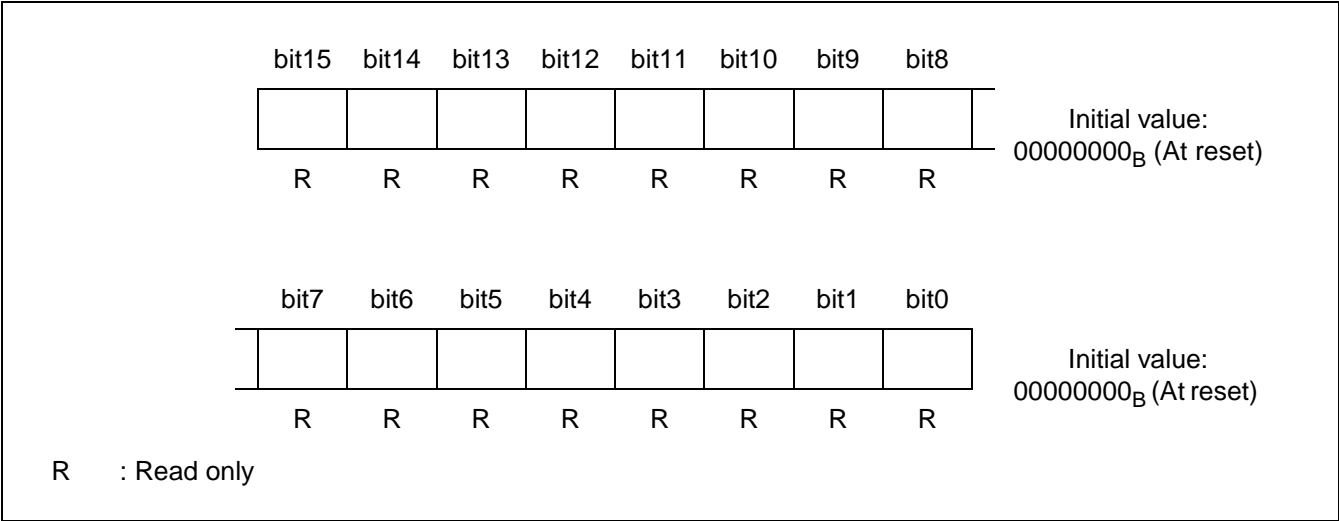
22.8.2.4 Timer Register (BTxTMR)

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 22.8-19 shows the bit configuration of the PPG timer register (BTxTMR).

Figure 22.8-19 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

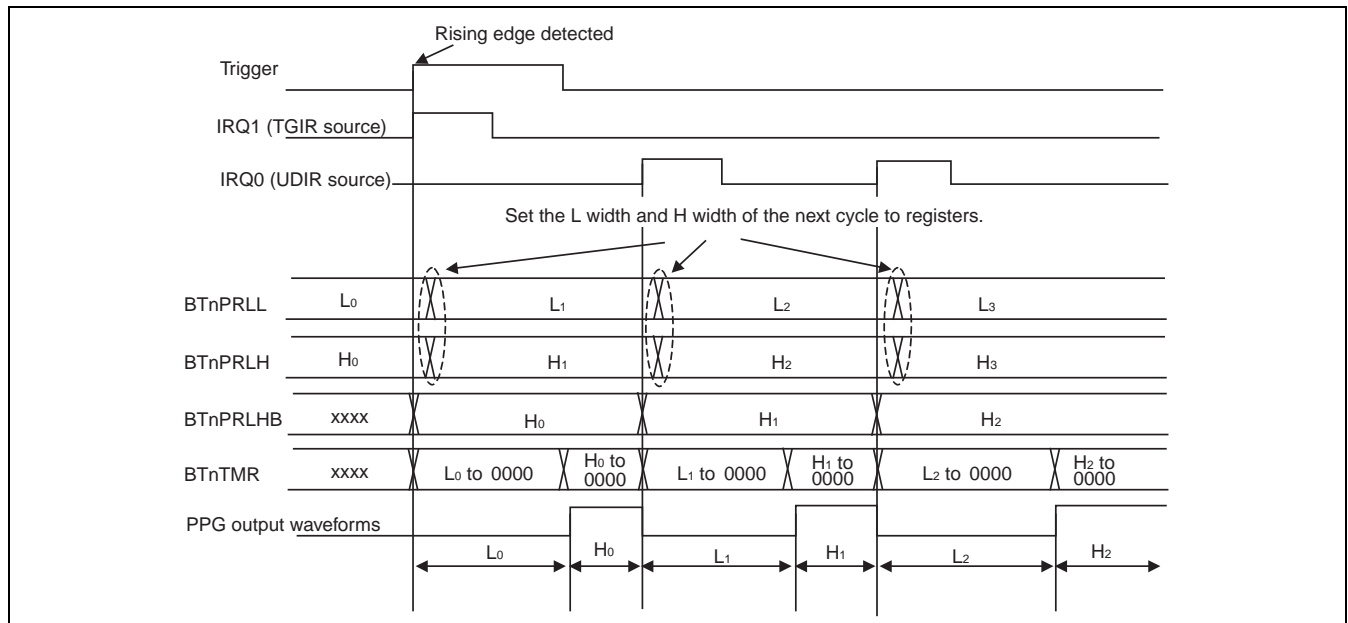
The PPG timer has two 16-bit reload registers for setting the "L" and "H" widths respectively and one "H" width setting buffer (BTxPRL, BTxPRLH, BTxPRLHB).

In response to the start trigger, the 16-bit down counter loads the BTxPRL 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 BTxPRL 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 BTxPRL 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 BTxPRL 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.



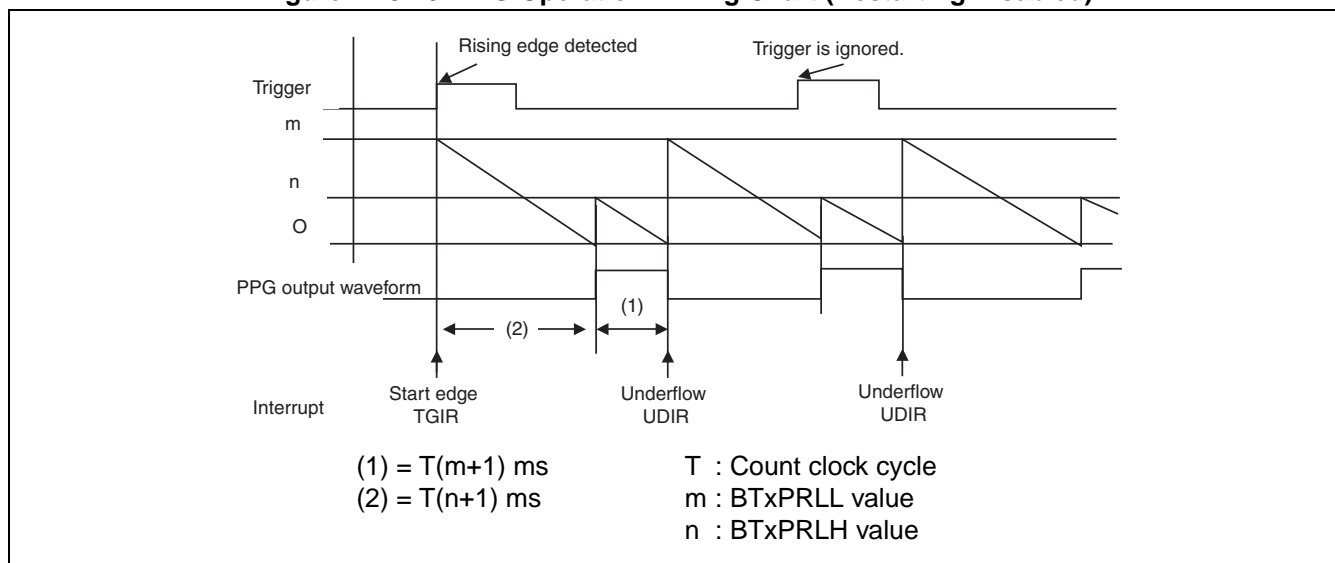
22.8.2.6 Continuous Operation

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

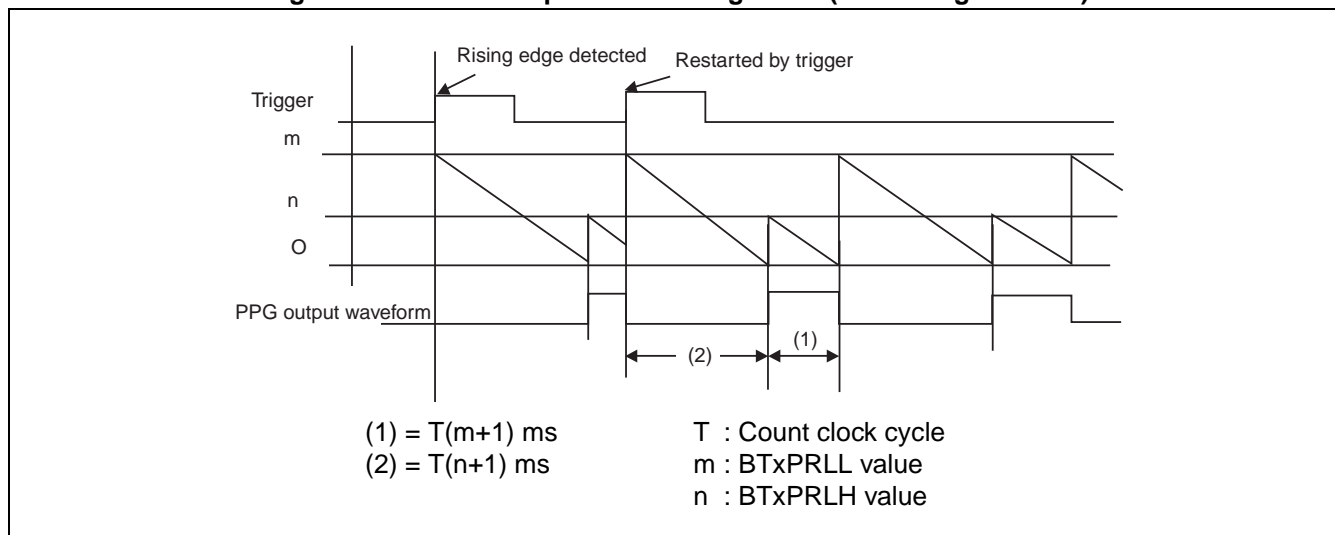
● When restarting is disabled (RTGEN = 0)

Figure 22.8-20 PPG Operation Timing Chart (Restarting Disabled)



● When restarting is enabled (RTGEN = 1)

Figure 22.8-21 PPG Operation Timing Chart (Restarting Enabled)



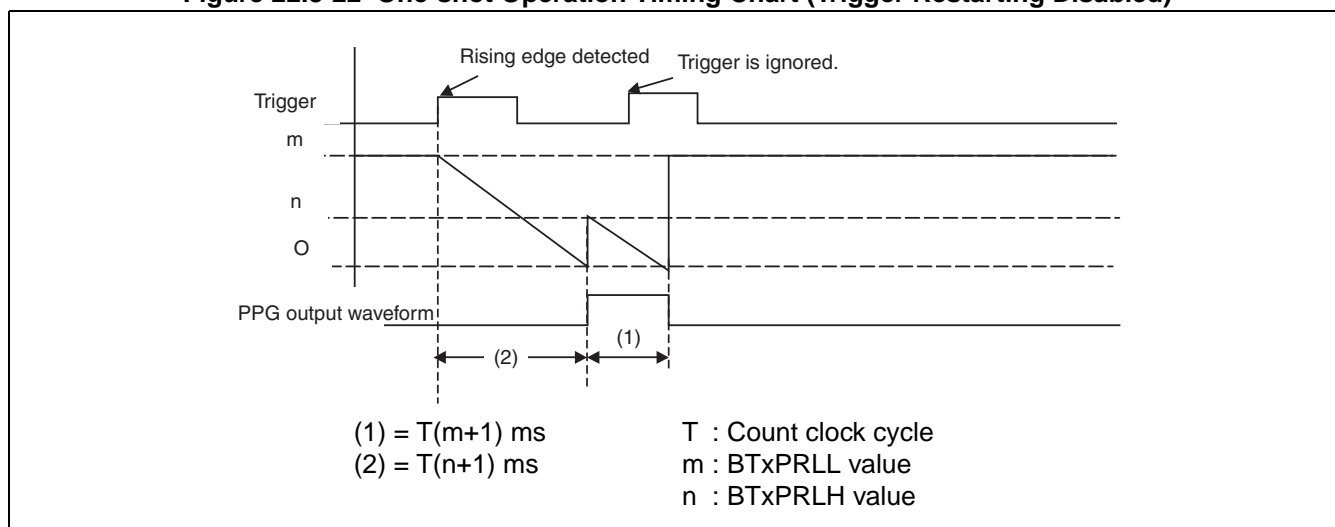
22.8.2.7 One-shot Operation

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

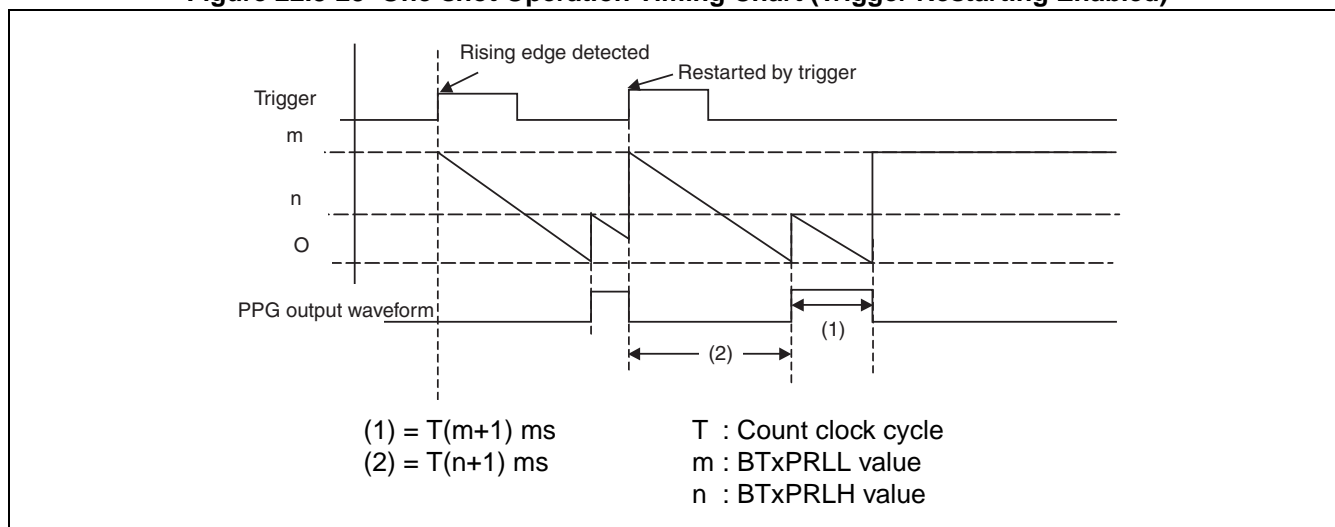
● When restarting is disabled (RTGEN = 0)

Figure 22.8-22 One-shot Operation Timing Chart (Trigger Restarting Disabled)



● When restarting is enabled (RTGEN = 1)

Figure 22.8-23 One-shot Operation Timing Chart (Trigger Restarting Enabled)



■ 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.

$$PL = T \times (L+1)$$

$$PH = T \times (H+1)$$

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.

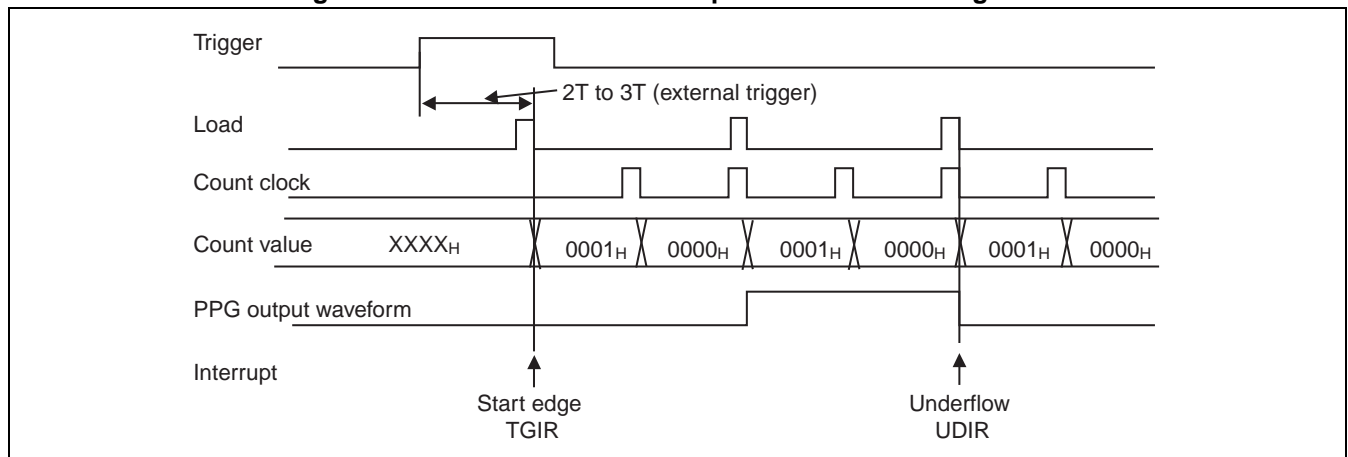
■ Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)

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 22.8-24 shows the interrupt factors and timing chart, assuming "L" width setting = 1 and "H" width setting = 1.

Figure 22.8-24 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 22.8-25 Timer Control Register (BTxTMCR Upper Byte)

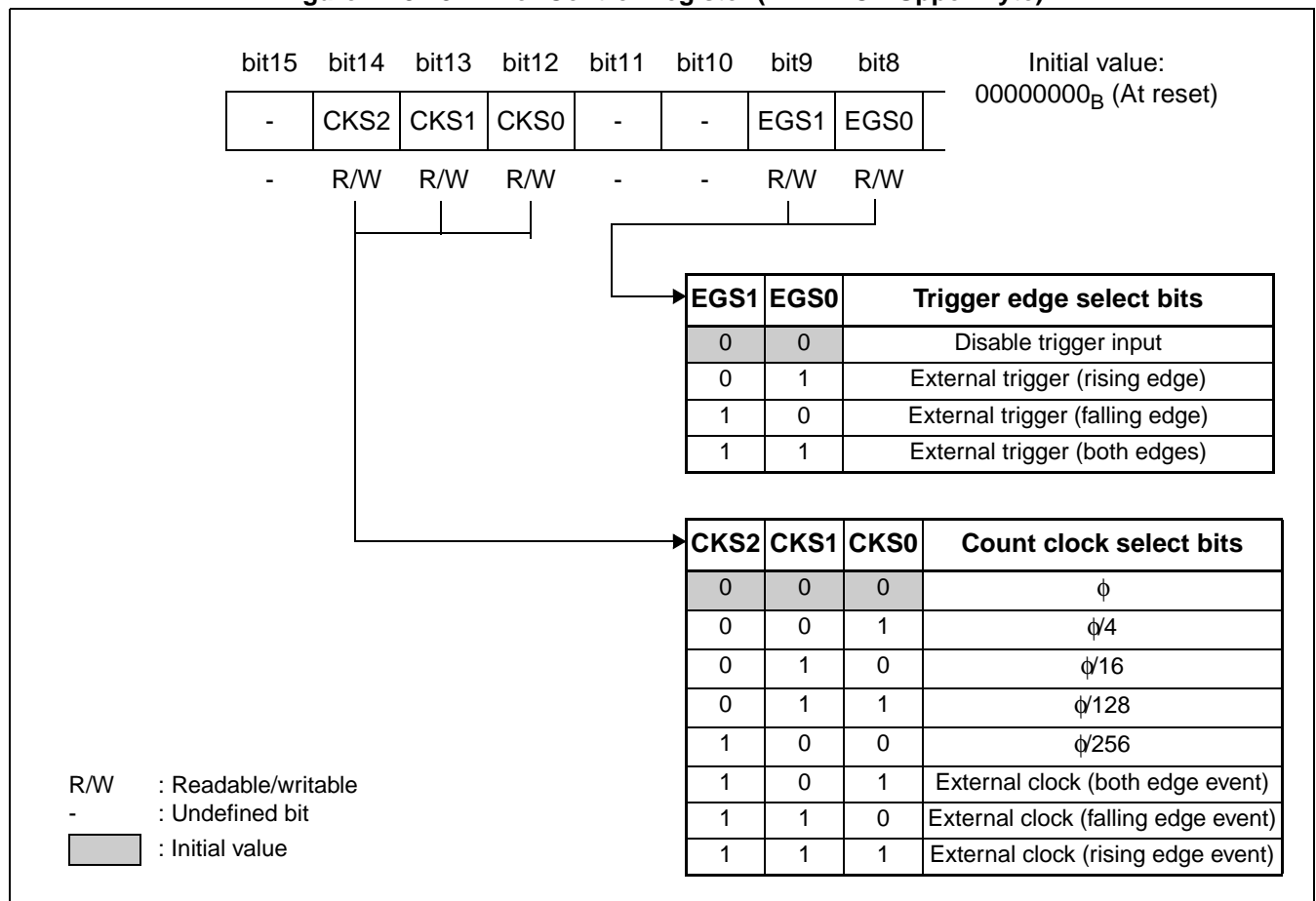


Table 22.8-7 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to these bits, write "0".
bit9, bit8	EGS1, EGS0: Trigger edge select bits	<ul style="list-style-type: none"> 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. <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0.</p> <ul style="list-style-type: none"> 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 22.8-26 Timer Control Register (BTxTMCR Lower Byte)

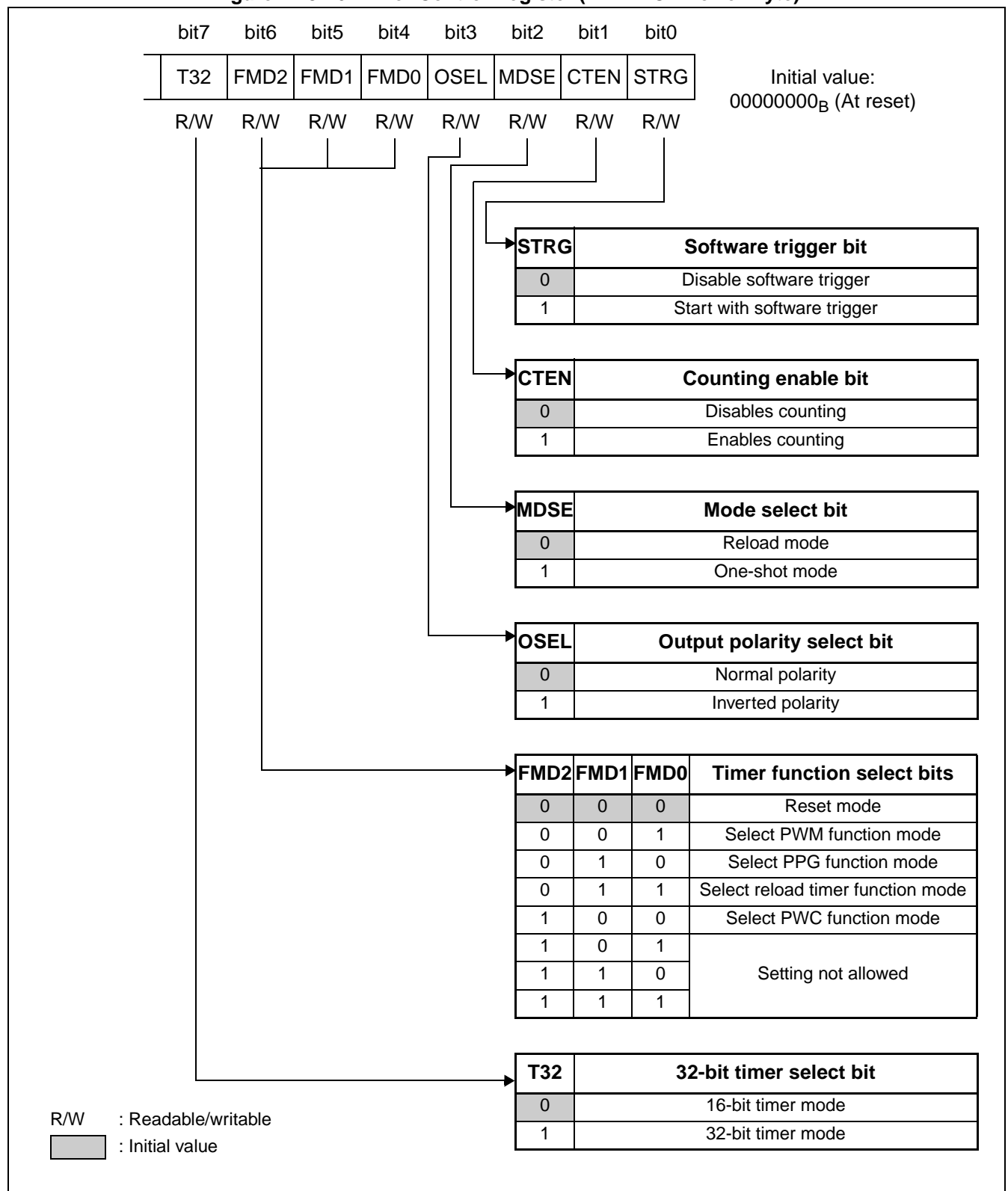


Table 22.8-8 Timer Control Register (BTxTMCR Lower Byte)

Bit name		Function															
bit7	T32: 32-bit timer select bit	<ul style="list-style-type: none"> 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". 															
bit6 to bit4	FMD2, FMD1, FMD0: Timer function select bits	<ul style="list-style-type: none"> 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. 															
bit3	OSEL: Output polarity select bit	<ul style="list-style-type: none"> 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 border="1"> <thead> <tr> <th>MDSE</th><th>OSEL</th><th>Output Waveforms</th></tr> </thead> <tbody> <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> </tbody> </table>	MDSE	OSEL	Output Waveforms	0	0	Toggle output of "L" at the count start	0	1	Toggle output of "H" at the count start	1	0	Rectangular wave of "H" during count	1	1	Rectangular wave of "L" during count
MDSE	OSEL	Output Waveforms															
0	0	Toggle output of "L" at the count start															
0	1	Toggle output of "H" at the count start															
1	0	Rectangular wave of "H" during count															
1	1	Rectangular wave of "L" during count															
bit2	MDSE: Mode select bit	<ul style="list-style-type: none"> 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. 															
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none"> This bit enables the down counter. Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter. 															
bit0	STRG: Software trigger bit	<ul style="list-style-type: none"> Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger. <p>Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.</p> <ul style="list-style-type: none"> The value read from the STRG bit is always "0". <p>Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</p>															

■ Status Control Register (BTxSTC)

Figure 22.8-27 Status Control Register (BTxSTC)

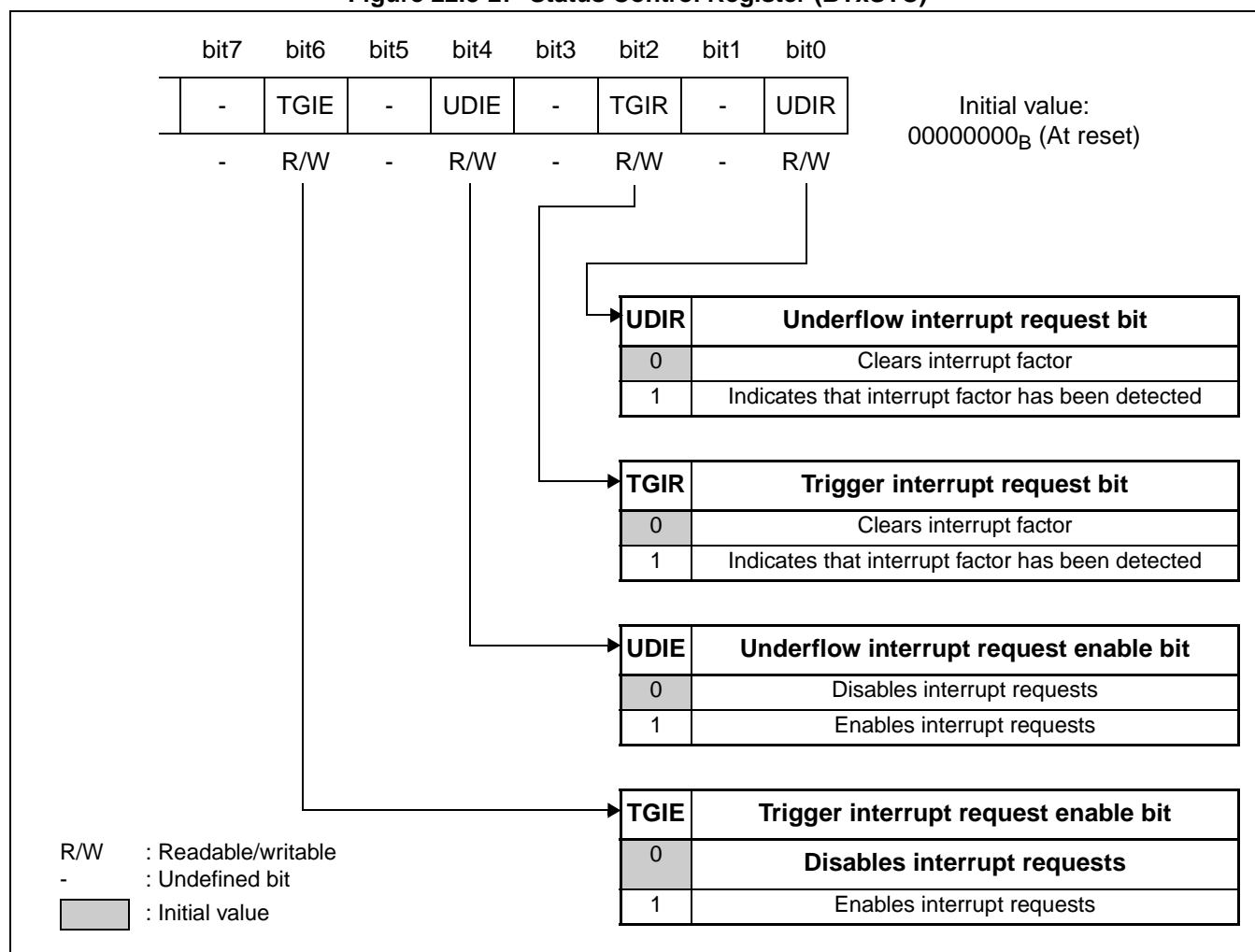


Table 22.8-9 Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	<ul style="list-style-type: none"> 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.

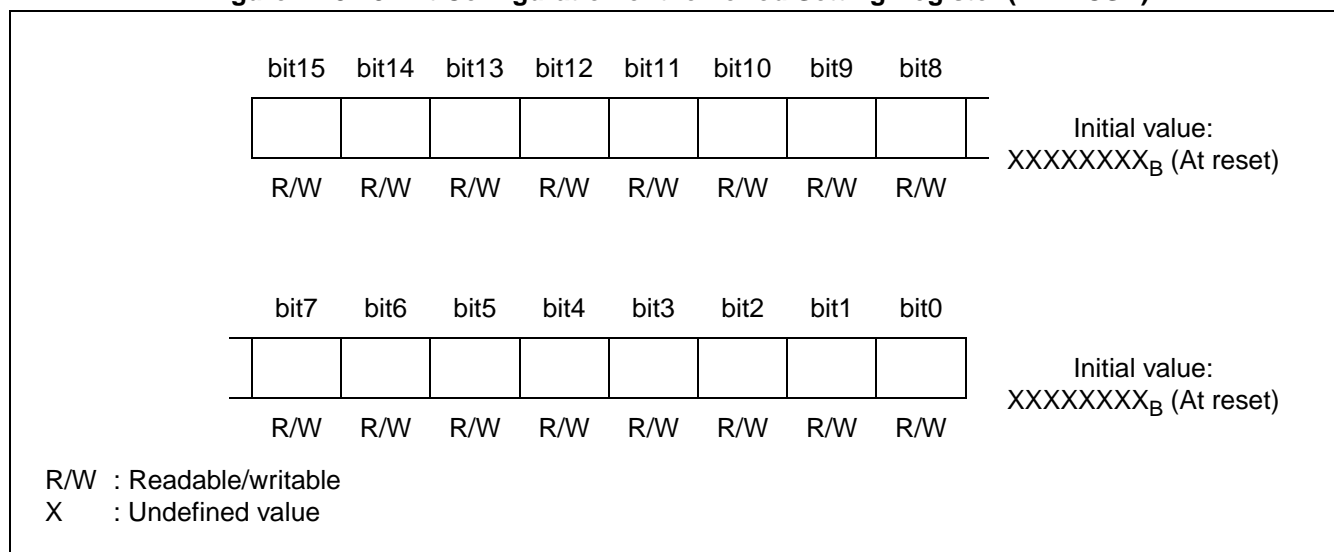
22.8.3.2 Period Setting Register (BTxPCSR)

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 22.8-28 shows the bit configuration of the period setting register (BTxPCSR).

Figure 22.8-28 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).

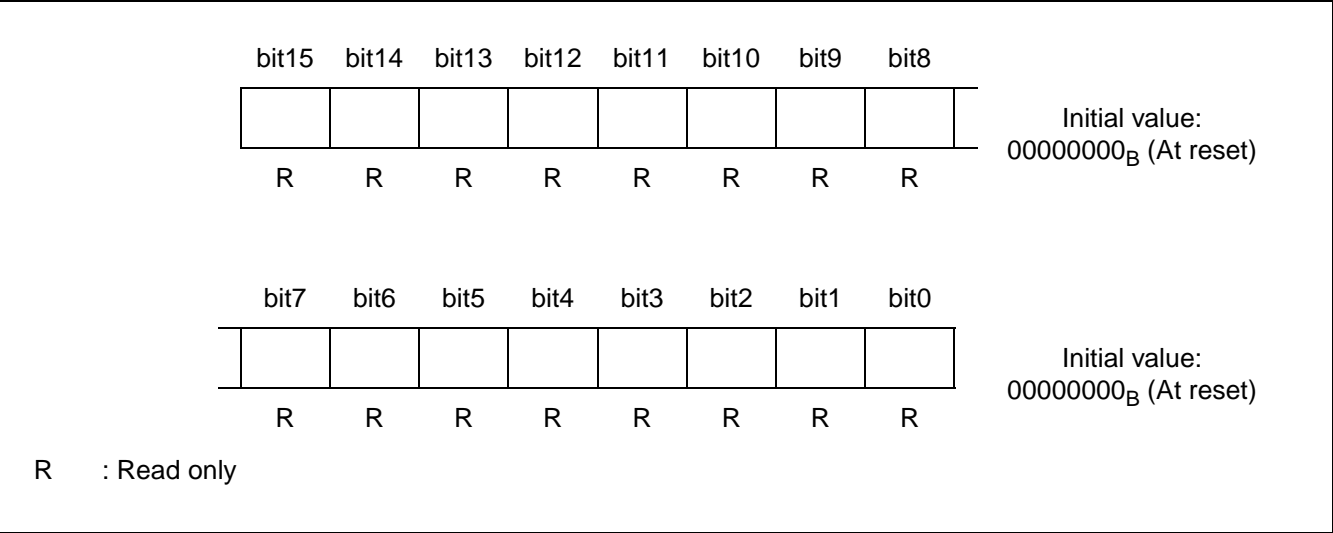
22.8.3.3 Timer Register (BTxTMR)

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 22.8-29 shows the bit configuration of the timer register (BTxTMR).

Figure 22.8-29 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

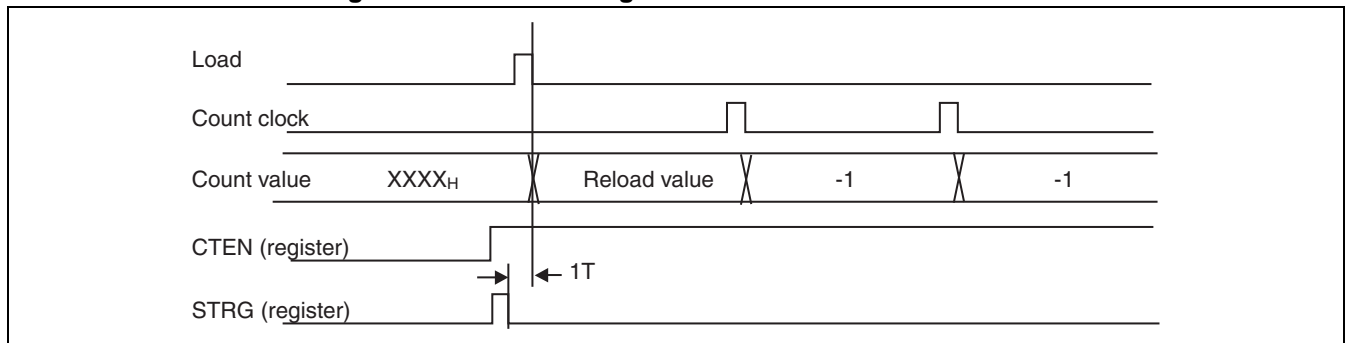
To start counting the moment counting is enabled, write "1" to both of the CTEN and STRG bits in the timer control register. The STRG bit maintains the trigger input always enabled irrespective of the operation mode as long as the timer is active (CNTE = 1).

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 22.8-30 illustrates how the counter is started by the software trigger and operates.

Figure 22.8-30 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.

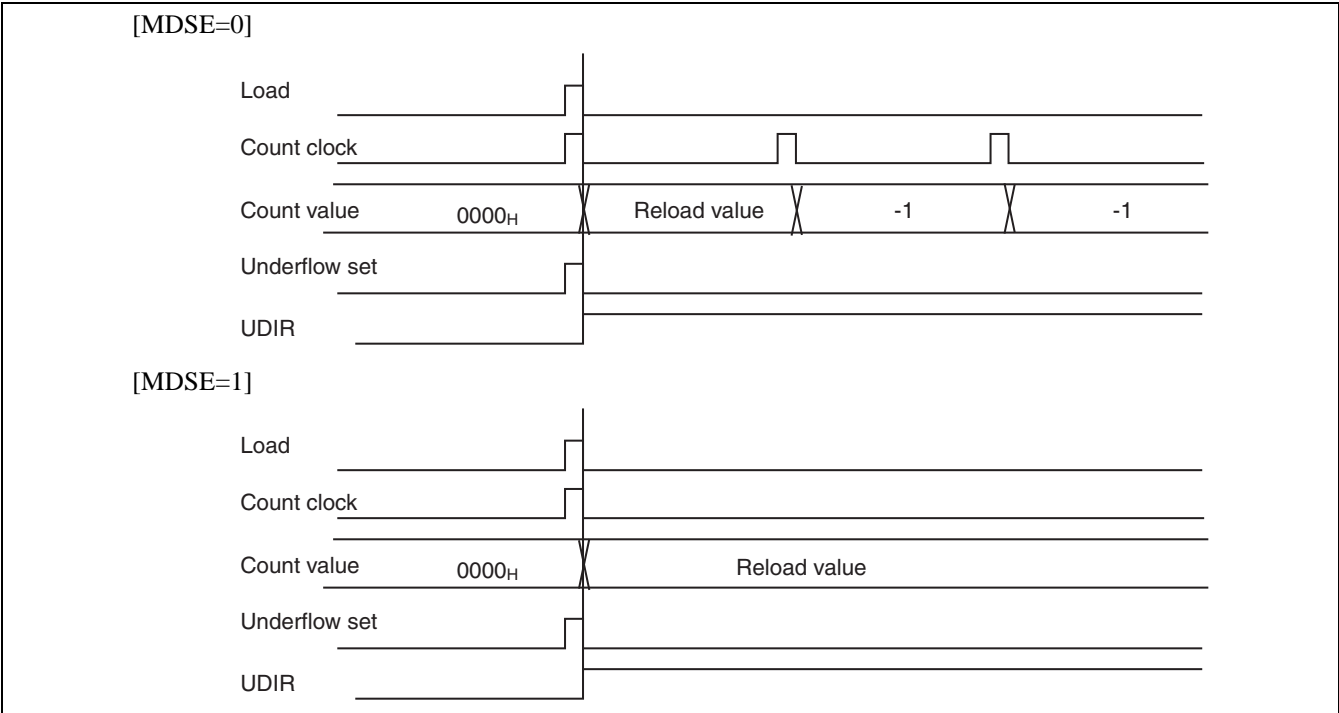
When the counter counts [period setting register value + 1], therefore, an underflow occurs.

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 22.8-31 is a timing chart of underflow operation.

Figure 22.8-31 Underflow Operation Timing Chart

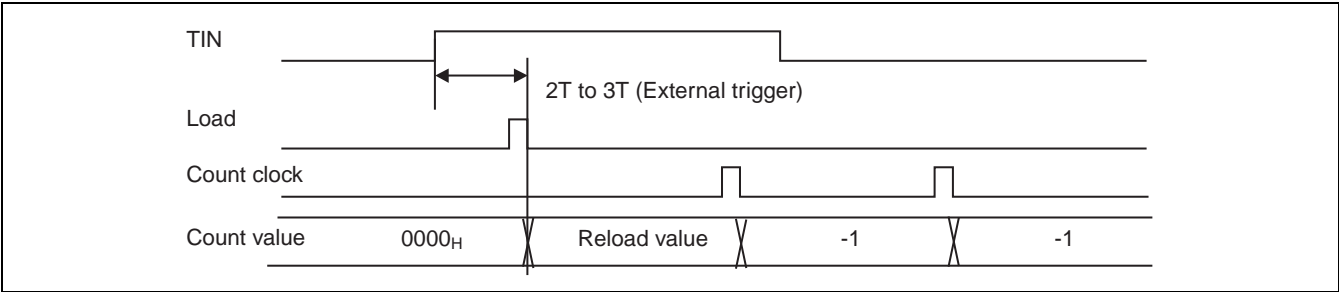


■ Input Pin Operation

The TIN pin can be used as a trigger input. When the effective edge is input to the TIN 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 22.8-32 illustrates the trigger input operation with the rising edge selected as the effective edge.

Figure 22.8-32 Trigger Input Operation

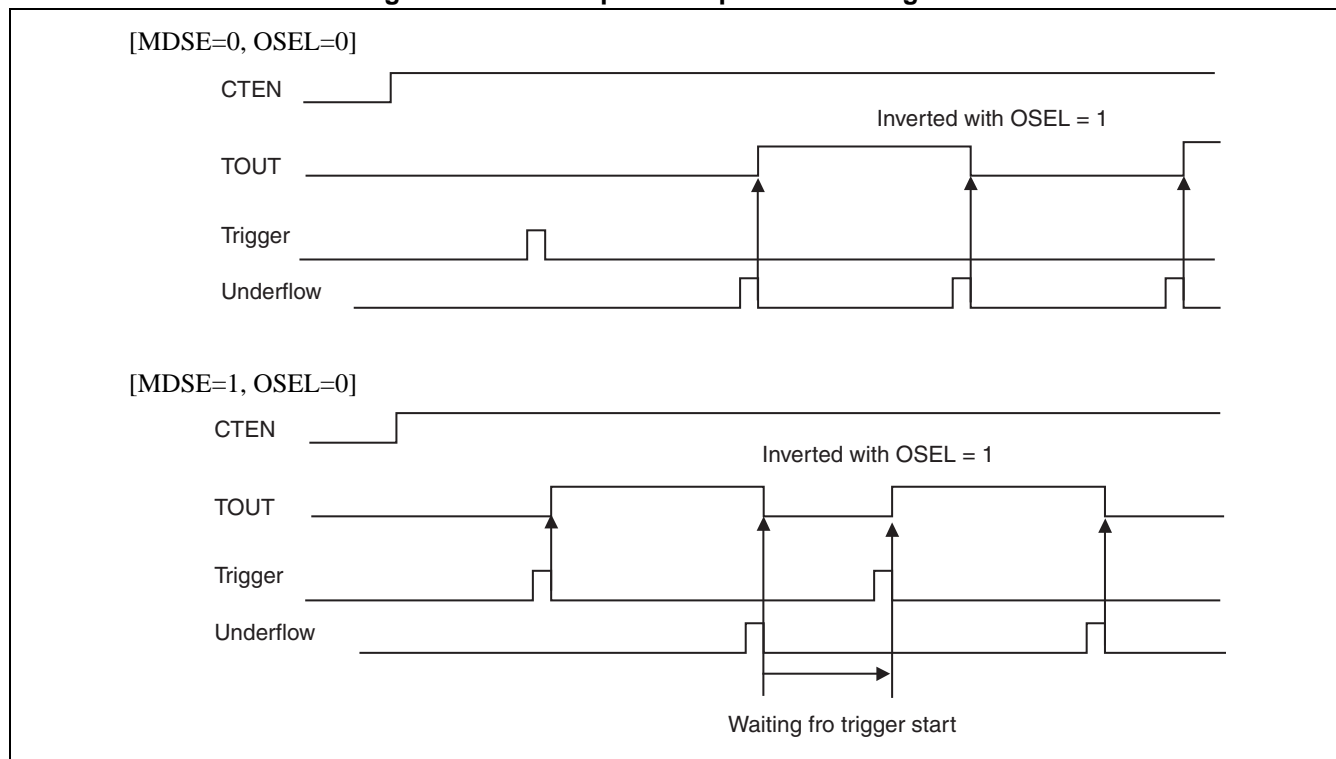


■ Output Pin Operation

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 22.8-33 is a timing chart of output pin operation.

Figure 22.8-33 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 22.8-34 Timer Control Register (BTxTMCR Upper Byte)

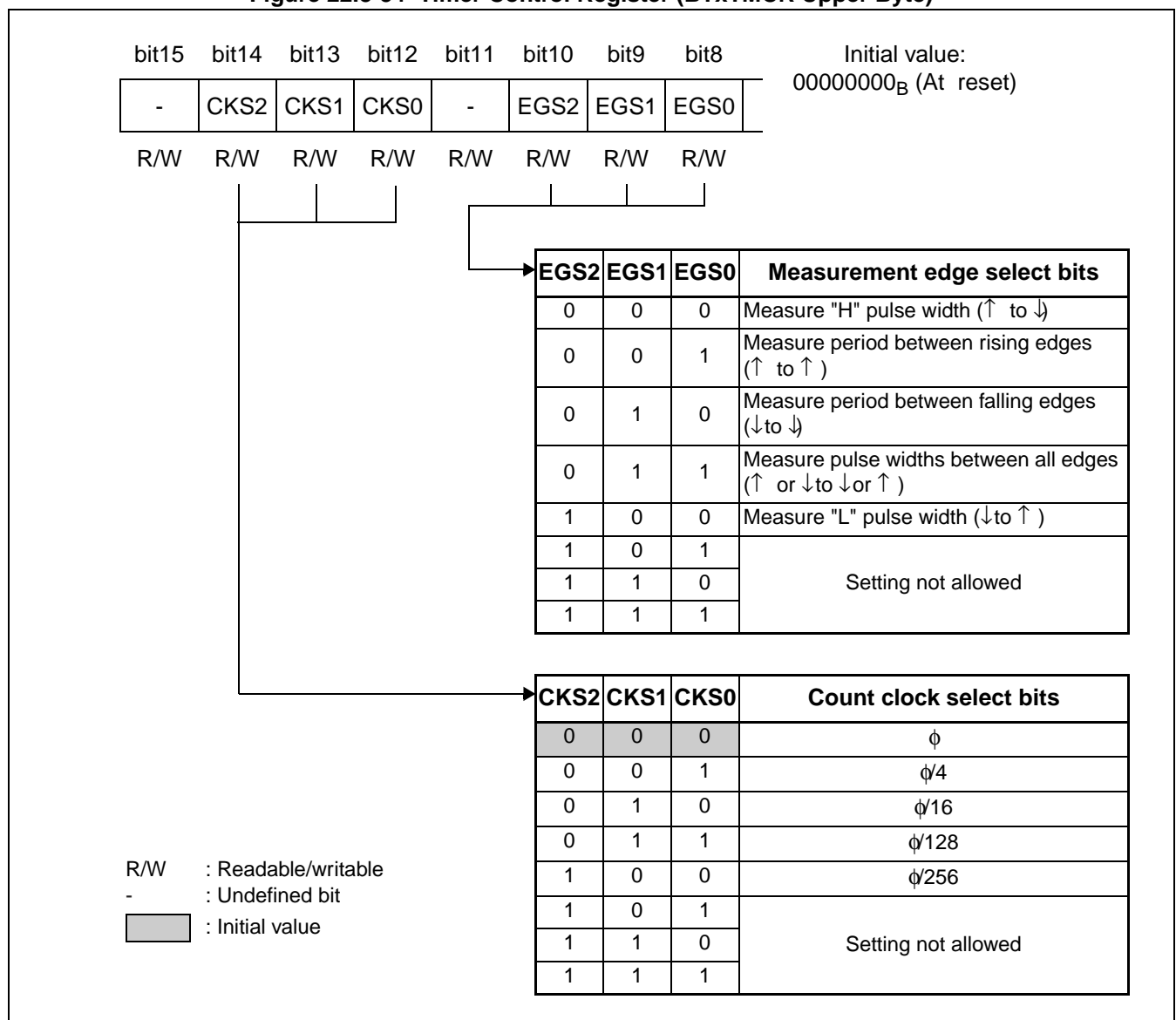


Table 22.8-10 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit10 to bit8	EGS2, EGS1, EGS0: Measurement edge select bits	<ul style="list-style-type: none"> 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 22.8-35 Timer Control Register (BTxTMCR Lower Byte)

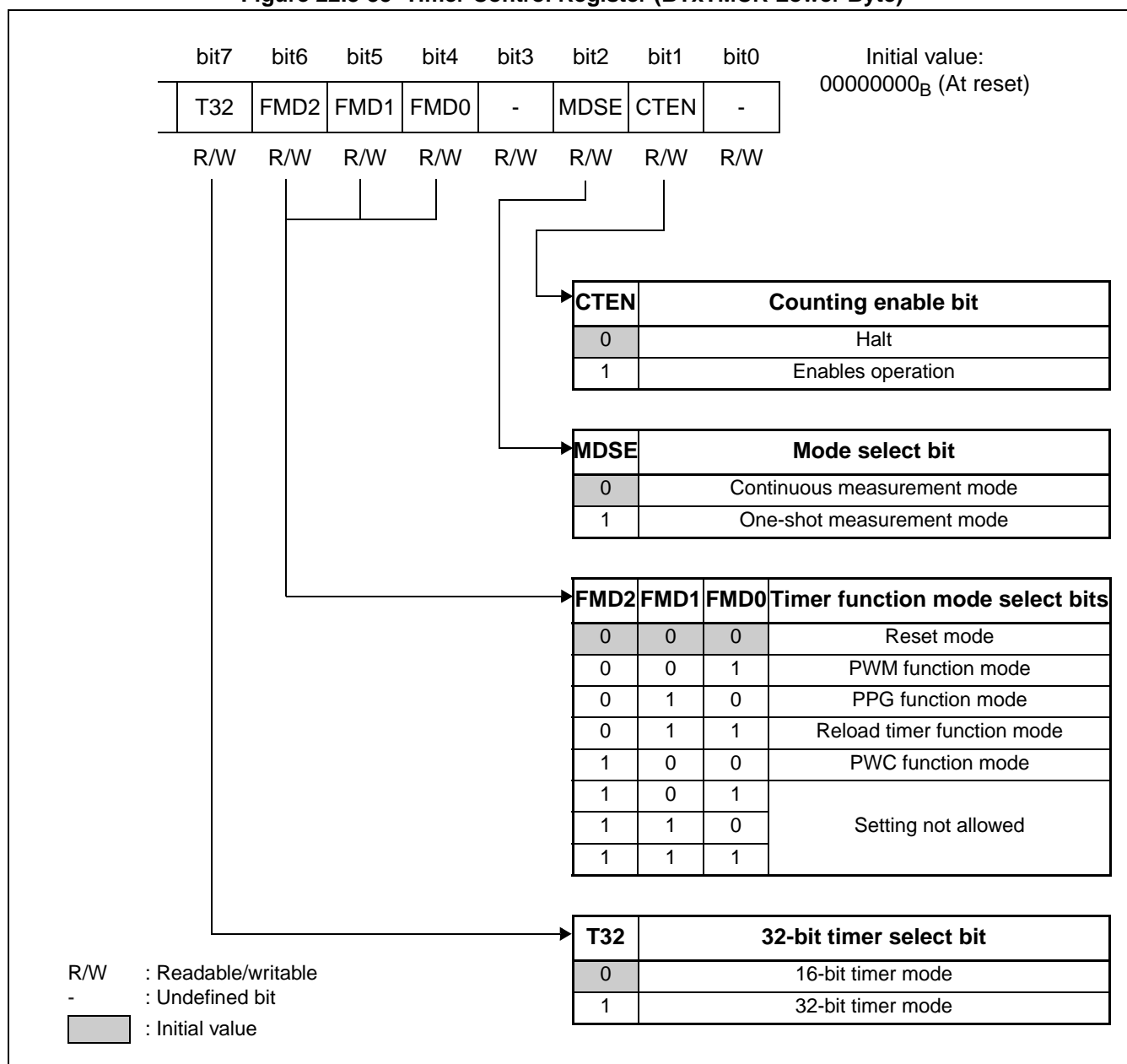


Table 22.8-11 Timer Control Register (BTxTMCR Lower Byte)

Bit name		Function									
bit7	T32: 32-bit timer select bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0". 									
bit2	MDSE: Mode select bit	<ul style="list-style-type: none"> Selects measurement mode as follows. <table border="1"> <thead> <tr> <th>MDSE</th><th>Mode</th><th>Operation</th></tr> </thead> <tbody> <tr> <td>0</td><td>Continuous measurement</td><td>Continuous measurement: buffer register enabled</td></tr> <tr> <td>1</td><td>One-shot measurement</td><td>Halts after each measurement</td></tr> </tbody> </table> <ul style="list-style-type: none"> 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. 	MDSE	Mode	Operation	0	Continuous measurement	Continuous measurement: buffer register enabled	1	One-shot measurement	Halts after each measurement
MDSE	Mode	Operation									
0	Continuous measurement	Continuous measurement: buffer register enabled									
1	One-shot measurement	Halts after each measurement									
bit1	CTEN: Counting enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0". 									

■ Status Control Register (BTxSTC)

Figure 22.8-36 Status Control Register (BTxSTC)

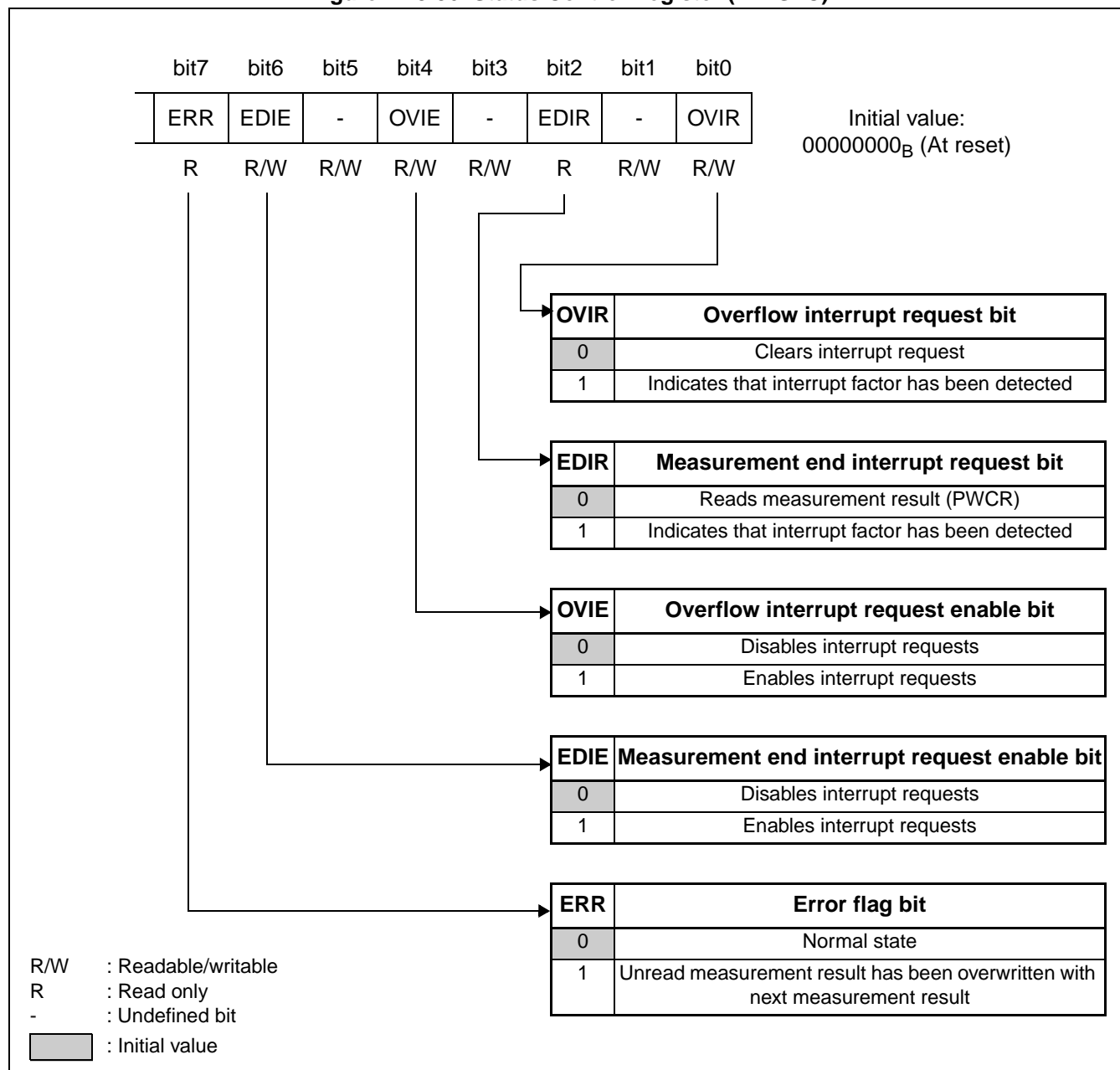


Table 22.8-12 Status Control Register (BTxSTC)

Bit name		Function
bit7	ERR: Error flag bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit4	OVIE: Overflow interrupt request enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit2	EDIR: Measurement end interrupt request bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> The value read is "0" When writing to this bit, write "0".
bit0	OVIR: Overflow interrupt request bit	<ul style="list-style-type: none"> 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.

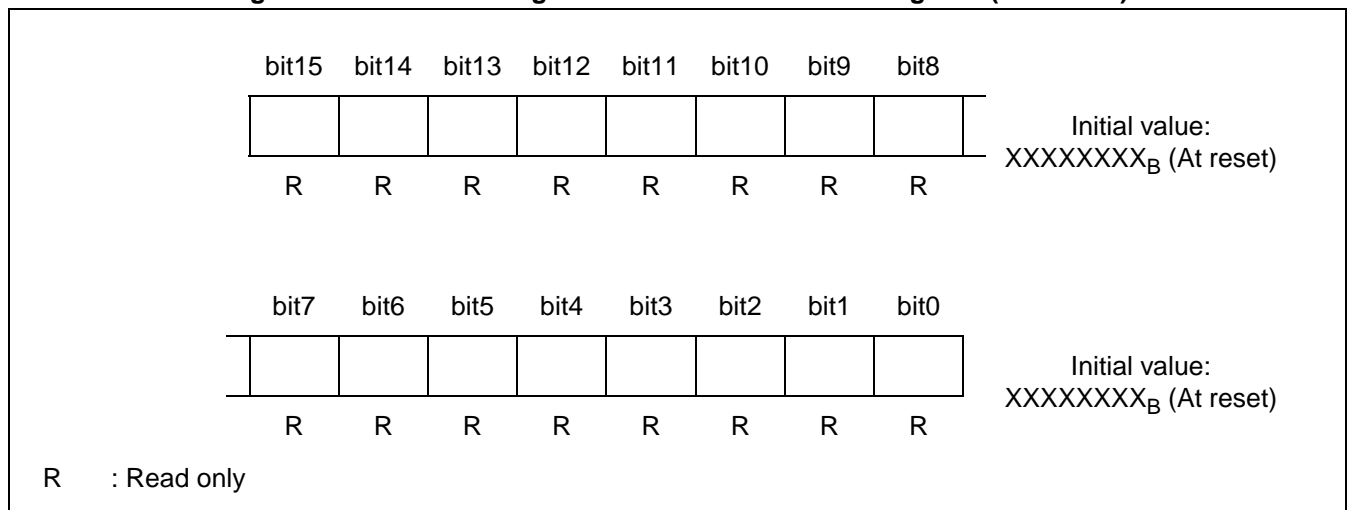
22.8.4.2 Data Buffer Register (BTxDTBF)

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 22.8-37 shows the bit configuration of the data buffer register (BTxDTBF).

Figure 22.8-37 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.

■ 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	Internal count clock selected
CKS2, CKS1, CKS0 bits	
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 BTxDTBFB 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 BTxDTBFB 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:

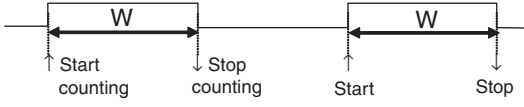
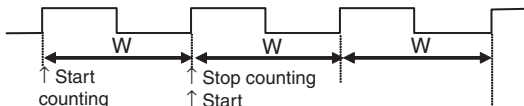
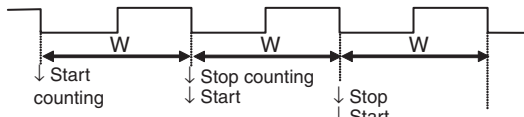
When measurement is completed, the measurement result in the counter is transferred to the BTxDTBF register.

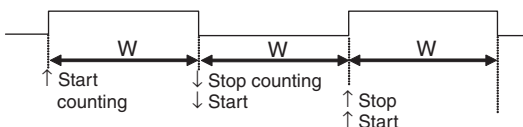
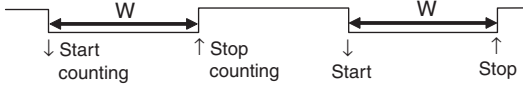
When the BTxDTBF register is read, the last measurement result is obtained. During measurement operation, the BTxDTBF register holds the result of preceding measurement. The count value being measured cannot be read.

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	 <p>Measure the width of "H" period. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of falling edge</p>
Measurement of period between rising edges	001 _B	 <p>Measure the period between rising edges. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of rising edge</p>
Measurement of period between falling edges	010 _B	 <p>Measure the period between falling edges. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of falling edge</p>

Measurement mode	EGS2, EGS1, EGS0	Measurement target (W: Pulse width to be measured)
Measurement of pulse widths between all edges	011 _B	 <p>Measure the width between continuously input edges. Start counting (measurement) : upon detection of edge Stop counting (measurement) : upon detection of edge</p>
Measurement of "L" pulse width	100 _B	 <p>Measure the width of the "L" period. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of rising edge</p>

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]
-------------------------	--

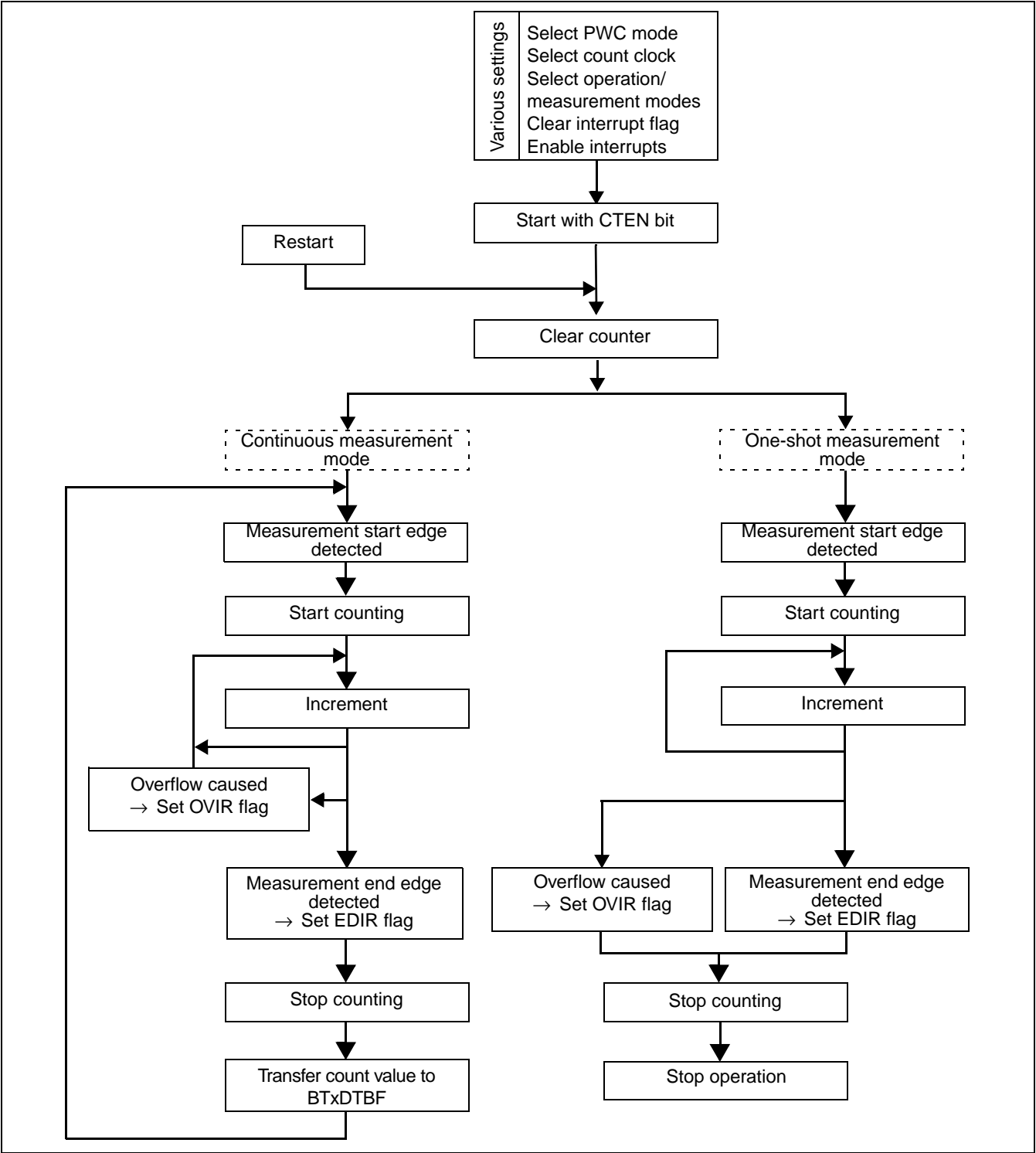
● Generating interrupt requests

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 22.8-40 Pulse Width Measurement Operation Flow



CHAPTER 23 10-bit A/D Converter

This chapter describes the functions and operations of I/O the 10-bit A/D converter.

- 23.1 Overview
- 23.2 Configuration
- 23.3 Pins
- 23.4 Registers
- 23.5 Interrupt
- 23.6 Explanation of Operations

23.1 Overview

The 10-bit A/D converter is a successive approximation A/D converter. It has two conversion modes: activation by software and activation by external trigger.

- Conversion time: Approximately 8.1 μ s (sampling 5.9 μ s, conversion 2.2 μ s) fch: @40MHz
- A/D conversion result register: Available for each channel
- Channel/skin function

This section shows the configuration of the 10-bit A/D converter.

Figure 23.2-1 shows the block diagram of the 10-bit A/D converter.

The diagram illustrates the internal architecture of the AD converter module. On the left, a vertical bus labeled 'M P X' receives 12 analog inputs: AN11, AN10, AN9, AN8, AN7, AN6, AN5, AN4, AN3, AN2, AN1, and AN0. This bus connects to a 'S/H' (Sample and Hold) block, which then feeds into a 'Comparator'. The 'Comparator' also receives input from a 'D/A converter'. The output of the 'Comparator' is sent to the 'Control Logic' block. The 'Control Logic' block is part of a larger module that also contains an 'A/D' (Analog-to-Digital) converter and 'ch & Status Control Logic'. The 'A/D' block is connected to an 'Internal data bus' on the right. The 'ch & Status Control Logic' block has two outputs: one to the 'M P X' bus and another labeled 'Interrupt request'. Additionally, 'External pins' (specifically ATRG) are connected to the 'ch & Status Control Logic' block. A 'Buffer x12' is connected to the 'Internal data bus' and the 'D/A converter'.

23.3 Pins

This section describes the pins of the 10-bit A/D converter.

■ Overview

The 10-bit A/D converter has the following pins:

- AVCC pin
10-bit A/D converter analog power input pin.
- AVRH pin
10-bit A/D converter reference voltage input pin.
- AVSS pin
10-bit A/D converter GND pin.
- AN0 to AN11
10-bit A/D converter analog input pins.

These are multiplexed pins. For details on using the AN0 to AN11 pins of the 10-bit A/D converter, see "17.4.5 ADER Control Register (ADER)".

- ATRG pins
10-bit A/D converter external trigger input pins. These are multiplexed pins.

23.4 Registers

This section describes the configurations and functions of registers used for the 10-bit A/D converter.

■ List of 10-bit A/D converter block registers

Table 23.4-1 shows a list of 10-bit A/D converter registers.

Table 23.4-1 10-bit A/D Converter Registers

Register Abbreviation	Register Name	See
ADCTH,ADCTL	A/D control register	23.4.1
ADCH	Software conversion analog input select register	23.4.2
ADAT0	A/D conversion result register ch.0	23.4.3
ADAT1	A/D conversion result register ch.1	23.4.3
ADAT2	A/D conversion result register ch.2	23.4.3
ADAT3	A/D conversion result register ch.3	23.4.3
ADAT4	A/D conversion result register ch.4	23.4.3
ADAT5	A/D conversion result register ch.5	23.4.3
ADAT6	A/D conversion result register ch.6	23.4.3
ADAT7	A/D conversion result register ch.7	23.4.3
ADAT8	A/D conversion result register ch.8	23.4.3
ADAT9	A/D conversion result register ch.9	23.4.3
ADAT10	A/D conversion result register ch.10	23.4.3
ADAT11	A/D conversion result register ch.11	23.4.3

23.4.1 A/D Control Register (ADCTH, ADCTL)

Figure 23.4-1 shows the bit configuration of the A/D control registers (ADCTH, ADCTL).

Figure 23.4-1 Bit Configuration of the A/D Control Registers (ADCTH, ADCTL)

ADCTH

bit	15	14	13	12	11	10	9	8
	-	-	-	-	-	-	TRG	STR
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

ADCTL

bit	7	6	5	4	3	2	1	0
	ASS3	ASS2	ASS1	ASS0	BUSY	-	INT	INTE
Attribute	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/write allowed

R: Read only

-: Not used

[bit15 to bit10]: Unused bits

The read value is always "0".

[bit9]: TRG

When TRG=1, A/D conversion is triggered by detecting the rising edge of external pin (ATRIG).

If edge is detected during A/D conversion, it is ignored.

Value	Description
0	Disable external pin trigger
1	Enable external pin trigger

[bit8]: STR

This is the A/D conversion start bit.

Value	Description
0	No effect
1	Software activation/reactivation (write during conversion)

The read value of this bit is always "0".

[bit7 to bit4]: ASS3 to ASS0

The selected analog channel can be read.

Data valid when BUSY=1 can be read.

Value	Description
0000 _B to 1011 _B	Selected channel
Others	Not settable

[bit3]: BUSY

This bit indicates that A/D conversion is in progress.

Value	Description
0	Not converting
1	Converting

[bit2]: Unused bit

The read value is always "0".

[bit1]: INT

This is the A/D conversion end bit.

Value	Description
0	No conversion or converting
1	Conversion complete

[bit0]: INTE

This is the A/D conversion interrupt enable bit.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Interrupt request is generated if INTE=1 when INT=1.

23.4.2 Software Conversion Analog Input Select Register (ADCH)

Figure 23.4-2 shows the bit configuration of the software conversion analog input select register (ADCH).

Figure 23.4-2 Bit Configuration of the Software Conversion Analog Input Select Register (ADCH)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	i11	i10	i9	i8	i7	i6	i5	i4	i3	i2	i1	i0
Attribute	R/W															
Initial value	0															
R/W: Read/write allowed																
-: Not used																

[bit15 to bit12]: Unused bits

The read value is always "0".

[bit11 to bit0]: i11 to i0

These are the software conversion analog input select bits.

Value	Description
0	Input not selected
1	Input selected

If multiple inputs are selected, conversion is performed successively for all selected inputs.

23.4.3 A/D Conversion Result Registers ch.0 to ch.11 (ADAT0 to ADAT11)

Figure 23.4-3 shows the bit configuration of the A/D conversion result registers ch.0 to ch.11 (ADAT0 to ADAT11).

Figure 23.4-3 Bit Configuration of the A/D Conversion Result Registers ch.0 to ch.11 (ADAT0 to ADAT11)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	"0"	"0"	"0"	"0"	"0"	"0"	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
Attribute	R															
Initial value	0															
R: Read only																

[bit15 to bit10]: Unused bits

The read value is always "0".

[bit9 to bit0]: d9 to d0

These bits store the A/D conversion result corresponding to each channel.

23.5 Interrupt

An interrupt request is generated when A/D conversion completes.

■ Overview

Table 23.5-1 shows the conditions where interrupt request occurs.

Table 23.5-1 Interrupt Request Generating Conditions

Interrupt request	A/D conversion completion interrupt request
Interrupt request flag	INT=1 in A/D control register (ADCTL)
Enable interrupt request	INTE=1 in A/D control register (ADCTL)
Clear interrupt request	Write "0" in INT bit of A/D control registers (ADCTL)

<Notes>

- When interrupt request flag is "1", an interrupt request occurs at the point the interrupt request is enabled.
Do one of the following to enable generation of interrupt request:
 - Clear interrupt request before enabling interrupt request.
 - Clear interrupt request and enable interrupt request at the same time.
- Clear interrupt request after disabling interrupt request or within the interrupt processing routine.
- See "APPENDIX C Interrupt Vectors" for information on interrupt vector number of each interrupt request.
- The interrupt level corresponding to the interrupt vector number is set in the interrupt control registers (ICR00 to ICR47). See "CHAPTER 12 Interrupt Controller" for the details on setting the interrupt level.

23.6 Explanation of Operations

This section describes the operations of the 10-bit A/D converter.

■ A/D operation by software conversion

In order to perform A/D conversion by software, first select the necessary channels from the twelve analog input pins AN0 to AN11. A channel is enabled by writing "1" in the corresponding bit of the ADCH register.

● In case of single channel

If only one channel is selected as analog input pin for conversion. When you write "1" in the STR bit of the ADCTH register, software conversion is started and "1" is written in the BUSY bit of the ADCTL register.

If you write "1" in the STR bit once more during conversion, the conversion operation is initialized and conversion is restarted.

When A/D conversion completes, the BUSY bit of the ADCTL register is reset to "0" and "1" is set in the INT bit of the ADCTL register. The end of conversion operation can be determined by reading these status bits. Also, if you want to generate a conversion completion interrupt, set "1" in the INTE bit of the ADCTL register.

● In case of multiple channels (scan conversion)

If multiple channels are selected as conversion analog input pin, availability of each channel is checked automatically, channels are switched and A/D conversion is triggered successively, and the conversion result is stored in the register corresponding to each register.

Conversion operation is started by writing "1" in the ADCH register bit corresponding to the channel to be converted and "1" in the STR bit of the ADCTH register. At this time, the BUSY bit of the ADCTL register is set to "1". The conversion channels are selected successively from 0 to 11. The channel not selected with the ADCH register is not converted and conversion proceeds to the next selected channel.

If you write "1" in the STR bit once more during conversion, the conversion operation is initialized and the selected channels are converted once more from channel 0 to 11.

When A/D conversion of all selected channels completes, the BUSY bit of the ADCTL register is reset to "0" and "1" is set in the INT bit of the ADCTL register. Also, if you want to generate a conversion completion interrupt, set "1" in the INTE bit of the ADCTL register.

The A/D conversion result is stored in each channel's register.

■ A/D operation by external trigger conversion

When an external trigger is enabled (ADCTH: TRG=1), conversion is started by detecting the rising edge of the external pin (ATRГ) input. Conversion can be activated by software while external trigger is enabled. Also, if the rising edge of external pin (ATRГ) input is detected once more during A/D conversion, conversion operation continues and the edge is ignored.

<Note>

Be sure A/D conversion is stopped when entering the stop mode of low power consumption mode.

CHAPTER 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 I²C Interface
- 24.20 Overview of I²C Interface
- 24.21 Registers of I²C Interface

24.22 Interrupts of I²C Interface

24.23 Dedicated Baud Rate Generator

24.24 Notes on I²C 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)
- I²C (I²C bus interface)

■ Switching the Interface Mode

To communicate through each serial interface, the serial mode registers (SMR) shown in Table 24.1-1 should be used to set the operation mode before starting the communication.

Table 24.1-1 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 ² C (I ² C 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 other registers will be initialized when the operation mode is changed. Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

■ Number of Channels

This product has 12 built-in channels for multi-function serial interface. There is no I²C 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.8 and ch.11.

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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Framing error Overrun error Parity error^{*2}
9	Interrupt request	<ul style="list-style-type: none"> 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.)

		Function
11	FIFO options	<ul style="list-style-type: none">• 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.8 and ch.11.
*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 24.4-1 Registers of the UART (1 / 5)

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
	FCR10	FIFO control register 10	24.4.7
	FCR00	FIFO control register 00	24.4.8
	FBYTE10	FIFO1 byte register 0	24.4.9
	FBYTE20	FIFO2 byte register 0	24.4.9
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
	FCR11	FIFO control register 11	24.4.7
	FCR01	FIFO control register 01	24.4.8
	FBYTE11	FIFO1 byte register 1	24.4.9
	FBYTE21	FIFO2 byte register 1	24.4.9

Table 24.4-1 Registers of the UART (2 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
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
	FCR12	FIFO control register 12	24.4.7
	FCR02	FIFO control register 02	24.4.8
	FBYTE12	FIFO1 byte register 2	24.4.9
	FBYTE22	FIFO2 byte register 2	24.4.9
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
	FCR13	FIFO control register 13	24.4.7
	FCR03	FIFO control register 03	24.4.8
	FBYTE13	FIFO1 byte register 3	24.4.9
	FBYTE23	FIFO2 byte register 3	24.4.9

Table 24.4-1 Registers of the UART (3 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
4	SCR4	Serial control register 4	24.4.1
	SMR4	Serial mode register 4	24.4.2
	ESCR4	Extended serial control register 4	24.4.4
	BGR4	Baud rate generator register 4	24.4.6
	SSR4	Serial status register 4	24.4.3
	RDR4	Received data register 4	24.4.5
	TDR4	Transmitted data register 4	24.4.5
	FCR14	FIFO control register 14	24.4.7
	FCR04	FIFO control register 04	24.4.8
	FBYTE14	FIFO1 byte register 4	24.4.9
	FBYTE24	FIFO2 byte register 4	24.4.9
5	SCR5	Serial control register 5	24.4.1
	SMR5	Serial mode register 5	24.4.2
	ESCR5	Extended serial control register 5	24.4.4
	BGR5	Baud rate generator register 5	24.4.6
	SSR5	Serial status register 5	24.4.3
	RDR5	Received data register 5	24.4.5
	TDR5	Transmitted data register 5	24.4.5
	FCR15	FIFO control register 15	24.4.7
	FCR05	FIFO control register 05	24.4.8
	FBYTE15	FIFO1 byte register 5	24.4.9
	FBYTE25	FIFO2 byte register 5	24.4.9

Table 24.4-1 Registers of the UART (4 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
6	SCR6	Serial control register 6	24.4.1
	SMR6	Serial mode register 6	24.4.2
	ESCR6	Extended serial control register 6	24.4.4
	BGR6	Baud rate generator register 6	24.4.6
	SSR6	Serial status register 6	24.4.3
	RDR6	Received data register 6	24.4.5
	TDR6	Transmitted data register 6	24.4.5
	FCR16	FIFO control register 16	24.4.7
	FCR06	FIFO control register 06	24.4.8
	FBYTE16	FIFO1 byte register 6	24.4.9
	FBYTE26	FIFO2 byte register 6	24.4.9
7	SCR7	Serial control register 7	24.4.1
	SMR7	Serial mode register 7	24.4.2
	ESCR7	Extended serial control register 7	24.4.4
	BGR7	Baud rate generator register 7	24.4.6
	SSR7	Serial status register 7	24.4.3
	RDR7	Received data register 7	24.4.5
	TDR7	Transmitted data register 7	24.4.5
	FCR17	FIFO control register 17	24.4.7
	FCR07	FIFO control register 07	24.4.8
	FBYTE17	FIFO1 byte register 7	24.4.9
	FBYTE27	FIFO2 byte register 7	24.4.9
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

Table 24.4-1 Registers of the UART (5 / 5)

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
10	SCRA	Serial control register A	24.4.1
	SMRA	Serial mode register A	24.4.2
	ESCRA	Extended serial control register A	24.4.4
	BGRA	Baud rate generator register A	24.4.6
	SSRA	Serial status register A	24.4.3
	RDRA	Received data register A	24.4.5
	TDRA	Transmitted data register A	24.4.5
11	SCRB	Serial control register B	24.4.1
	SMRB	Serial mode register B	24.4.2
	ESCRB	Extended serial control register B	24.4.4
	BGRB	Baud rate generator register B	24.4.6
	SSRB	Serial status register B	24.4.3
	RDRB	Received data register B	24.4.5
	TDRB	Transmitted data register B	24.4.5

Table 24.4-2 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 24.4-3 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 24.4-1 shows the bit structure of the serial control register (SCR), and Table 24.4-4 describes the function of each bit.

Figure 24.4-1 Bit Structure of Serial Control Register (SCR)

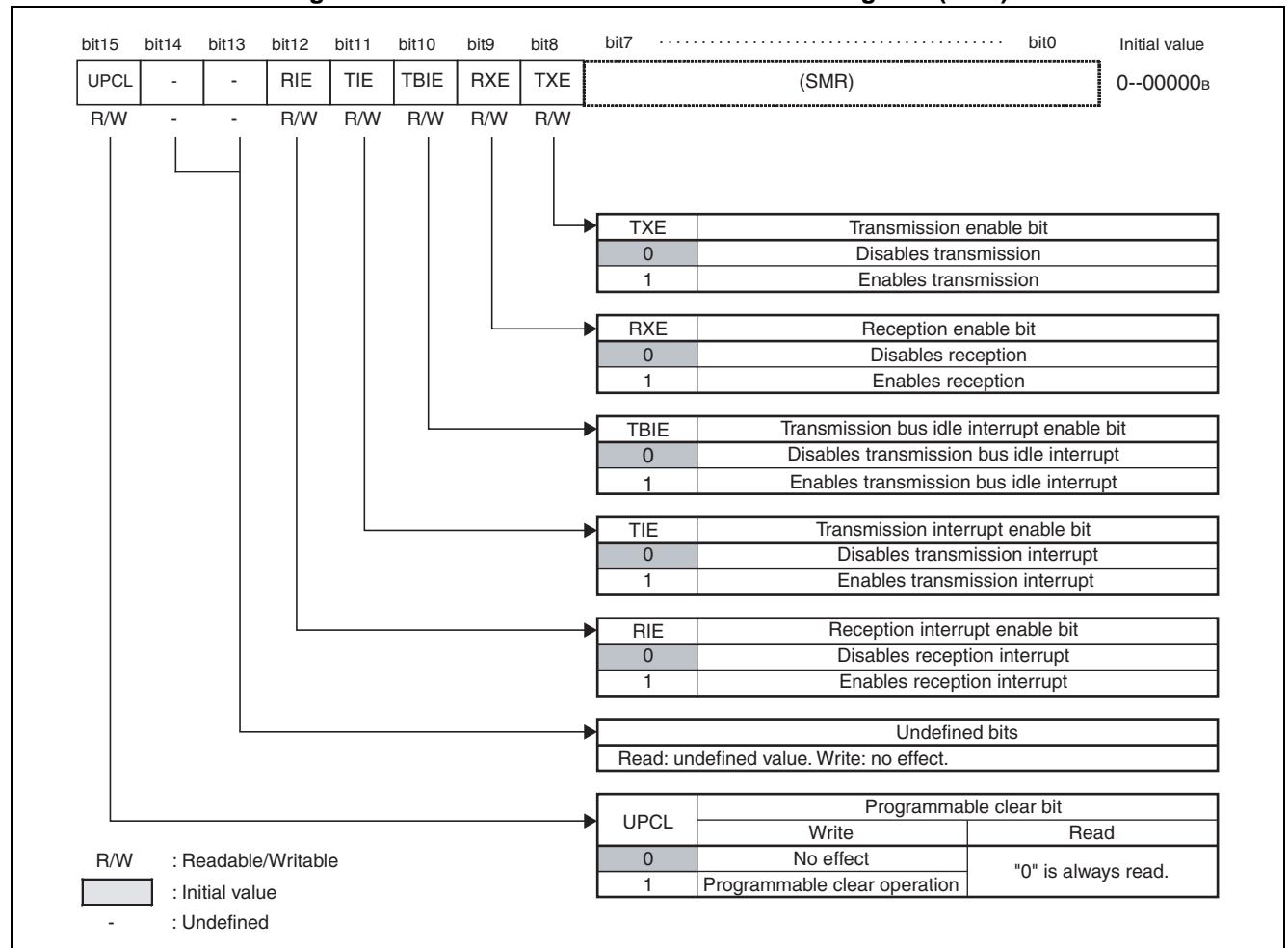


Table 24.4-4 Functional Description of Each Bit of Serial Control Register (SCR)

Bit name		Function
bit15	UPCL: Programmable clear bit	<p>This bit is used to initialize the internal state of the UART.</p> <p>Setting the bit to "1":</p> <ul style="list-style-type: none"> 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). <p>Setting the bit to "0": No effect on the operation</p> <p>Reading this bit always returns "0".</p> <p>Note:</p> <p>Execute the programmable clear operation after disabling interrupts.</p> <p>Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.</p>
bit14, bit13	Undefined bits	<p>Read: undefined value</p> <p>Write: no effect</p>
bit12	RIE: Reception interrupt enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<p>This bit is used to enable/disable UART reception operation.</p> <ul style="list-style-type: none"> Setting the bit to "0" disables the reception operation. Setting the bit to "1" enables the reception operation. <p>Note:</p> <p>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.)</p> <p>If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.</p>
bit8	TXE: Transmission enable bit	<p>This bit is used to enable/disable UART transmission operation.</p> <ul style="list-style-type: none"> Setting the bit to "0" disables the transmission operation. Setting the bit to "1" enables the transmission operation. <p>Note:</p> <p>If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.</p>

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 24.4-2 shows the bit structure of the serial mode register (SMR), and Table 24.4-5 describes the function of each bit.

Figure 24.4-2 Bit Structure of Serial Mode Register (SMR)

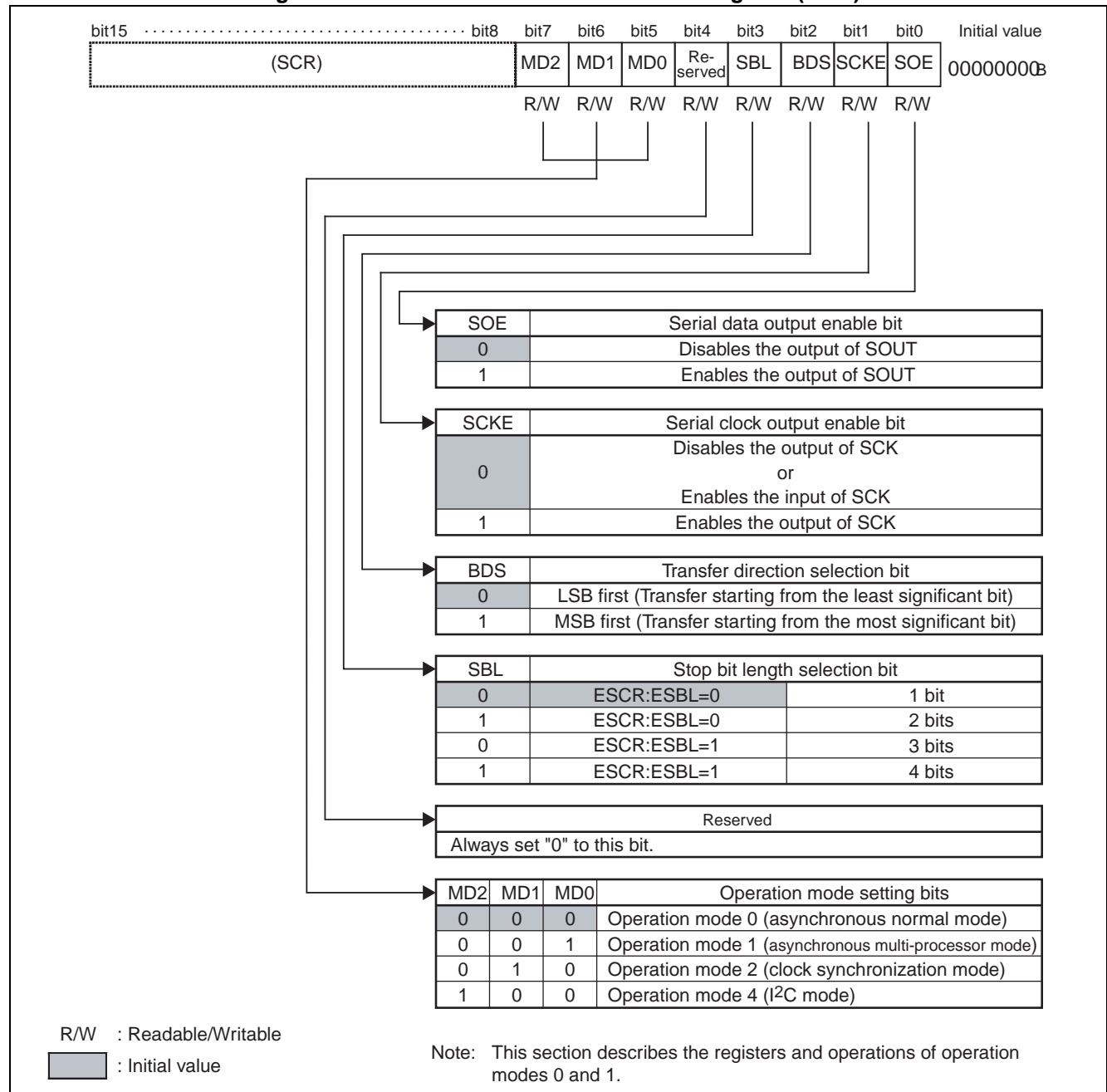


Table 24.4-5 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2, MD1, MD0: Operation mode setting bits	<p>These bits set the operation mode for the asynchronous serial interface.</p> <p>"000_B": Selects operation mode 0 (asynchronous normal mode)</p> <p>"001_B": Selects operation mode 1 (asynchronous multi-processor mode)</p> <p>"010_B": Selects operation mode 2 (clock synchronization mode)</p> <p>"100_B": Selects operation mode 4 (I²C mode)</p> <p>This section describes the registers and operations of operation mode 0 (asynchronous normal mode) and operation mode 1 (asynchronous multi-processor mode).</p> <p>Note:</p> <p>Settings other than above are prohibited.</p> <p>To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1). And then, after setting the operation mode, set each register.</p>
bit4	Reserved bit	Always set "0" to this bit.
bit3	SBL: Stop bit length selection bit	<p>This bit is used to select a bit length for a stop bit (frame end mark of transmission data).</p> <p>Setting the bit to SBL=0, ESCR:ESBL=0 sets the stop bit to 1 bit in length.</p> <p>Setting the bit to SBL=1, ESCR:ESBL=0 sets the stop bit to 2 bits in length.</p> <p>Setting the bit to SBL=0, ESCR:ESBL=1 sets the stop bit to 3 bits in length.</p> <p>Setting the bit to SBL=1, ESCR:ESBL=1 sets the stop bit to 4 bits in length.</p> <p>Note:</p> <ul style="list-style-type: none"> 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	<p>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).</p> <p>Note:</p> <p>Set this bit when transmission and reception are disabled (TXE = RXE = 0).</p>
bit1	SCKE: Serial clock output enable bit	<p>This bit is used to control the I/O port of the serial clock.</p> <p>Setting the bit to "0":</p> <p>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.</p> <p>Setting the bit to "1" enables the output of SCK.</p>
bit0	SOE: Serial data output enable bit	<p>This bit is used to enable/disable the output of serial data.</p> <p>Setting the bit to "0" disables the output.</p> <p>Setting the bit to "1" enables the output of SOUT.</p>

<Note>

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. 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 24.4-3 shows the bit structure of the serial status register (SSR) and Table 24.4-6 describes the function of each bit.

Figure 24.4-3 Bit Structure of Serial Status Register (SSR)

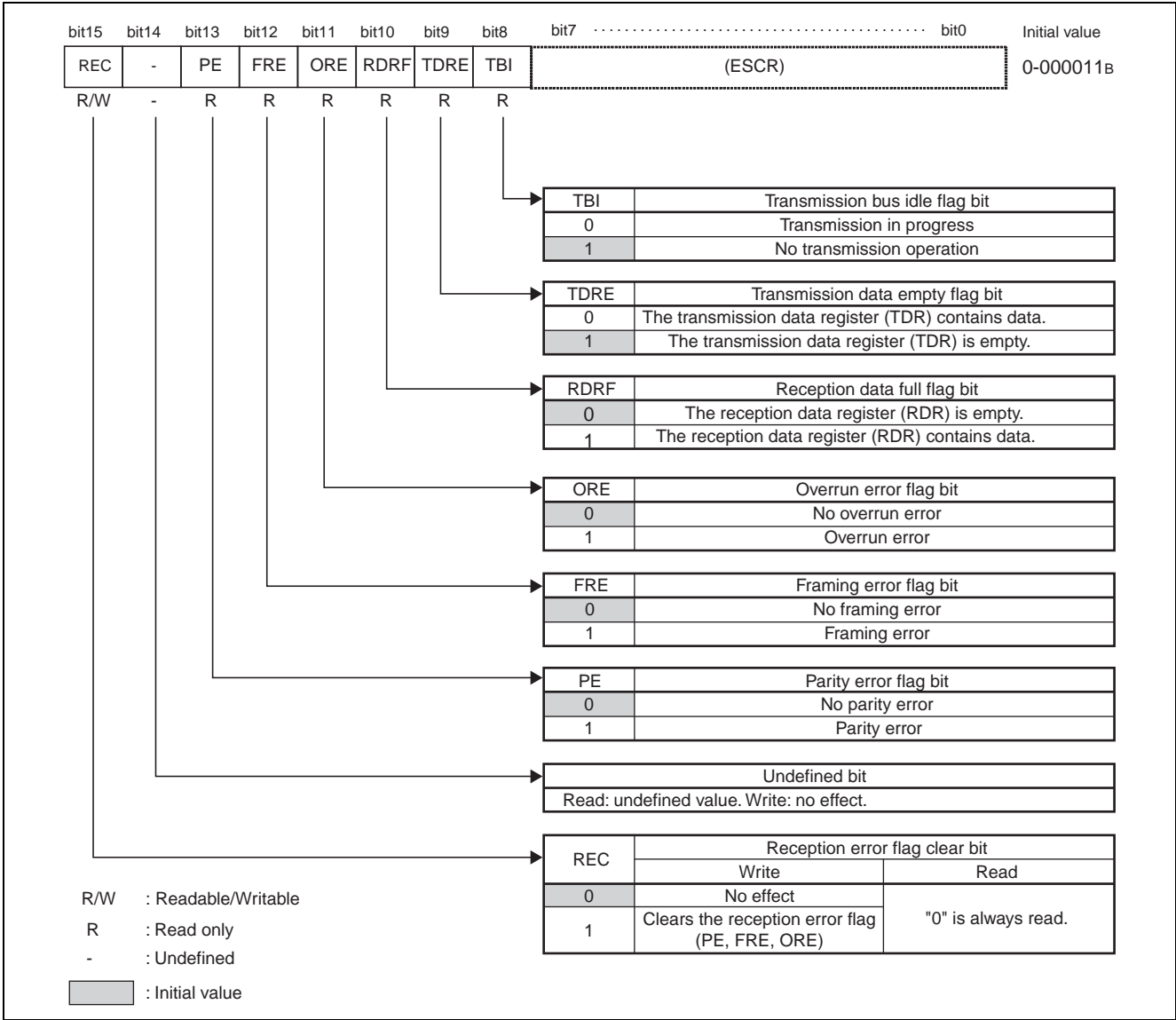


Table 24.4-6 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	<p>This bit is used to clear the PE, FRE and ORE flag in the serial status register (SSR).</p> <ul style="list-style-type: none"> Writing "1" clears the error flag. Writing "0" has no effect. <p>Reading this bit always returns "0".</p>
bit14	Undefined bit	<p>Read: undefined value</p> <p>Write: no effect</p>
bit13	PE: Parity error flag bit (only available in operation mode 0)	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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.

Table 24.4-6 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit10	RDRF: Reception data full flag bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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 24.4-4 shows the bit structure of the extended serial control register (ESCR) and Table 24.4-7 describes the function of each bit.

Figure 24.4-4 Bit Structure of Extended Serial Control Register (ESCR)

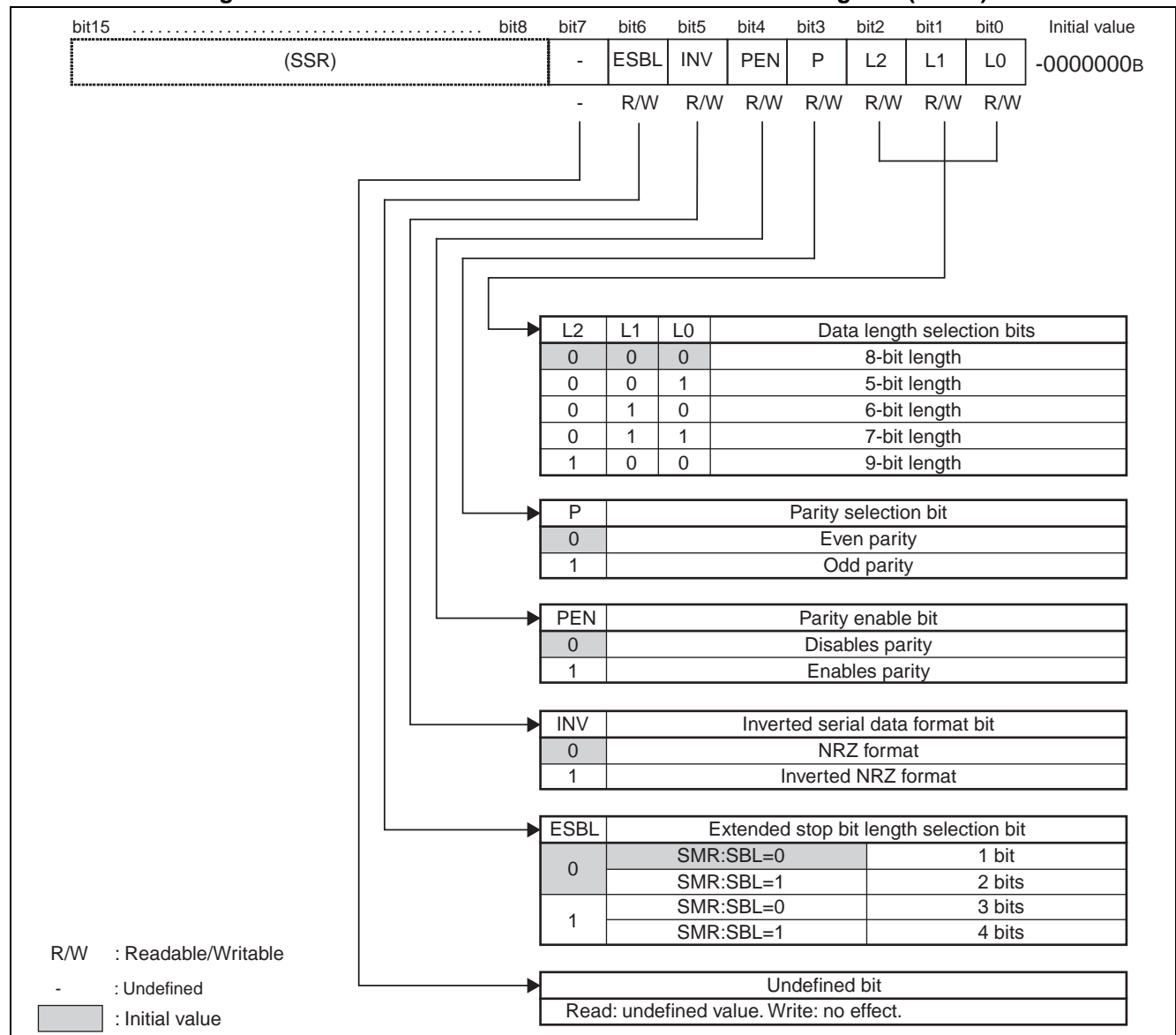


Table 24.4-7 Functional Description of Each Bit of Extended Serial Control Register (ESCR)

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: <ul style="list-style-type: none"> 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). <ul style="list-style-type: none"> 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). <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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)

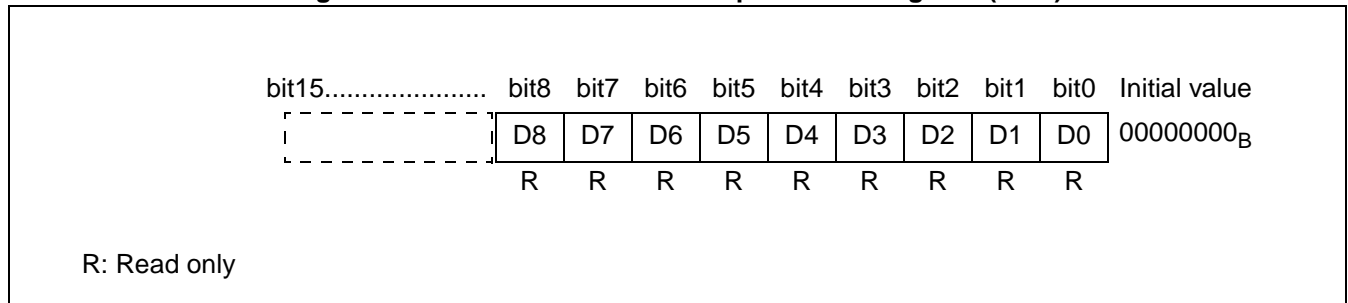
The reception data register and transmission data register are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

When FIFO operation is enabled, the RDR/TDR address becomes the read/write address for the FIFO.

■ Reception Data Register (RDR)

Figure 24.4-5 illustrates the bit structure of the serial reception register (RDR).

Figure 24.4-5 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 indicates the 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.

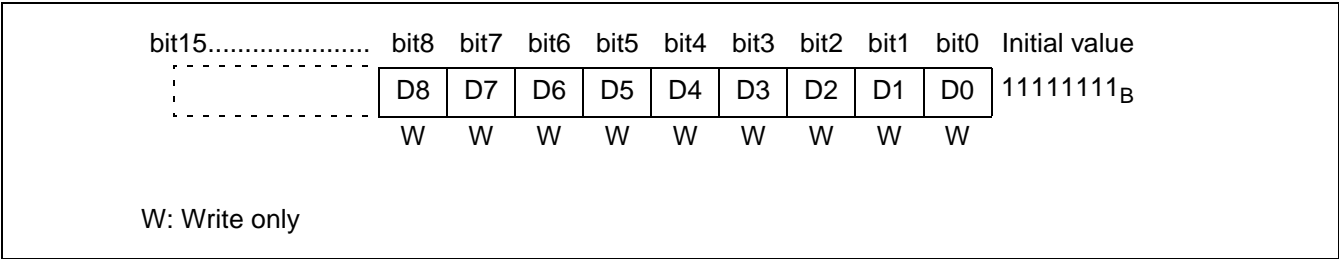
<Notes>

- 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 24.4-6 illustrates the bit structure of the transmission data register.

Figure 24.4-6 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 indicates the 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".
-

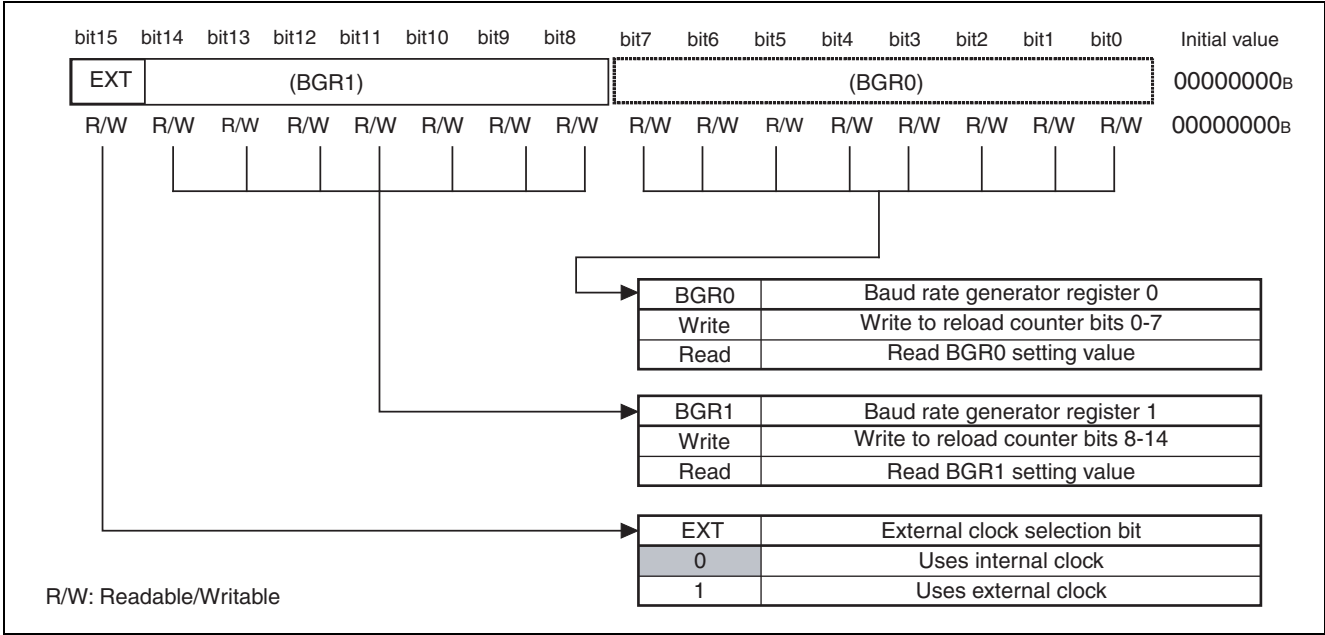
24.4.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

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 24.4-7 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 24.4-7 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 24.4-8 shows the bit structure of the FIFO control register 1 (FCR1) and Table 24.4-8 describes the function of each bit.

Figure 24.4-8 Bit Structure of FIFO Control Register 1 (FCR1)

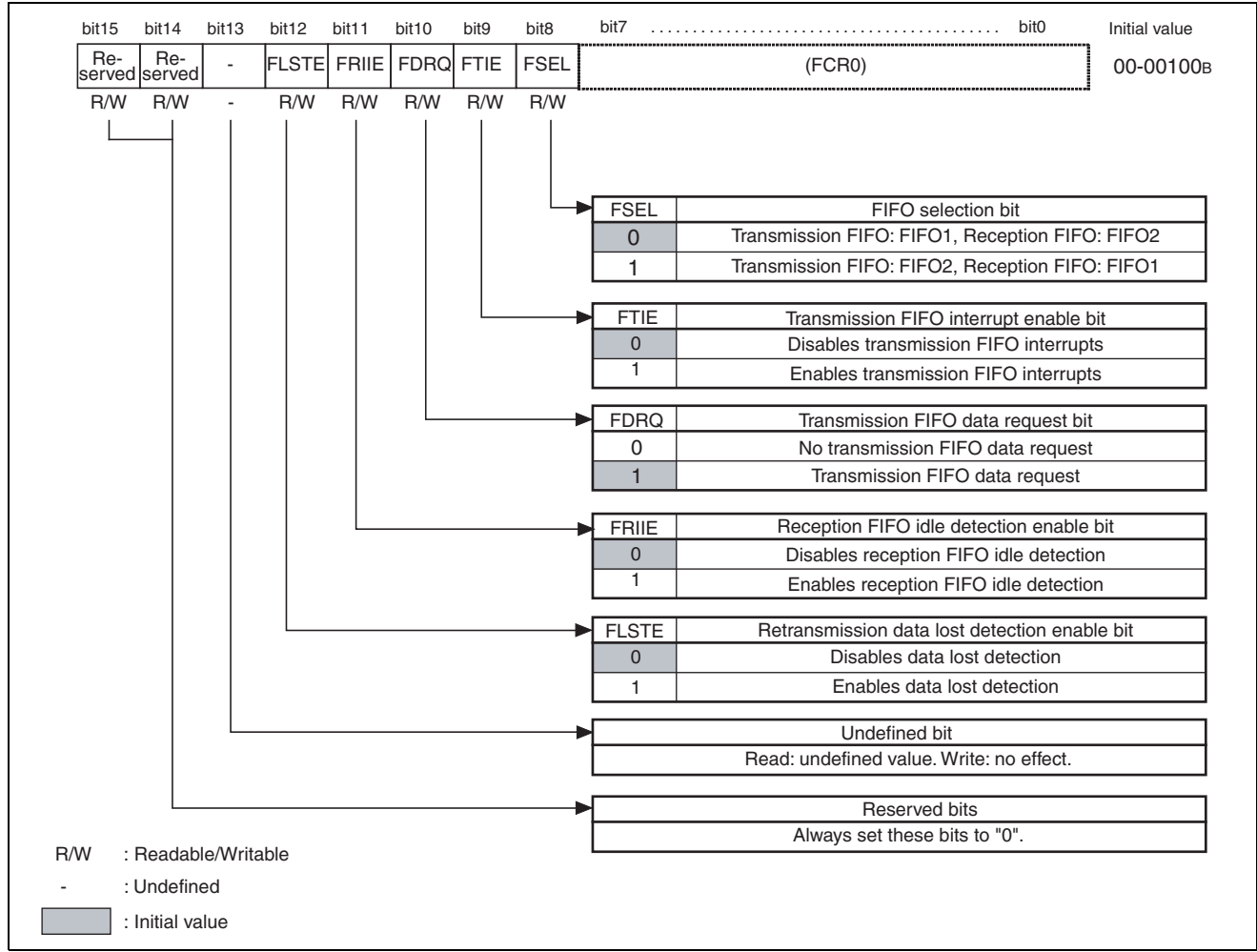


Table 24.4-8 Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

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 <ul style="list-style-type: none"> 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 24.4-9 shows the bit structure of the FIFO control register 0 (FCR0) and Table 24.4-9 describes the function of each bit.

Figure 24.4-9 Bit Structure of FIFO Control Register 0 (FCR0)

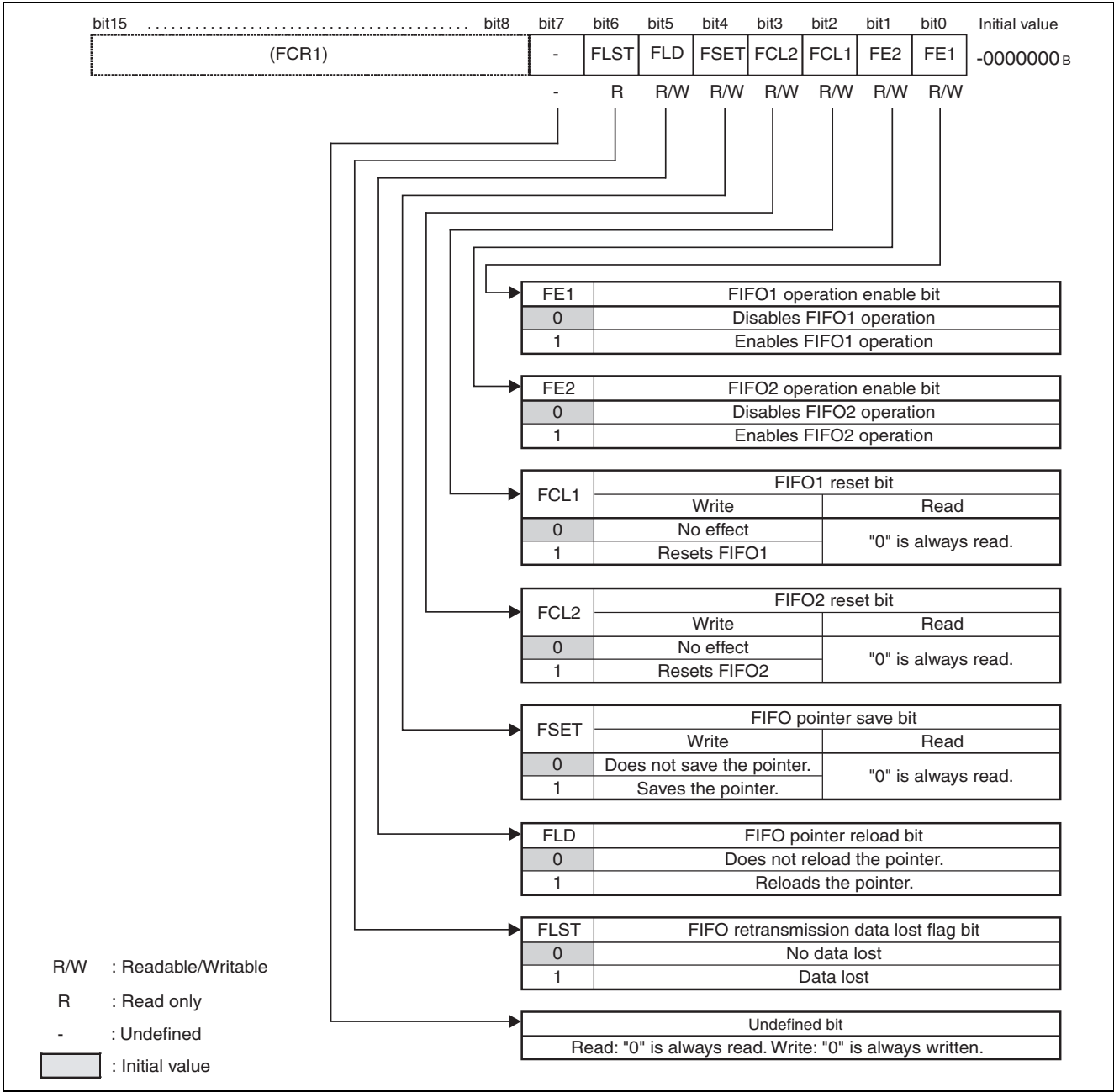


Table 24.4-9 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 2)

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 (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 <ul style="list-style-type: none"> 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".

Table 24.4-9 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit2	FCL1: FIFO1 reset bit	<p>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.</p> <p>Note:</p> <ul style="list-style-type: none"> 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	<p>This bit is used to enable/disable FIFO2 operation.</p> <ul style="list-style-type: none"> 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	<p>This bit is used to enable/disable FIFO1 operation.</p> <ul style="list-style-type: none"> 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.

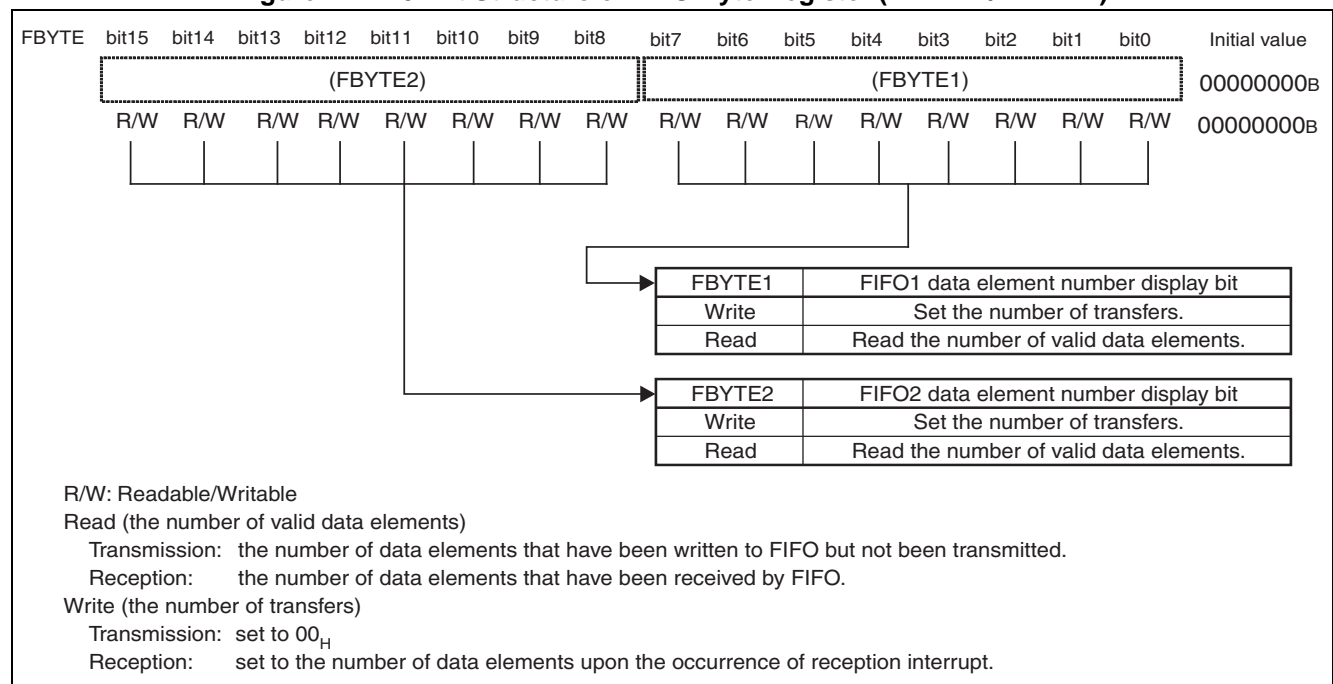
24.4.9 FIFO Byte Register (FBYTE1/FBYTE2)

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 24.4-10 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 24.4-10 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 24.4-10 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 24.5-1 shows the interrupt control bits and interrupt sources of the UART.

Table 24.5-1 Interrupt Control Bits and Interrupt Sources of UART (1 / 2)

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		

Table 24.5-1 Interrupt Control Bits and Interrupt Sources of UART (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
			0	1			
Trans- mission	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)*
	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)*
	FDRQ	FCR1	○	○	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

*: 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 24.5-1 Timing for Setting RDRF (Reception Data Full) Flag Bit

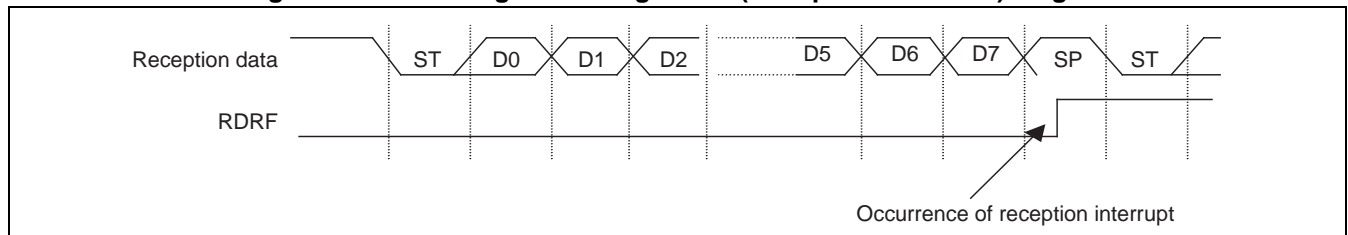


Figure 24.5-2 Timing for Setting FRE (Framing Error) Flag Bit

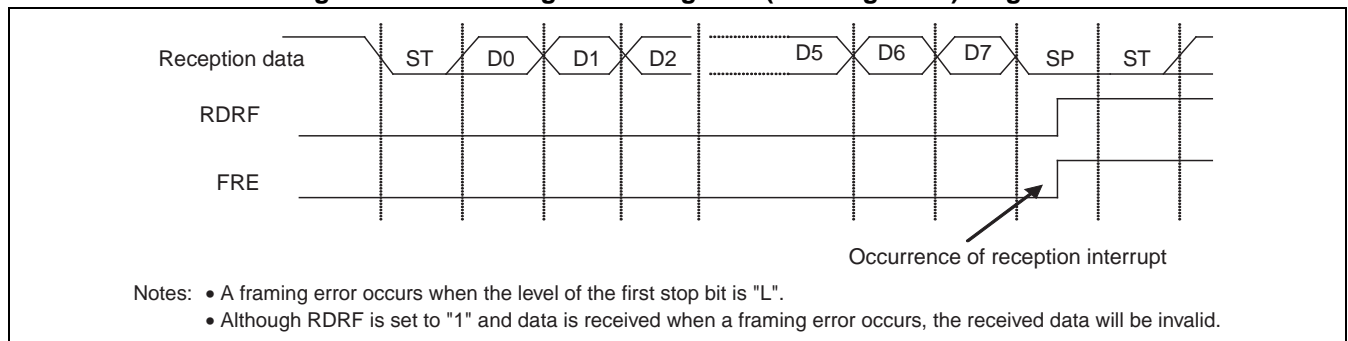
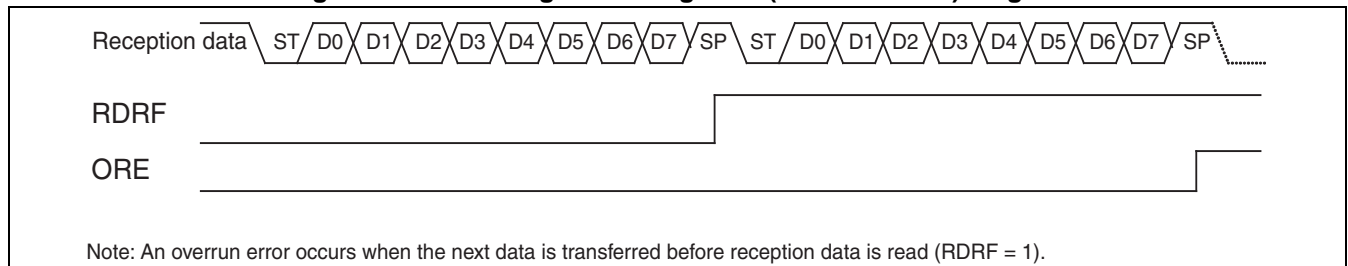


Figure 24.5-3 Timing for Setting ORE (Overrun Error) Flag Bit



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 24.5-4 Timing for Generating Reception Interrupt when Reception FIFO is Used

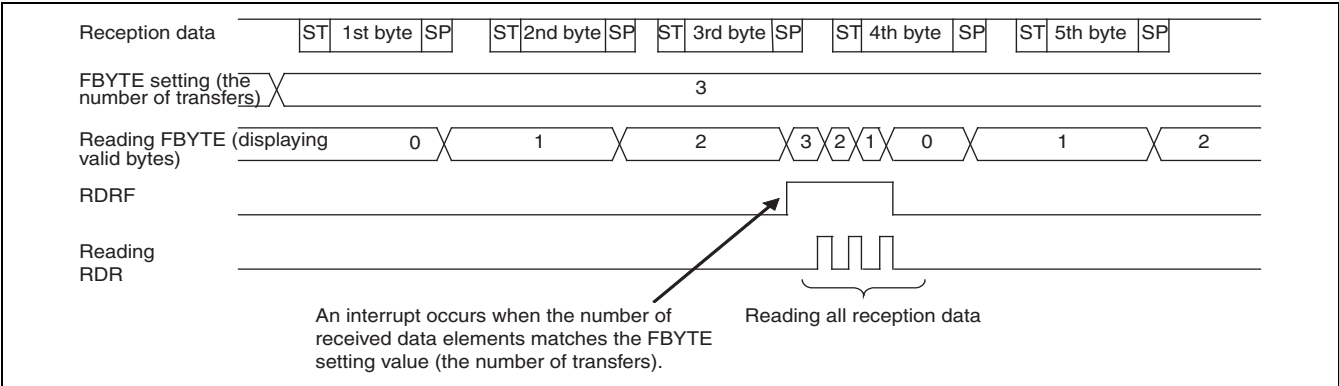
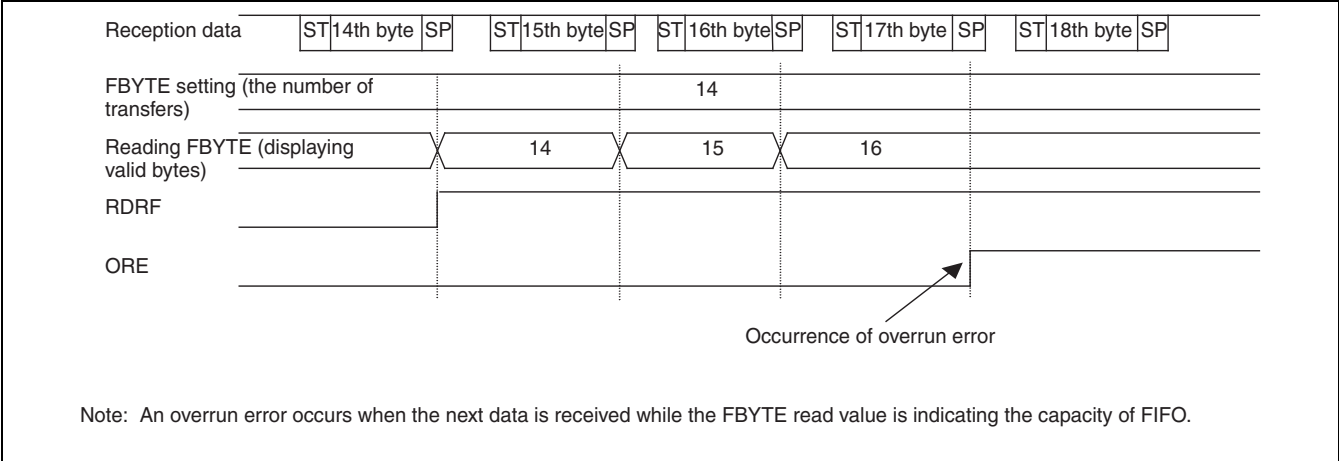


Figure 24.5-5 Timing for Setting ORE (Overrun Error) Flag Bit



24.5.3 Occurrence of Transmission Interrupts and Flag Set Timing

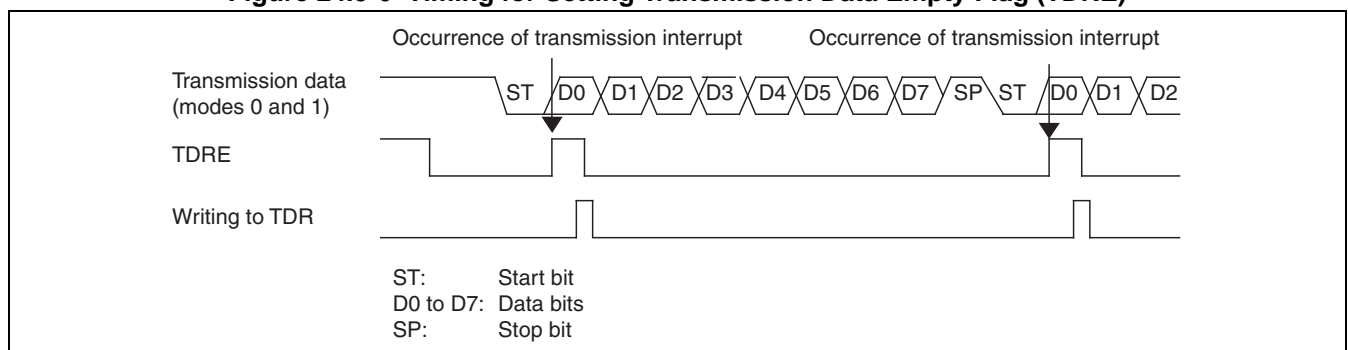
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

■ 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).

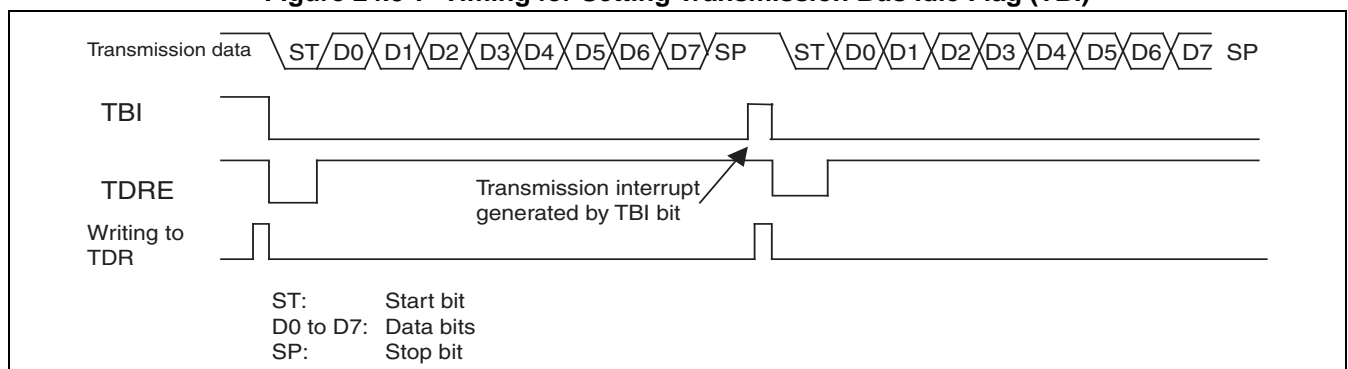
Figure 24.5-6 Timing for Setting Transmission Data Empty Flag (TDRE)



● 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 24.5-7 Timing for Setting Transmission Bus Idle Flag (TBI)



24.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

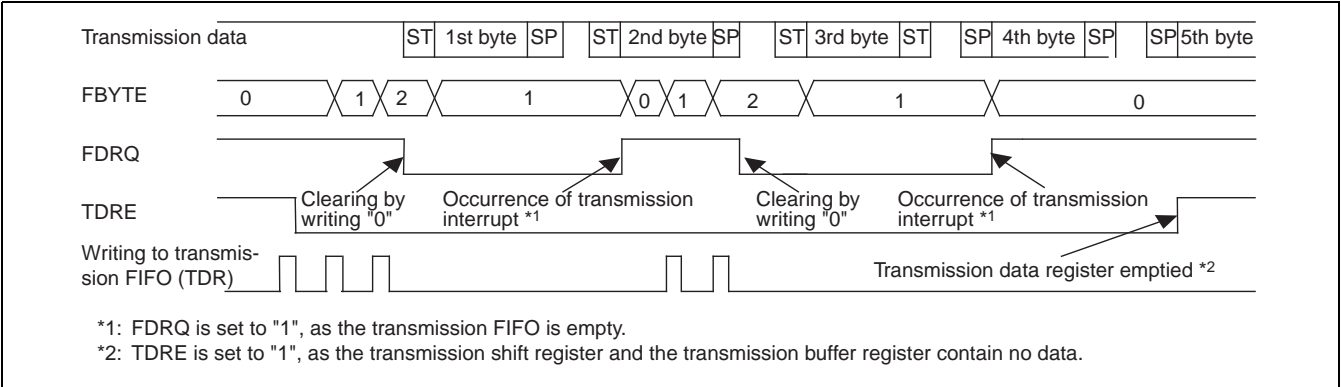
When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

■ 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 24.5-8 Occurrence Timing for Transmission Interrupts when Transmission FIFO is Used



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.

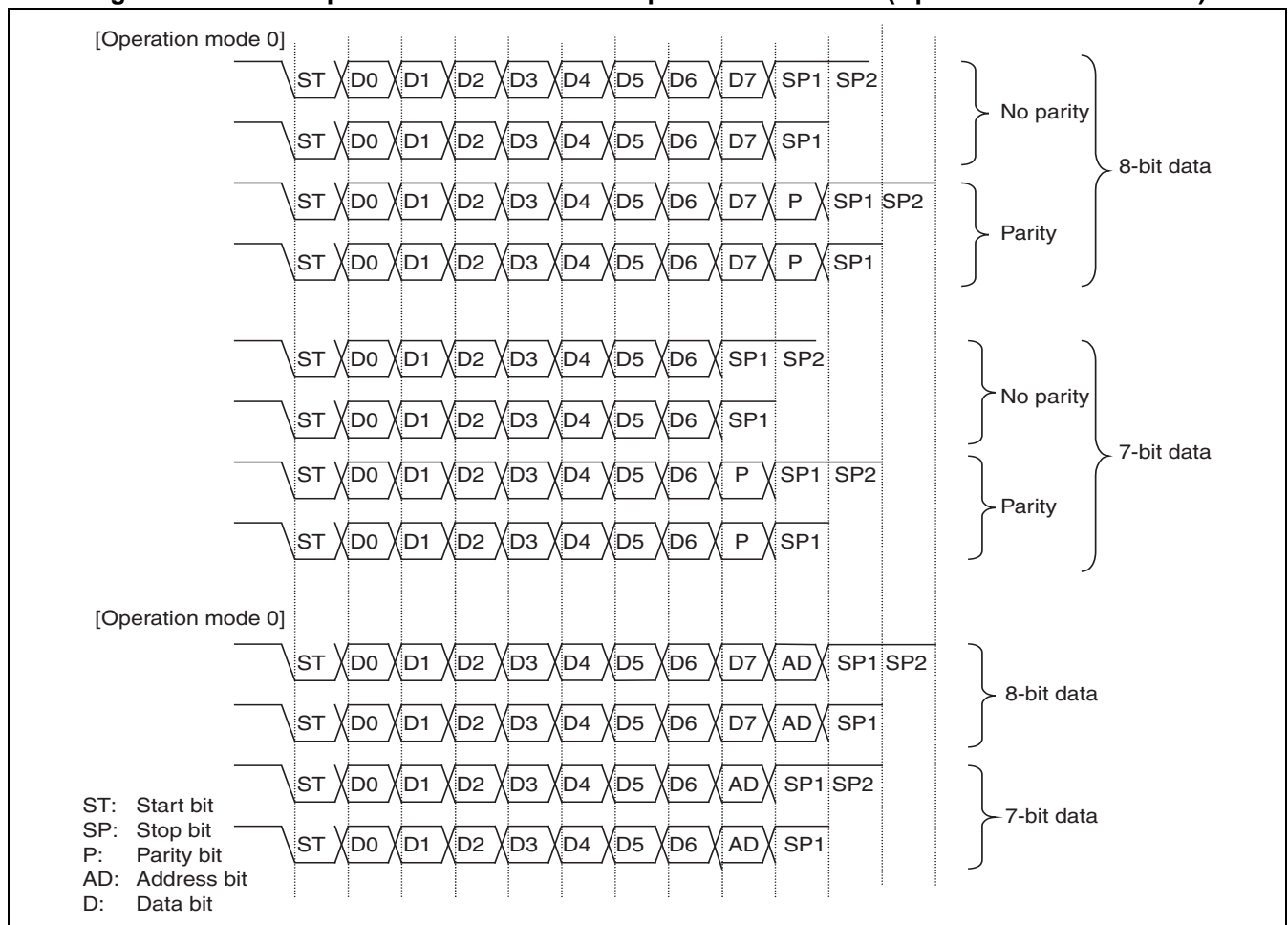
■ Operation of UART

● 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 24.6-1 shows transmission/reception data formats for operation modes 0 and 1.

Figure 24.6-1 Examples of Transmission/Reception Data Formats (Operation Modes 0 and 1)



<Notes>

- Figure 24.6-1 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>

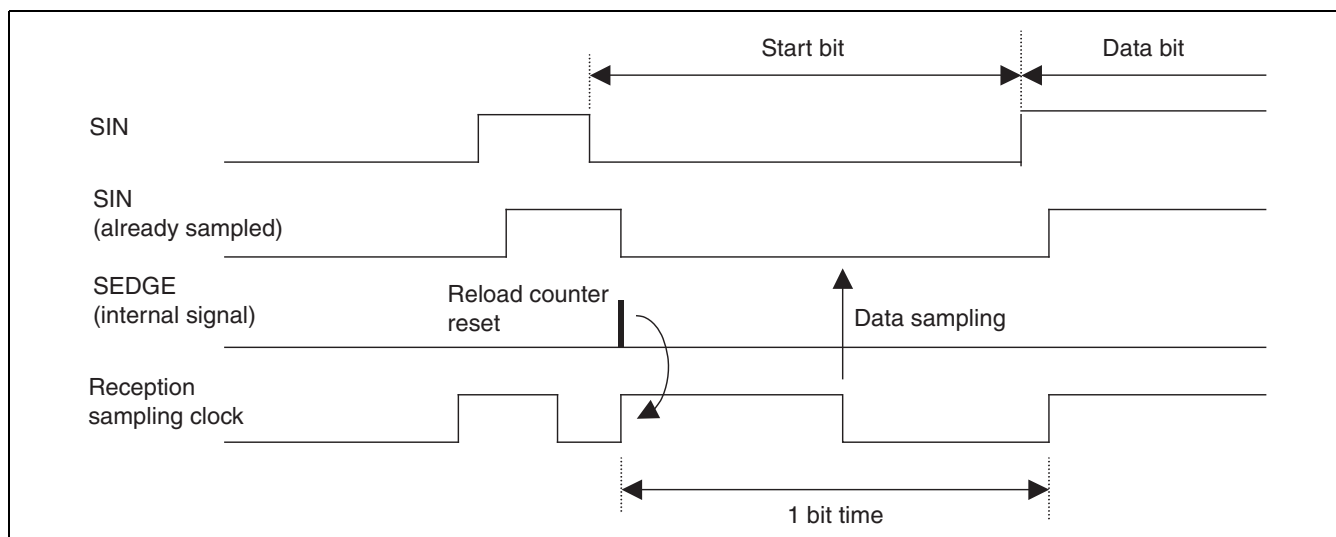
The data in the reception data register (RDR) will become valid, if no reception error occurs (SSR:PE, ORE, FRE = 0) when the reception data register full flag bit (SSR:RDRF) is set to "1".

● 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

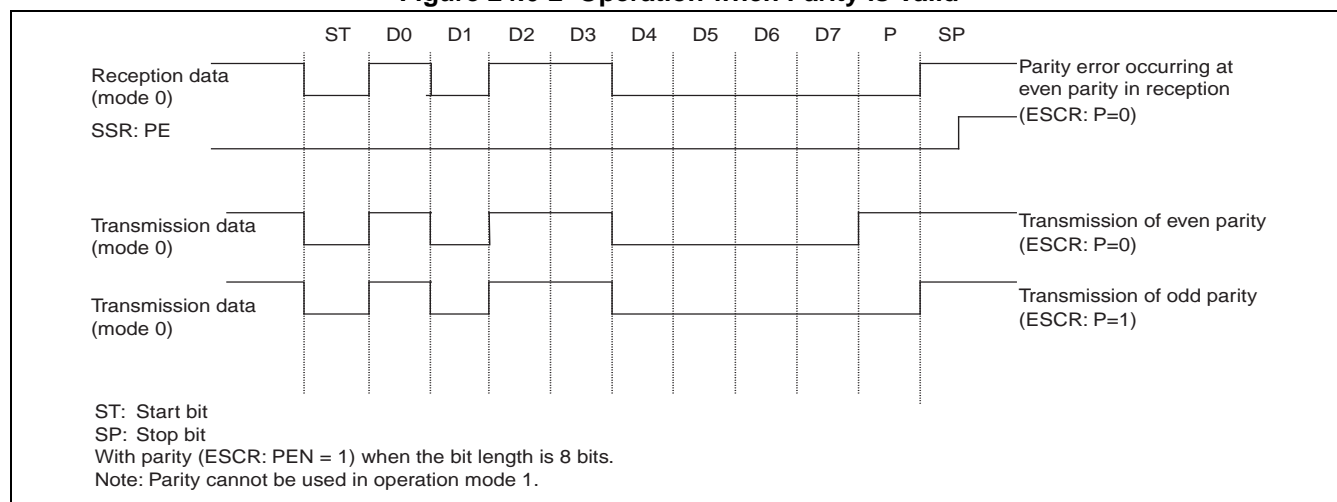
- In operation mode 0, parity errors, overrun errors and frame errors can be detected.
- In operation mode 1, overrun errors and frame errors can be detected. Parity errors, on the other hand, cannot be detected.

● 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 24.6-2 shows transmission/reception data when parity is valid.

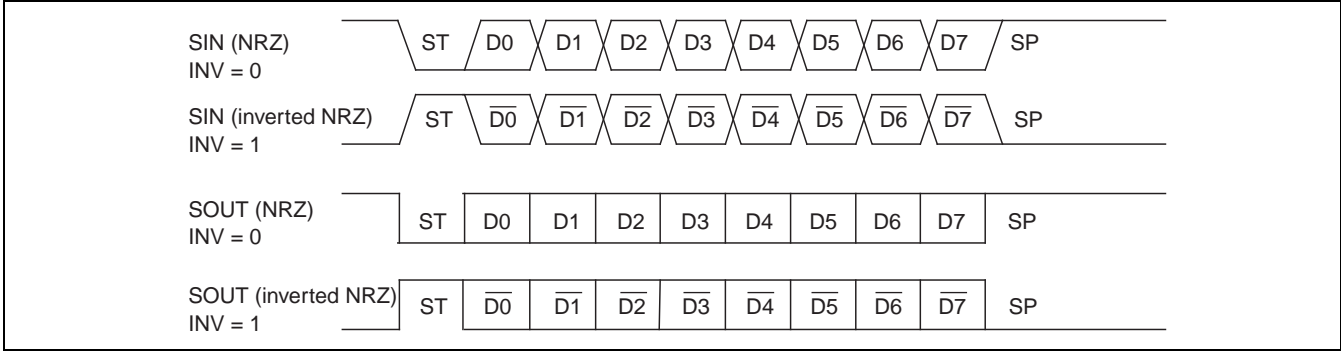
Figure 24.6-2 Operation when Parity is Valid



● 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 24.6-3 shows the NRZ and inverted NRZ signaling systems.

Figure 24.6-3 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.

(1) Reload value:

$$V = \phi / b - 1$$

V: Reload value

b: Baud rate

ϕ : Peripheral clock (PCLK), external clock frequency

(2) 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:

$$b = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

(3) Baud rate error

The following formula is used to calculate a baud rate error.

$$\text{Error}(\%) = (\text{calculated value} - \text{target value}) / \text{target value} \times 100$$

Example: peripheral clock (PCLK) = 20MHz, target baud rate = 153600bps

$$\text{Reload value} = (20 \times 1000000) / 153600 - 1 = 129$$

$$\text{Baud rate (calculated value)} = (20 \times 1000000) / (129 + 1) = 153846 \text{ (bps)}$$

$$\text{Error}(\%) = (153846 - 153600) / 153600 \times 100 = 0.16 \text{ (\%)}$$

<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 24.7-1 Reload Values and Baud Rates

Baudrate (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	-	-	-	-	-	-	-	-	-	-

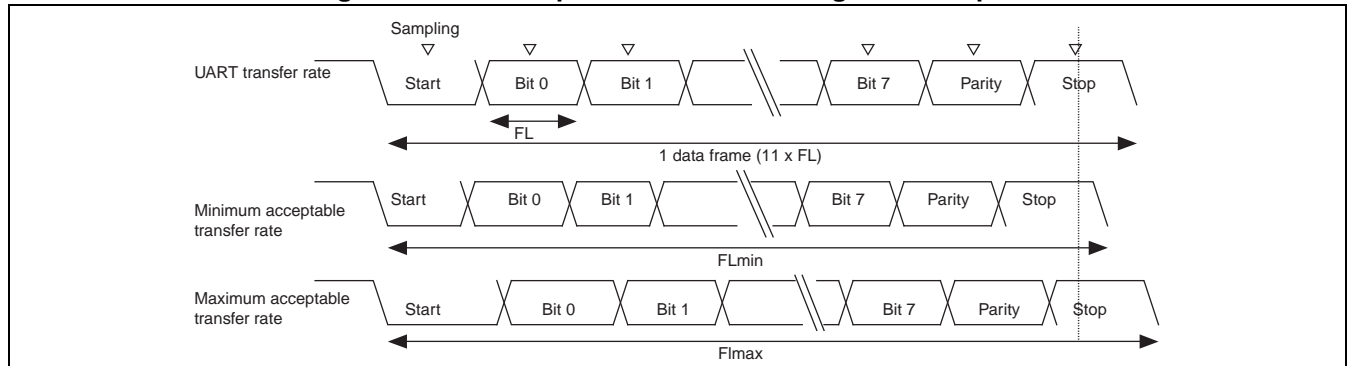
- Value: the value set in BGR1/BGR0 registers (decimal)
- ERR: baud rate error (%)

■ 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 24.7-1 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:

$$FLmin = (11 \text{ bits} \times (V + 1) - (V + 1)/2 + 2)/\phi = (21V + 25)/2\phi (s)$$

V: reload value ϕ : peripheral clock (PCLK)

Consequently, the maximum receivable baud rate at the transmission destination (BGmax) is as follows:

$$BGmax = 11/FLmin = 22\phi/(21V+25) \text{ (bps)}$$

V: reload value ϕ : peripheral clock (PCLK)

Likewise, the maximum acceptable transfer rate (FLmax) can be calculated as shown below:

$$FLmax = (11 \text{ bits} \times (V + 1) + (V + 1)/2 - 2)/\phi = (23V + 19)/2\phi (s)$$

V: reload value ϕ : peripheral clock (PCLK)

Consequently, the minimum receivable baud rate at the transmission destination (BGmin) is as follows:

$$BGmin = 11/FLmax = 22\phi/(23V+19) \text{ (bps)}$$

V: reload value ϕ : 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 24.7-2 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

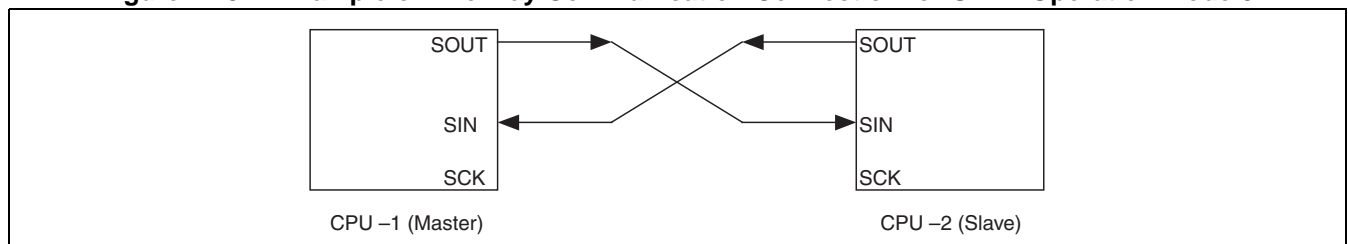
24.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)

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 24.8-1.

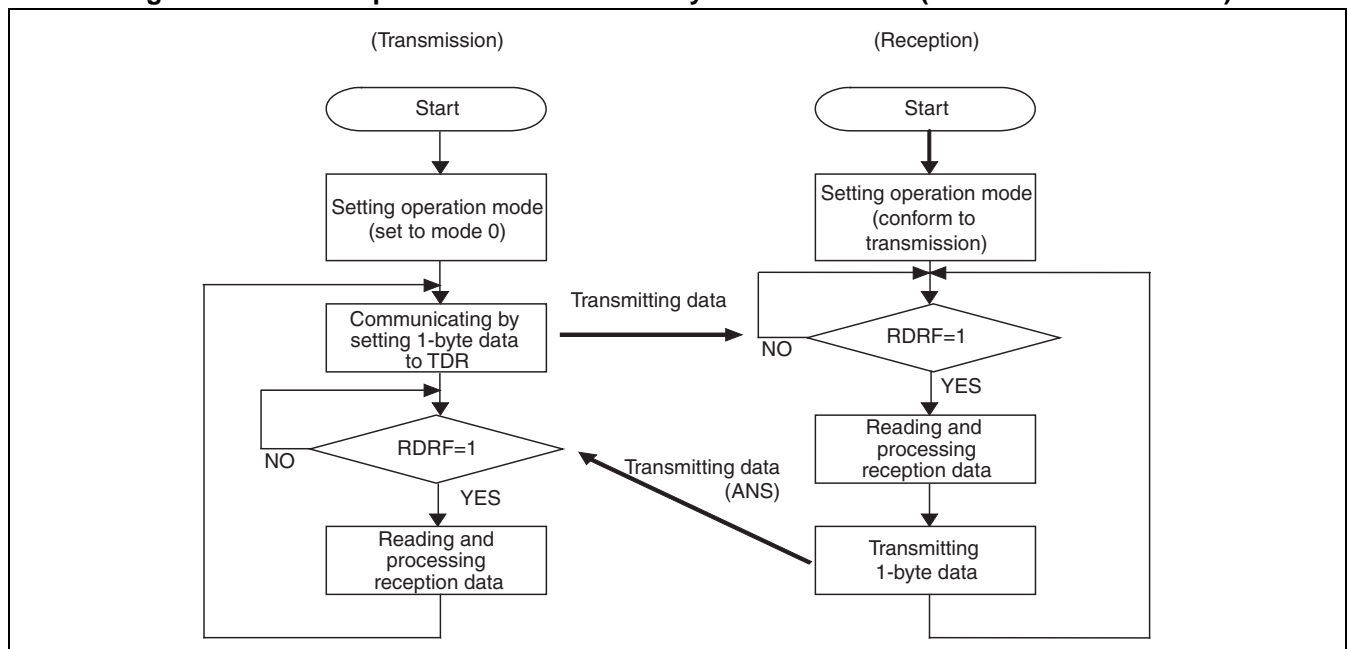
Figure 24.8-1 Example of Two-way Communication Connection for UART Operation Mode 0



■ Flowchart

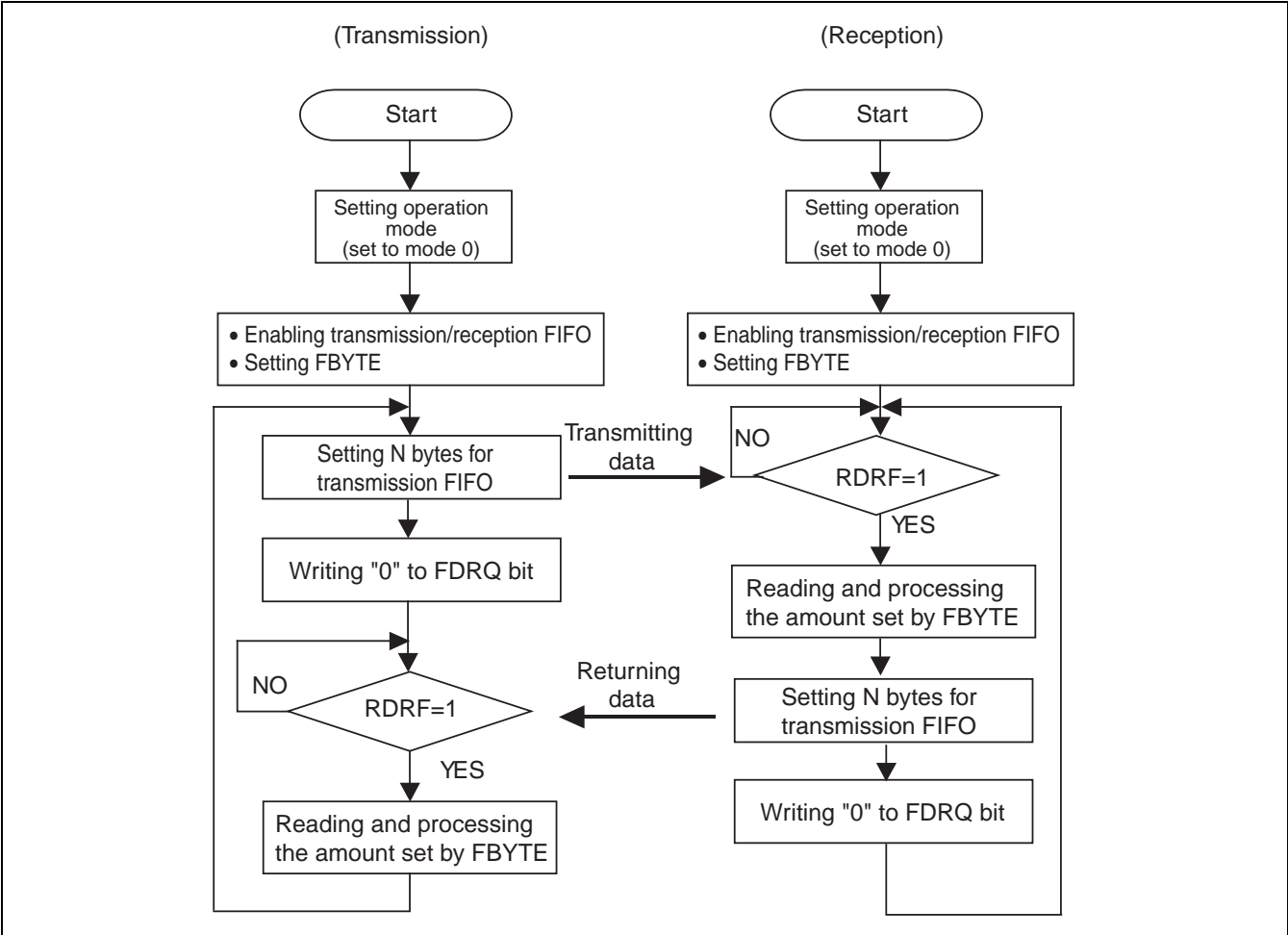
● When FIFO is not used

Figure 24.8-2 Example Flowchart for Two-way Communication (When FIFO is Not Used)



● When FIFO is used

Figure 24.8-3 Example Flowchart for Two-way Communication (When FIFO is Used)



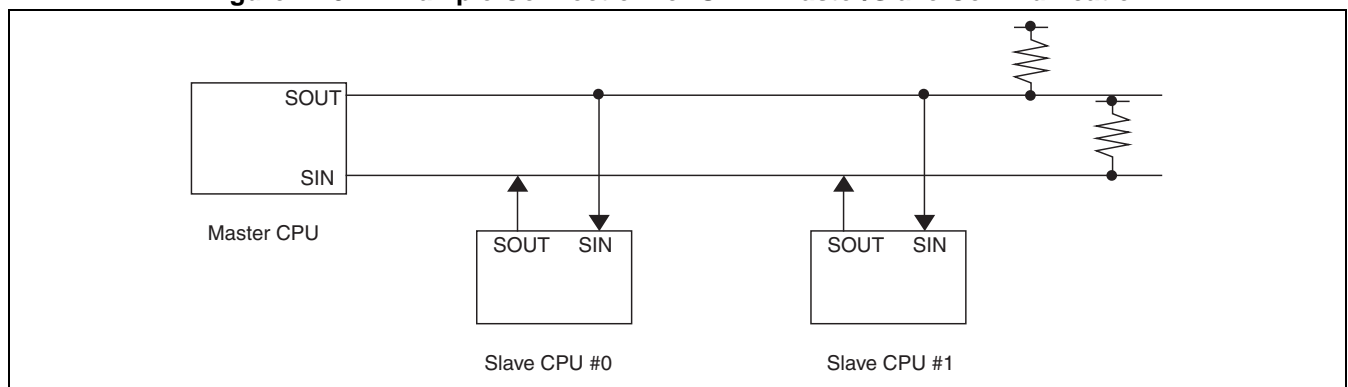
24.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

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 24.9-1, to configure a communication system. The UART can be used by both the master and slaves.

Figure 24.9-1 Example Connection for UART Master/Slave Communication



■ Function Selection

For master/slave communication, select the operation mode and data transfer system shown in Table 24.9-1.

Table 24.9-1 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 24.9-2 and Figure 24.9-3 show flowcharts for the master/slave communication (multi-processor mode).

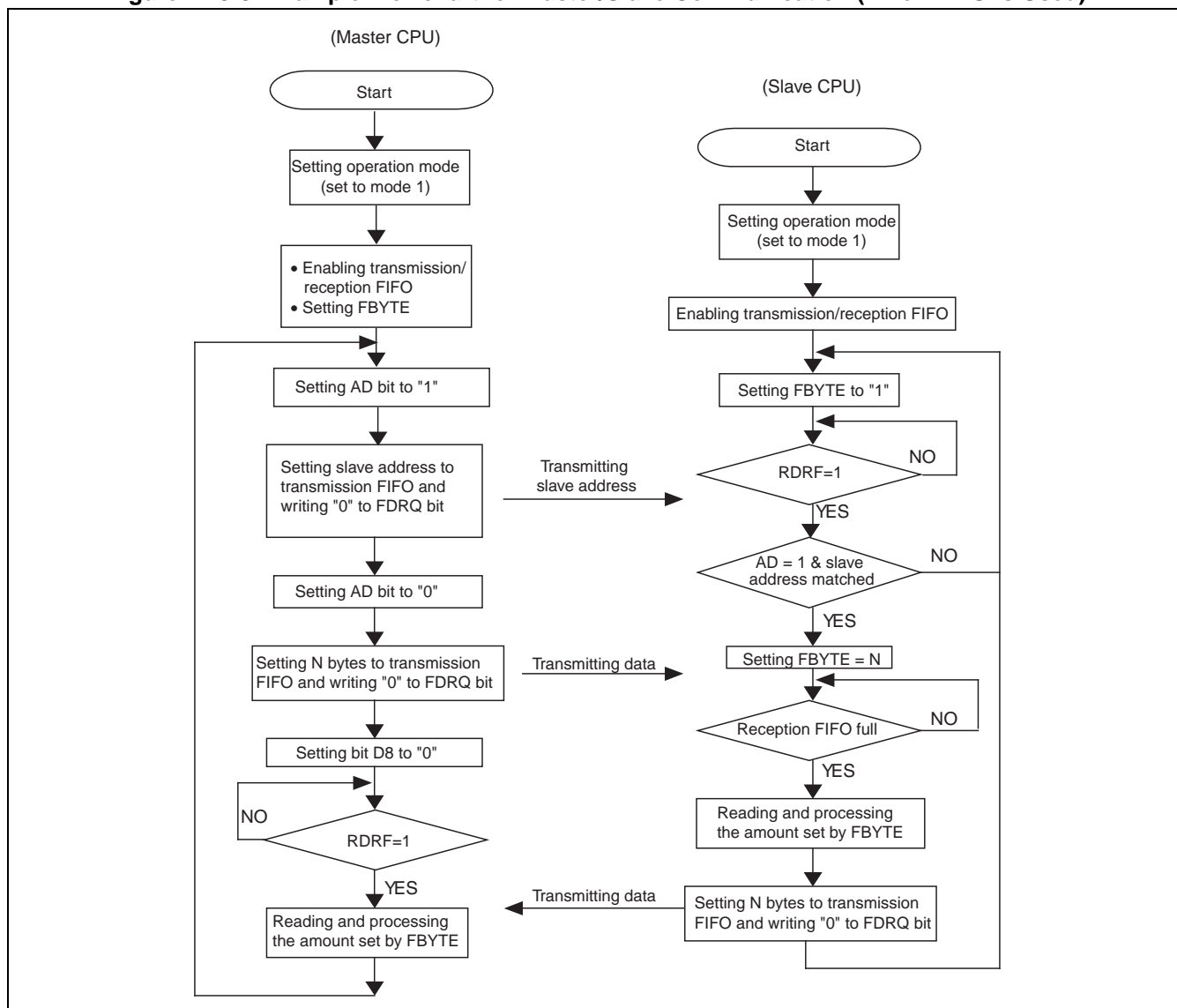
● **When FIFO is not used:**

```
graph TD
    subgraph Master_CPU [Master CPU]
        M_Start([Start]) --> M_SetMode[Setting operation mode  
(set to mode 1)]
        M_SetMode --> M_SetPins[Setting SIN pin to serial  
data input  
Setting SOUT pin to serial  
data output]
        M_SetPins --> M_SelectBits[Selecting 7 or 8 data bits  
Selecting 1 or 2 stop bits]
        M_SelectBits --> M_SetD8_1[Setting bit D8 to "1"]
        M_SetD8_1 --> M_EnableTxRx[Enabling transmission/  
reception operation]
        M_EnableTxRx --> M_TransmitAddr[Transmitting slave address]
        M_TransmitAddr --> M_SetD8_0[Setting bit D8 to "0"]
        M_SetD8_0 --> M_CommSlave[Communicating with  
slave CPU]
        M_CommSlave --> M_CommComp1{Communication  
completed?}
        M_CommComp1 -- NO --> M_CommSlave
        M_CommComp1 -- YES --> M_CommSlave2{Communicating  
with another slave  
CPU}
        M_CommSlave2 -- NO --> M_EndComm([End of communication])
        M_CommSlave2 -- YES --> M_DisableTxRx[Disabling transmission/  
reception operation]
        M_DisableTxRx --> M_EndComm
    end

    subgraph Slave_CPU [Slave CPU]
        S_Start([Start]) --> S_SetMode[Setting operation mode  
(set to mode 1)]
        S_SetMode --> S_SetSIN[Setting SIN pin to serial  
data input]
        S_SetSIN --> S_SelectBits[Selecting 7 or 8 data bits  
Selecting 1 or 2 stop bits]
        S_SelectBits --> S_EnableTxRx[Enabling transmission/  
reception operation]
        S_EnableTxRx --> S_ReceiveByte[Reception byte]
        S_ReceiveByte --> S_BitD8_1{Bit D8 = 1}
        S_BitD8_1 -- NO --> S_ReceiveByte
        S_BitD8_1 -- YES --> S_AddrMatch{Slave address  
matched}
        S_AddrMatch -- NO --> S_ReceiveByte
        S_AddrMatch -- YES --> S_SetSOUT[Setting SOUT pin to serial  
data output]
        S_SetSOUT --> S_CommMaster[Communicating with  
master CPU]
        S_CommMaster --> S_CommComp2{Communication  
completed?}
        S_CommComp2 -- NO --> S_CommMaster
        S_CommComp2 -- YES --> S_EndComm([End of communication])
    end
```

● When FIFO is used

Figure 24.9-3 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 (SSEL89AB)
 - 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	<ul style="list-style-type: none"> Full-duplex double buffer (when FIFO is not used) Transmission/reception FIFO (up to 16 bytes each) (when FIFO is used)*
2	Transfer system	<ul style="list-style-type: none"> Clock synchronization (no start bit / no stop bit) Master/slave function SPI supported (both master & slaves supported)
3	Baud rate	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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.8 to ch.11.
10	FIFO options	<ul style="list-style-type: none"> 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.

*: There is no FIFO between ch.8 and ch.11

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 24.13-1 Registers of the CSIO (1 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
Common to 8 to 11	SSEL89AB	Serial mode select register 89AB	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
	FCR10	FIFO control register 10	24.13.7
	FCR00	FIFO control register 00	24.13.8
	FBYTE10	FIFO1 byte register 0	24.13.9
	FBYTE20	FIFO2 byte register 0	24.13.9
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
	FCR11	FIFO control register 11	24.13.7
	FCR01	FIFO control register 01	24.13.8
	FBYTE11	FIFO1 byte register 1	24.13.9
	FBYTE21	FIFO2 byte register 1	24.13.9

Table 24.13-1 Registers of the CSIO (2 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
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
	FCR12	FIFO control register 12	24.13.7
	FCR02	FIFO control register 02	24.13.8
	FBYTE12	FIFO1 byte register 2	24.13.9
	FBYTE22	FIFO2 byte register 2	24.13.9
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
	FCR13	FIFO control register 13	24.13.7
	FCR03	FIFO control register 03	24.13.8
	FBYTE13	FIFO1 byte register 3	24.13.9
	FBYTE23	FIFO2 byte register 3	24.13.9

Table 24.13-1 Registers of the CSIO (3 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
4	SCR4	Serial control register 4	24.13.1
	SMR4	Serial mode register 4	24.13.2
	ESCR4	Extended serial control register 4	24.13.4
	BGR4	Baud rate generator register 4	24.13.6
	SSR4	Serial status register 4	24.13.3
	RDR4	Received data register 4	24.13.5
	TDR4	Transmitted data register 4	24.13.5
	FCR14	FIFO control register 14	24.13.7
	FCR04	FIFO control register 04	24.13.8
	FBYTE14	FIFO1 byte register 4	24.13.9
	FBYTE24	FIFO2 byte register 4	24.13.9
5	SCR5	Serial control register 5	24.13.1
	SMR5	Serial mode register 5	24.13.2
	ESCR5	Extended serial control register 5	24.13.4
	BGR5	Baud rate generator register 5	24.13.6
	SSR5	Serial status register 5	24.13.3
	RDR5	Received data register 5	24.13.5
	TDR5	Transmitted data register 5	24.13.5
	FCR15	FIFO control register 15	24.13.7
	FCR05	FIFO control register 05	24.13.8
	FBYTE15	FIFO1 byte register 5	24.13.9
	FBYTE25	FIFO2 byte register 5	24.13.9

Table 24.13-1 Registers of the CSIO (4 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
6	SCR6	Serial control register 6	24.13.1
	SMR6	Serial mode register 6	24.13.2
	ESCR6	Extended serial control register 6	24.13.4
	BGR6	Baud rate generator register 6	24.13.6
	SSR6	Serial status register 6	24.13.3
	RDR6	Received data register 6	24.13.5
	TDR6	Transmitted data register 6	24.13.5
	FCR16	FIFO control register 16	24.13.7
	FCR06	FIFO control register 06	24.13.8
	FBYTE16	FIFO1 byte register 6	24.13.9
	FBYTE26	FIFO2 byte register 6	24.13.9
7	SCR7	Serial control register 7	24.13.1
	SMR7	Serial mode register 7	24.13.2
	ESCR7	Extended serial control register 7	24.13.4
	BGR7	Baud rate generator register 7	24.13.6
	SSR7	Serial status register 7	24.13.3
	RDR7	Received data register 7	24.13.5
	TDR7	Transmitted data register 7	24.13.5
	FCR17	FIFO control register 17	24.13.7
	FCR07	FIFO control register 07	24.13.8
	FBYTE17	FIFO1 byte register 7	24.13.9
	FBYTE27	FIFO2 byte register 7	24.13.9
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
	RDRM8	Received data mirror register 8	24.13.11
	TDRM8	Transmitted data mirror register 8	24.13.11

Table 24.13-1 Registers of the CSIO (5 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
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
	RDRM9	Received data mirror register 9	24.13.11
	TDRM9	Transmitted data mirror register 9	24.13.11
10	SCRA	Serial control register A	24.13.1
	SMRA	Serial mode register A	24.13.2
	ES CRA	Extended serial control register A	24.13.4
	BGRA	Baud rate generator register A	24.13.6
	SSRA	Serial status register A	24.13.3
	RDRA	Received data register A	24.13.5
	TDRA	Transmitted data register A	24.13.5
	RDRMA	Received data mirror register A	24.13.11
	TDRMA	Transmitted data mirror register A	24.13.11
11	SCRB	Serial control register B	24.13.1
	SMRB	Serial mode register B	24.13.2
	ESCRB	Extended serial control register B	24.13.4
	BGRB	Baud rate generator register B	24.13.6
	SSRB	Serial status register B	24.13.3
	RDRB	Received data register B	24.13.5
	TDRB	Transmitted data register B	24.13.5
	RDRMB	Received data mirror register B	24.13.11
	TDRMB	Transmitted data mirror register B	24.13.11

Table 24.13-2 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 24.13-1 shows the bit structure of the serial control register (SCR), and Table 24.13-3 describes the function of each bit.

Figure 24.13-1 Bit Structure of Serial Control Register (SCR)

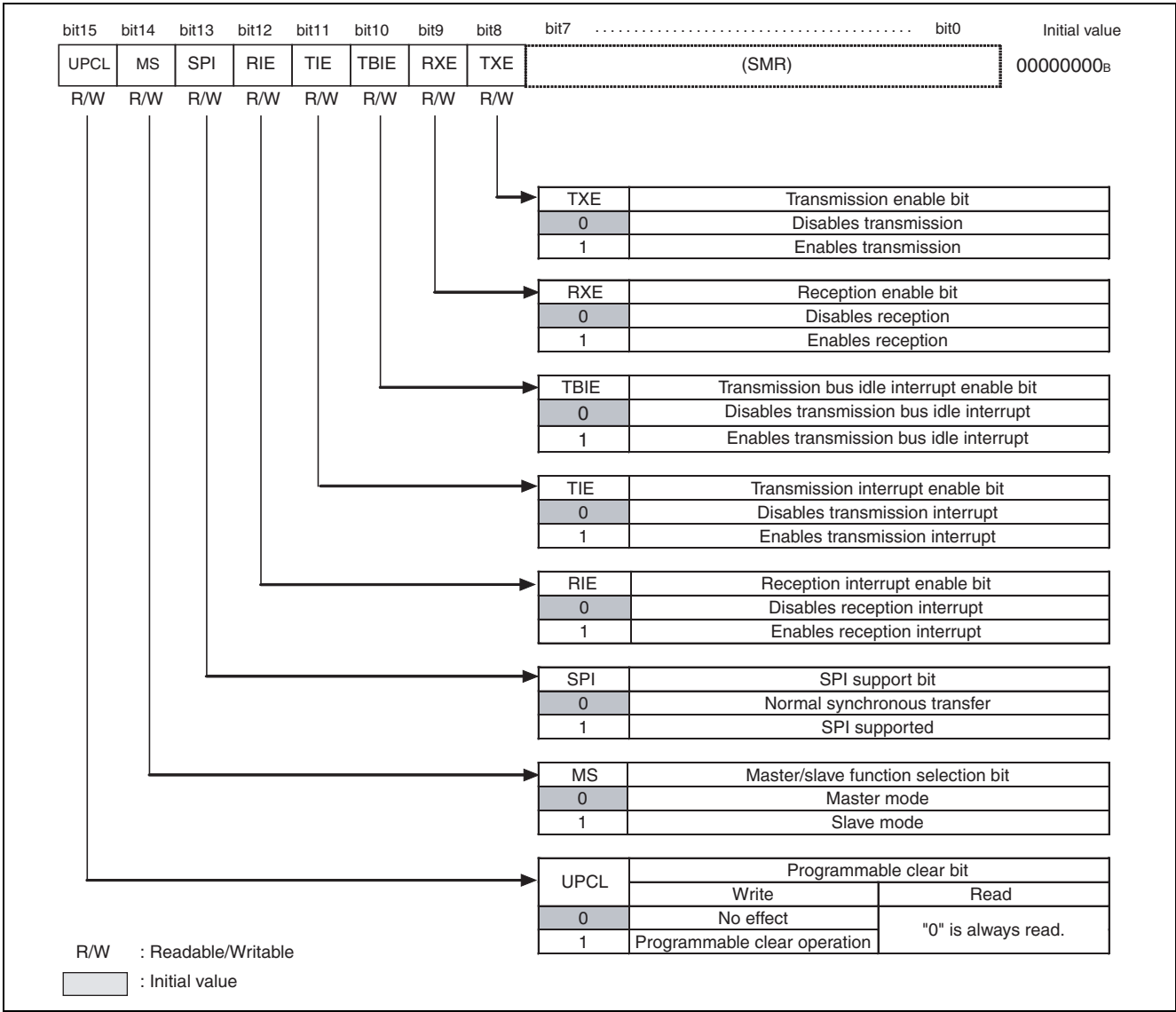


Table 24.13-3 Functional Description of Each Bit of Serial Control Register (SCR) (1 / 2)

Bit name		Function
bit15	UPCL: Programmable clear bit	<p>This bit is used to initialize the internal state of the CSIO.</p> <p>Setting the bit to "1":</p> <ul style="list-style-type: none"> 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". <p>Note:</p> <p>Execute the programmable clear operation after disabling interrupts.</p> <p>Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.</p>
bit14	MS: Master/slave function selection bit	<p>This bit is used to select master or slave mode.</p> <p>Setting the bit to "0" selects master mode.</p> <p>Setting the bit to "1" selects slave mode.</p> <p>Note:</p> <p>The external clock will be input directly, if SMR:SCKE is set to "0" when slave mode is selected.</p>
bit13	SPI: SPI support bit	<p>This bit is used to enable communication supporting SPI.</p> <p>Setting the bit to "0" enables normal synchronous communication.</p> <p>Setting the bit to "1" enables SPI support.</p>
bit12	RIE: Reception interrupt enable bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<p>This bit is used to enable/disable CSIO reception operation.</p> <p>Setting the bit to "0" disables data frame reception operation.</p> <p>Setting the bit to "1" enables data frame reception operation.</p> <p>Note:</p> <p>If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.</p>

Table 24.13-3 Functional Description of Each Bit of Serial Control Register (SCR) (2 / 2)

Bit name		Function
bit8	TXE: Transmission enable bit	<p>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.</p> <p>Note:</p> <p>If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.</p>

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 24.13-2 shows the bit structure of the serial mode register (SMR), and Table 24.13-4 describes the function of each bit.

Figure 24.13-2 Bit Structure of Serial Mode Register (SMR)

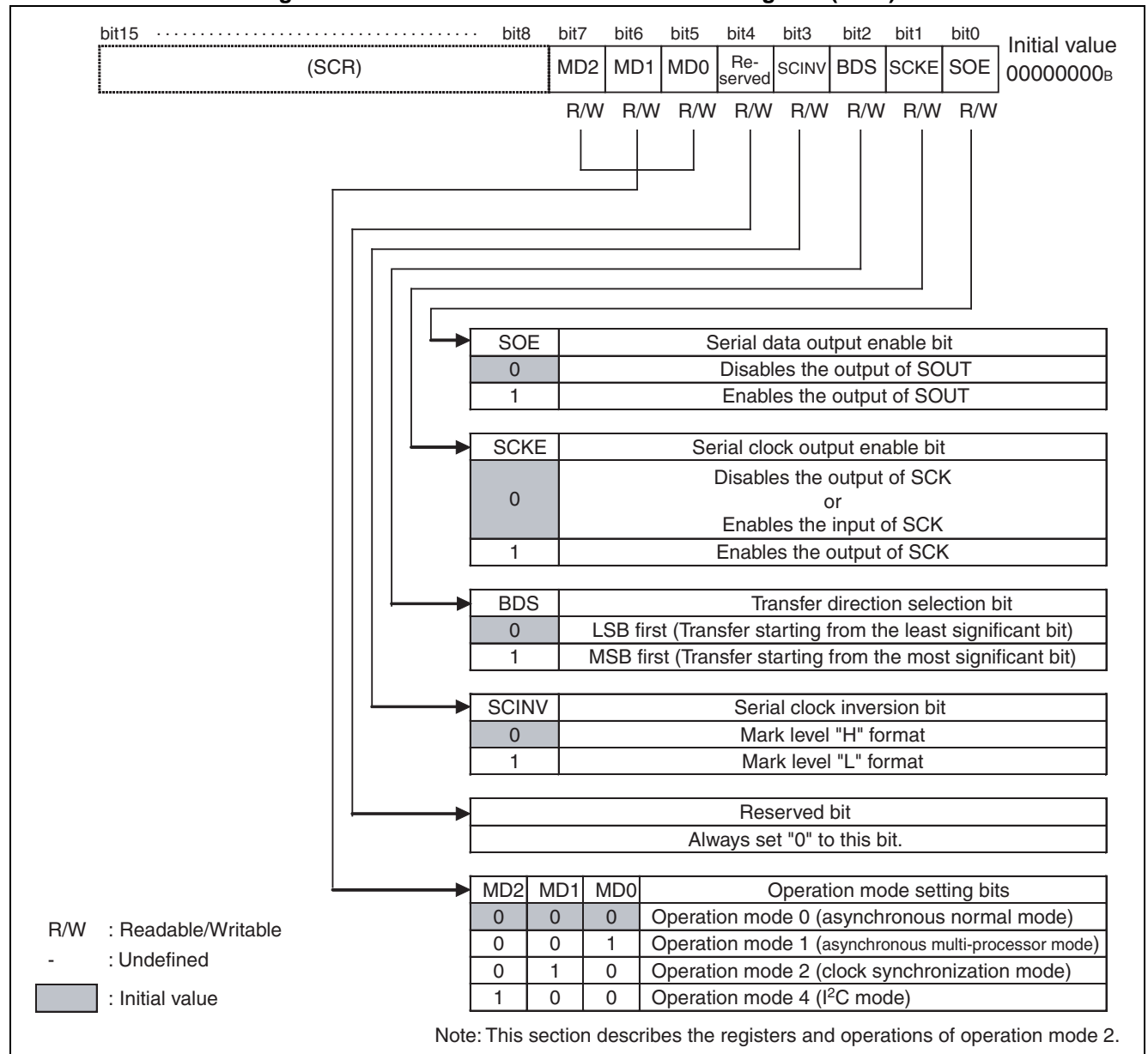


Table 24.13-4 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	<p>These bits are used to select the operation mode.</p> <p>"000_B": Selects operation mode 0 (asynchronous normal mode)</p> <p>"001_B": Selects operation mode 1 (asynchronous multi-processor mode)</p> <p>"010_B": Selects operation mode 2 (clock synchronization mode)</p> <p>"100_B": Selects operation mode 4 (I²C mode)</p> <p>This section describes the registers and operations of operation mode 2 (clock synchronization mode).</p> <p>Note:</p> <p>Settings other than above are prohibited.</p> <p>To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1).</p> <p>And then, after setting the operation mode, set each register.</p>
bit4	Reserved bit	Always set "0" to this bit.
bit3	SCINV: Serial clock inversion bit	<p>This bit is used to invert the serial clock format.</p> <p>Setting the bit to "0":</p> <ul style="list-style-type: none"> 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. <p>Setting the bit to "1":</p> <ul style="list-style-type: none"> 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. <p>Note:</p> <p>Set this bit when transmission and reception are disabled (TXE = RXE = 0).</p>
bit2	BDS: Transfer direction selection bit	<p>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).</p> <p>Note:</p> <p>Set this bit when transmission and reception are disabled (TXE = RXE = 0).</p>
bit1	SCKE: Serial clock output enable bit	<p>This bit is used to control the I/O port of the serial clock.</p> <p>Setting the bit to "0":</p> <p>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.</p> <p>Setting the bit to "1" enables the output of SCK.</p>
bit0	SOE: Serial data output enable bit	<p>This bit is used to enable/disable the output of serial data.</p> <p>Setting the bit to "0" enables the output of the "H" level of SOUT.</p> <p>Setting the bit to "1" enables the output of SOUT.</p>

<Note>

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. 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 24.13-3 shows the bit structure of the serial status register (SSR) and Table 24.13-5 describes the function of each bit.

Figure 24.13-3 Bit Structure of Serial Status Register (SSR)

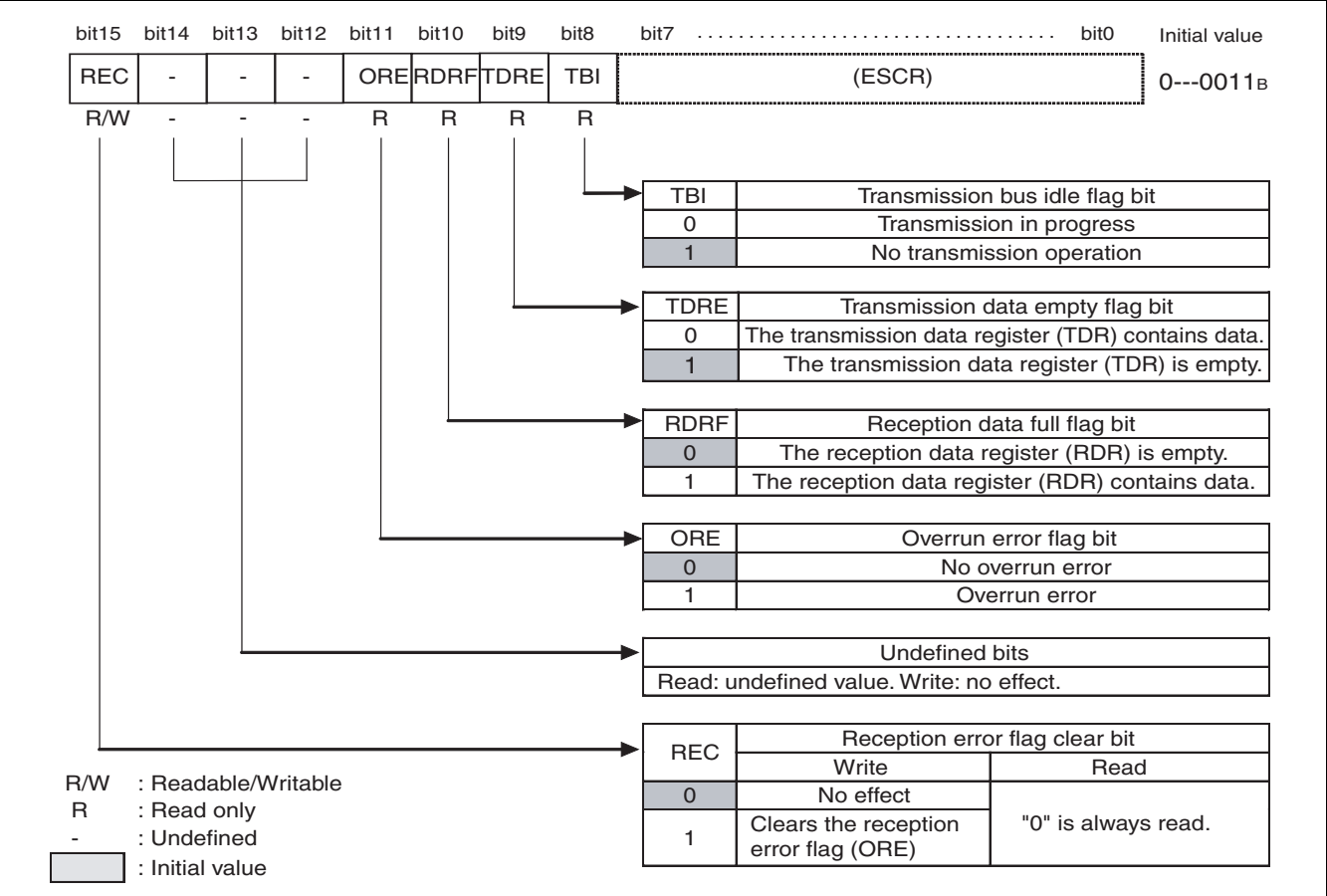


Table 24.13-5 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

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). <ul style="list-style-type: none">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

Table 24.13-5 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit11	ORE: Overrun error flag bit	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 24.13-4 shows the bit structure of the extended serial control register (ESCR) and Table 24.13-6 describes the function of each bit.

Figure 24.13-4 Bit Structure of Extended Serial Control Register (ESCR)

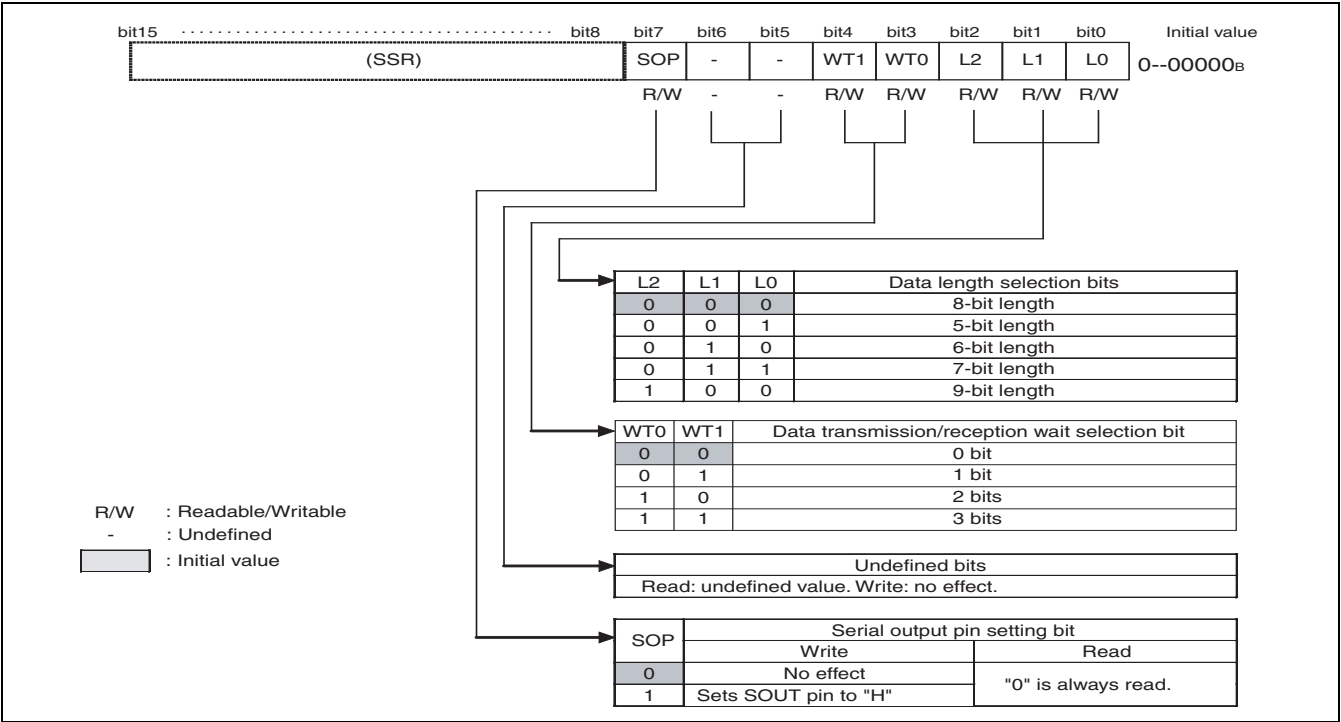


Table 24.13-6 Functional Description of Each Bit of Extended Serial Control Register (ESCR)

Bit name		Function
bit7	SOP: Serial output pin setting bit	<ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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.

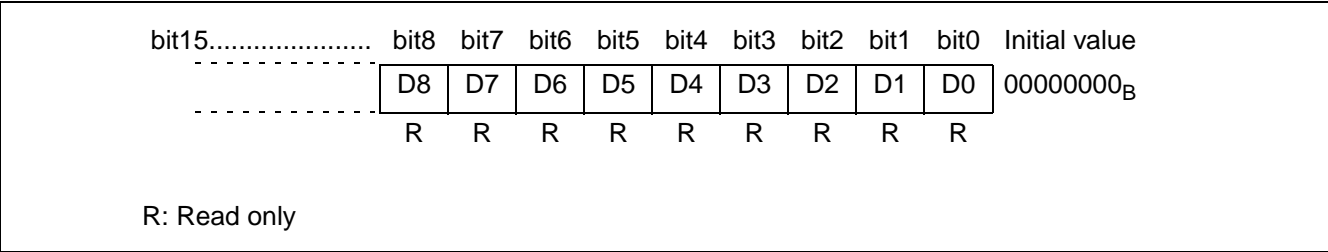
24.13.5 Reception Data Register / Transmission Data Register (RDR/TDR)

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 24.13-5 illustrates the bit structure of the serial reception register (RDR).

Figure 24.13-5 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 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 indicates the 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.

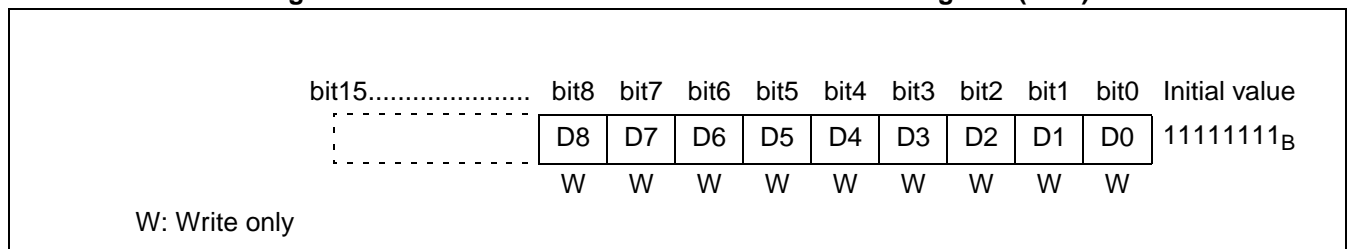
<Notes>

- 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 24.13-6 illustrates the bit structure of the transmission data register.

Figure 24.13-6 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 indicates the 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".
-

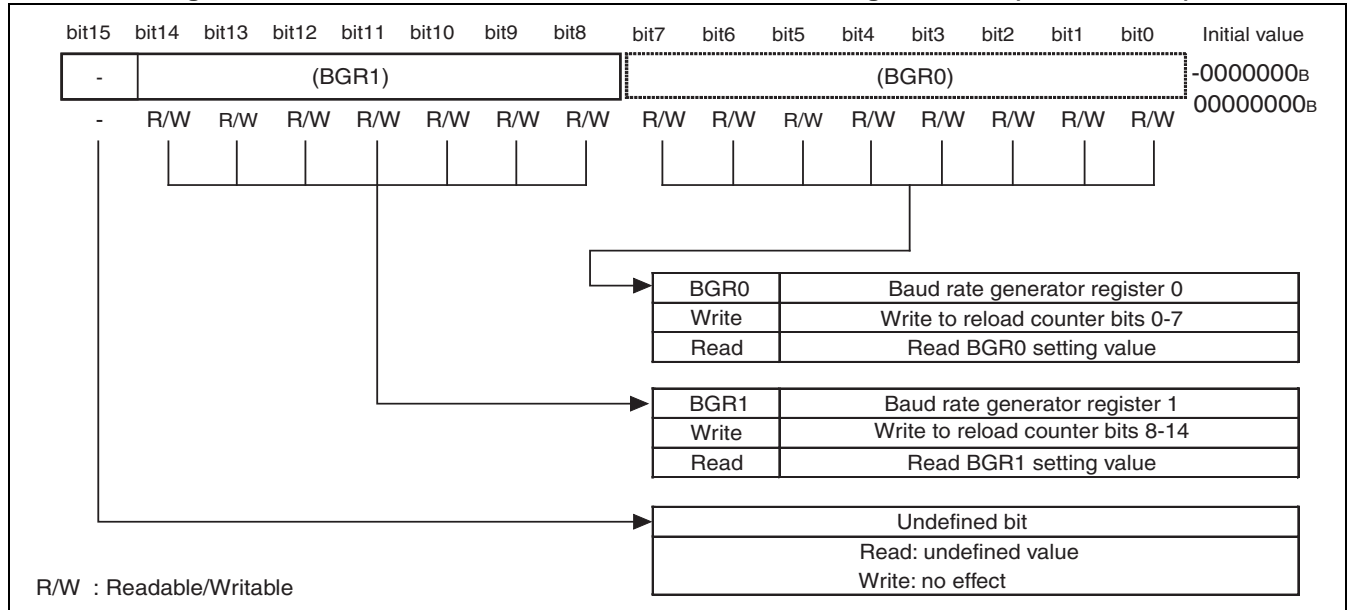
24.13.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

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 24.13-7 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 24.13-7 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 24.13-8 shows the bit structure of the FIFO control register 1 (FCR1) and Table 24.13-7 describes the function of each bit.

Figure 24.13-8 Bit Structure of FIFO Control Register 1 (FCR1)

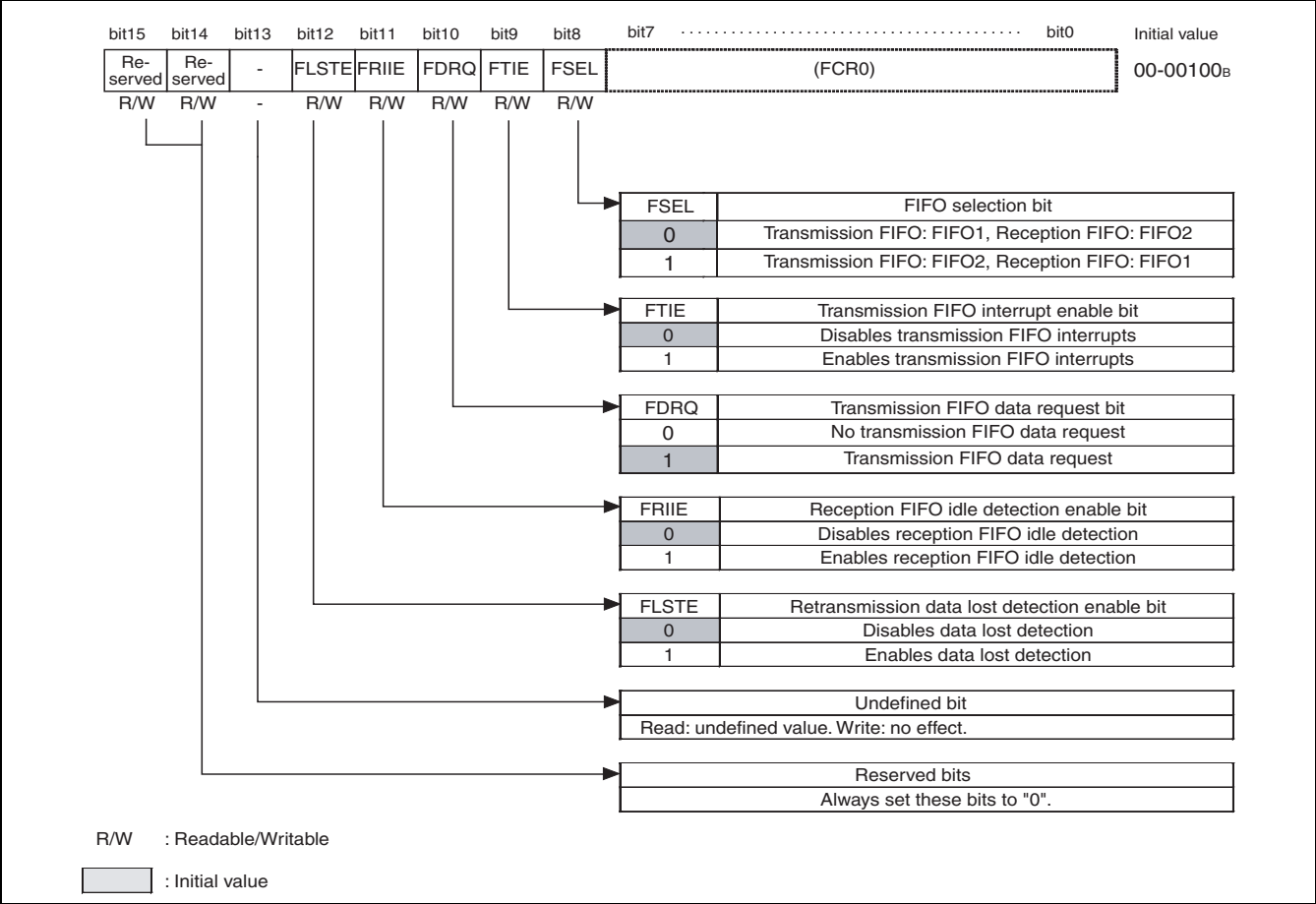


Table 24.13-7 Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

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 <ul style="list-style-type: none"> • FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) • Transmission FIFO reset FDRQ reset condition <ul style="list-style-type: none"> • 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 24.13-9 shows the bit structure of the FIFO control register 0 (FCR0) and Table 24.13-8 describes the function of each bit.

Figure 24.13-9 Bit Structure of FIFO Control Register 0 (FCR0)

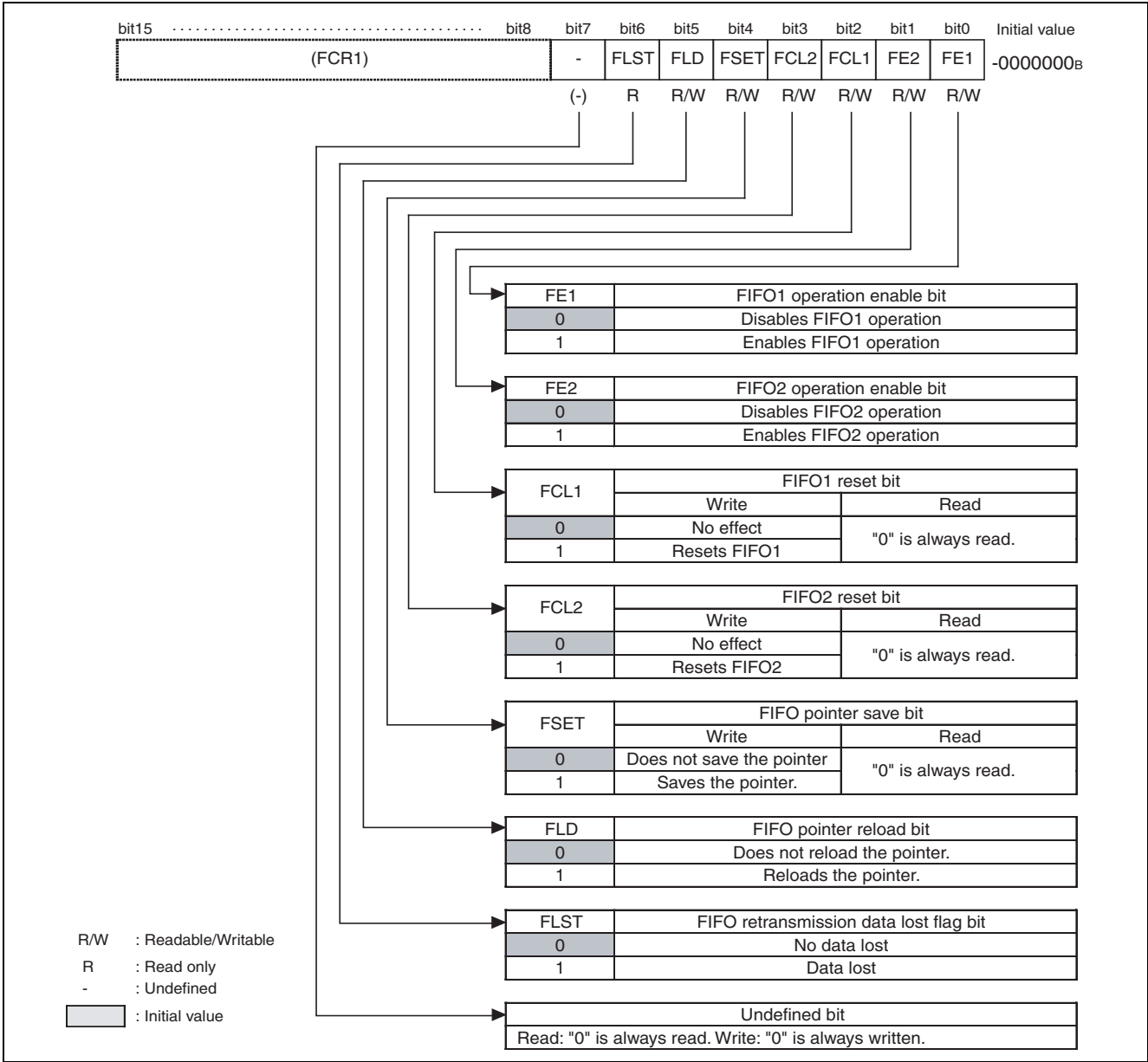


Table 24.13-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 2)

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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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".

Table 24.13-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 2)

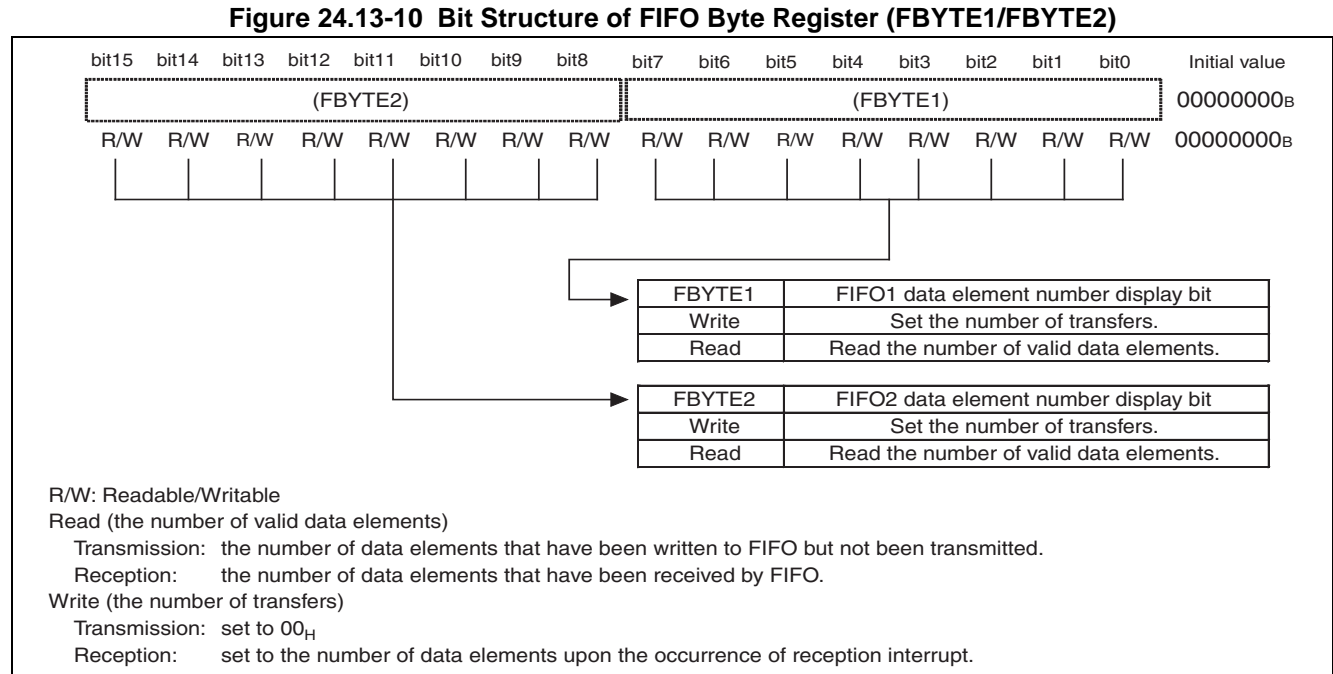
Bit name		Function
bit2	FCL1: FIFO1 reset bit	<p>This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR1:FLST1 bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained.</p> <p>Note:</p> <ul style="list-style-type: none"> 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	<p>This bit is used to enable/disable FIFO2 operation.</p> <ul style="list-style-type: none"> 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	<p>This bit is used to enable/disable FIFO1 operation.</p> <ul style="list-style-type: none"> 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 24.13-10 shows the 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 24.13-9 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>

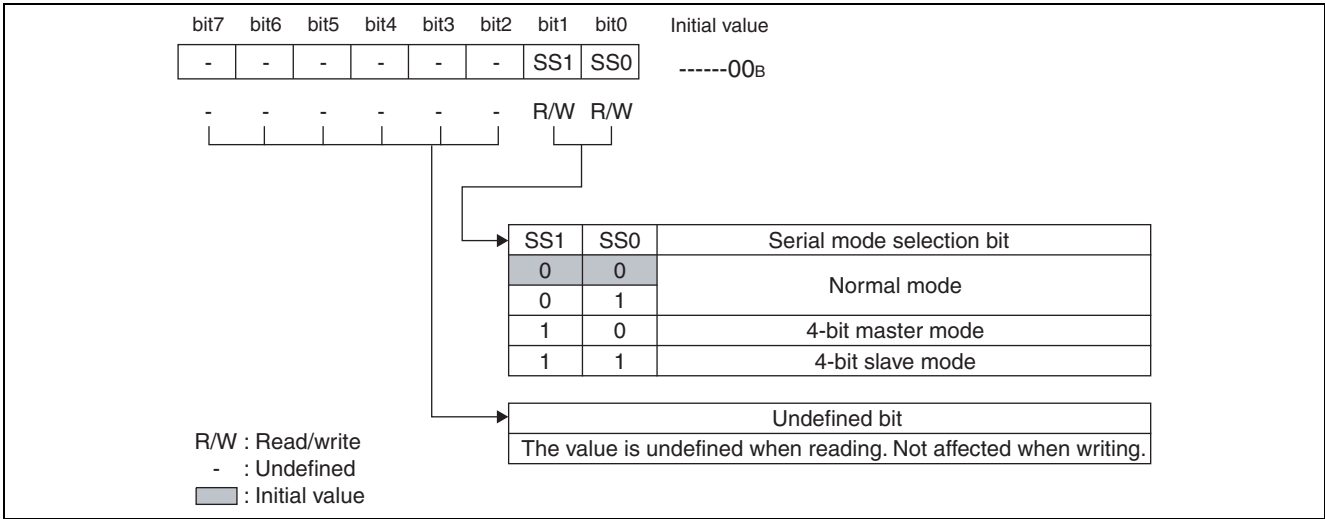
- In master operation, set "00_H" to FBYTE1/FBYTE2 of the transmission FIFO except when receiving data.
 - Set the number of transmission data elements when receiving data in master operation, when the transmission FIFO is empty and the TIE and TBIE bits are set to "0".
 - To disable reception (RXE = 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 (SSEL89AB)

These registers operate the CSIO with 4-channels by using 1 clock simultaneously, enabling 4-bit serial communication.
The combination of ch.8 to ch.11 enable 4-channel simultaneous communication.

Figure 24.13-11 shows the bit configuration of the serial mode select registers (SSEL89AB).

Figure 24.13-11 Bit configuration of the serial mode select registers (SSEL89AB)



<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 either ch.8 to ch.11 is allowed in master mode.
- 4-bit slave mode: 4-channel simultaneous communication using either ch.8 to ch.11 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 (SCR).
 - ch.8 to ch.10: Slave mode
 - ch.11: 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 (SCR).
-

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 8 (RDRM8) corresponds to lower 8 bits of the received data register 8 (RDR8) while the received data mirror register B (RDRMB) corresponds to lower 8 bits of the received data register B (RDRB).

The received data mirror registers (RDRM) of ch.8 to ch.11 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 8 (TDRM8) corresponds to lower 8 bits of the transmitted data register 8 (TDR8) while the transmitted data mirror register B (TDRMB) corresponds to lower 8 bits of the transmitted data register B (TDRB).

The transmitted data mirror registers (TDRM) of ch.8 to ch.11 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 24.14-1 lists the interrupt control bits and interrupt sources of the CSIO.

Table 24.14-1 Interrupt Control Bits and Interrupt Sources of CSIO (1 / 2)

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	Overflow error		Writing "1" to the reception error flag clear bit (SSR:REC)

Table 24.14-1 Interrupt Control Bits and Interrupt Sources of CSIO (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
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)*
	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)*
	FDRQ	FCR1	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

*: 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 24.14-1 Reception Operation and Flag Set Timing

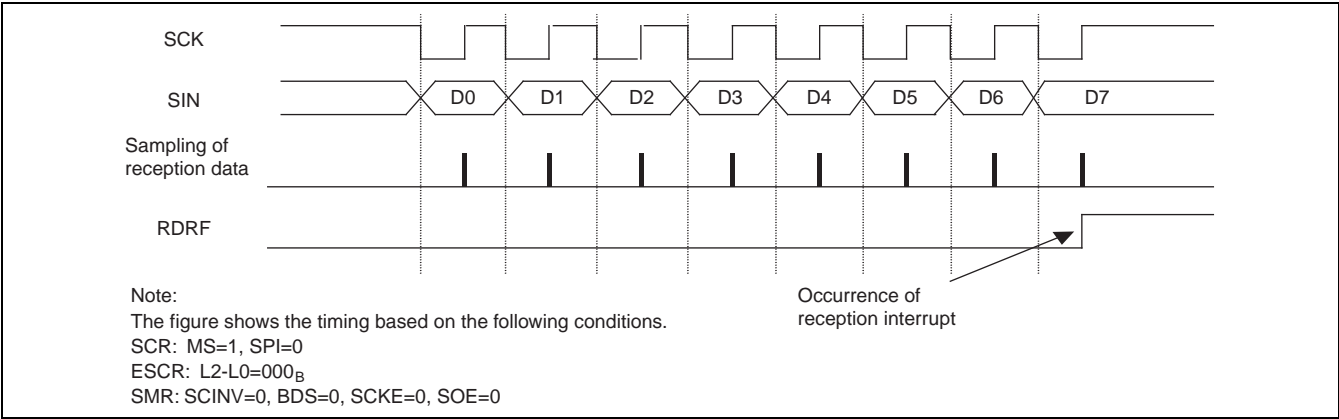
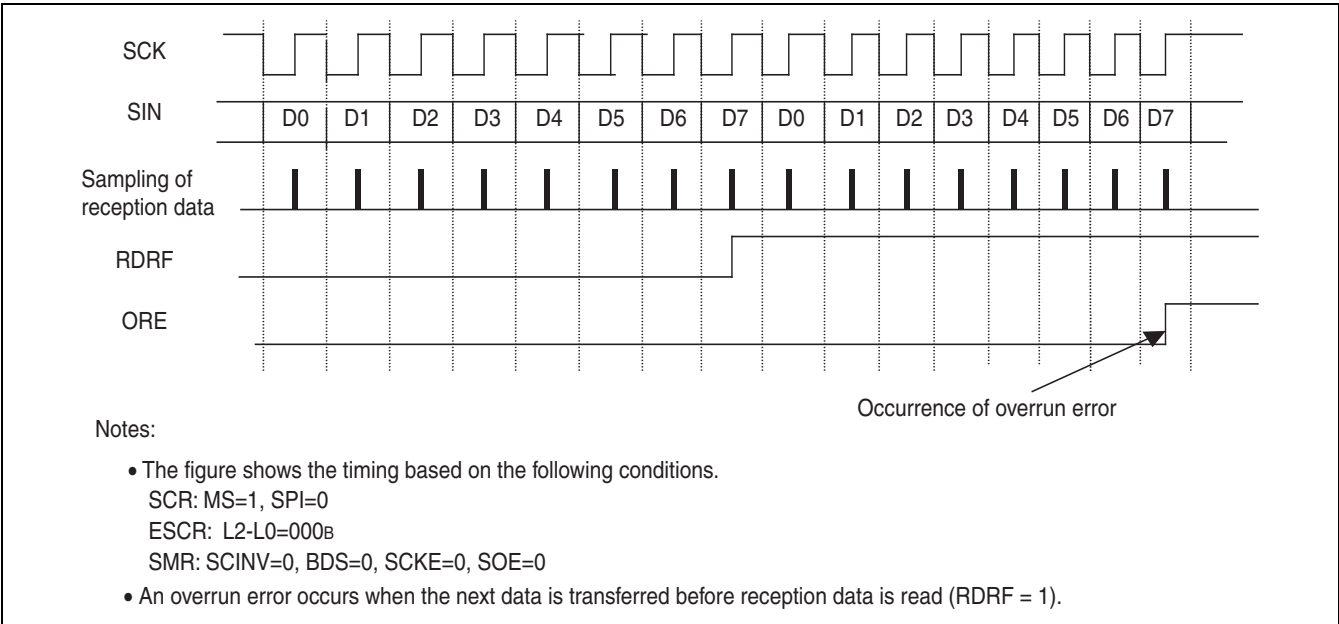


Figure 24.14-2 Timing for Setting ORE (Overrun Error) Flag



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 24.14-3 Timing for Generating Reception Interrupt when Reception FIFO is Used

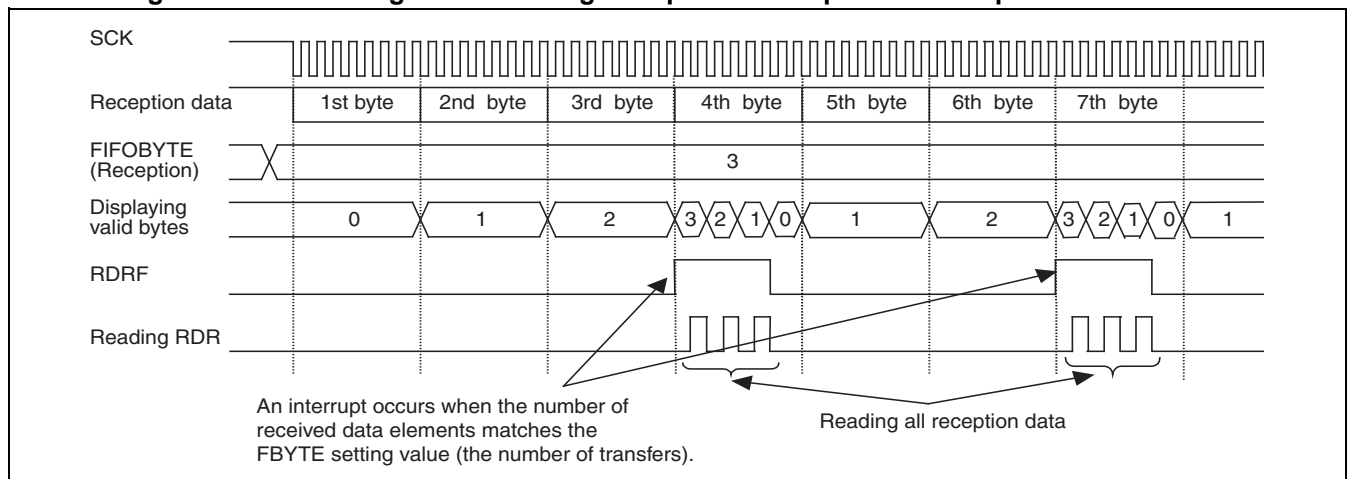
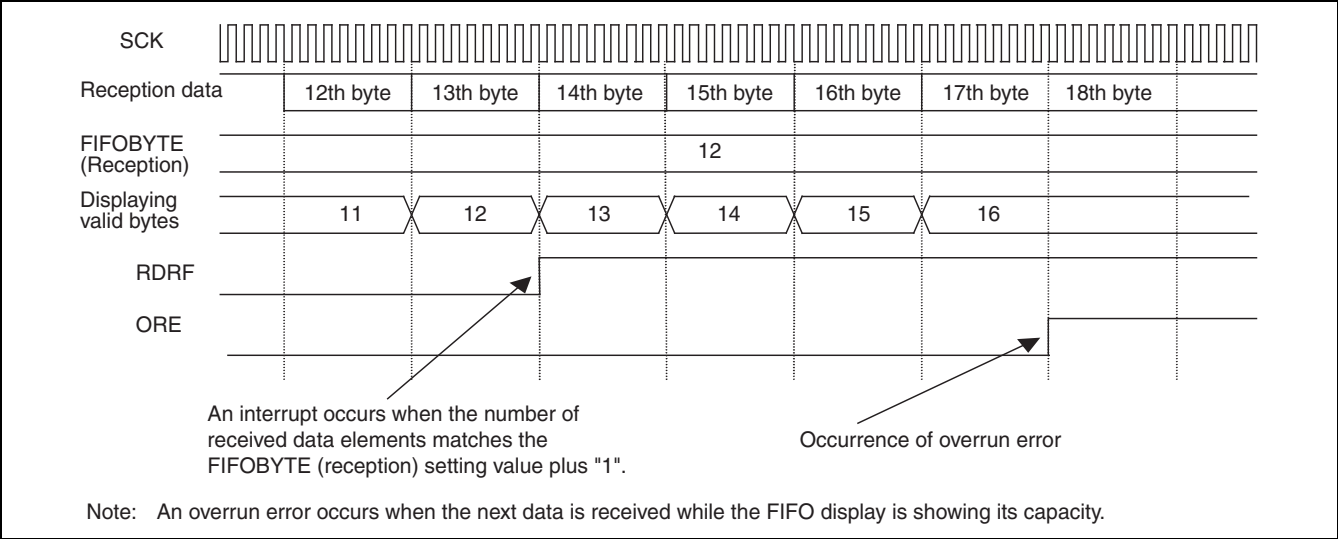


Figure 24.14-4 Timing for Setting ORE (Overrun Error) Flag Bit



24.14.3 Occurrence of Transmission Interrupts and Flag Set Timing

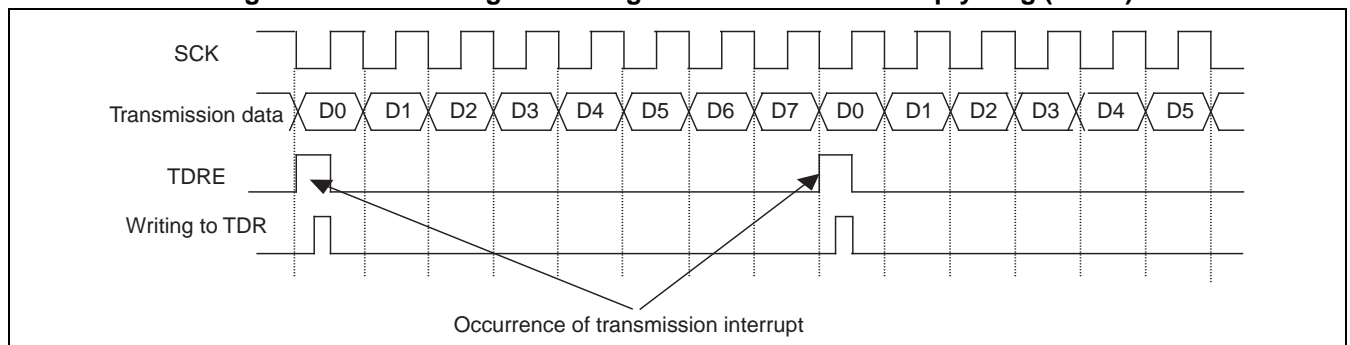
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

■ 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).

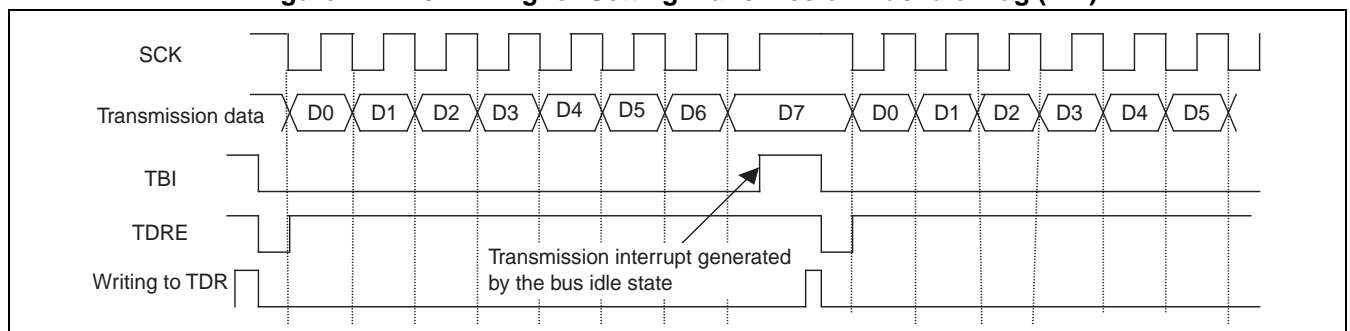
Figure 24.14-5 Timing for Setting Transmission Data Empty Flag (TDRE)



● 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 24.14-6 Timing for Setting Transmission Bus Idle Flag (TBI)



24.14.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

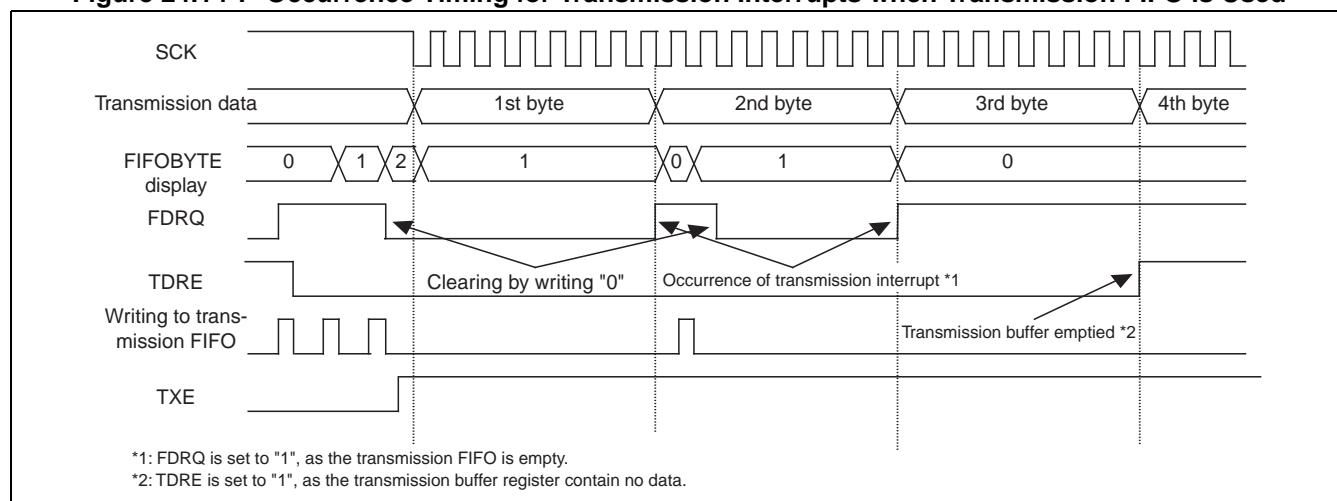
When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

■ 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 24.14-7 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 24.15-1 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 24.15-2 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	*	*	*	*	*	0	1	0	0	0	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
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
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

<Note>

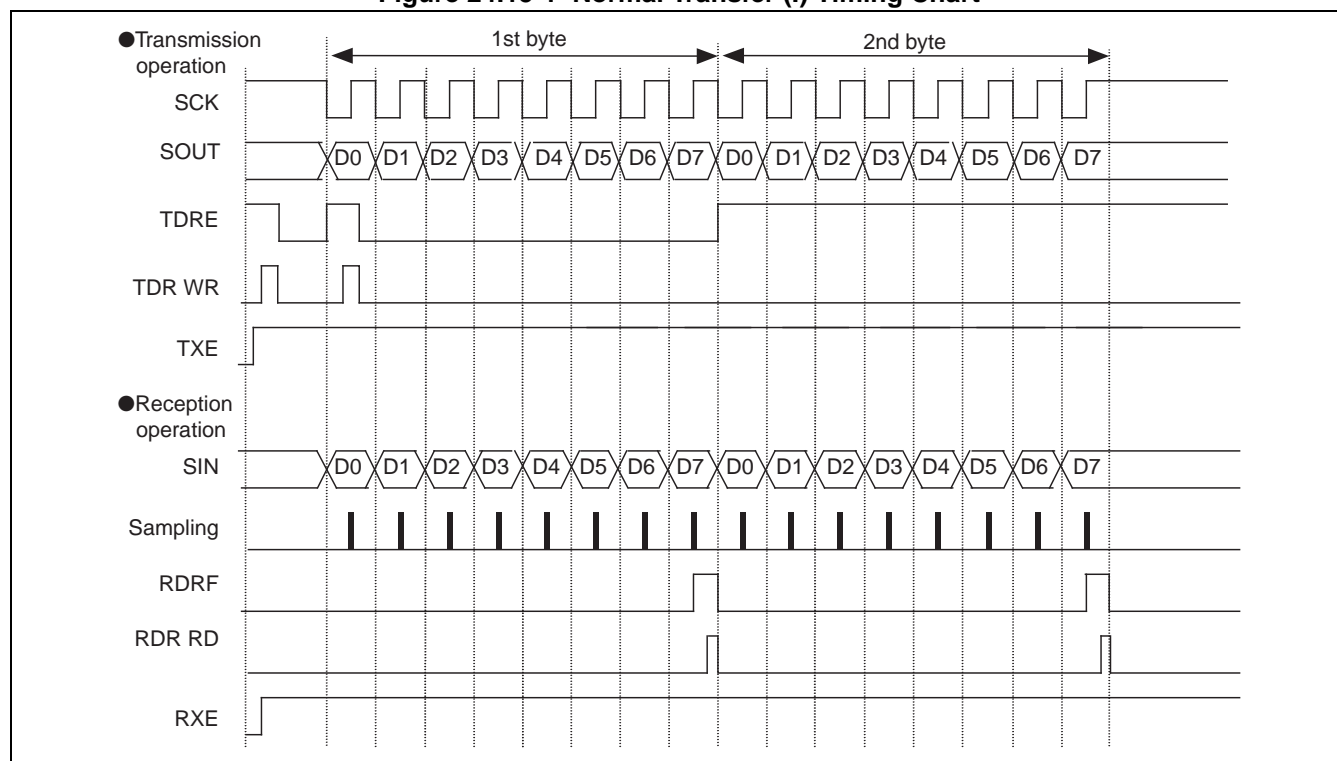
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation : SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

● Normal transfer (I) timing chart

Figure 24.15-1 Normal Transfer (I) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

• Transmission operation

- (1) 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.
- (2) 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.

- Reception operation
 - (1) 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).
 - (2) 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.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.
-

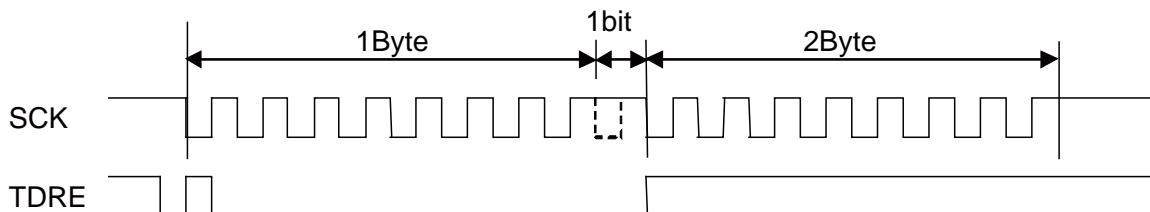
<Notes>

- 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.
-

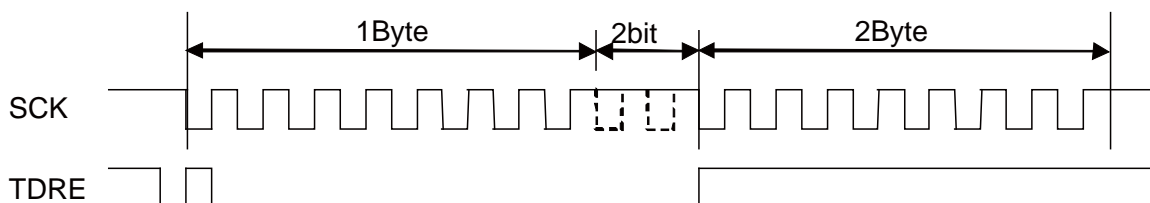
- Transmission/reception operation
 - (1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).
 - (2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the 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.
 - (3) 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.

- Sequential data transmission/reception wait operation
 - (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.

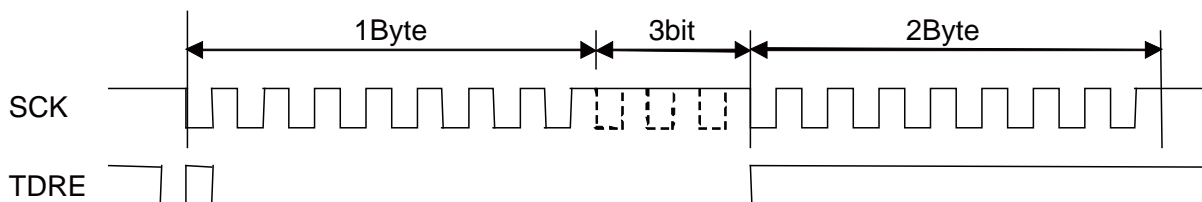
●ESCR:WT1=0, ESCR:WT0=1 (master)



●ESCR:WT1=1, ESCR:WT0=0 (master)



●ESCR:WT1=1, ESCR:WT0=1 (master)



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) 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.
 - (2) 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.
- Reception operation
 - (1) 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).
 - (2) 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.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation
 - (1) 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).
 - (2) 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.
 - (3) 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 24.15-3 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 24.15-4 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	*	*	*	*	*	0	1	0	0	1	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
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
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

<Note>

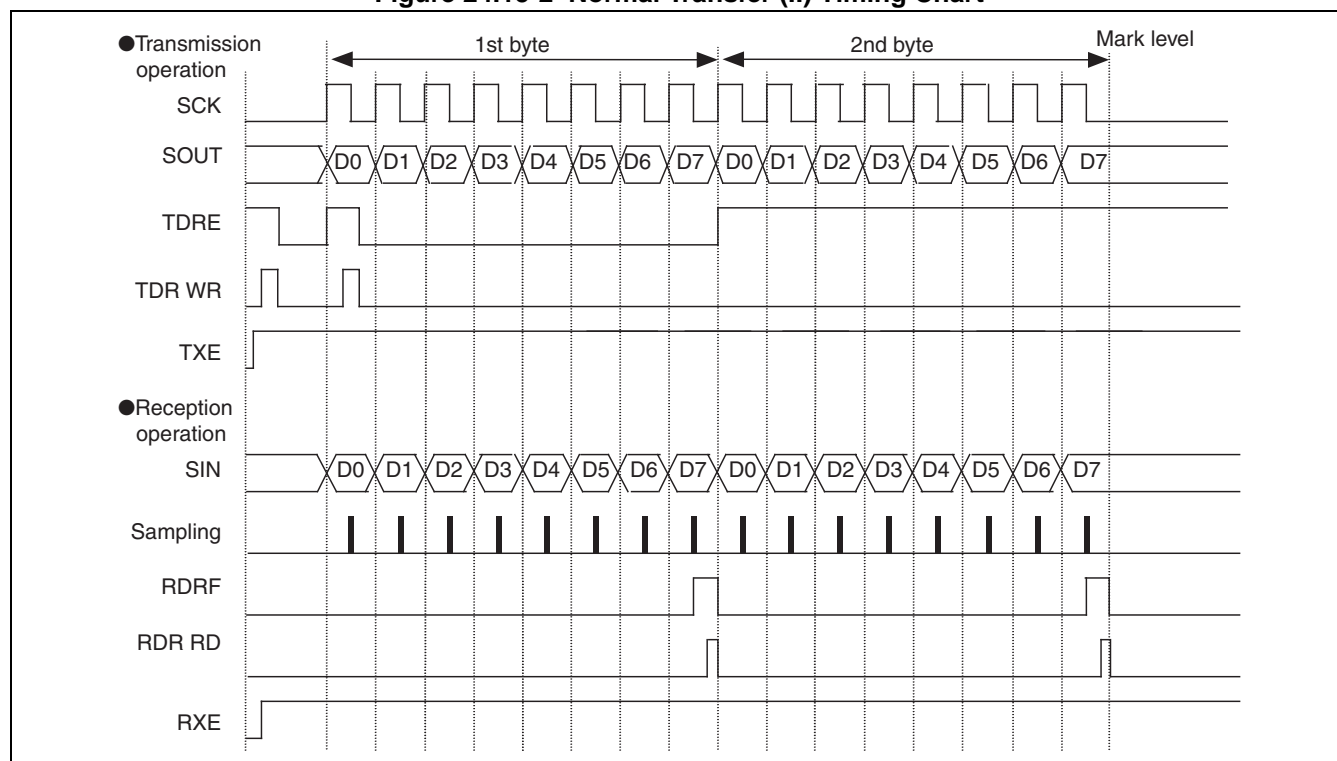
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation : SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

■ Normal Transfer (II) Timing Chart

Figure 24.15-2 Normal Transfer (II) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation

- (1) 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.
- (2) 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.

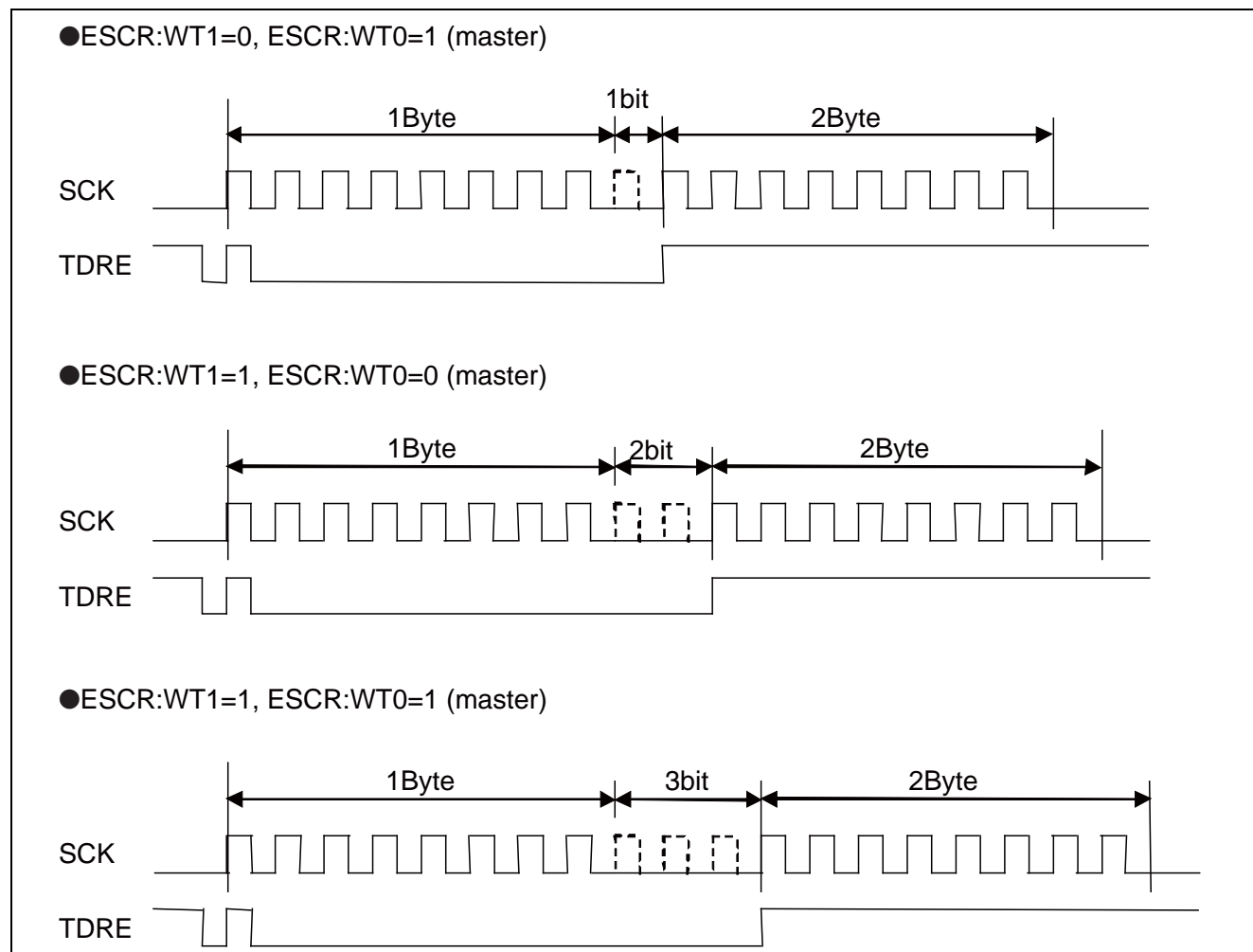
- Reception operation
 - (1) If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).
 - (2) SSR:RDRF 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.
 - (3) 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.
-

- Transmission/reception operation
 - (1) 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).
 - (2) 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.
 - (3) 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.

- Sequential data transmission/reception wait operation
 - (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) 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.
 - (2) 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.
- Reception operation
 - (1) 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).
 - (2) 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.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation
 - (1) 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).
 - (2) 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.
 - (3) 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

Table 24.15-5 Features of SPI Transfer (I)

	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 24.15-6 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	*	*	*	*	*	0	1	0	0	0	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
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
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

<Note>

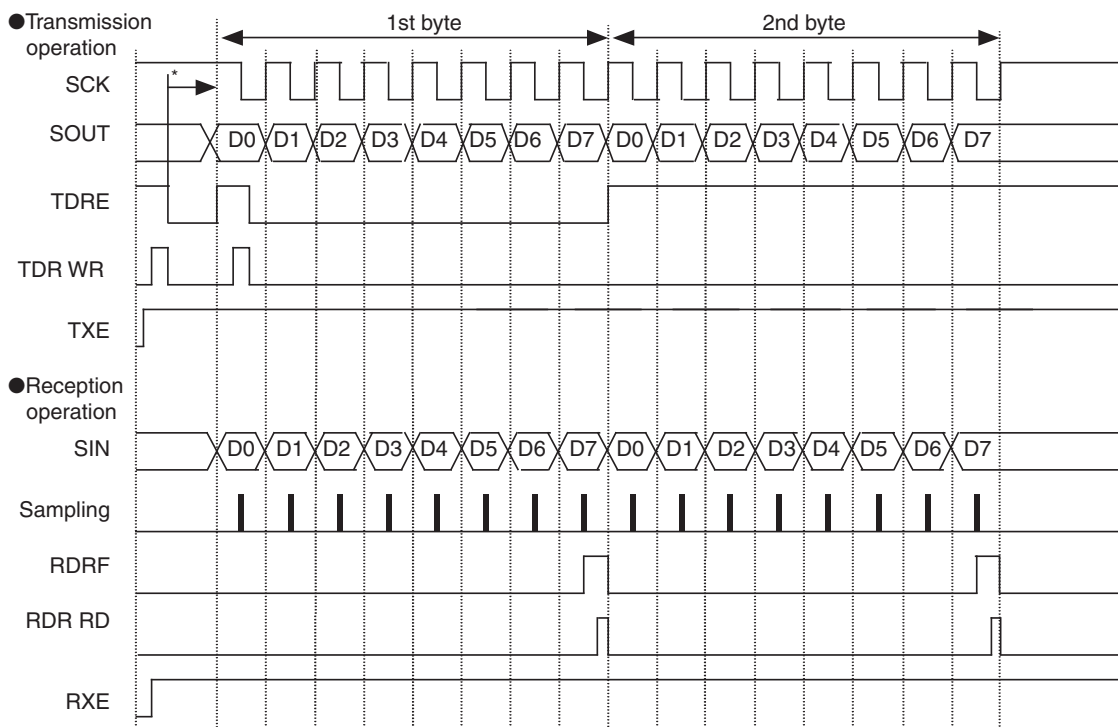
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation : SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

● SPI transfer (I) timing chart

Figure 24.15-3 SPI Transfer (I) Timing Chart



* : During slave transmission (MS=1, SCKE=0, SCE=1), a duration of 4 or more peripheral clocks (PCLK) is required after data is written to TDR.

● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation
 - (1) 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.
 - (2) 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.

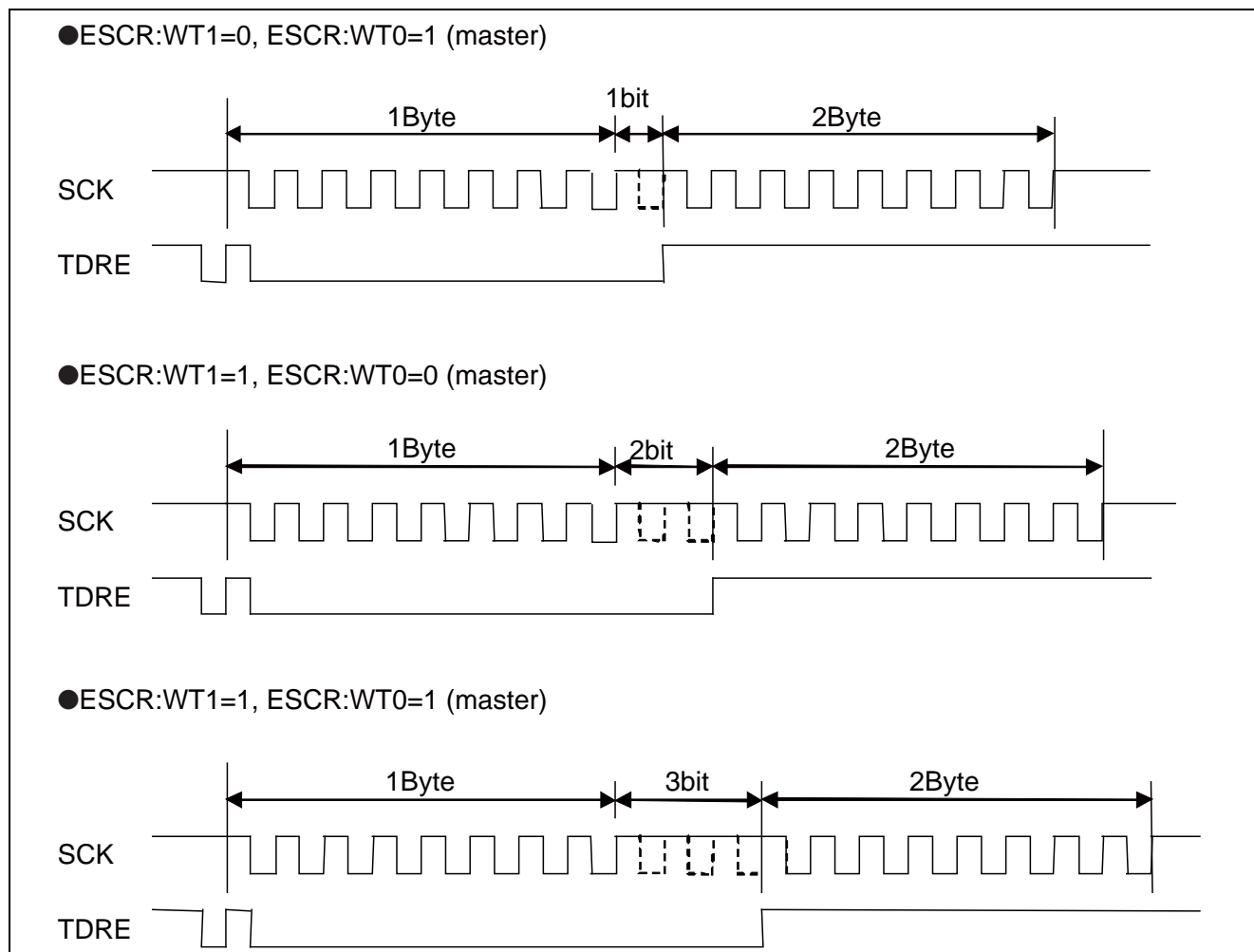
- Reception operation
 - (1) If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).
 - (2) SSR:RDRF 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.
 - (3) 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.
-

- Transmission/reception operation
 - (1) 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).
 - (2) 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.
 - (3) 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.

- Sequential data transmission/reception wait operation
 - (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) 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.
 - (2) 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.
- Reception operation
 - (1) 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).
 - (2) 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.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation
 - (1) 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).
 - (2) 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.
 - (3) 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 24.15-7 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 24.15-8 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	*	*	*	*	*	0	1	0	0	1	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
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
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

<Note>

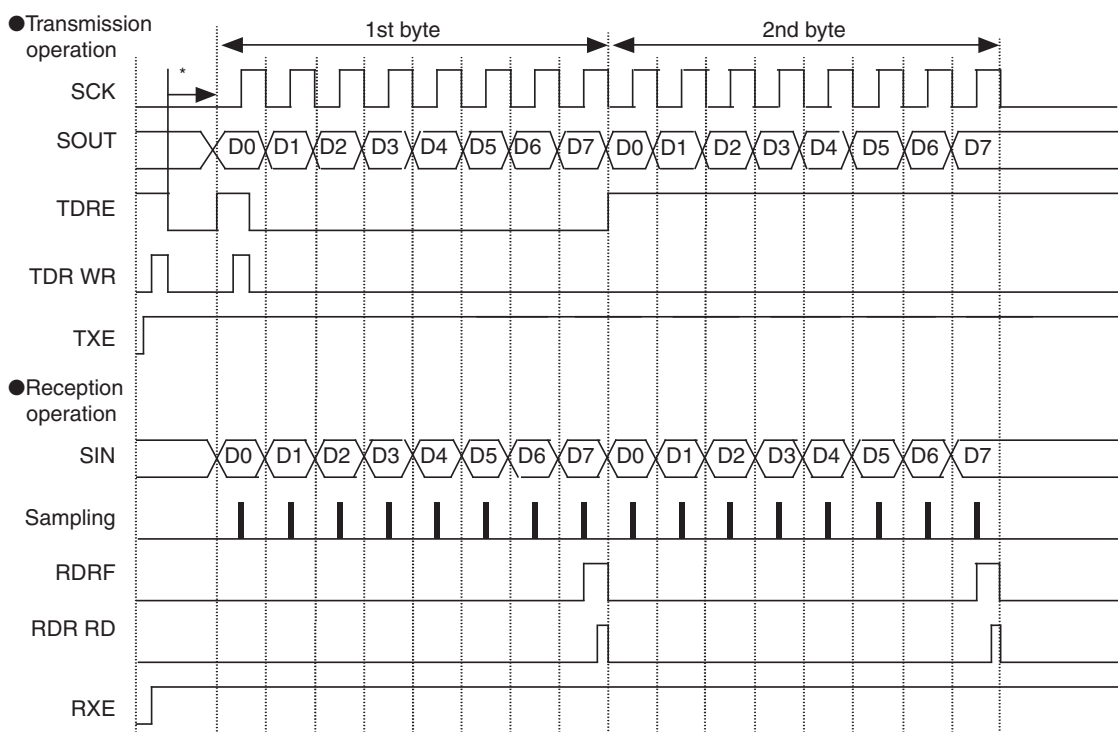
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation : SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

● SPI transfer (II) timing chart

Figure 24.15-4 SPI Transfer (II) Timing Chart



*: During slave transmission (MS=1, SCKE=0, SCE=1), a duration of 4 or more peripheral clocks (PCLK) is required after data is written to TDR.

● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

• Transmission operation

- (1) 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.
- (2) 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.

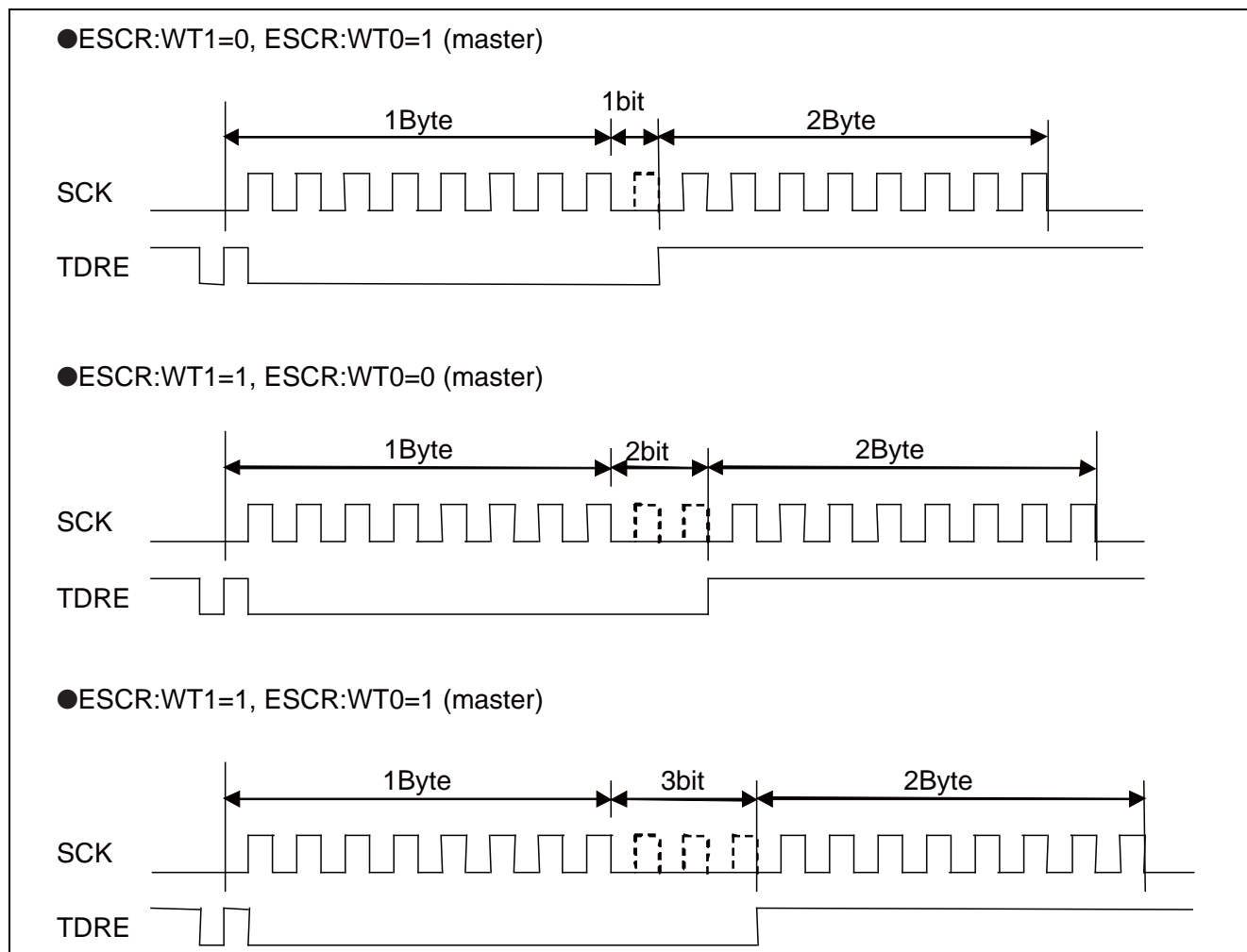
- Reception operation
 - (1) If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock output (SCK).
 - (2) SSR:RDRF 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.
 - (3) 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.
-

- Transmission/reception operation
 - (1) 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).
 - (2) 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.
 - (3) 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.

- Sequential data transmission/reception wait operation
 - (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation
 - (1) 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.
 - (2) 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.
- Reception operation
 - (1) 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).
 - (2) 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.
 - (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation
 - (1) 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).
 - (2) 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.
 - (3) 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.8 to ch.11 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 (SSEL89AB).

In addition, the required settings vary according to whether communication is executed in master mode or in slave mode.

Table 24.15-9 shows the settings required for the 4-channel simultaneous communication mode.

Table 24.15-9 Settings in 4-channel simultaneous communication mode

Mode	Setting		ch.8	ch.9	ch.10	ch.11
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 (SSEL89AB)

SCR: Serial control register (SCR)

The serial clock input methods vary according to whether the mode is 4-bit master mode or 4-bit slave mode.

Table 24.15-10 lists the input sources of the serial clock.

Table 24.15-10 Input Sources of the Serial Clock

Mode	ch.8	ch.9	ch.10	ch.11
4-bit master (SS1, SS0 = 10)	Output from ch.11	Output from ch.11	Output from ch.11	SCK11 pin
4-bit slave (SS1, SS0 = 11)	SCK11 pin	SCK11 pin	SCK11 pin	SCK11 pin

Figure 24.15-5 shows the input sources of the serial clock in 4-bit master mode and in 4-bit slave mode.

Figure 24.15-5 The Input Sources of the Serial Clock

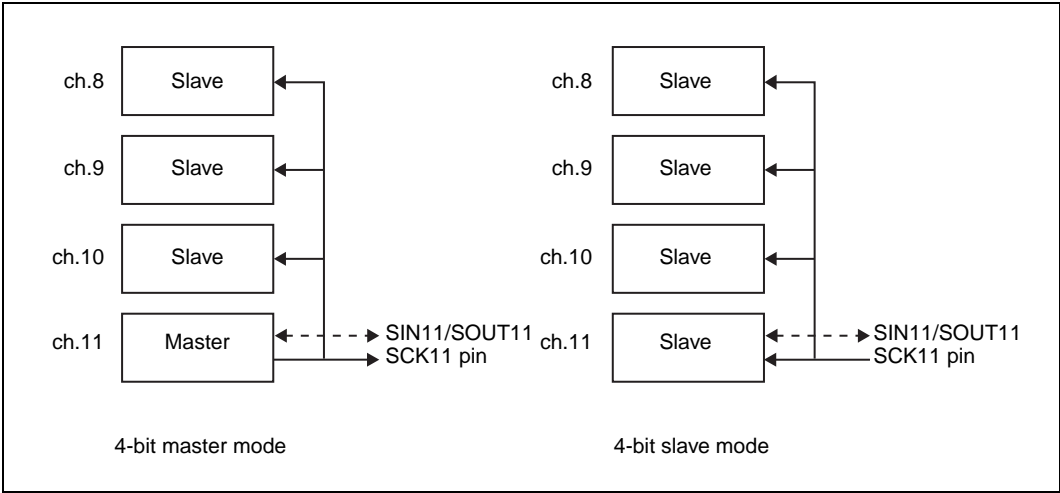


Table 24.15-11 shows the available pin combination for the four-channel simultaneous communication mode.

Table 24.15-11 Available Pin Combination

	ch.8	ch.9	ch.10	ch.11
Combination 1	SCK8 SIN8 SOUT8	SCK9 SIN9 SOUT9	SCK10 SIN10 SOUT10	SCK11 SIN11 SOUT11

● 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 (RDRM)
- Transmitted data mirror registers (TDRM)

Access to these registers allows access to the received data register (RDR) lower 8 bits or the transmitted data register (TDR) lower 8 bits.

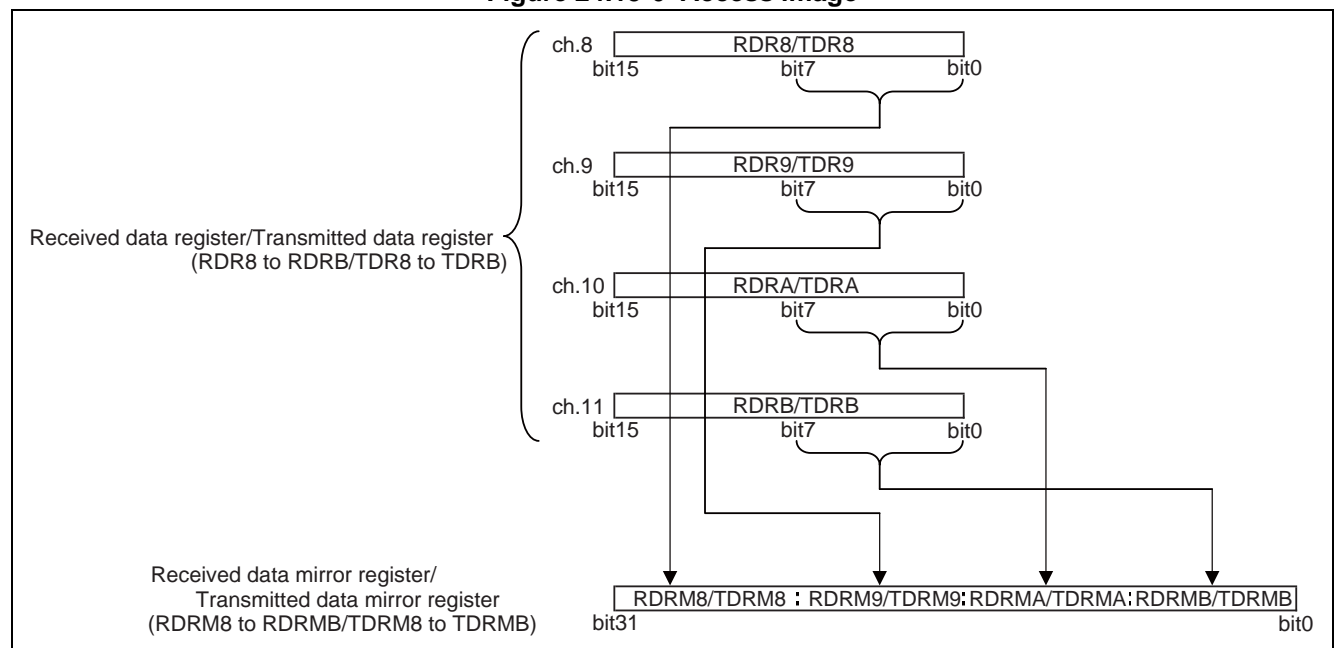
The received data mirror registers (RDRM)/transmitted data mirror registers (TDRM) of ch.8 to ch.11 are arranged in line. So, word access allows the registers to be written to at one time. Use this for DMA transfer and other purposes.

<Note>

The interrupt during 4 channels are simultaneously activating, it is recommended to allow using 1 channel only out of 4 channels.

Figure 24.15-6 shows the images of the received data mirror register (RDRM)/transmitted data mirror register (TDRM).

Figure 24.15-6 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 rate 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.

(1) Reload value:

$$V = \phi / b - 1$$

V: Reload value

b: Baud rate

ϕ : Peripheral clock (PCLK) frequency

(2) 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:

$$b = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

(3) Baud rate error

The following formula is used to calculate a baud rate error.

$$\text{Error (\%)} = (\text{calculated value} - \text{target value}) / \text{target value} \times 100$$

Example: peripheral clock (PCLK) = 20MHz, target baud rate = 153600bps

$$\text{Reload value} = (20 \times 1000000) / 153600 - 1 = 129$$

$$\text{Baud rate (calculated value)} = (20 \times 1000000) / (129 + 1) = 153846 \text{ (bps)}$$

$$\text{Error (\%)} = (153846 - 153600) / 153600 \times 100 = 0.16 \text{ (\%)}$$

<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 SCINV is set to "0", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer.
 - When SCINV is set to "1", the "L" 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 24.16-1 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)

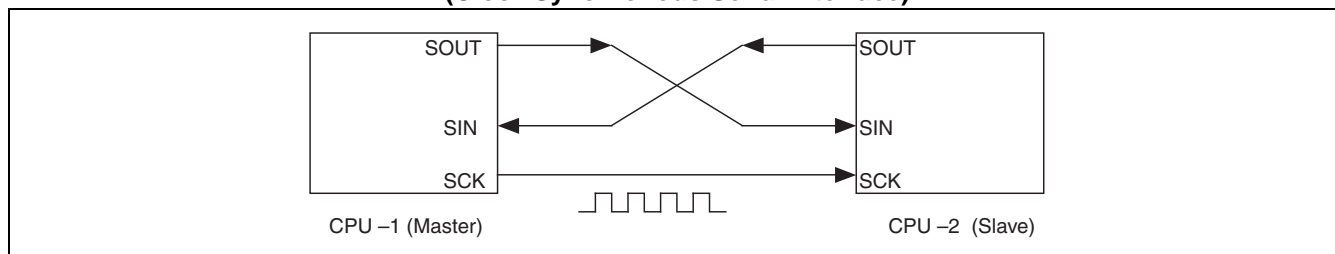
24.17 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

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 24.17-1.

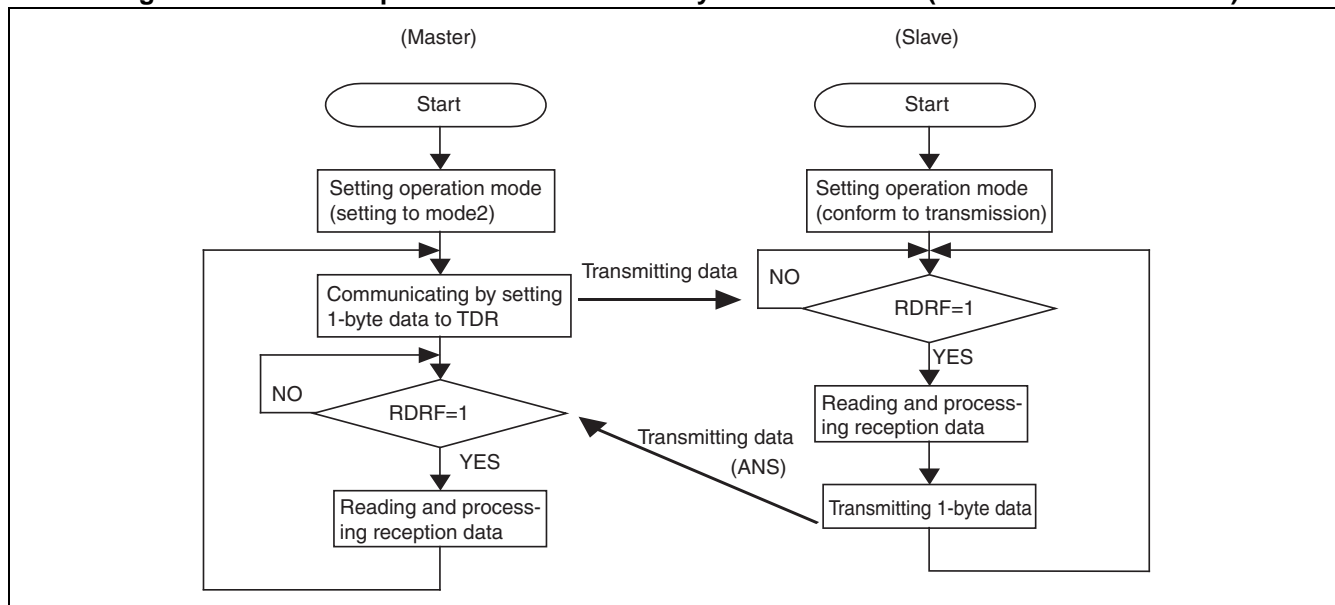
**Figure 24.17-1 Example of Two-way Communication Connection for CSIO
(Clock Synchronous Serial Interface)**



■ Flowchart

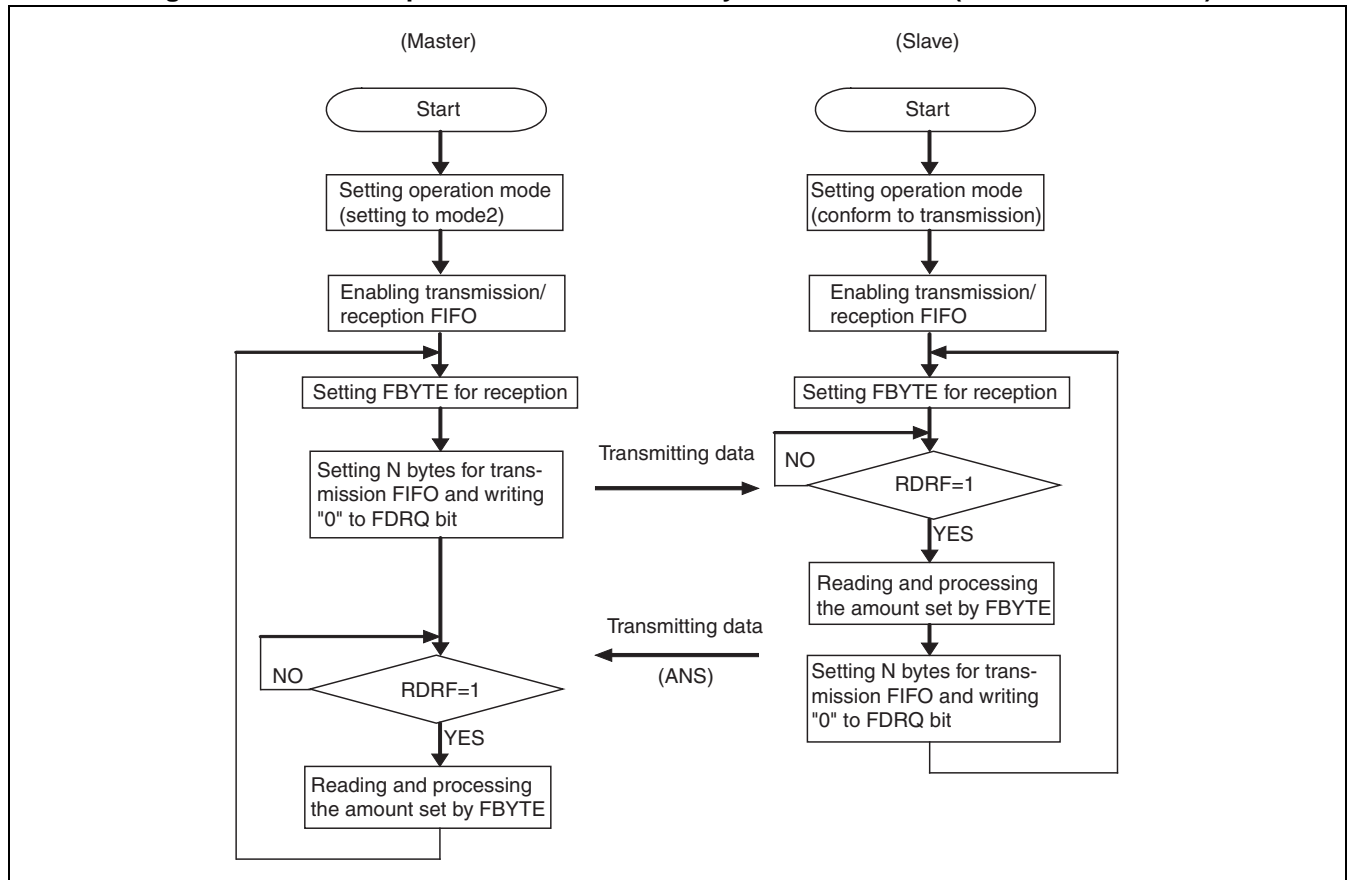
● When FIFO is not used

Figure 24.17-2 Example Flowchart for Two-way Communication (When FIFO is Not Used)



● When FIFO is used

Figure 24.17-3 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)*
- 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.

*: There is no FIFO between ch.8 and ch.11.

24.21 Registers of I²C Interface

This section lists the registers of the I²C interface.

■ List of Registers of I²C Interface

Table 24.21-1 Registers of the I²C (1 / 6)

Channel	Abbreviated Register Name	Register Name	Reference
0	IBCR0	I ² C bus control register 0	24.21.1
	SMR0	Serial mode register 0	24.21.2
	IBSR0	I ² C bus status register 0	24.21.3
	BGR0	Baud rate generator register 0	24.21.8
	SSR0	Serial status register 0	24.21.4
	RDR0	Received data register 0	24.21.5
	TDR0	Transmitted data register 0	24.21.5
	FCR10	FIFO control register 10	24.21.9
	FCR00	FIFO control register 00	24.21.10
	FBYTE10	FIFO1 byte register 0	24.21.11
	FBYTE20	FIFO2 byte register 0	24.21.11
	ISMK0	7-bit slave address mask register 0	24.21.6
	ISBA0	7-bit slave address register 0	24.21.7

Table 24.21-1 Registers of the I²C (2 / 6)

Channel	Abbreviated Register Name	Register Name	Reference
1	IBCR1	I ² C bus control register 1	24.21.1
	SMR1	Serial mode register 1	24.21.2
	IBSR1	I ² C 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
	FCR11	FIFO control register 11	24.21.9
	FCR01	FIFO control register 01	24.21.10
	FBYTE11	FIFO1 byte register 1	24.21.11
	FBYTE21	FIFO2 byte register 2	24.21.11
	ISMK1	7-bit slave address mask register 1	24.21.6
	ISBA1	7-bit slave address register 1	24.21.7
2	IBCR2	I ² C bus control register 2	24.21.1
	SMR2	Serial mode register 2	24.21.2
	IBSR2	I ² C 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
	FCR12	FIFO control register 12	24.21.9
	FCR02	FIFO control register 02	24.21.10
	FBYTE12	FIFO1 byte register 2	24.21.11
	FBYTE22	FIFO2 byte register 2	24.21.11
	ISMK2	7-bit slave address mask register 2	24.21.6
	ISBA2	7-bit slave address register 2	24.21.7

Table 24.21-1 Registers of the I²C (3 / 6)

Channel	Abbreviated Register Name	Register Name	Reference
3	IBCR3	I ² C bus control register 3	24.21.1
	SMR3	Serial mode register 3	24.21.2
	IBSR3	I ² C 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
	FCR13	FIFO control register 13	24.21.9
	FCR03	FIFO control register 03	24.21.10
	FBYTE13	FIFO1 byte register 3	24.21.11
	FBYTE23	FIFO2 byte register 3	24.21.11
	ISMK3	7-bit slave address mask register 3	24.21.6
	ISBA3	7-bit slave address register 3	24.21.7
4	IBCR4	I ² C bus control register 4	24.21.1
	SMR4	Serial mode register 4	24.21.2
	IBSR4	I ² C bus status register 4	24.21.3
	BGR4	Baud rate generator register 4	24.21.8
	SSR4	Serial status register 4	24.21.4
	RDR4	Received data register 4	24.21.5
	TDR4	Transmitted data register 4	24.21.5
	FCR14	FIFO control register 14	24.21.9
	FCR04	FIFO control register 04	24.21.10
	FBYTE14	FIFO1 byte register 4	24.21.11
	FBYTE24	FIFO2 byte register 4	24.21.11
	ISMK4	7-bit slave address mask register 4	24.21.6
	ISBA4	7-bit slave address register 4	24.21.7

Table 24.21-1 Registers of the I²C (4 / 6)

Channel	Abbreviated Register Name	Register Name	Reference
5	IBCR5	I ² C bus control register 5	24.21.1
	SMR5	Serial mode register 5	24.21.2
	IBSR5	I ² C bus status register 5	24.21.3
	BGR5	Baud rate generator register 5	24.21.8
	SSR5	Serial status register 5	24.21.4
	RDR5	Received data register 5	24.21.5
	TDR5	Transmitted data register 5	24.21.5
	FCR15	FIFO control register 15	24.21.9
	FCR05	FIFO control register 05	24.21.10
	FBYTE15	FIFO1 byte register 5	24.21.11
	FBYTE25	FIFO2 byte register 5	24.21.11
	ISMK5	7-bit slave address mask register 5	24.21.6
	ISBA5	7-bit slave address register 5	24.21.7
6	IBCR6	I ² C bus control register 6	24.21.1
	SMR6	Serial mode register 6	24.21.2
	IBSR6	I ² C bus status register 6	24.21.3
	BGR6	Baud rate generator register 6	24.21.8
	SSR6	Serial status register 6	24.21.4
	RDR6	Received data register 6	24.21.5
	TDR6	Transmitted data register 6	24.21.5
	FCR16	FIFO control register 16	24.21.9
	FCR06	FIFO control register 06	24.21.10
	FBYTE16	FIFO1 byte register 6	24.21.11
	FBYTE26	FIFO2 byte register 6	24.21.11
	ISMK6	7-bit slave address mask register 6	24.21.6
	ISBA6	7-bit slave address register 6	24.21.7

Table 24.21-1 Registers of the I²C (5 / 6)

Channel	Abbreviated Register Name	Register Name	Reference
7	IBCR7	I ² C bus control register 7	24.21.1
	SMR7	Serial mode register 7	24.21.2
	IBSR7	I ² C bus status register 7	24.21.3
	BGR7	Baud rate generator register 7	24.21.8
	SSR7	Serial status register 7	24.21.4
	RDR7	Received data register 7	24.21.5
	TDR7	Transmitted data register 7	24.21.5
	FCR17	FIFO control register 17	24.21.9
	FCR07	FIFO control register 07	24.21.10
	FBYTE17	FIFO1 byte register 7	24.21.11
	FBYTE27	FIFO2 byte register 7	24.21.11
	ISMK7	7-bit slave address mask register 7	24.21.6
	ISBA7	7-bit slave address register 7	24.21.7
8	IBCR8	I ² C bus control register 8	24.21.1
	SMR8	Serial mode register 8	24.21.2
	IBSR8	I ² C 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
	ISMK8	7-bit slave address mask register 8	24.21.6
	ISBA8	7-bit slave address register 8	24.21.7

Table 24.21-1 Registers of the I²C (6 / 6)

Channel	Abbreviated Register Name	Register Name	Reference
9	IBCR9	I ² C bus control register 9	24.21.1
	SMR9	Serial mode register 9	24.21.2
	IBSR9	I ² C 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
	ISMK9	7-bit slave address mask register 9	24.21.6
	ISBA9	7-bit slave address register 9	24.21.7
10	IBCRA	I ² C bus control register A	24.21.1
	SMRA	Serial mode register A	24.21.2
	IBSRA	I ² C bus status register A	24.21.3
	BGRA	Baud rate generator register A	24.21.8
	SSRA	Serial status register A	24.21.4
	RDRA	Received data register A	24.21.5
	TDRA	Transmitted data register A	24.21.5
	ISMKA	7-bit slave address mask register A	24.21.6
	ISBAA	7-bit slave address register A	24.21.7
11	IBCRB	I ² C bus control register B	24.21.1
	SMRB	Serial mode register B	24.21.2
	IBSRB	I ² C bus status register B	24.21.3
	BGRB	Baud rate generator register B	24.21.8
	SSRB	Serial status register B	24.21.4
	RDRB	Received data register B	24.21.5
	TDRB	Transmitted data register B	24.21.5
	ISMKB	7-bit slave address mask register B	24.21.6
	ISBAB	7-bit slave address register B	24.21.7

Table 24.21-2 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 24.21-1 shows the bit structure of the I²C bus control register (IBCR), and Table 24.21-3 describes the function of each bit.

Figure 24.21-1 Bit Structure of I²C Bus Control Register (IBCR)

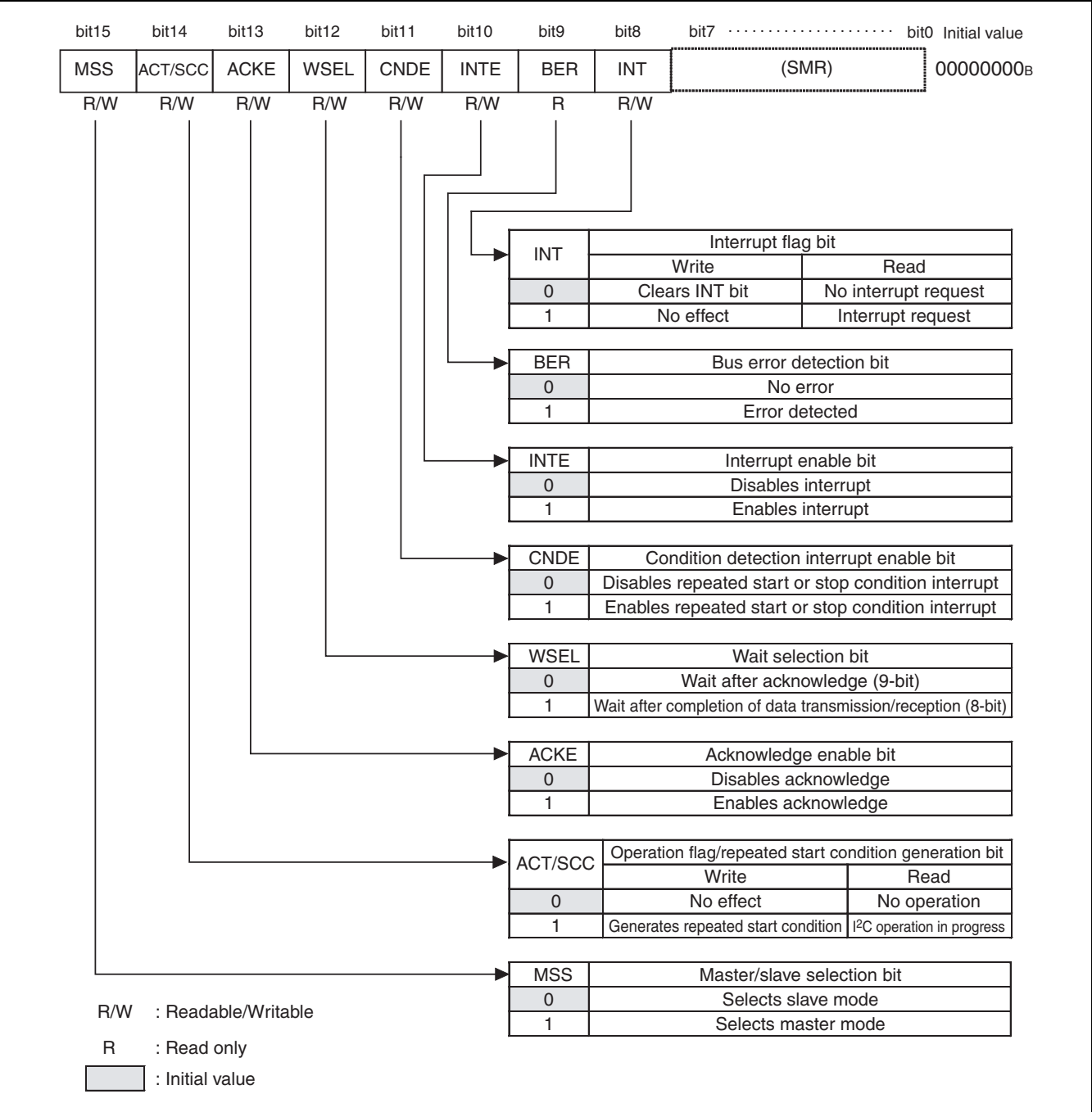


Table 24.21-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (1 / 4)

Bit name		Function															
bit15	MSS: Master/slave selection bit	<ul style="list-style-type: none"> 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". <p>The MSS bit is cleared under the following conditions.</p> <ul style="list-style-type: none"> 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". <p>The relationship between the MSS and ACT bits is shown below.</p> <table border="1"> <thead> <tr> <th>MSS bit</th><th>ACT bit</th><th>Status</th></tr> </thead> <tbody> <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* 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> </tbody> </table> <p>*: ACK response: indicates that SDA of the I²C bus is at "L" during acknowledge.</p> <p>Note:</p> <p>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".</p> <p>Writing "0" to the MSS bit returns "1" during master operation, as long as the ACT bit is set to "1".</p>	MSS bit	ACT bit	Status	0	0	Idle	0	1	Slave operation in progress (slave mode) due to slave address match or ACK response* to reserved address	1	0	Master operation on standby	1	1	Master operation in progress (master mode)
MSS bit	ACT bit	Status															
0	0	Idle															
0	1	Slave operation in progress (slave mode) due to slave address match or ACK response* to reserved address															
1	0	Master operation on standby															
1	1	Master operation in progress (master mode)															

Table 24.21-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (2 / 4)

Bit name		Function				
bit14	ACT/SCC: Operation flag / repeated start condition generation bit	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>	Read	Write	ACT bit	SCC bit
		Read	Write			
ACT bit	SCC bit					
		<p>The ACT bit indicates that the device is operating in master or slave mode.</p> <p>Setting conditions for the ACT bit:</p> <ul style="list-style-type: none">• 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). <p>Reset conditions for the ACT bit:</p> <p><Master mode></p> <ul style="list-style-type: none">• A stop condition is detected.• An arbitration lost condition is detected.• A bus error is detected.• The I²C interface is disabled (EN bit = 0). <p><Slave mode></p> <ul style="list-style-type: none">• 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). <p>A repeated start is performed when "1" is written to this bit during master mode.</p> <p>Writing "0" is invalid.</p> <p>Note:</p> <p>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".</p> <p>In slave mode (MSS = 0, ACT = 1), it is prohibited to write "1" to this bit.</p> <p>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.</p> <p>The SCC bit is read when a read modify write (RMW) instruction is used.</p>				

Table 24.21-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (3 / 4)

Bit name		Function
bit13	ACKE: Acknowledge enable bit	<ul style="list-style-type: none"> 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". <p>This bit is invalid under the following conditions.</p> <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> This bit is used to determine whether an interrupt should occur (INT = 1) before or after an acknowledgement to put the I²C bus in a wait state. The WSEL bit is invalid under the following conditions. <ul style="list-style-type: none"> 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. <p>^{*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 acknowledgement.</p>
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	<p>This bit indicates that an error is detected on the I²C bus.</p> <p>Setting conditions for the BER bit:</p> <ul style="list-style-type: none"> A start condition or stop condition is detected during the transfer of the first byte[*]. A (repeated) start condition or stop condition is detected at the 2nd bit - 9th (acknowledge) bit of data in the second or succeeding byte. <p>Reset conditions for the BER bit:</p> <ul style="list-style-type: none"> "0" is written to the INT bit when BER is set to "1". The I²C interface is disabled (EN = 0). <p>[*]: First byte: indicates the data after (repeated) start condition</p> <p>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.</p>

Table 24.21-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (4 / 4)

bit8	Bit name	Function
	INT: Interrupt flag bit	<p>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.</p> <p>Setting conditions for the INT bit:</p> <p><8th bit></p> <ul style="list-style-type: none"> • 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. <p><9th bit></p> <ul style="list-style-type: none"> • 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. <p><Other condition></p> <p>A bus error is detected.</p> <p>Reset conditions for the INT bit:</p> <ul style="list-style-type: none"> • (1) "0" is written to the INT bit. • (2) "0" is written to the MSS bit when the INT bit is "1" and the ACT bit is "1". • (3) "1" is written to the SCC bit when the INT bit is "1" and the ACT bit is "1". <p>Writing "1" to the INT bit is invalid.</p> <p>Note:</p> <p>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.</p> <p>"1" is read when a read modify write (RMW) instruction is used.</p> <p>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.</p>

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 24.21-2 shows the bit structure of the serial mode register (SMR), and Table 24.21-4 describes the function of each bit.

Figure 24.21-2 Bit Structure of Serial Mode Register (SMR)

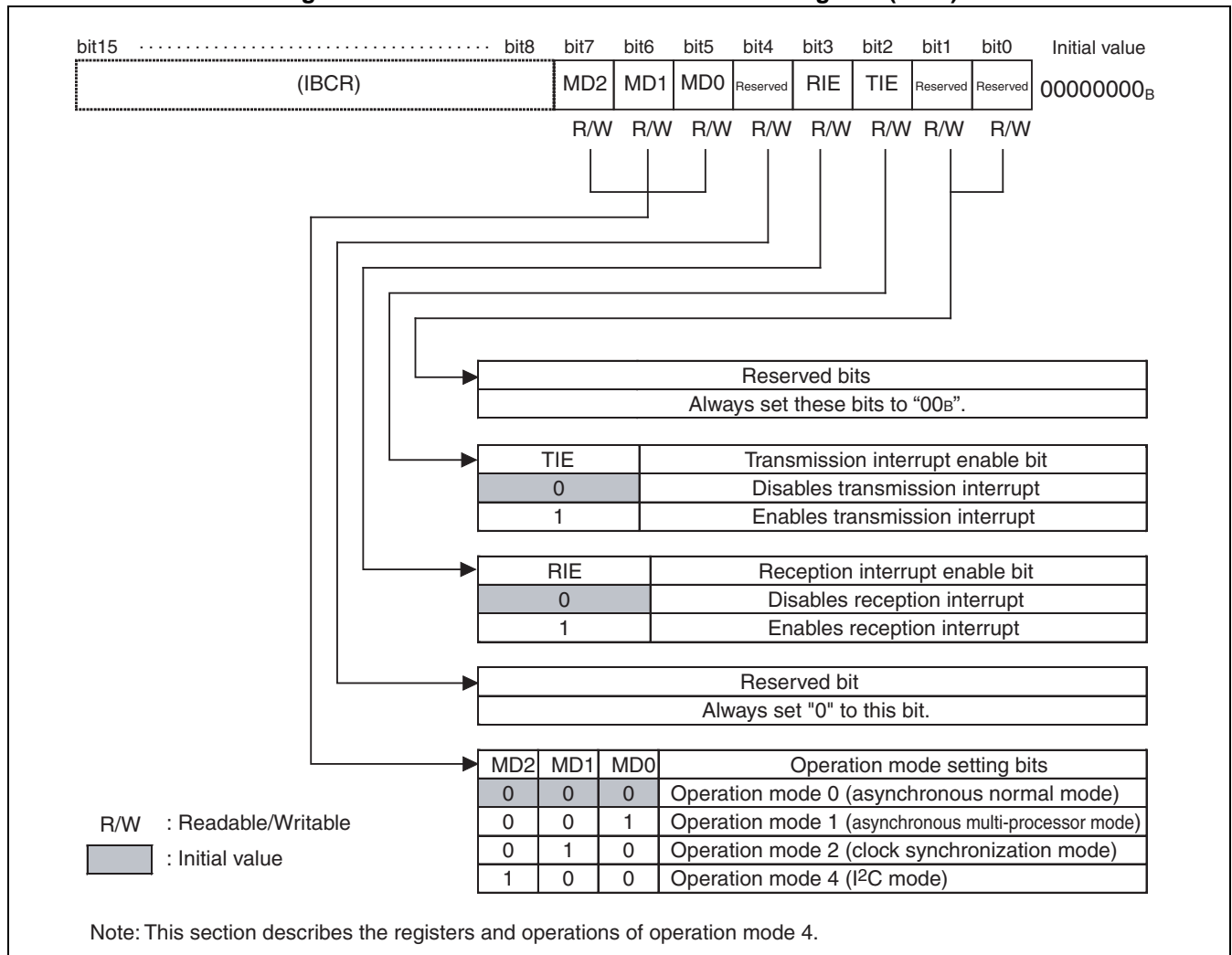


Table 24.21-4 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	<p>These bits are used to select the operation mode.</p> <p>"000_B": Selects operation mode 0 (asynchronous normal mode)</p> <p>"001_B": Selects operation mode 1 (asynchronous multi-processor mode)</p> <p>"010_B": Selects operation mode 2 (clock synchronization mode)</p> <p>"100_B": Selects operation mode 4 (I²C mode)</p> <p>This section describes the registers and operations of operation mode 4 (I²C mode).</p> <p>Note:</p> <p>Settings other than above are prohibited.</p> <p>To switch the operation mode, disable I²C first (ISMK:EN = 0).</p> <p>Set each register after selecting the operation mode.</p>
bit4	Reserved bit	Always set "0" to this bit.
bit3	RIE: Reception interrupt enable bit	<ul style="list-style-type: none"> 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". <p>Note:</p> <p>Set this bit to "0" when receiving data using the INT bit in the I²C bus control register (IBCR).</p>
bit2	TIE: Transmission interrupt enable bit	<ul style="list-style-type: none"> 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". <p>Note:</p> <p>Set this bit to "0" when transmitting data using the INT bit in the I²C bus control register (IBCR).</p>
bit1, bit0	Reserved bits	Always set these bits to "00 _B ".

<Note>

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. 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 24.21-3 shows the bit structure of the I²C bus status register (IBSR) and Table 24.21-5 describes the function of each bit.

Figure 24.21-3 Bit Structure of I²C Bus Status Register (IBSR)

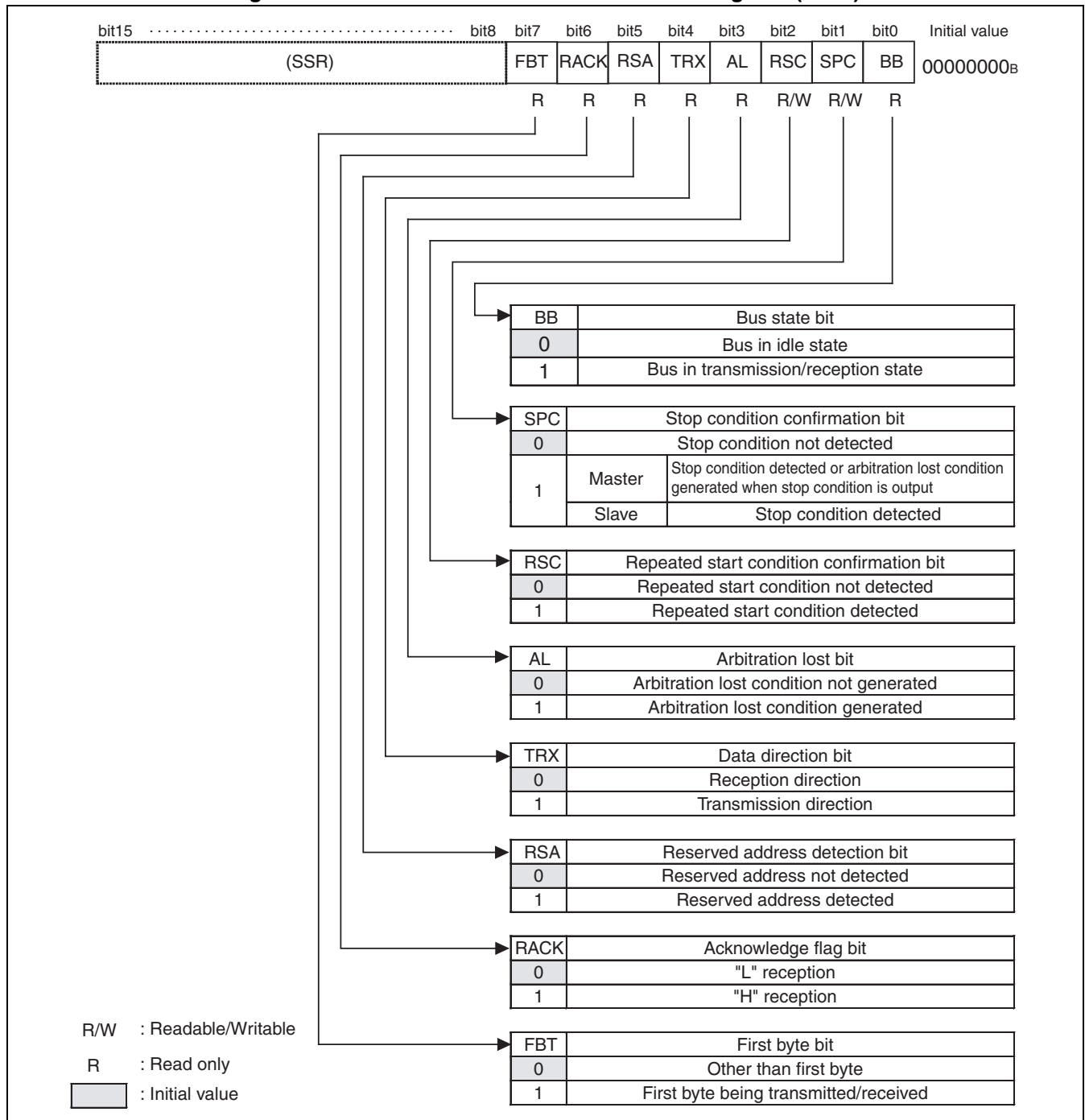


Table 24.21-5 Functional Description of Each Bit of I²C Bus Status Register (1 / 3)

Bit name		Function
bit7	FBT: First byte bit	<p>This bit indicates the first byte.</p> <p>Setting condition for the FBT bit: A (repeated) start condition is detected.</p> <p>Clearing conditions for the FBT bit:</p> <ol style="list-style-type: none"> (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	<p>This bit is used to indicate the acknowledge received for the first byte during master or slave mode.</p> <p>Update conditions for the RACK bit</p> <ol style="list-style-type: none"> (1) Acknowledge for the first byte (2) Acknowledge for data in master or slave mode <p>Clearing conditions for the RACK bit (RACK bit = 0)</p> <ol style="list-style-type: none"> (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	<p>This bit indicates the detection of a reserved address.</p> <p>Setting condition for the RSA bit (RSA = 1) The first byte is set to "0000XXXX" or "1111XXXX". "X" can be "0" or "1".</p> <p>Reset conditions for the RSA bit (RSA = 0)</p> <ol style="list-style-type: none"> (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). <p>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.</p> <p>Note:</p> <p>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.</p> <p>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".</p>

Table 24.21-5 Functional Description of Each Bit of I²C Bus Status Register (2 / 3)

Bit name		Function
bit4	TRX: Data direction bit	<p>This bit indicates the data direction.</p> <p>Setting conditions for the TRX bit:</p> <ul style="list-style-type: none"> (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). <p>Reset conditions for the TRX bit:</p> <ul style="list-style-type: none"> (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).
bit3	AL: Arbitration lost bit	<p>This bit indicates an arbitration lost condition.</p> <p>Setting conditions for the AL bit:</p> <ul style="list-style-type: none"> (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. <p>Reset conditions for the AL bit:</p> <ul style="list-style-type: none"> (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).

Table 24.21-5 Functional Description of Each Bit of I²C Bus Status Register (3 / 3)

Bit name		Function
bit2	RSC: Repeated start condition confirmation bit	<p>This bit indicates that a repeated start condition has been detected in master or slave mode.</p> <p>Setting condition for the RSC bit: A repeated start condition is detected after acknowledgement during slave or master mode operation.</p> <p>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.</p> <p>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.</p>
bit1	SPC: Stop condition confirmation bit	<p>This bit indicates that a stop condition has been detected in master or slave mode.</p> <p>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.</p> <p>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.</p> <p>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.</p>
bit0	BB: Bus state bit	<p>This bit indicates the bus state.</p> <p>Setting condition for the BB bit: "L" is detected at SDA or SCL of the I²C bus.</p> <p>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).</p>

24.21.4 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status.

■ Serial Status Register (SSR)

Figure 24.21-4 shows the bit structure of the serial status register (SSR) and Table 24.21-6 describes the function of each bit.

Figure 24.21-4 Bit Structure of Serial Status Register (SSR)

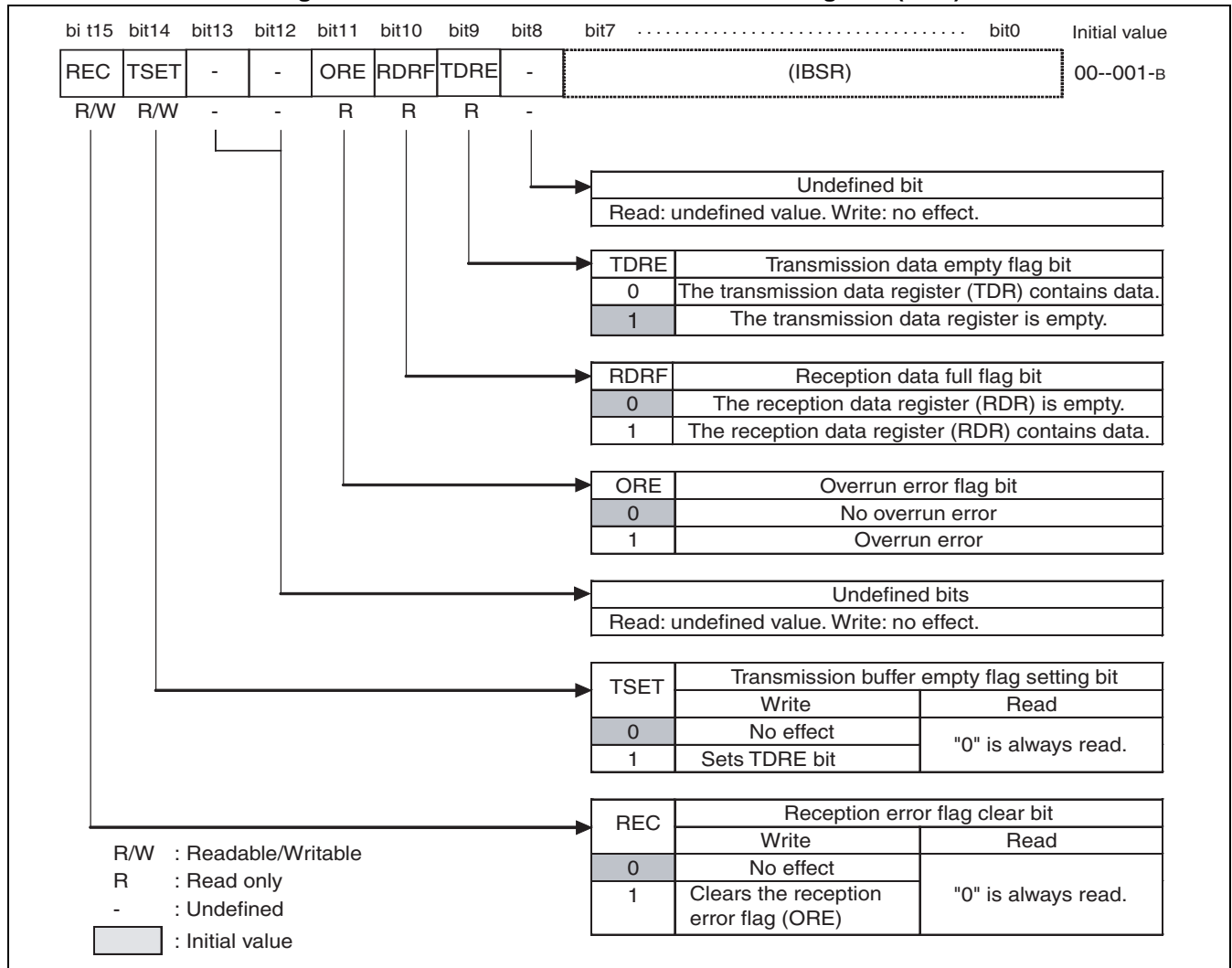


Table 24.21-6 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

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). <ul style="list-style-type: none"> 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). <ul style="list-style-type: none"> Writing "1" sets the TDRE bit. Writing "0" has no effect. Reading this bit always returns "0".

Table 24.21-6 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit13, bit12	Undefined bits	Read: undefined value Write: no effect
bit11	ORE: Overrun error flag bit	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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. <p>Note: NACK response: indicates that SDA of the I²C bus is at "H" during acknowledge.</p>
bit9	TDRE: Transmission data empty flag bit	<ul style="list-style-type: none"> 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

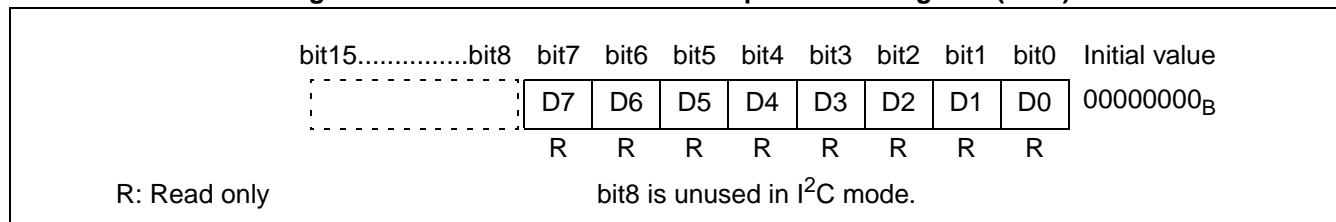
24.21.5 Reception Data Register / Transmission Data Register (RDR/TDR)

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 24.21-5 illustrates the bit structure of the serial reception register (RDR).

Figure 24.21-5 Bit Structure of Reception Data Register (RDR)



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* 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.

*: Indicates the data after a (repeated) start condition.

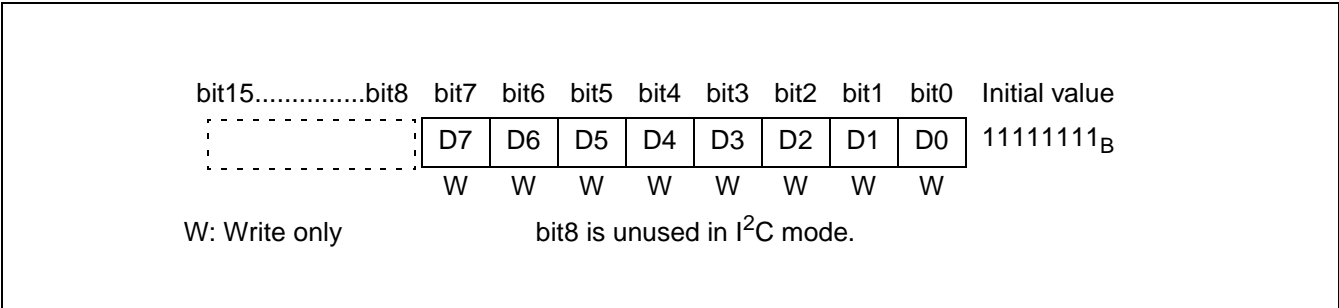
<Notes>

- 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 24.21-6 illustrates the bit structure of the transmission data register.

Figure 24.21-6 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 acknowledgement).
- 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 24.21-7 shows the bit structure of the 7-bit slave address mask register (ISMK) and Table 24.21-7 describes the function of each bit.

Figure 24.21-7 Bit Structure of 7-bit Slave Address Mask Register (ISMK)

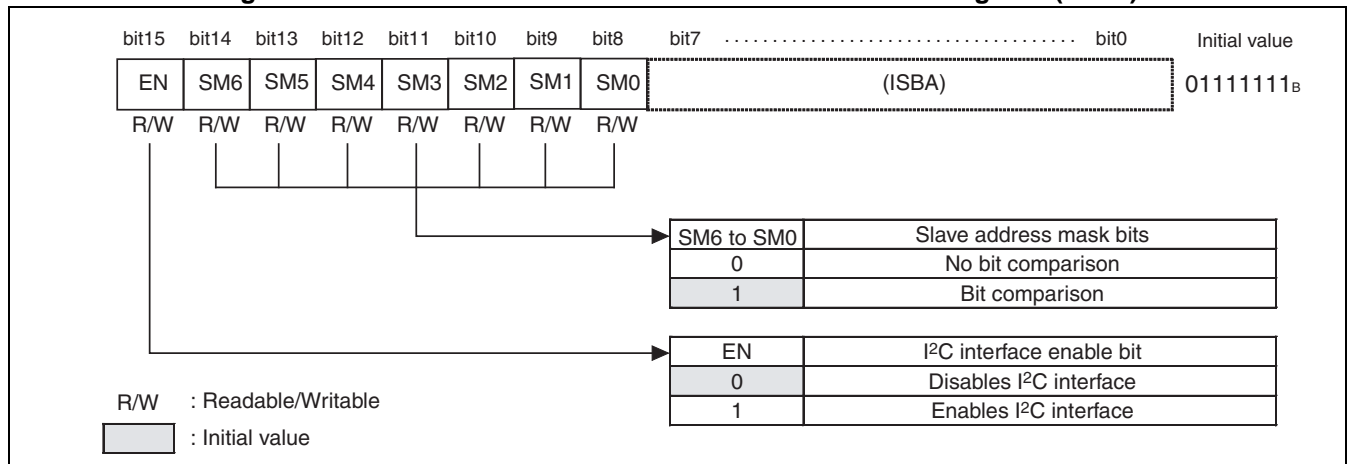


Table 24.21-7 Functional Description of Each Bit of 7-bit Slave Address Mask Register (ISMK)

Bit name		Function
bit15	EN: I ² C interface enable bit	<p>This bit is used to enable or disable the operation of the I²C interface.</p> <p>Setting the bit to "0" disables the operation of the I²C interface.</p> <p>Setting the bit to "1" enables the operation of the I²C interface.</p> <p>Note:</p> <p>This bit is not cleared to "0" even when the BER bit in the IBSR register is set to "1".</p> <p>Set the baud rate generator when this bit is set to "0".</p> <p>Set a 7-bit slave address and the 7-bit slave mask register when this bit is set to "0".</p> <p>Setting the EN bit to "0" during transmission may generate a pulse at SDA/SCL of the I²C bus.</p> <p>If FIFO has been enabled, write "0" to the EN bit after disabling FIFO.</p>
bit14 to bit8	SM6 to SM0: Slave address mask bits	<p>These bits are used to determine whether to compare the 7-bit slave address with the received address.</p> <p>Bit set to "1": compared</p> <p>Bit set to "0": handled as matched</p> <p>Note:</p> <p>Set this register when the EN bit is "0".</p>

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 24.21-8 shows the bit structure of the 7-bit slave address register (ISBA) and Table 24.21-8 describes the function of each bit.

Figure 24.21-8 Bit Structure of 7-bit Slave Address Register (ISBA)

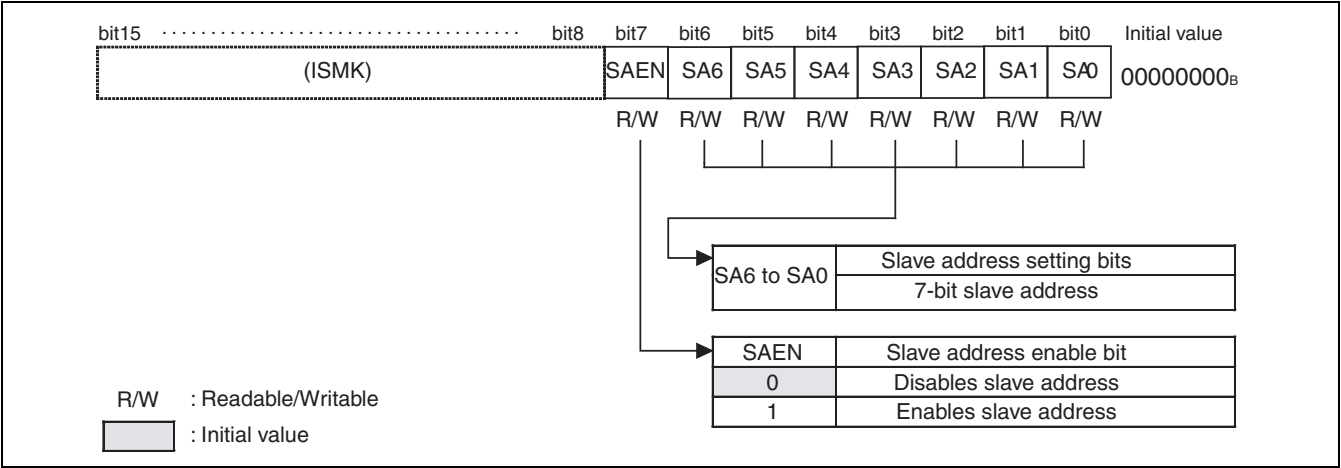


Table 24.21-8 Functional Description of Each Bit of 7-bit Slave Address Register (ISBA)

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".

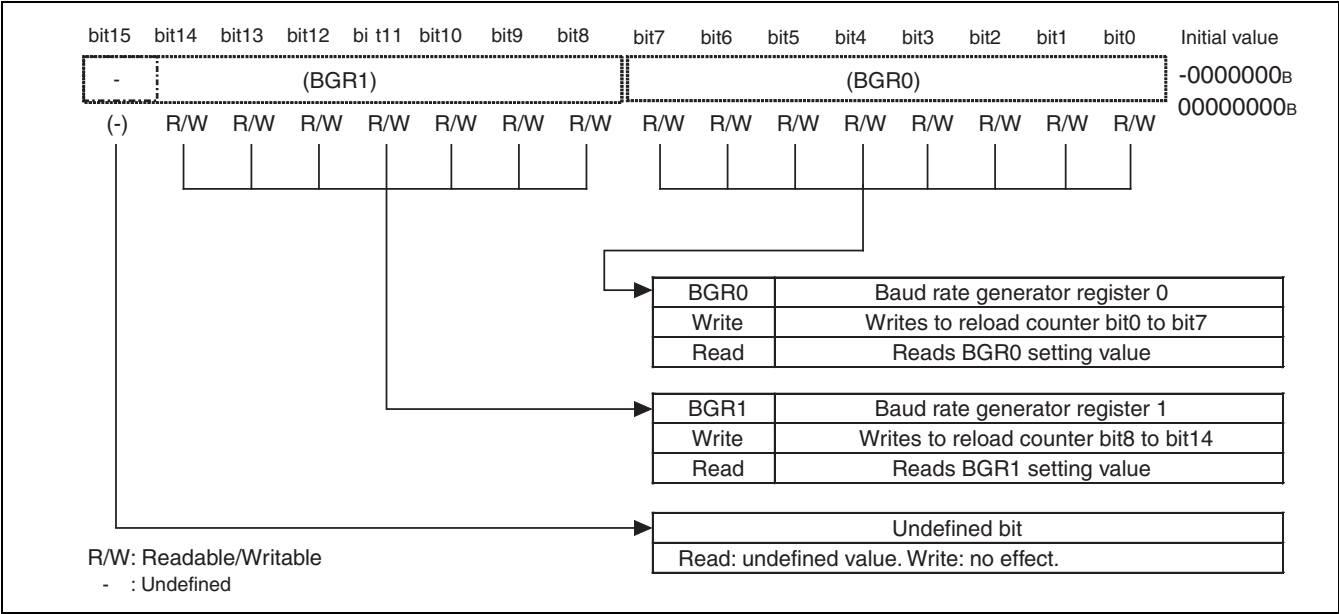
24.21.8 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

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 24.21-9 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 24.21-9 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 (I²C 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 24.21-10 shows the bit structure of the FIFO control register 1 (FCR1) and Table 24.21-9 describes the function of each bit.

Figure 24.21-10 Bit Structure of FIFO Control Register 1 (FCR1)

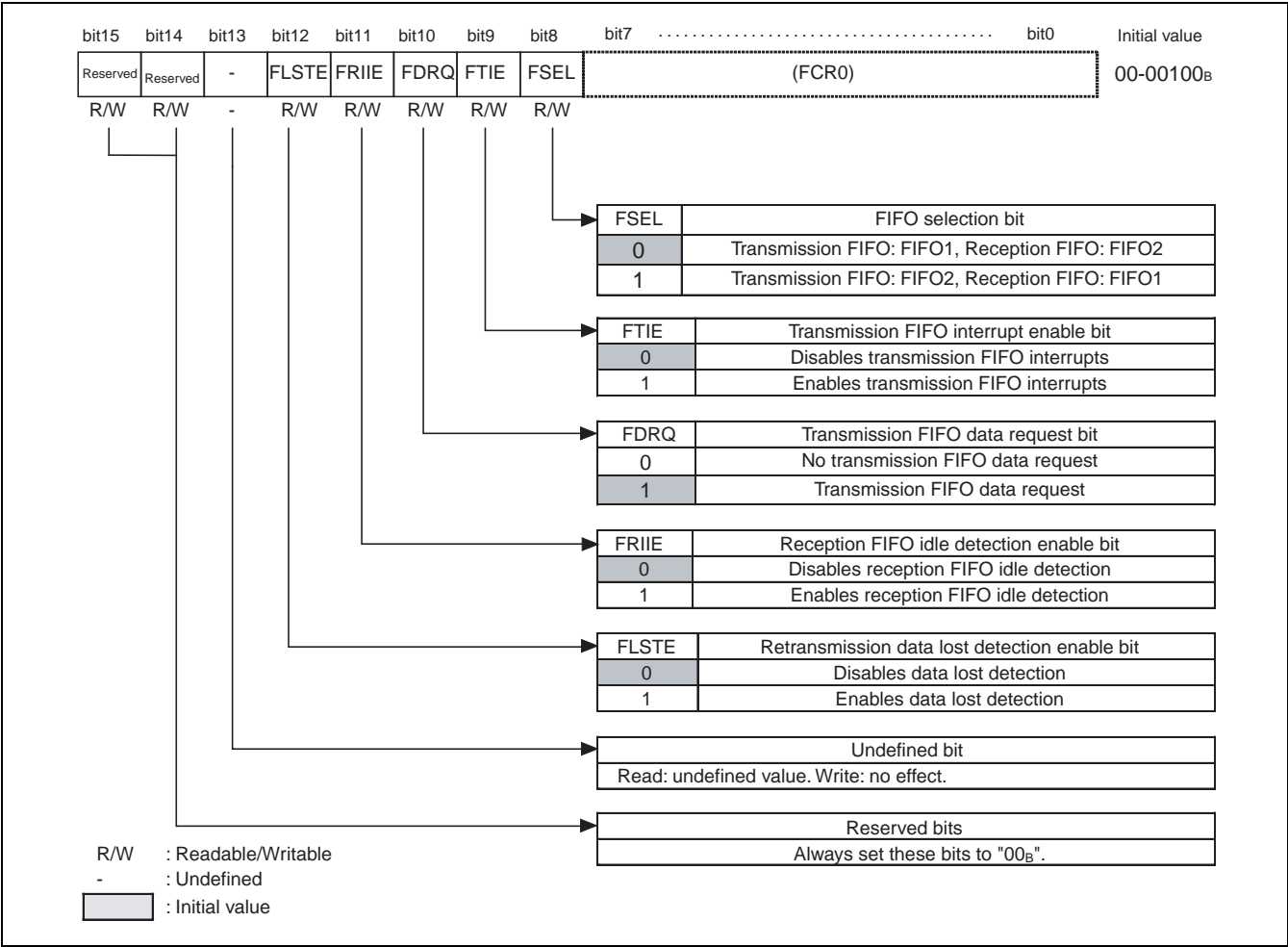


Table 24.21-9 Functional Description of Each Bit of FIFO Control Register 1 (FCR1) (1 / 2)

Bit name		Function
bit15, bit14	Reserved bits	Always set these bits to "00 _B ".
bit13	Undefined bit	Read: undefined value Write: no effect

**Table 24.21-9 Functional Description of Each Bit of FIFO Control Register 1
(FCR1) (2 / 2)**

Bit name		Function
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 <ul style="list-style-type: none"> • FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) • Transmission FIFO reset FDRQ reset condition <ul style="list-style-type: none"> • 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 24.21-11 shows the bit structure of the FIFO control register 0 (FCR0) and Table 24.21-10 describes the function of each bit.

Figure 24.21-11 Bit Structure of FIFO Control Register 0 (FCR0)

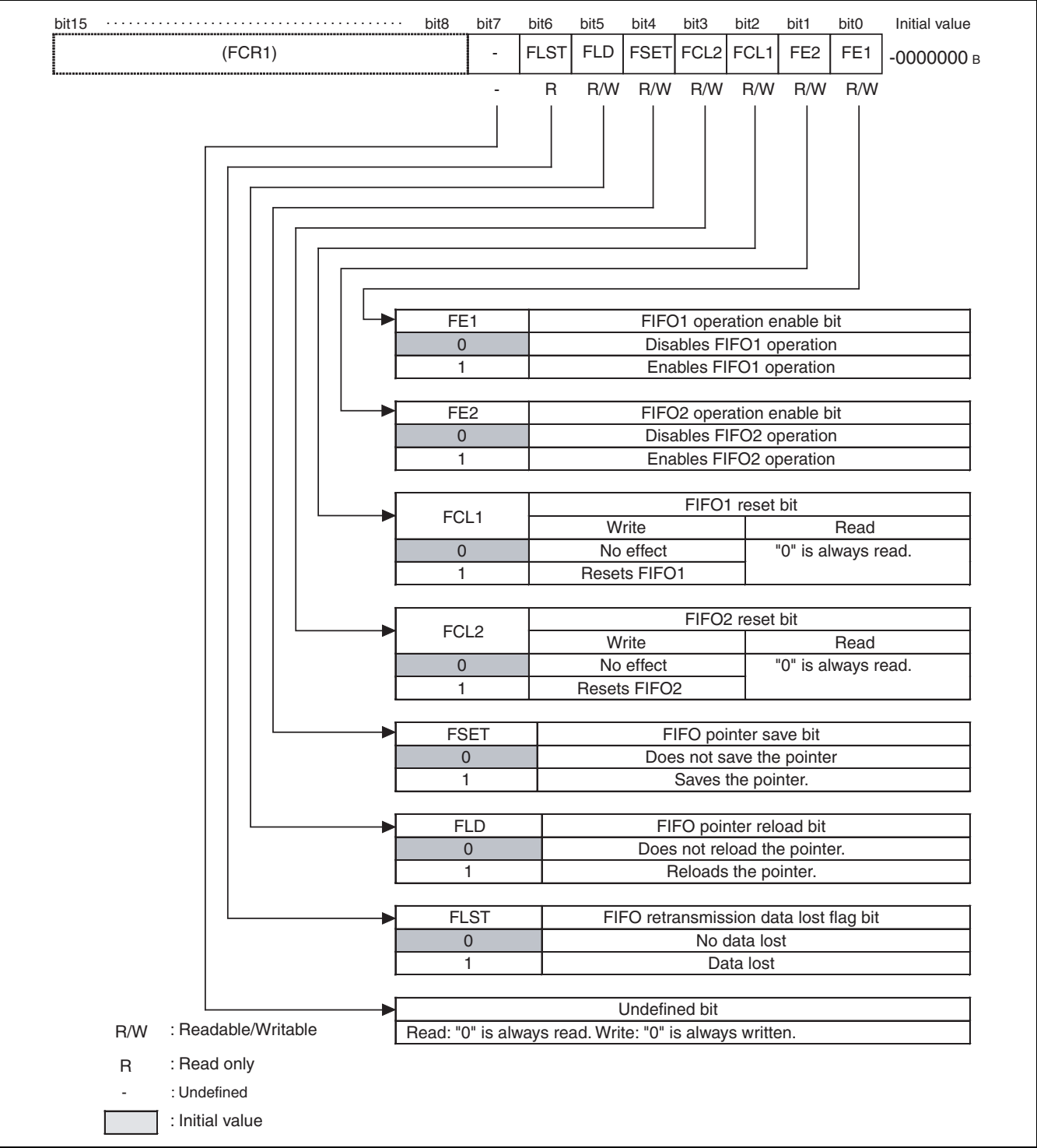


Table 24.21-10 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 3)

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 <ul style="list-style-type: none"> 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".

**Table 24.21-10 Functional Description of Each Bit of FIFO Control Register 0
(FCR0) (2 / 3)**

Bit name		Function
bit2	FCL1: FIFO1 reset bit	<p>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.</p> <p>Note:</p> <ul style="list-style-type: none"> 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	<p>This bit is used to enable/disable FIFO2 operation.</p> <ul style="list-style-type: none"> 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. <p>Note:</p> <p>Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1".</p> <p>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".</p> <p>If 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".</p> <p>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.</p>

Table 24.21-10 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (3 / 3)

Bit name		Function
bit0	FE1: FIFO1 operation enable bit	<p>This bit is used to enable/disable FIFO1 operation.</p> <ul style="list-style-type: none"> 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. <p>Note:</p> <p>Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1".</p> <p>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".</p> <p>If 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".</p> <p>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.</p>

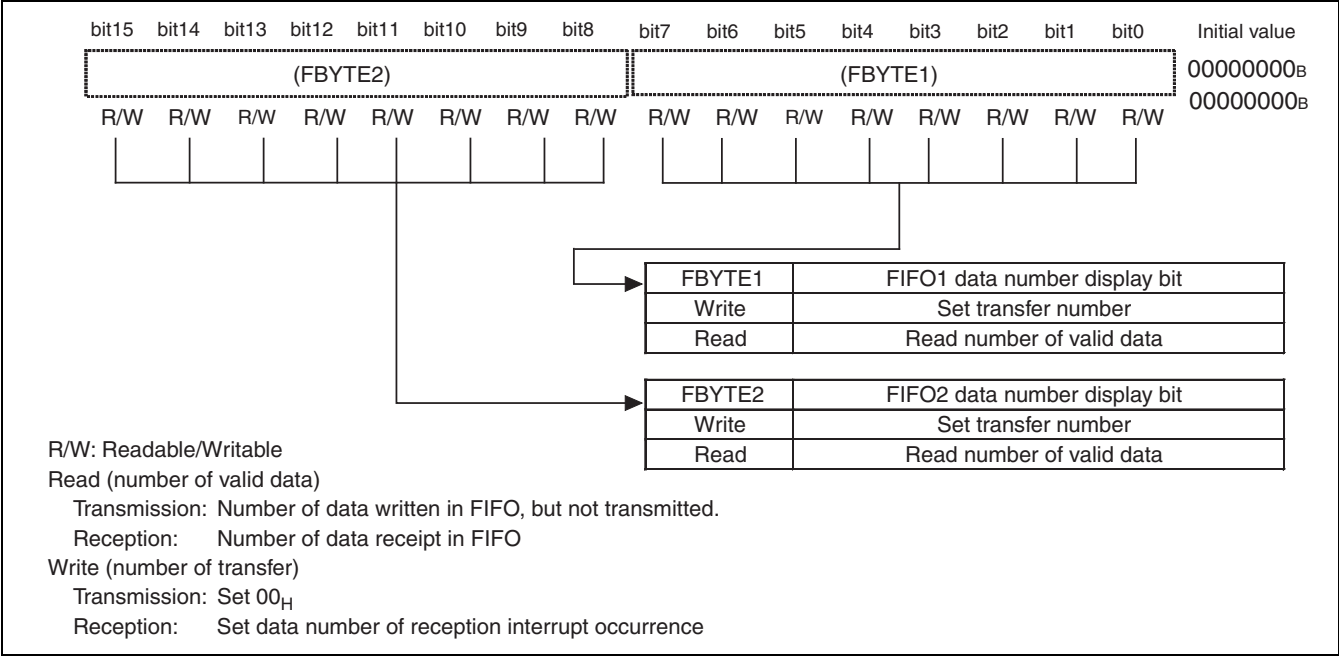
24.21.11 FIFO Byte Register (FBYTE1/FBYTE2)

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 24.21-12 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 24.21-12 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 24.21-11 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 "08H".
- 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 I²C 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 24.22-1 shows the interrupt control bits and interrupt sources of the I²C interface.

Table 24.22-1 Interrupt Control Bits and Interrupt Sources of I²C Interface (1 / 2)

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)

Table 24.22-1 Interrupt Control Bits and Interrupt Sources of I²C Interface (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
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.
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".

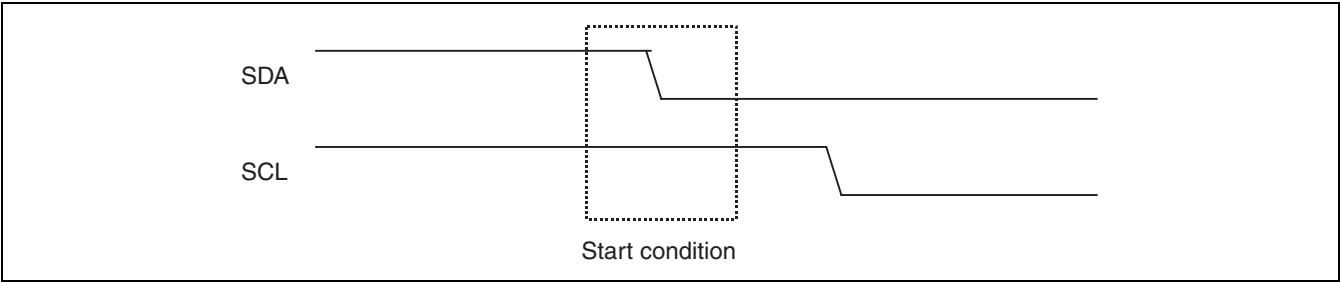
24.22.1 Operation of I²C Interface Communication

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.

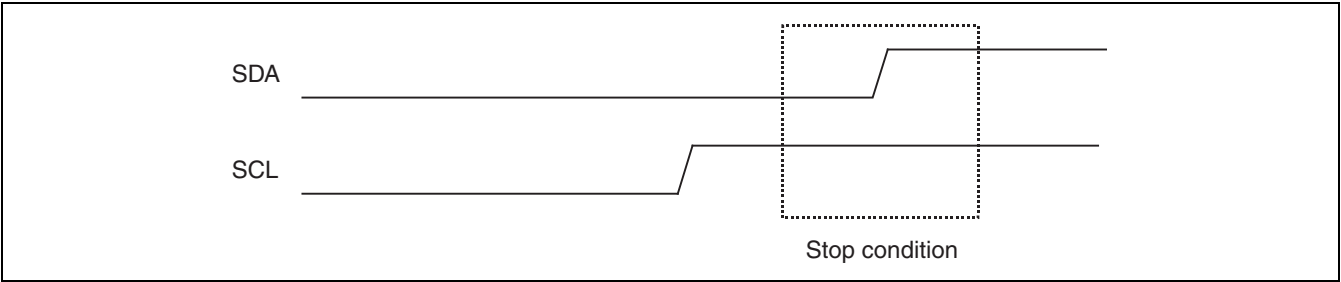
Figure 24.22-1 Start Condition



■ Stop Condition for I²C Bus

The stop condition for the I²C bus is shown below.

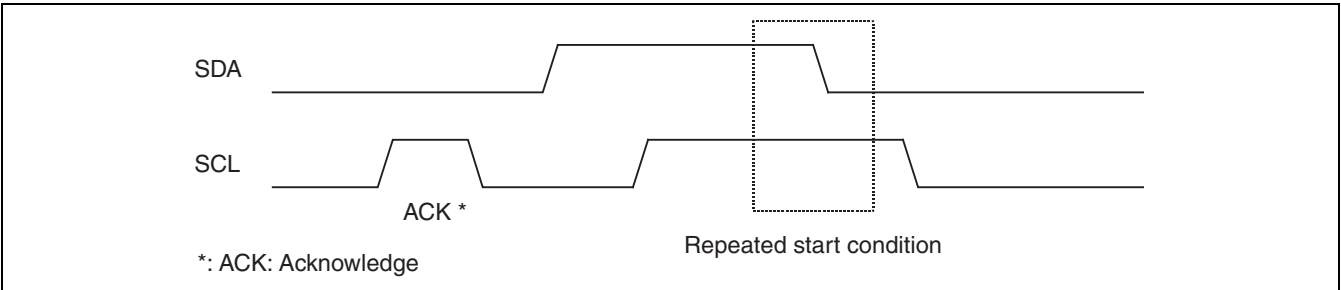
Figure 24.22-2 Stop Condition



■ Repeated Start Condition for I²C Bus

The repeated start condition for the I²C bus is shown below.

Figure 24.22-3 Repeated Start Condition



24.22.2 Master Mode

Master mode generates a start condition for the I²C bus and outputs a clock to the I²C 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 I²C bus is in an idle state (SCL = "H", SDA = "H").

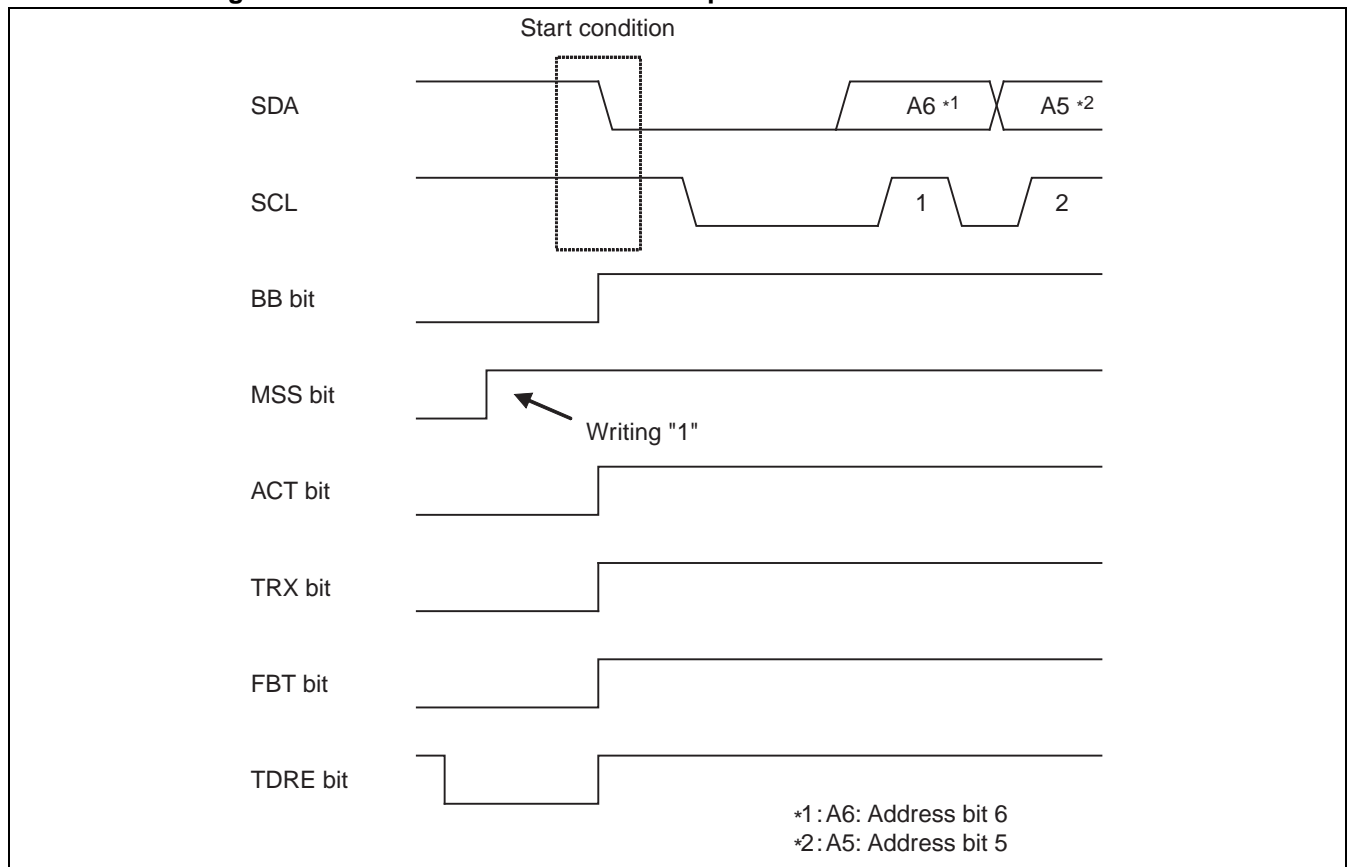
■ Generating a Start Condition

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 I²C bus sets the ACT bit to "1". After that, the BB bit is set to "1", indicating that the I²C bus is in the middle of communication, once the start condition is received (see Figure 24.22-4).

Figure 24.22-4 Correlation between Output of Start Condition and Each Bit



<Note>

Use the peripheral clock (PCLK) at 8 MHz or higher in operation mode 4 (I²C 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 24.22-5 and Figure 24.22-6 show the address and data direction output timings.

Figure 24.22-5 Address and Data Direction (When FIFO is Disabled)

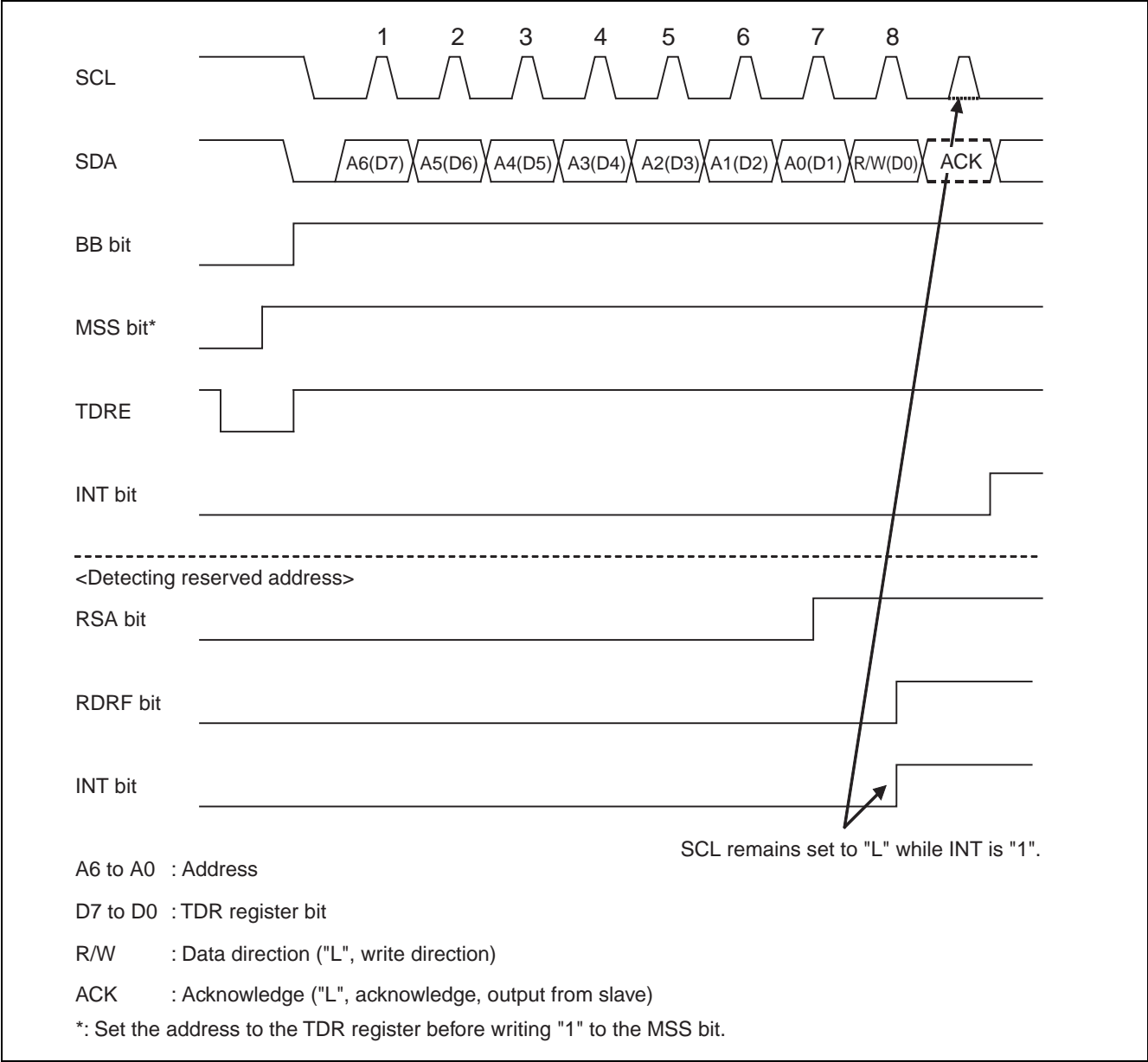
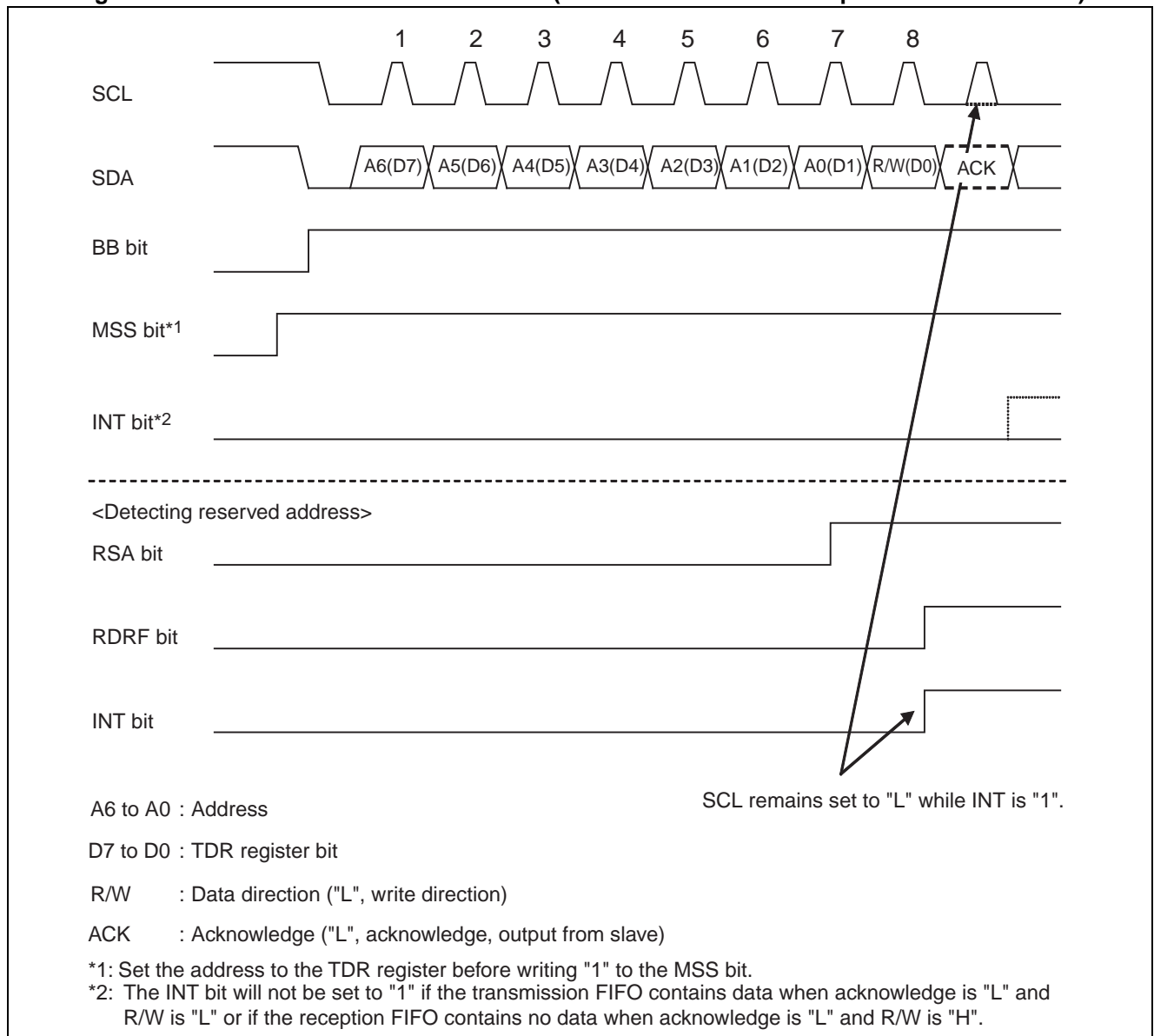


Figure 24.22-6 Address and Data Direction (When Transmission/Reception FIFO is Enabled)



■ Receiving Acknowledge after Transmitting 1st Byte

The I²C 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 24.22-2 Operations after Reception of Acknowledge (RSA Bit = 0)

Trans- mission FIFO	Reception FIFO	Trans- mission 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.	The INT bit is set to "1", causing a wait
				1	When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	
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.	The INT bit is set to "1", causing a wait
				1	When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	
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	

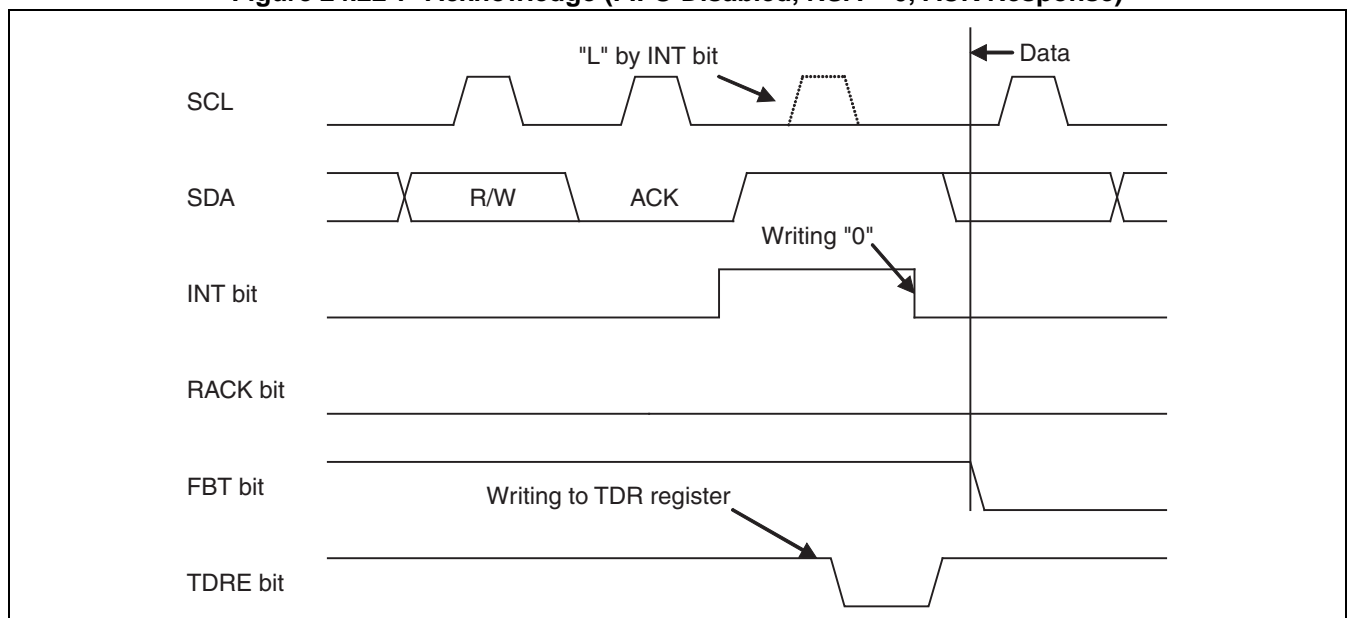
● FIFO disabled (both transmission FIFO and reception FIFO disabled)

- 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".
 - For transmission to the slave (data direction bit = 0), data including a slave address should be set to the transmission FIFO.
 - For reception of data from the slave (data direction bit = 1)
 1. Set the number of receptions to the FIFO byte setting register, after writing the slave address and data direction bit to the transmission data register.
 2. Write the dummy data for the number of data elements to be received to the transmission data register, after writing the slave address and data direction bit to the transmission data register.
- 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 24.22-7 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 24.22-8 Acknowledge (FIFO Disabled, RSA = 0, NACK Response)

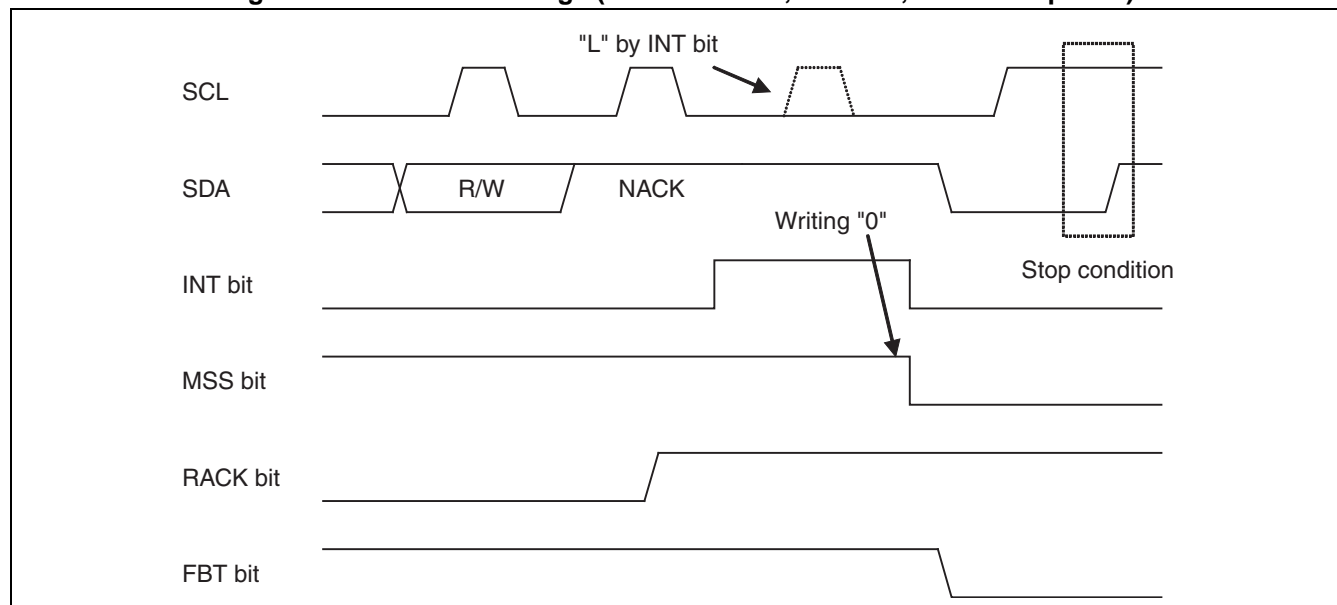


Figure 24.22-9 Acknowledge (FIFO Disabled, RSA = 1, ACK Response)

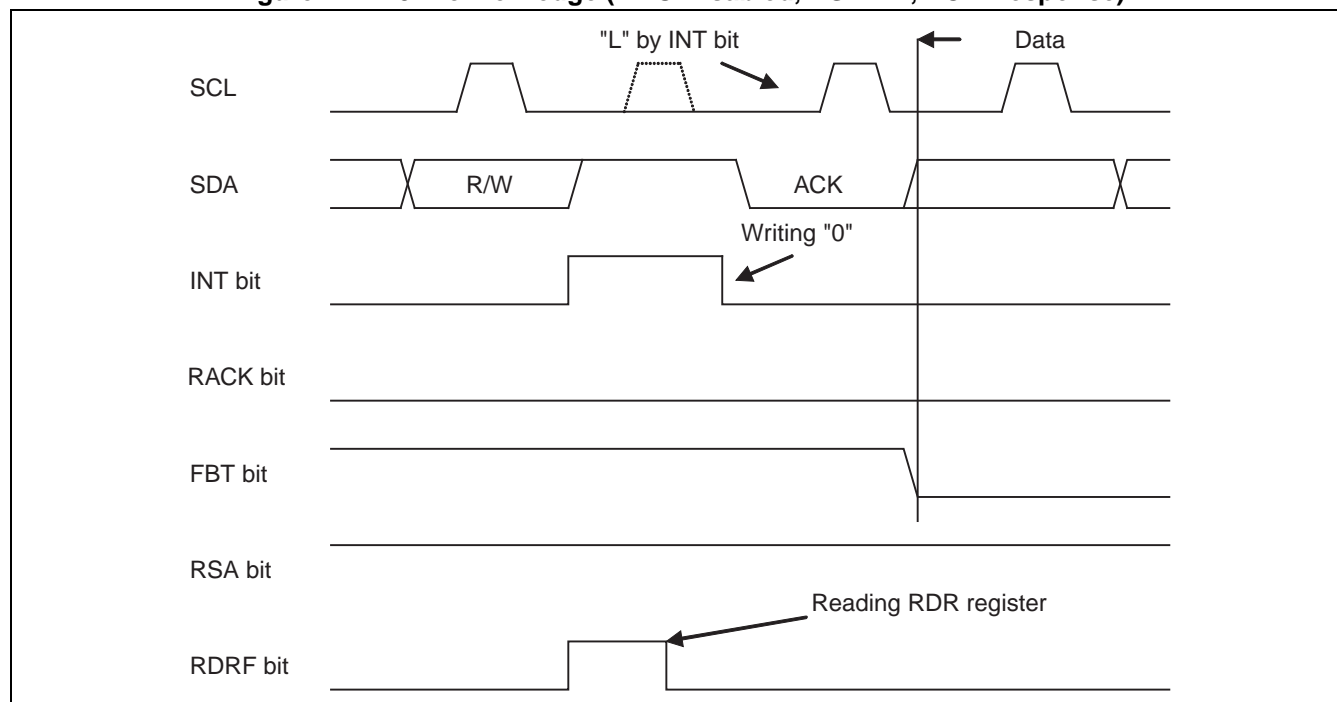


Figure 24.22-10 Acknowledge (FIFO Disabled, RSA = 1, NACK Response)

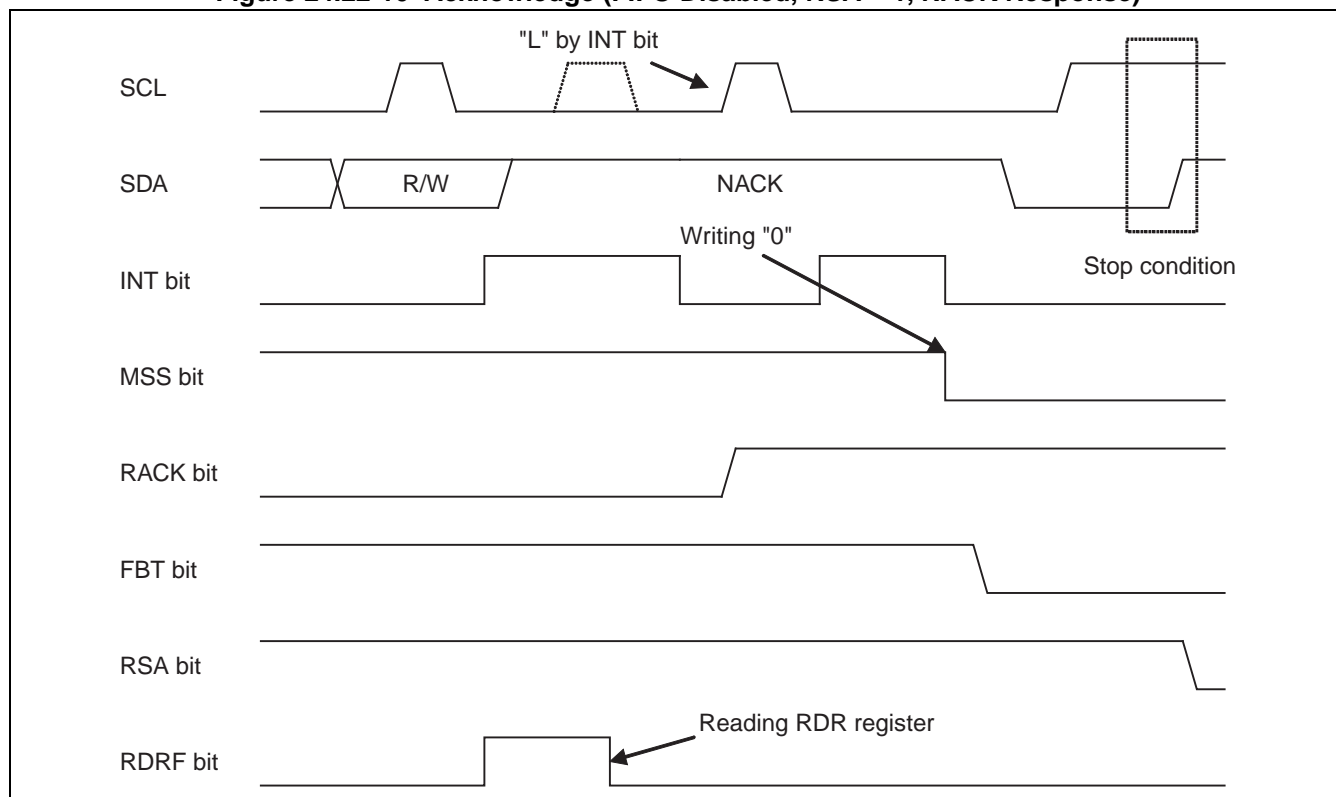
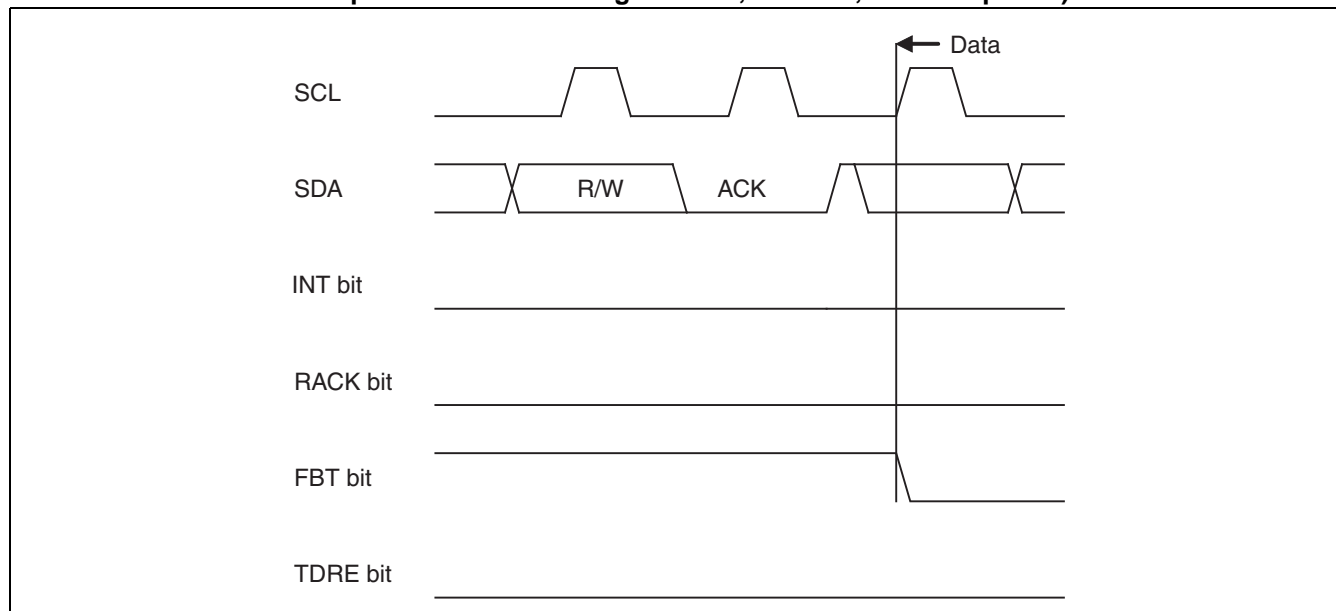


Figure 24.22-11 Acknowledge (FIFO Enabled, Transmission FIFO Containing Data, Reception FIFO Containing No Data, RSA = 0, ACK Response)



■ 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 24.22-3 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 I²C bus.
 - (5) Put the I²C 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 cancelled 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 I²C 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 I²C bus in a wait.
 - (4) Write "0" to the MSS bit to generate a stop condition.

● Transmitting data to reserved address

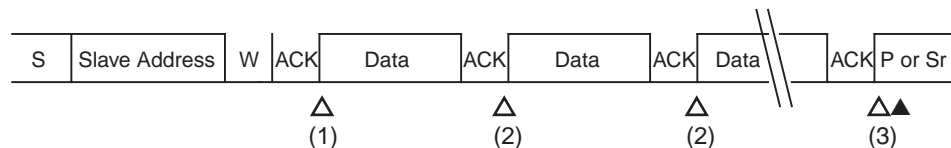
- When transmission FIFO is disabled:
 - (1) Set the reserved address to the TDR register as the slave address and write "1" to the MSS bit.
 - (2) The interrupt flag (INT) will be set to "1" once the slave address has been transmitted.
 - (3) Read from the RDR register to check the reserved address.*
 - (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 I²C bus.
 - (6) Put the I²C 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 cancelled 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.*
 - (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 I²C 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 I²C 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.

*: 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 24.22-12 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

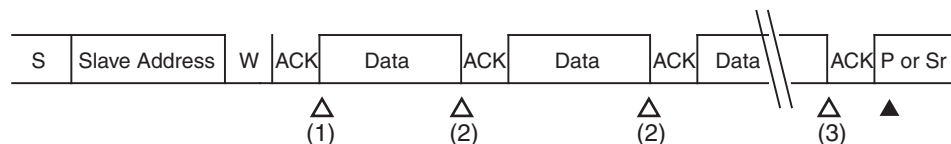
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register

(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 24.22-13 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

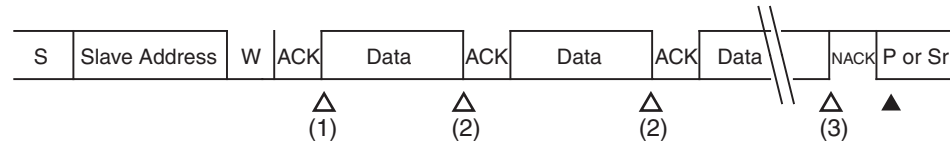
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register

(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 24.22-14 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

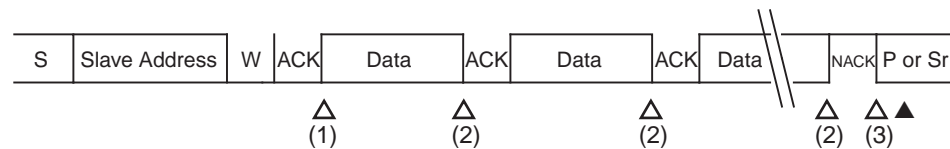
Writing "0" to INT after writing transmission data to TDR register

(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 24.22-15 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

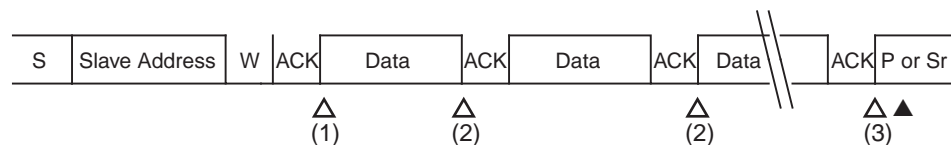
Writing "0" to INT after writing transmission data to TDR register

(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 24.22-16 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

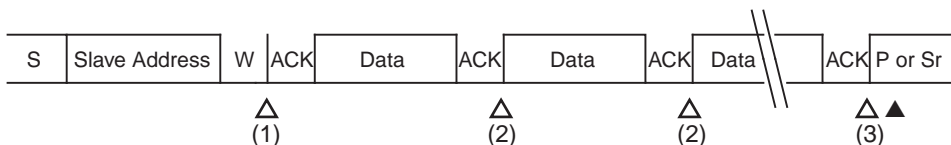
Writing "0" to WSEL and INT after writing transmission data to the transmission buffer

(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 24.22-17 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

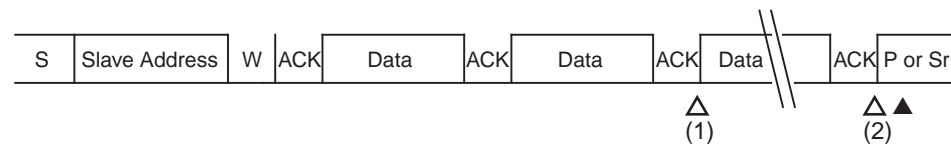
Writing "0" to INT after writing transmission data to TDR register

(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 24.22-18 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

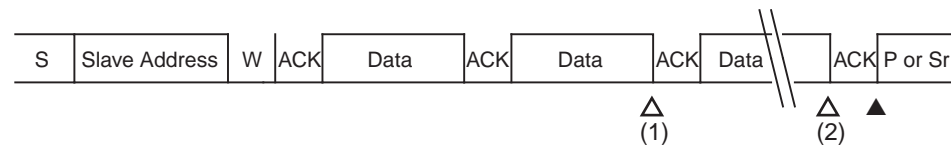
- Writing "0" to INT after writing transmission data to transmission FIFO

(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 24.22-19 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

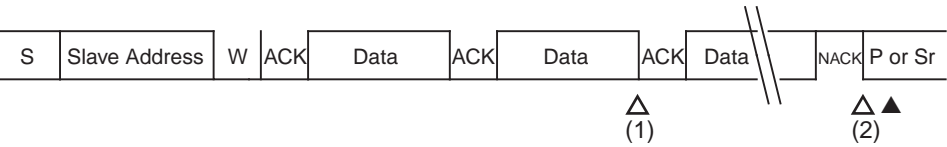
(1) Interrupt generated by the empty state of transmission FIFO

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 24.22-20 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 I²C 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 24.22-4 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.

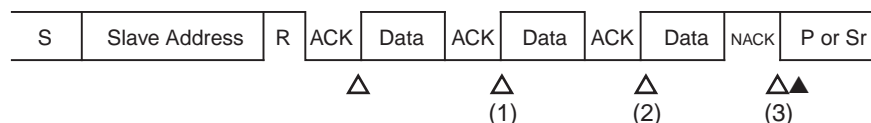
- When reception FIFO is disabled:
 - Set the slave address (including the data direction bit) to the TDR register and write "1" to the MSS bit.
 - An ACK will be received after the slave address is transmitted, and then the interrupt flag (INT) will be set to "1".
 - Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel a wait for the I²C bus.

- (4) Put the I²C 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 I²C 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 24.22-21 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 24.22-22 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 24.22-23 Master Reception Interrupt (3) - when FIFO is Enabled (WSEL = 0, ACKE = 0, RSA = 0)



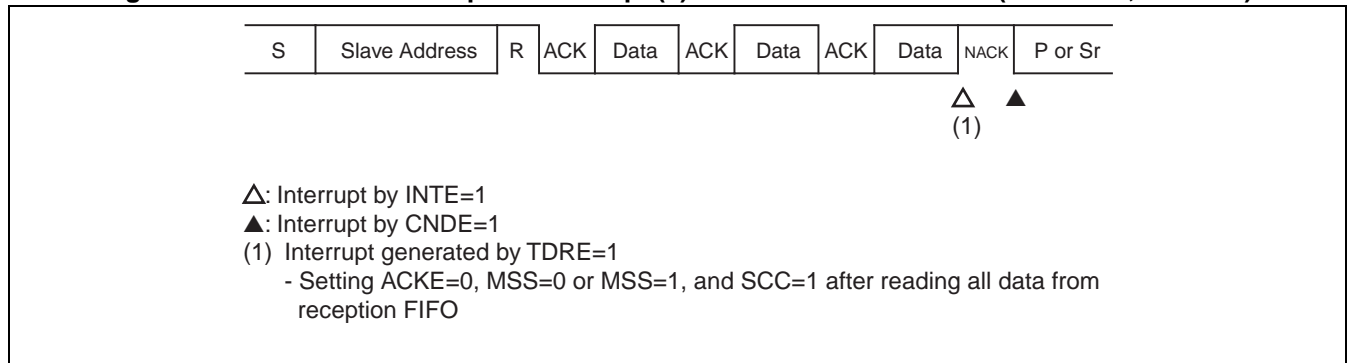
△: Interrupt by INTE=1

▲: 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 24.22-24 Master Reception Interrupt (4) - when FIFO is Enabled (WSEL = 1, RSA = 0)



■ 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 I²C 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 I²C 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 24.22-5 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 I²C 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

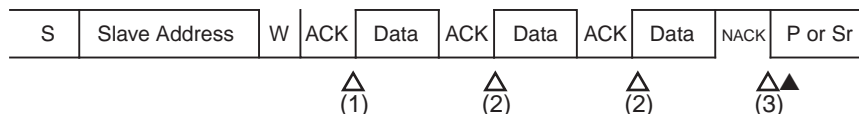
The data direction bit, which determines data transmission or reception, is received after an address is received. When this bit is set to "0", this indicates transmission from the master, therefore, as a slave, the device will receive data.

■ 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 I²C 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 I²C bus wait. (Refer to Table 24.22-5.)
 - (2) After 1-byte data is received, set the interrupt flag (INT) to "1" according to the WSEL setting to put the I²C 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 I²C 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 I²C 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 I²C 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 I²C 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 24.22-25 Slave Reception Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)

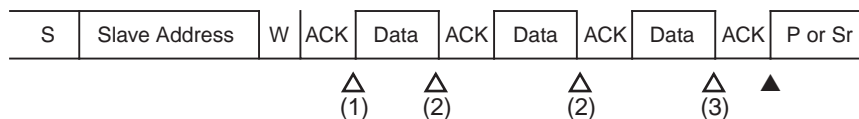


△: Interrupt by INTE=1

▲: 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 24.22-26 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

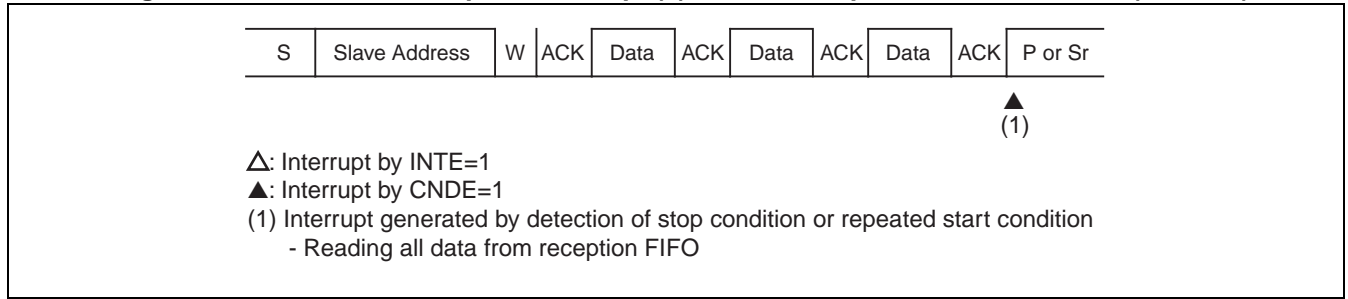
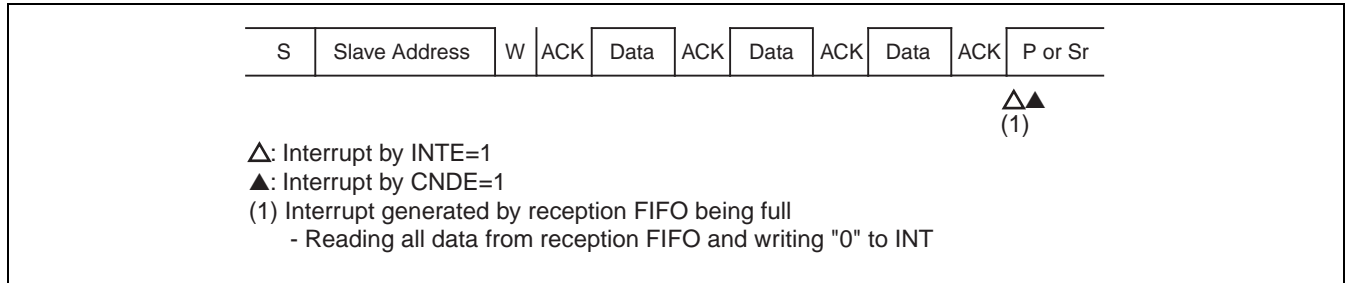
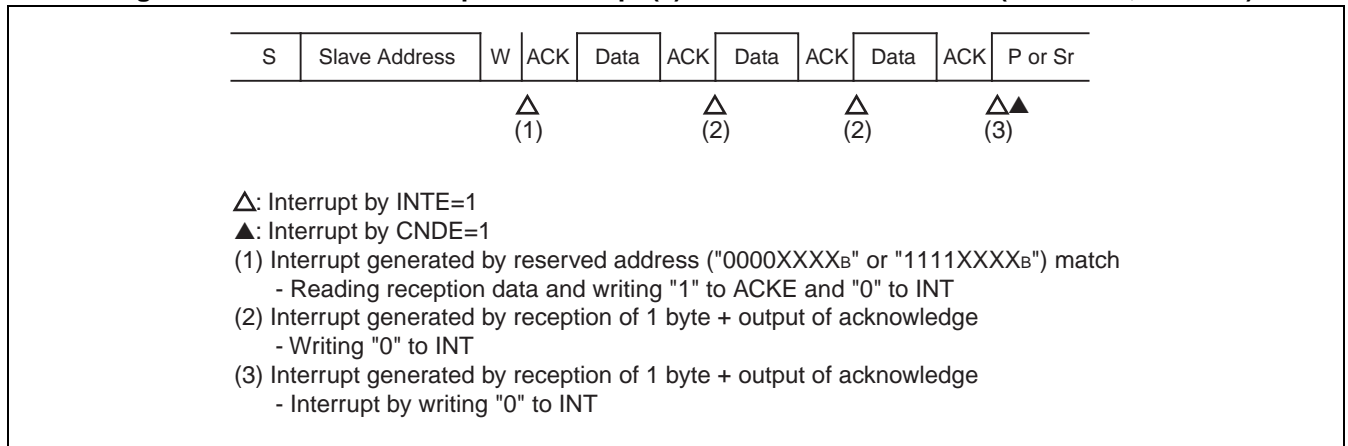
Figure 24.22-27 Slave Reception Interrupt (3) - when FIFO is Disabled (WSEL = 1, RSA = 0)



△: 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 24.22-28 Slave Reception Interrupt (4) - when Reception FIFO is Enabled (RSA = 0)**Figure 24.22-29 Slave Reception Interrupt (5) - when Reception FIFO is Enabled (RSA = 0)****Figure 24.22-30 Slave Reception Interrupt (6) - when FIFO is Disabled (WSEL = 0, RSA = 1)**

■ 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 24.22-5).

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 I²C 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 I²C 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.

(1) Reload value:

$$V = \phi / b - 1$$

V: Reload value b: Baud rate ϕ : Peripheral clock (PCLK) frequency

Note that the set baud rate may not be generated depending on the SCL rising time of the I²C bus. In that case, the reload value must be adjusted.

(2) 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:

$$b = (16 \times 1000000) / (39 + 2) = 400 \text{ kbps}$$

<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 (I²C 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 24.23-1 Reload Values and Baud Rates

Baud rate [bps]	8 MHz Reload value	10 MHz Reload value	16 MHz Reload value	20 MHz Reload value	24 MHz Reload value	32MHz 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 I²C 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 24.23-1 Master Reception Main Settings

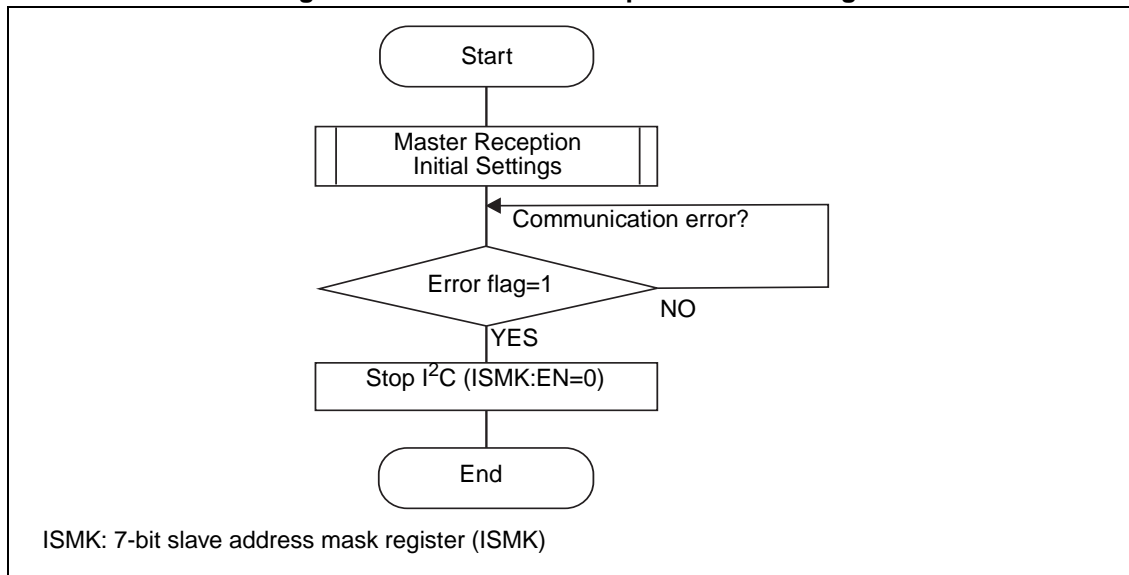


Figure 24.23-2 Master Reception Initial Settings

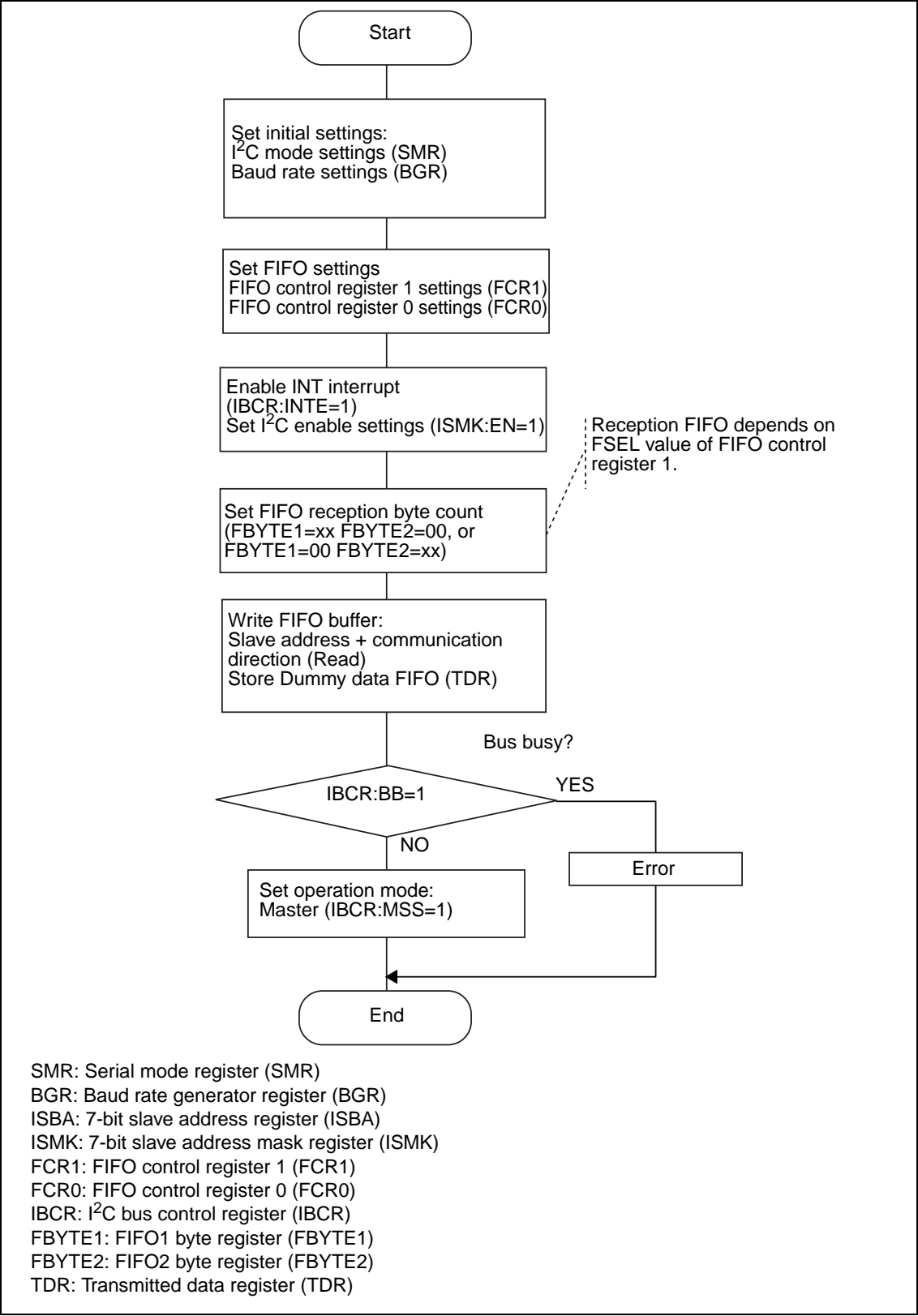


Figure 24.23-3 Master Reception Interrupt Process

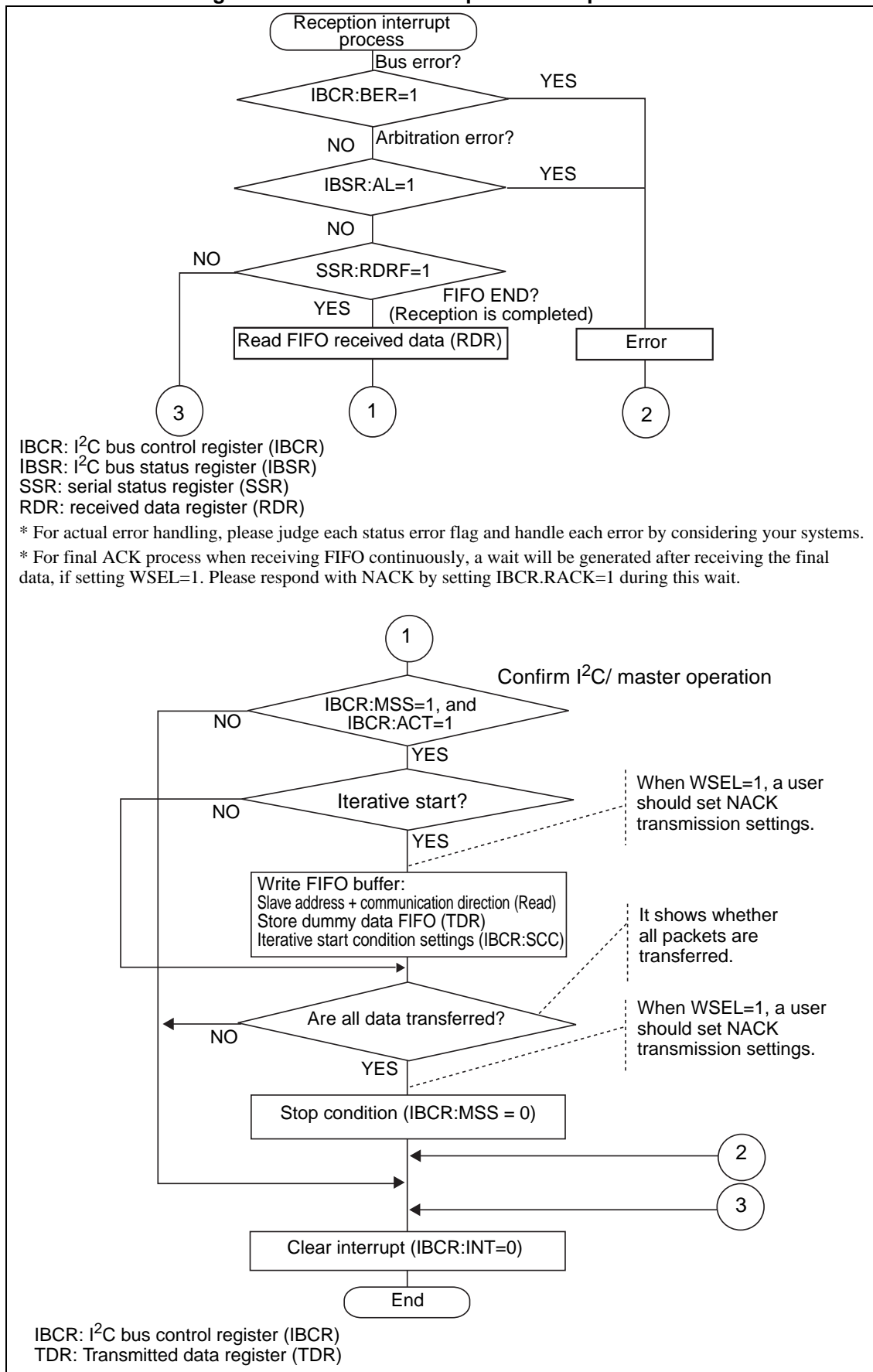


Figure 24.23-4 Slave Transmission Main Settings

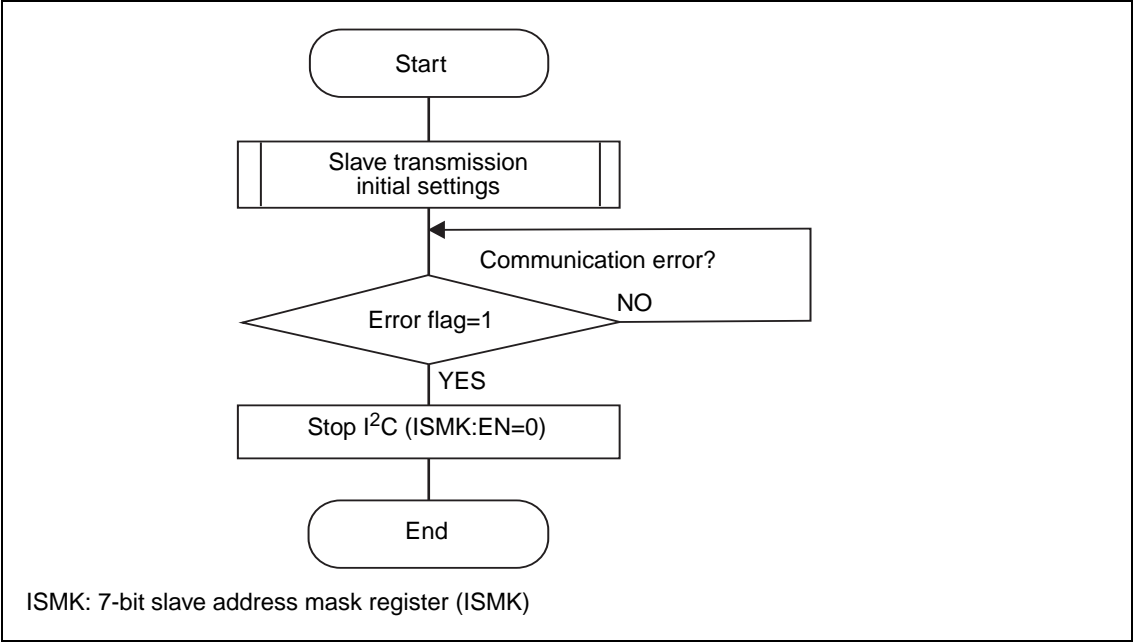


Figure 24.23-5 Slave Transmission Initial Settings

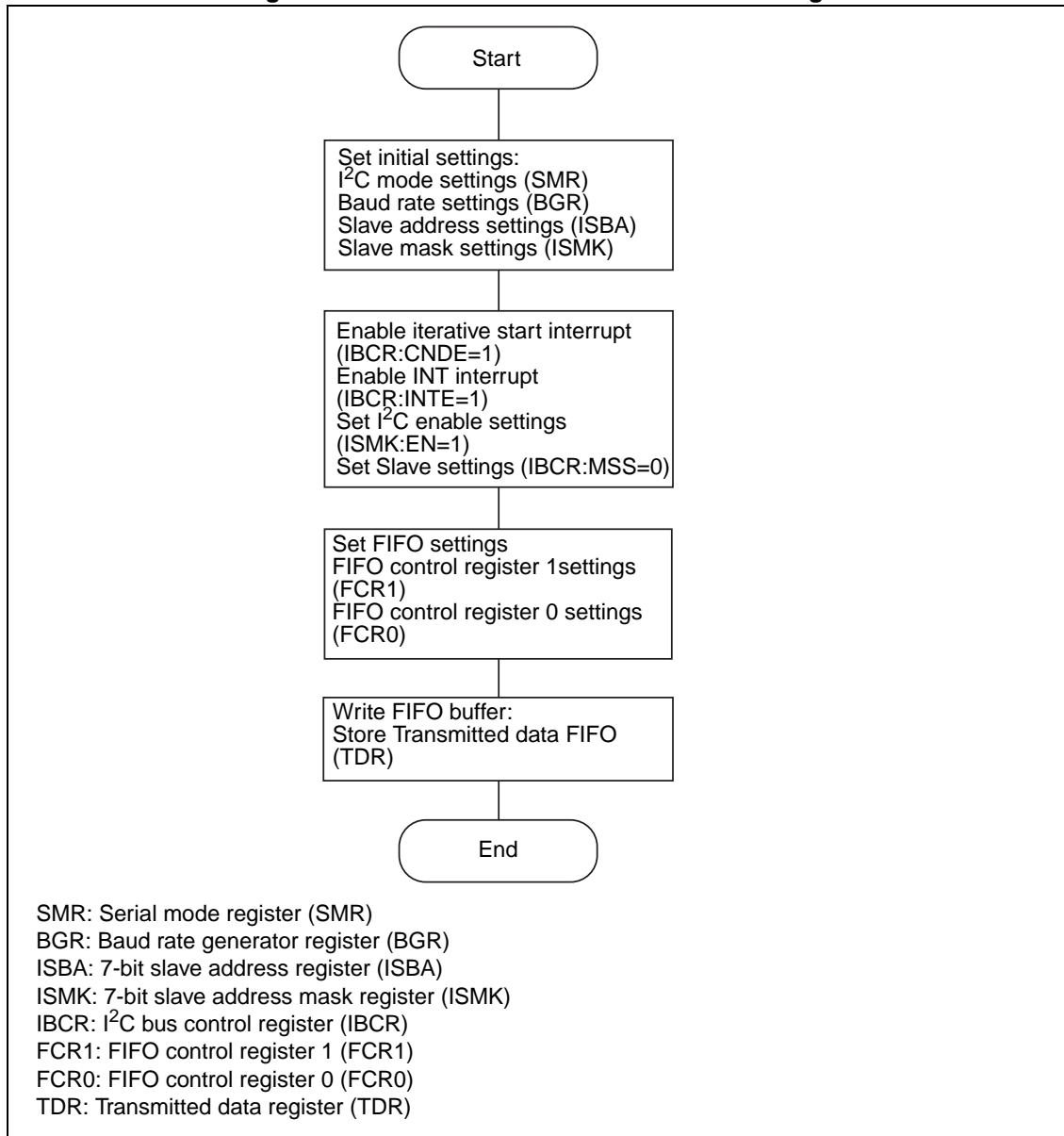
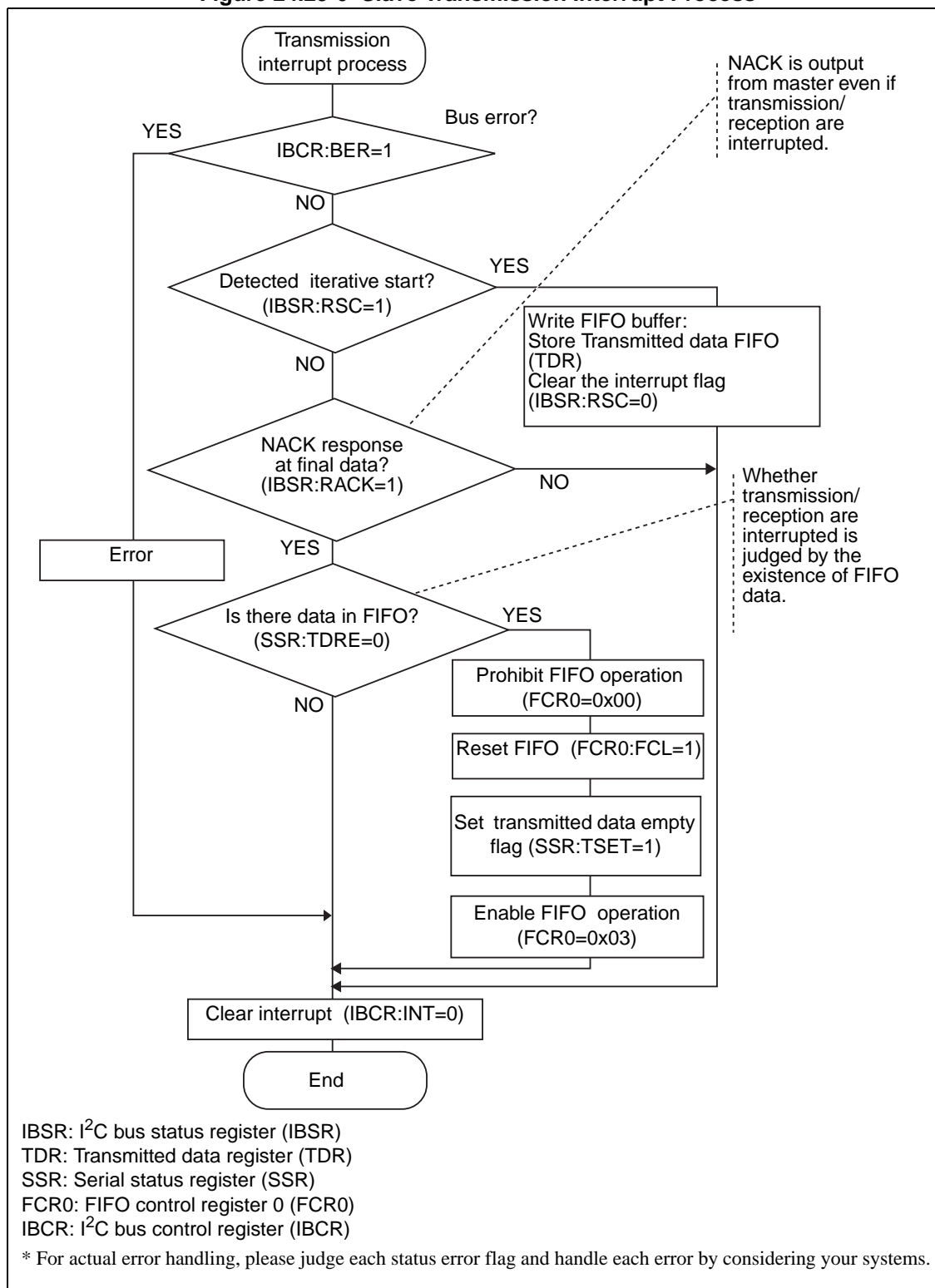


Figure 24.23-6 Slave Transmission Interrupt Process



■ I²C Master Transmission/ Slave Reception FIFO Communication Flow

Figure 24.23-7 Master Transmission Main Settings

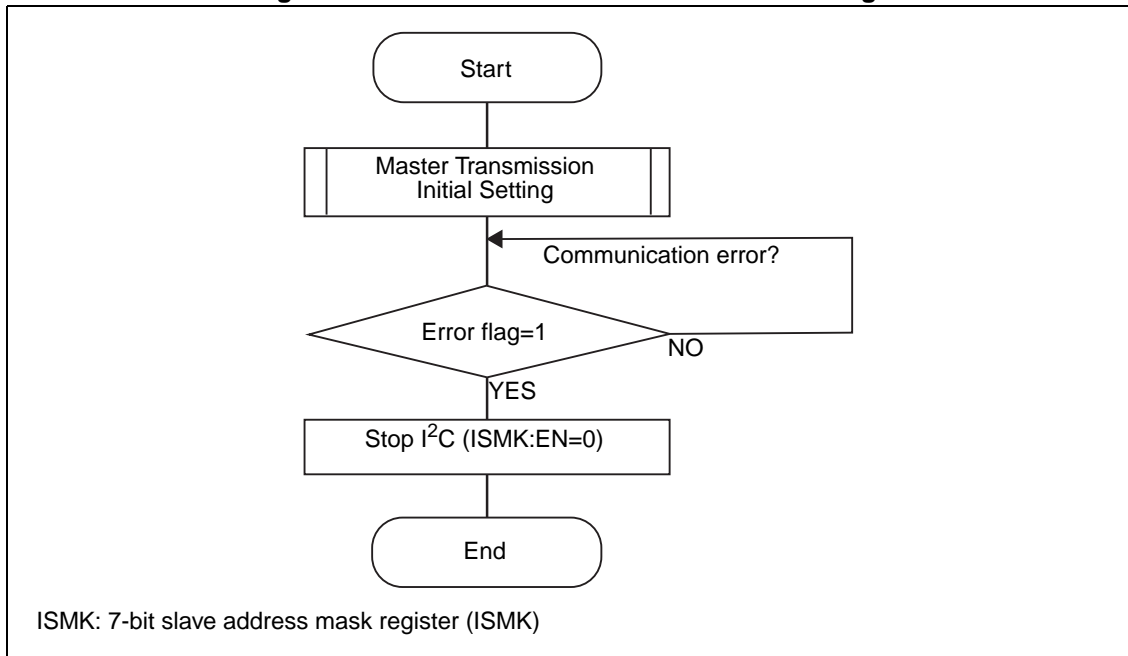


Figure 24.23-8 Master Transmission Initial Settings

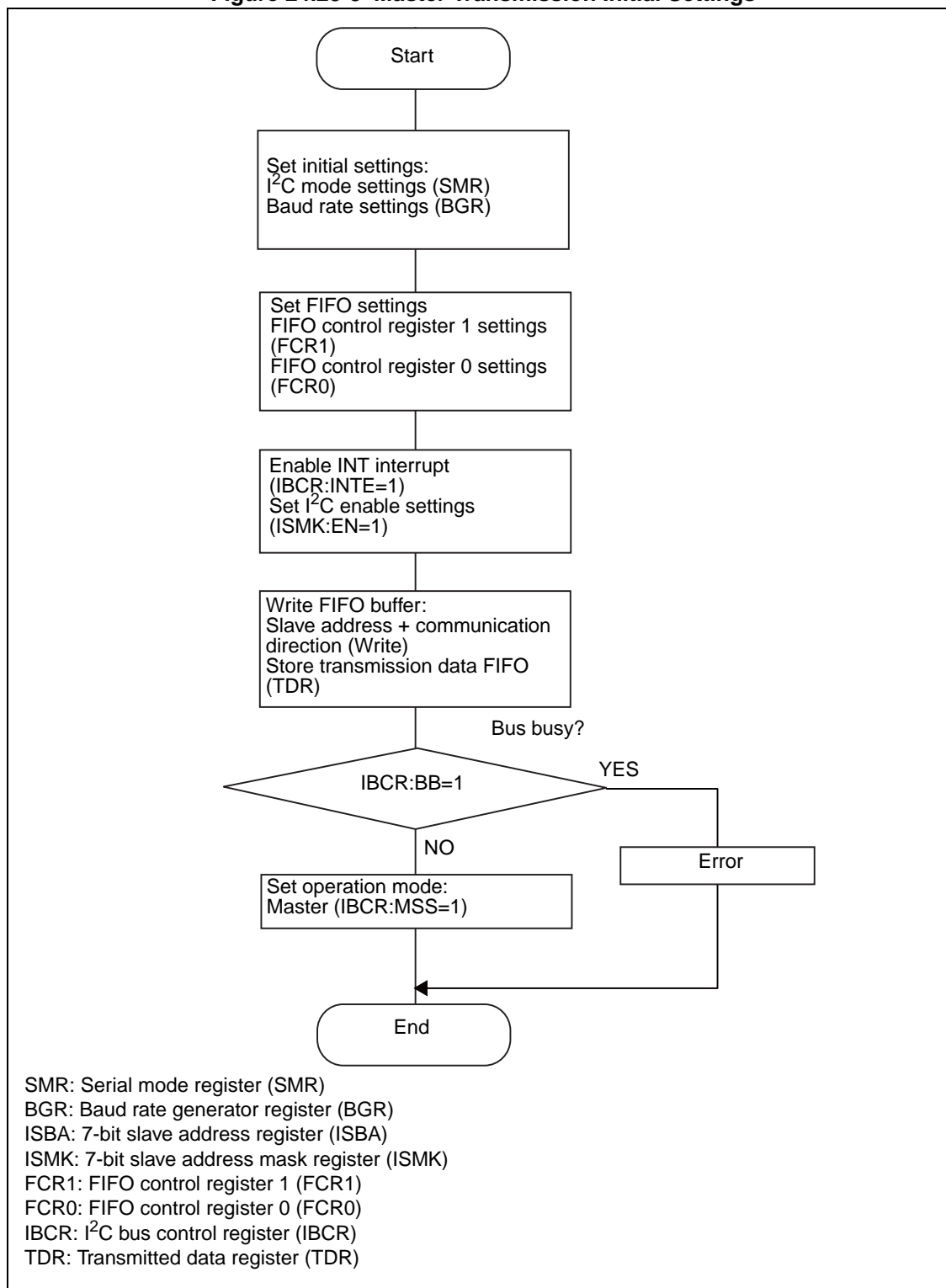


Figure 24.23-9 Master Transmission Interrupt Process

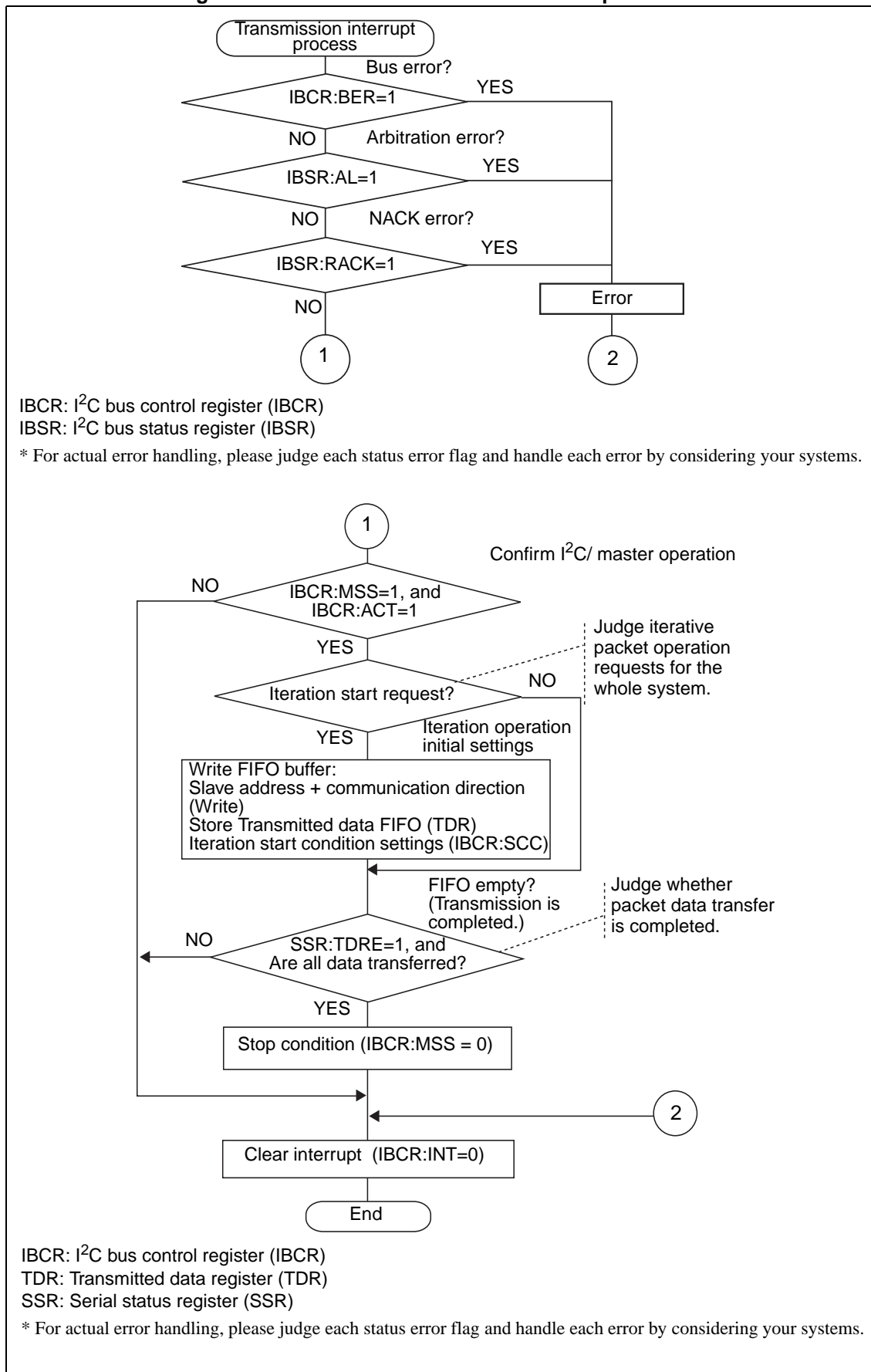


Figure 24.23-10 Slave Reception Main Settings

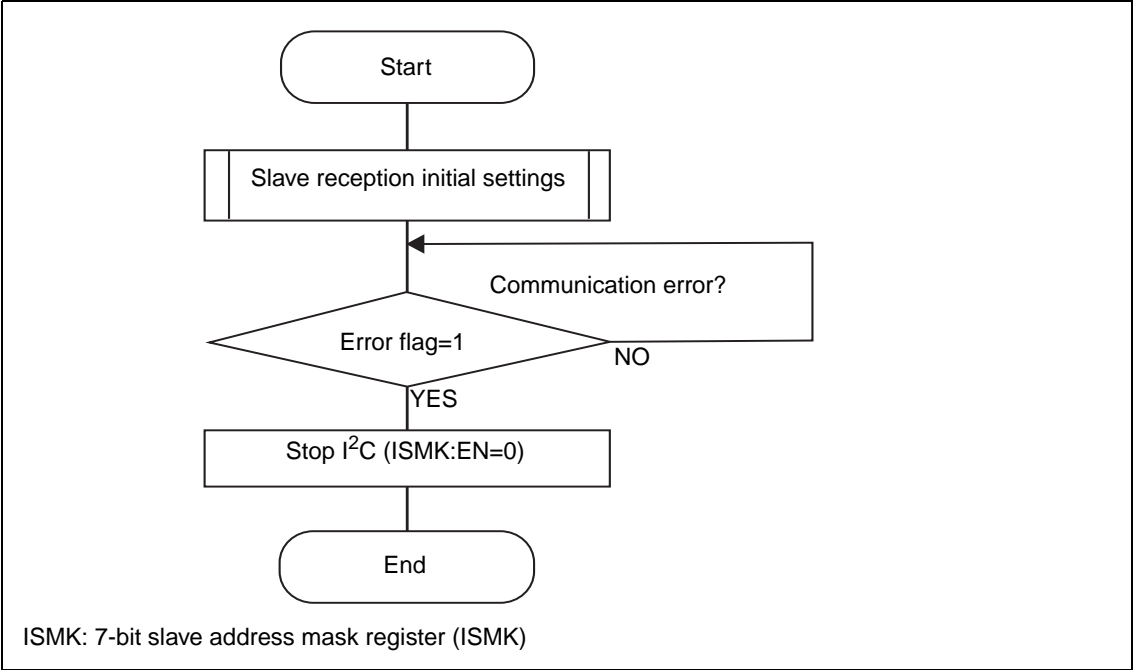


Figure 24.23-11 Slave Reception Initial Settings

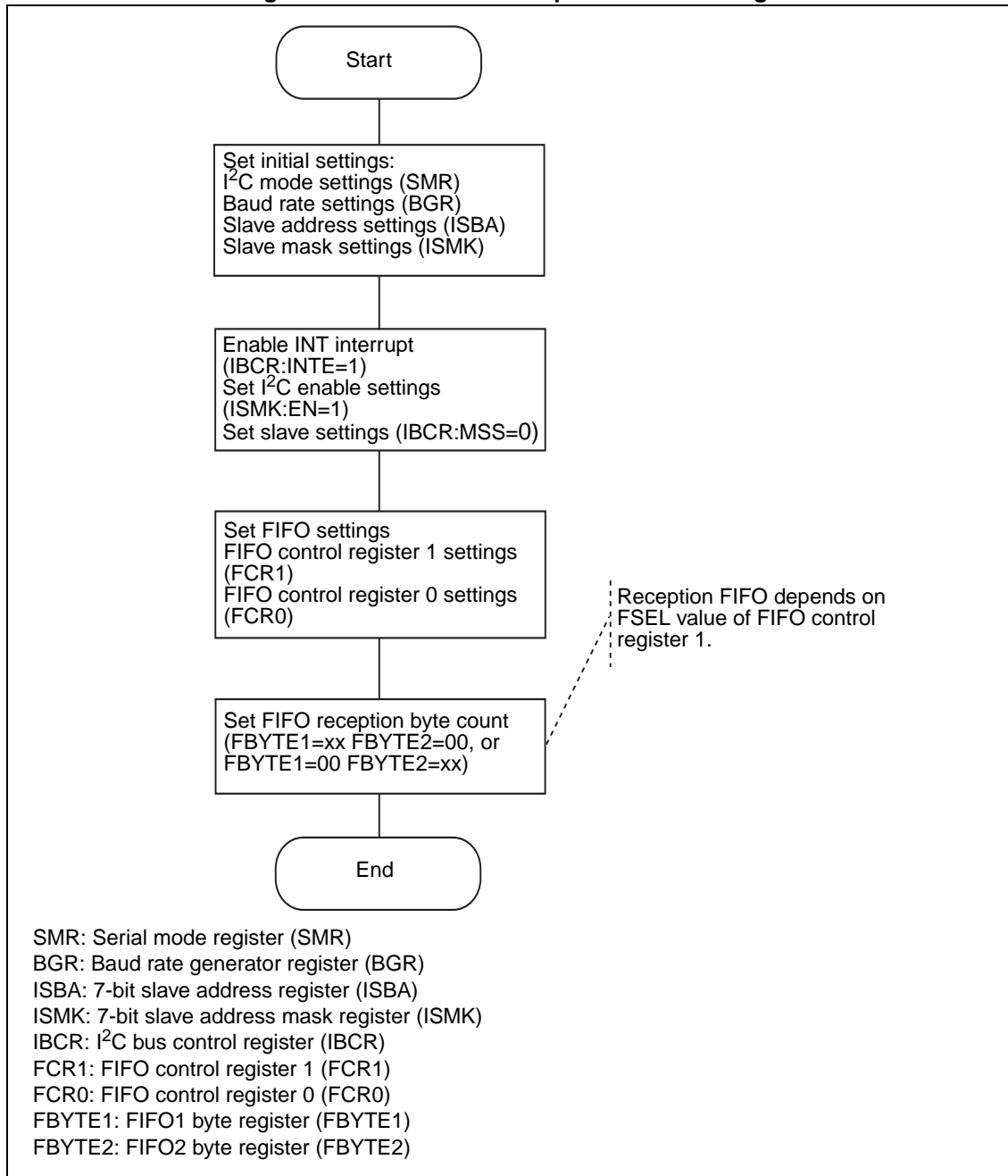
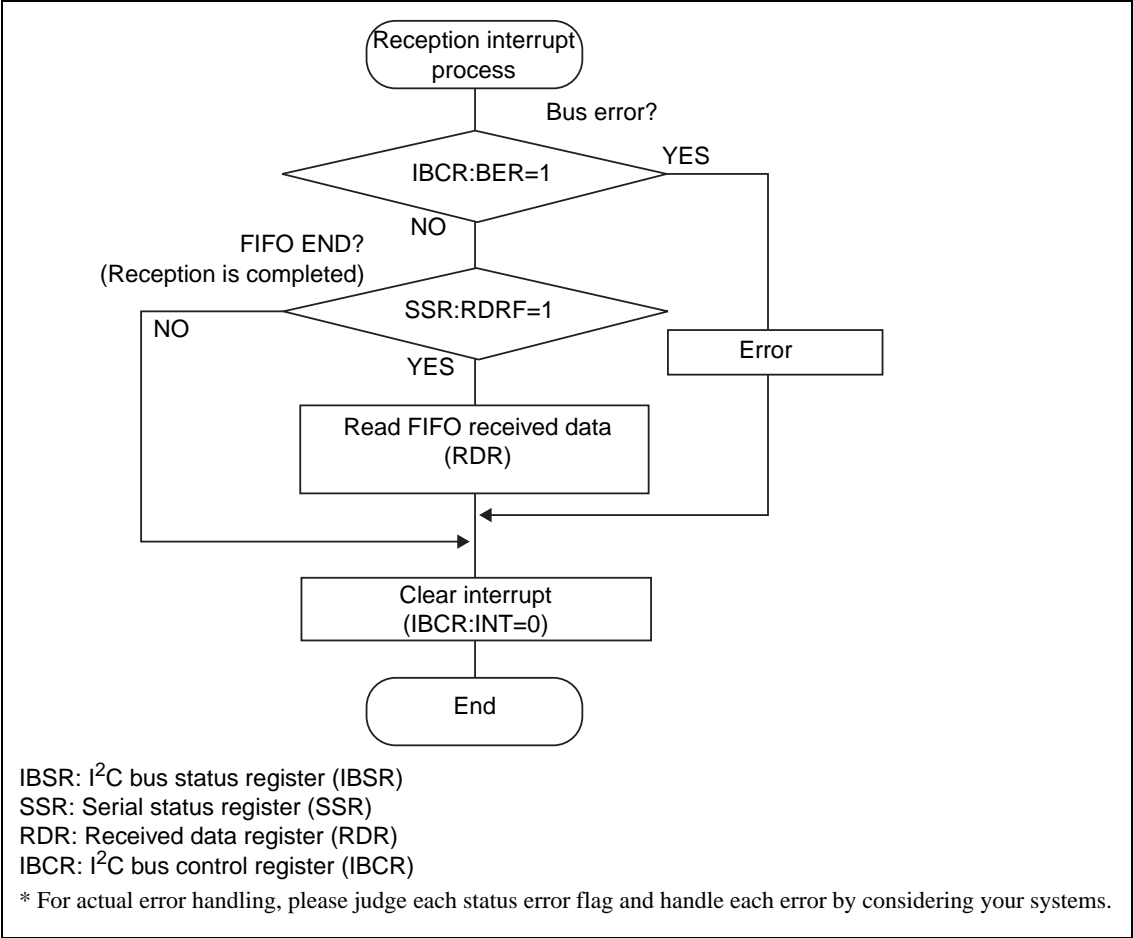


Figure 24.23-12 Slave Reception Interrupt Process

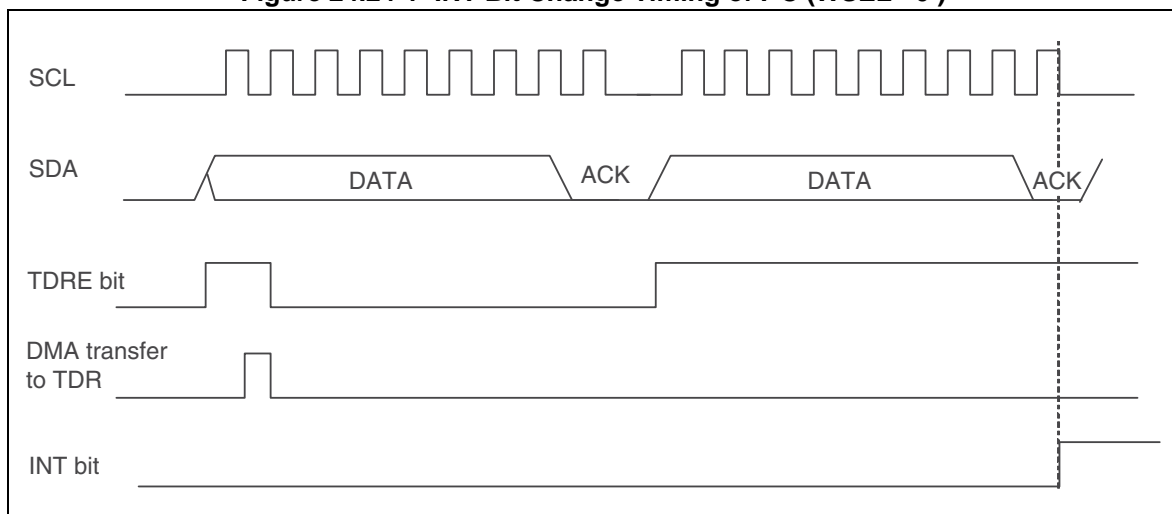


24.24 Notes on I²C Mode

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 24.24-1 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, master reception, and slave reception.)

Figure 24.24-1 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).

CHAPTER 25 DMA Controller (DMAC)

This chapter describes the functions and operations of the DMA controller (DMAC).

- 25.1 Overview
- 25.2 Configuration
- 25.3 Registers
- 25.4 Explanation of Operations
- 25.5 Control Flow
- 25.6 Notes on Use

25.1 Overview

The DMA controller is a module to enable DMA (Direct Memory Access) transfer with this product type. The DMA controller is connected to the on-chip bus of this product type. The DMA transfer controlled by this module enables high-speed data transfer without CPU intervention to increase system performance.

■ Hardware configuration

This module has the following configuration:

- On-chip bus interface
- Independent DMA channel \times 4
- 32-bit address registers (base address and index address etc. for each channel for a total of five)
- 32-bit transfer byte count registers (one per channel)
- Data buffers (32 bytes per channel)

■ Main functions

The following functions are available for data transfer by this module:

- Multiple channel (4 channels) independent DMA transfer
- Priority
 - Fixed priority
Fixed to $\text{ch.0} > \text{ch.1} > \text{ch.2} > \text{ch.3}$.
 - Round-robin priority
The next priority is determined so that the priority of the channel that performed transfer becomes the lowest and the priority of channels that has not performed data transfer is raised.
- DMA transfer trigger
 - Internal (constant) request
 - Request from ICH (request by ICH built-in peripheral function)
- DMA transfer mode
 - Transfer request mode: Burst transfer/Block transfer
 - Channel mode: DA mode/SCA mode/DCA mode/2-D mode, address increment/decrement/fixed
 - Transfer data size: 1, 2, 4, and 32 bytes
 - Total transferred bytes: Up to 2^{24} bytes
 - DMA halt by interrupt
 - Interrupt level to halt DMA transfer when interrupt occurs is settable

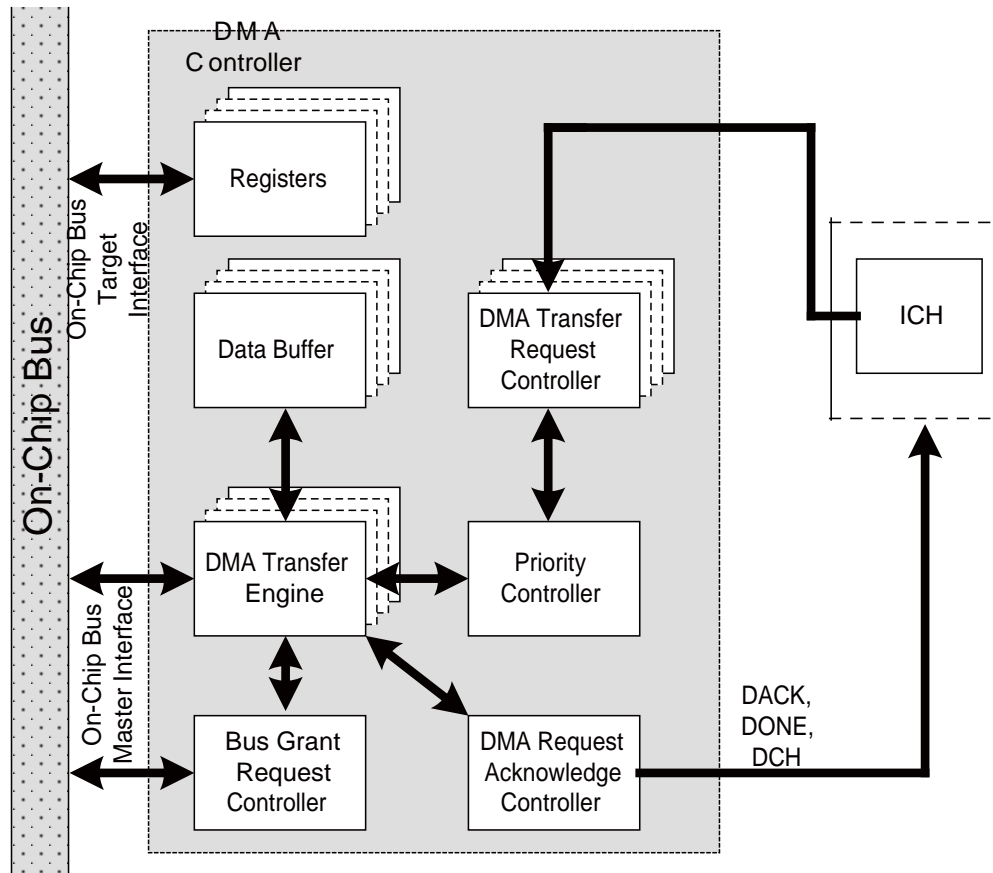
25.2 Configuration

This section describes the configuration of the DMA controller.

■ Block Diagram of DMA controller

Figure 25.2-1 shows a block diagram of the DMA controller.

Figure 25.2-1 Block Diagram of DMA Controller



25.3 Registers

This section describes the configurations and functions of registers used by the DMA controller.

■ List of Registers

Table 25.3-1 shows a list of registers for DMA controller.

Table 25.3-1 List of Registers for DMA Controller (1 / 2)

Channel	Register Abbreviation	Register Name	See
Common	GCFR	Global configuration register	25.3.1
	DNMIR	DMA-halt by NMI flag register	25.3.11
	DILVR	DMA-halt by interrupt level register	25.3.12
0	CCFR0	Channel configuration register 0	25.3.2
	CSTR0	Channel status register 0	25.3.3
	CCTR0	Channel control register 0	25.3.4
	SBA0	Source base address register 0	25.3.5
	DBA0	Destination base address register 0	25.3.6
	PIX0	Primary index register 0	25.3.7
	SIX0	Secondary index register 0	25.3.8
	BCL0	Byte count limit register 0	25.3.9
	APR0	Alternate pointer register 0	25.3.10
1	CCFR1	Channel configuration register 1	25.3.2
	CSTR1	Channel status register 1	25.3.3
	CCTR1	Channel control register 1	25.3.4
	SBA1	Source base address register 1	25.3.5
	DBA1	Destination base address register 1	25.3.6
	PIX1	Primary index register 1	25.3.7
	SIX1	Secondary index register 1	25.3.8
	BCL1	Byte count limit register 1	25.3.9
	APR1	Alternate pointer register 1	25.3.10

Table 25.3-1 List of Registers for DMA Controller (2 / 2)

Channel	Register Abbreviation	Register Name	See
2	CCFR2	Channel configuration register 2	25.3.2
	CSTR2	Channel status register 2	25.3.3
	CCTR2	Channel control register 2	25.3.4
	SBA2	Source base address register 2	25.3.5
	DBA2	Destination base address register 2	25.3.6
	PIX2	Primary index register 2	25.3.7
	SIX2	Secondary index register 2	25.3.8
	BCL2	Byte count limit register 2	25.3.9
	APR2	Alternate pointer register 2	25.3.10
3	CCFR3	Channel configuration register 3	25.3.2
	CSTR3	Channel status register 3	25.3.3
	CCTR3	Channel control register 3	25.3.4
	SBA3	Source base address register 3	25.3.5
	DBA3	Destination base address register 3	25.3.6
	PIX3	Primary index register 3	25.3.7
	SIX3	Secondary index register 3	25.3.8
	BCL3	Byte count limit register 3	25.3.9
	APR3	Alternate pointer register 3	25.3.10

25.3.1 Global Configuration Register (GCFR)

This register controls the entire DMA controller.

Figure 25.3-1 shows the bit configuration of the global configuration register (GCFR).

Figure 25.3-1 Bit Configuration of DMA Global Configuration Register (GCFR)

bit	31	30	0
	AT	Reserved	
Attribute	R/W		
Initial value	0		
R/W: Read/write allowed			

[bit 31]: AT (priority type bit)

This bit specifies how priority between channels is determined.

Written Value	Priority Type
0	Round-robin priority The next priority is determined so that the priority of the channel that performed transfer becomes the lowest and the priority of channels that has not performed data transfer is raised.
1	Fixed priority Fixed to the order of ch.0 (highest) > ch.1 > ch.2 > ch.3 (lowest).

[bit 30 to bit 0]: Reserved bits

Write	Always write "0".
Read	"0" is read.

25.3.2 Channel Configuration Registers (CCFR0 to CCFR3)

These are 16-bit registers that determine the channel configuration. It is available for each channel.

Figure 25.3-2 shows the bit configuration of the channel configuration registers (CCFR0 to CCFR3).

Figure 25.3-2 Bit Configuration of Channel Configuration Registers (CCFR0 to CCFR3)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	Reserved	Reserved	TM1	TM0	CM1	CM0
Attribute	R/W							
Initial value	0							
bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	RS3	RS2	RS1	RS0
Attribute	R/W							
Initial value	0							
R/W: Read/write allowed								

[bit 15 to bit 12]: Reserved bits

Write	Always write "0".
Read	"0" is read.

[bit 11, bit 10]: TM1, TM0 (transfer mode configuration bits)

These bits are used to set the transfer mode of the channel. Two different modes are available.

TM1	TM0	Transfer Mode
0	0	Block transfer
0	1	Burst transfer
1	0	Forbidden to set
1	1	Forbidden to set

[bit 9, bit 8]: CM1, CM0 (channel mode configuration bits)

These bits are used to set the operation mode of the channel. Four different modes are available.

CM1	CM0	Channel Mode
0	0	DA mode (Dual Addressing Mode)
0	1	SCA mode (Source Circular Addressing Mode)
1	0	DCA mode (Destination Circular Addressing Mode)
1	1	2D mode (2-Dimensional Addressing Mode)

[bit 7 to bit 4]: Reserved bits

Write	Always set to "0001".
-------	-----------------------

[bit 3 to bit 0]: RS3 to RS0 (DMA transfer request source configuration bits)

These bits are used to select the source of the transfer request for the channel.
However, specification of internal (constant) request is ignored when the channel is SCA mode or DCA mode

RS3	RS2	RS1	RS0	DMA Transfer Source
0	0	0	0	Internal (constant) request
0	0	0	1	Set with IORR register
Other				Forbidden to set

25.3.3 Channel Status Registers (CSTR0 to CSTR3)

These are 16-bit registers that show the channel status. They are available for each channel.

Figure 25.3-3 shows the bit configuration of the channel status registers (CSTR0 to CSTR3).

Figure 25.3-3 Bit Configuration of Channel Status Registers (CSTR0 to CSTR3)

bit	15	14	13	12	11	10	9	8
	BUSY	INT	CE	RER	WER	STP	FED	NE
Attribute	R/W							
Initial value	0							
bit	7	6	5	4	3	2	1	0
	Reserved	FS6	FS5	FS4	FS3	FS2	FS1	FS0
Attribute	R/W							
Initial value	0							
R/W: Read/write allowed								

[bit 15]: BUSY (busy status flag)

This is a flag that indicates the channel is busy.

Set Request	Start of DMA transfer by writing "1" in ACT of CCTR register
Clear Request	<ul style="list-style-type: none"> DMA transfer completion Error occurrence End of DMA transfer by writing "0" in ACT of CCTR register

Writing to this bit has no effect.

[bit 14]: INT (interrupt request flag)

This is a flag that indicates an interrupt request is detected.

Set Request	<ul style="list-style-type: none"> Normal ended Force ended Compare ended DMA transfer completion due to detection of read error/write error Writing "1" in this bit
Clear Request	<ul style="list-style-type: none"> Writing "0" in this bit

"1" can be written in this flag by software.

This flag must be accessed with a memory store instruction.

Do not access with read-modify-write instruction (bit manipulation instruction).

[bit 13]: CE (compare ended flag)

This is a flag that indicates DMA transfer has stopped because address match was detected in SCA or DCA mode.

Set Request	<ul style="list-style-type: none"> DMA transfer stop ended because address match was detected in SCA/DCA mode Writing "1" in this bit
Clear Request	<ul style="list-style-type: none"> Writing "0" in this bit

"1" can be written in this flag by software.

This flag must be accessed with a memory store instruction.

Do not access with read-modify-write instruction (bit manipulation instruction).

[bit 12]: RER (read error flag)

This is a flag that indicates an error has occurred while reading the source side data.

Set Request	<ul style="list-style-type: none"> Read error in source side data Writing "1" in this bit
Clear Request	<ul style="list-style-type: none"> Writing "0" in this bit

"1" can be written in this flag by software.

This flag must be accessed with a memory store instruction.

Do not access with read-modify-write instruction (bit manipulation instruction).

[bit 11]: WER (write error flag)

This is flag that indicates an error has occurred while writing to the destination side.

Set Request	<ul style="list-style-type: none"> Write error when writing to destination side Writing "1" in this bit
Clear Request	<ul style="list-style-type: none"> Writing "0" in this bit

"1" can be written in this flag by software.

This flag must be accessed with a memory store instruction.

Do not access with read-modify-write instruction (bit manipulation instruction).

[bit 10]: STP (stop ended flag)

This flag indicates that DMA transfer was force ended because DSTP input has become activate. After DSTP input has become active, this flag changes to "1" and DMA transfer is stop ended when the current transfer completes.

Set Request	<ul style="list-style-type: none"> Stop ended (DSTP input active) Writing "1" in this bit
Clear Request	<ul style="list-style-type: none"> Writing "0" in this bit

"1" can be written in this flag by software.

This flag must be accessed with a memory store instruction.

Do not access with read-modify-write instruction (bit manipulation instruction).

[bit 9]: FED (force ended flag)

This bit indicates that DMA transfer was force ended because "0" was written in the ACT bit of the CCTR register. After writing "0" in the ACT bit, this flag changes to "1" and DMA transfer is forced ended when the current transfer completes.

Set Request	<ul style="list-style-type: none"> Force ended ("0" written in ACT bit of CCTR register) Writing "1" in this bit
Clear Request	<ul style="list-style-type: none"> Writing "0" in this bit

"1" can be written in this flag by software.

This flag must be accessed with a memory store instruction.

Do not access with read-modify-write instruction (bit manipulation instruction).

[bit 8]: NE (normal ended flag)

This bit indicates that the configured transfer operation has completed normally. This bit is set as follows according to the channel mode (this bit is not set by end of transfer in SCA and DCA mode).

Set Request	<ul style="list-style-type: none"> DA mode: Transfer of bytes specified by BCL has completed 2D mode: Transfer of scan lines specified by SIX has completed Writing "1" in this bit
Clear Request	<ul style="list-style-type: none"> Writing "0" in this bit

"1" can be written in this flag by software.

This flag must be accessed with a memory store instruction.

Do not access with read-modify-write instruction (bit manipulation instruction).

[bit 7]: Reserved bit

Write	Always write "0".
Read	"0" is read.

[bit 6 to bit 0]: FS6 to FS0 (FIFO status bit)

These bits indicate the size of normally completed transfer.

Writing to these bits is ignored.

- Transfer completes with error

Type of Error	Register Content
Read Error	Normally completed read size
Write Error	Normally completed write size
Both	Normally completed read size

- Force ended

SSIZ relation*1	Register Content
$SSIZ < DSIZ$	Indicates the completed read size. It becomes 0 when write completes. It is 0 at start of transfer.
$SSIZ > DSIZ$	Indicates the completed write size. It becomes 0 when read completes. It is the value of SSIZ at start of transfer
$SSIZ = DSIZ$	Indicates the value of SSIZ when read completes. It becomes 0 when write completes. It is 0 at start of transfer.

*1 The SSIZ and DSIZ bits are in the CCTR register.

25.3.4 Channel Control Registers (CCTR0 to CCTR3)

These are 16-bit registers that control the operation of the channel. They are available for each channel.

Figure 25.3-4 shows the bit configuration of the channel control registers (CCTR0 to CCTR3).

Figure 25.3-4 Bit Configuration of Channel Control Registers (CCTR0 to CCTR3)

bit	15	14	13	12	11	10	9	8
	ACT	IE	BUF	ICE	FC	Reserved	SAU1	SAU0
Attribute	R/W							
Initial value	0							
bit	7	6	5	4	3	2	1	0
	SSIZ2	SSIZ1	SSIZ0	DAU1	DAU0	DSIZ2	DSIZ1	DSIZ0
Attribute	R/W							
Initial value	0							
R/W: Read/write allowed								

[bit 15]: ACT (channel operation bit)

Channel operation starts when this bit is set to "1". Also, operation halts temporarily when "0" is written while operating, but resumes when "1" is written once more.

This bit changes to "0" when transfer completes. It also changes to "0" and channel operation stops when transfer is canceled due to error.

Written Value	Instruction to Channel
0	Stop operation
1	Start operation

[bit 14]: IE (interrupt enable bit)

This bit specifies whether to issue an interrupt when the INT bit of the CSTR register is "1".

Written Value	Interrupt Operation
0	Disable interrupt
1	Enable interrupt

[bit 13]: BUF (buffer access enable bit)

This bit specifies whether the DMA transfer internal bus access is a buffer enabled access.

Written Value	DMA Transfer Access
0	DMA transfer bus access disabled
1	DMA transfer bus access enabled

[bit 12]: ICE (index compare enable bit)

This bit specifies whether to stop DMA transfer upon address match when channel mode is SCA or DCA. When fix to circular side is specified, "1" cannot be set.

Written Value	Address Compare Operation
0	Disable address compare
1	Enable address compare (stop DMA transfer when address match)

[bit 11]: FC (FIFO clear bit)

The channel FIFO buffer is cleared when "1" is written. After setting "1", to start channel operation set "0" with software.

When the ACT bit in the CCTR register and this bit are both "0", the information left in the FIFO buffer is written if "1" is written to both bits at the same time.

Written Value	FIFO Clear Operation
0	Do not clear FIFO
1	Clear FIFO

[bit 10]: Reserved bit

Write	Always write "0".
Read	"0" is read.

[bit 9, bit 8]: SAU1, SAU0 (transfer source address update bit)

These bits specify the amount of source side address update (in reality, the PIX register or the SIX register is updated depending on the channel mode and that status is reflected in the address during transfer).

SAU1	SAU0	Source Address Update
0	0	Hold. Address is fixed.
0	1	Increment. Add size specified with SSIZ.
1	0	Decrement. Subtract size specified with SSIZ. (If decrement is specified, set so that SSIZ=DSIZ.)
1	1	Forbidden to set

[bit 7 to bit 5]: SSIZ2 to SSIZ0 (transfer source size configuration bit)

These bits specify the source side transfer size.

SSIZ2	SSIZ1	SSIZ0	Source Side Transfer Size
0	0	0	1 byte
0	0	1	2 bytes
0	1	0	4 bytes
0	1	1	Forbidden to set
1	0	0	Forbidden to set
1	0	1	32 bytes
1	1	0	Forbidden to set
1	1	1	Forbidden to set

[bit 4, bit 3]: DAU1, DAU0 (transfer destination address update bit)

These bits specify the amount of destination side address update (in reality, the PIX register or the SIX register is updated depending on the channel mode and that status is reflected in the address during transfer).

DAU1	DAU0	Destination Address Update
0	0	Hold. Address is fixed.
0	1	Increment. Add size specified with DSIZ.
1	0	Decrement. Subtract size specified with DSIZ. (If decrement is specified, set so that DSIZ=SSIZ.)
1	1	Forbidden to set

[bit 2 to bit 0]: DSIZ2 to DSIZ0 (transfer destination size configuration bit)

These bits specify the destination side transfer size.

DSIZ2	DSIZ1	DSIZ0	Destination Side Transfer Size
0	0	0	1 byte
0	0	1	2 bytes
0	1	0	4 bytes
0	1	1	Forbidden to set
1	0	0	Forbidden to set
1	0	1	32 bytes
1	1	0	Forbidden to set
1	1	1	Forbidden to set

25.3.5 Source Base Address Registers (SBA0 to SBA3)

These are 32-bit registers that indicate the size of source side base address. They are available for each channel.

Figure 25.3-5 shows the bit configuration of the source base address registers (SBA0 to SBA3).

Figure 25.3-5 Bit Configuration of Source Base Address Registers (SBA0 to SBA3)

bit	31		0
	<div>SBA</div>		
Attribute	R/W		
Initial value	0		
R/W: Read/write allowed			

[bit 31 to bit 0]: SBA (source base address)

These bits store the source side base address.
The address used for source side transaction is calculated as described below according to mode.
Furthermore, this register value must be aligned on size specified by the CCTR register bits SSIZ2 to SSIZ0 and DSIZ2 to DSIZ0. See "■ Settings specific to each channel" for the alignment method.

- DA, SCA mode
Address calculation is performed as follows according to the CCTL register bits SAU1 and SAU0.

SAU1	SAU0	Address Calculation
0	0	Hold SBA (fixed)
0	1	Increment SBA+PIX
1	0	Decrement SBA-PIX

- DCA mode
Address is fixed to SBA. Address calculation value=SBA (fixed)

- 2D mode

Horizontal direction address calculation is performed according to SAU as shown in table A below.
Furthermore, when the number of bytes indicated by BCL is transferred, the vertical direction address calculation is performed according to SAU as shown in table B below.

Table A horizontal direction

SAU1	SAU0	Address Calculation
0	0	Hold SBA (fixed)
0	1	Increment SBA+PIX
1	0	Decrement SBA-PIX

Table B vertical direction

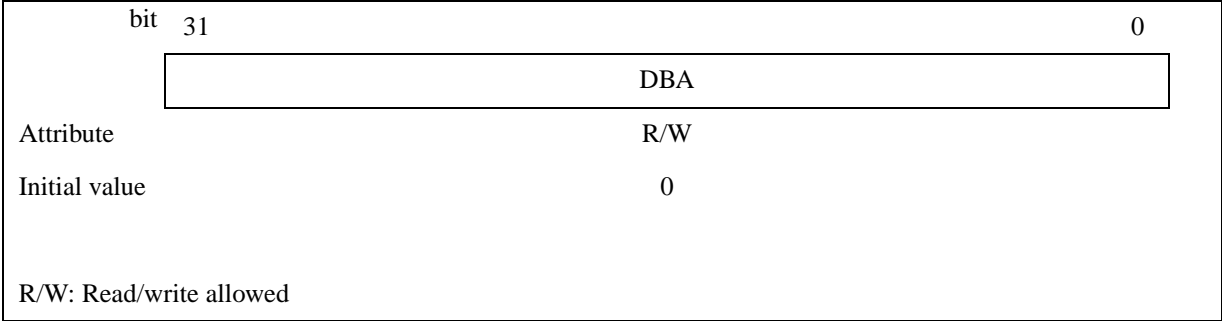
SAU1	SAU0	Address Calculation
0	0	Hold SBA (fixed)
0	1	Increment SBA+AP
1	0	Increment SBA-AP

25.3.6 Destination Base Address Registers (DBA0 to DBA3)

These are 32-bit registers that indicate the size of destination side base address. They are available for each channel.

Figure 25.3-6 shows the bit configuration of the destination base address registers (DBA0 to DBA3).

Figure 25.3-6 Bit Configuration of Destination Base Address Registers (DBA0 to DBA3)



[bit 31 to bit 0]: DBA (destination base address)

These bits store the destination side base address.
The address used for destination side transaction is calculated as described below according to mode.
Furthermore, this register value must be aligned on size specified by the CCTR register bits SSIZ2 to SSIZ0 and DSIZ2 to DSIZ0. See "■ Settings specific to each channel" for the alignment method.

- DA mode
Address calculation is performed as follows according to the CCTL register bits DAU1 and DAU0.

DAU1	DAU0	Address Calculation
0	0	Hold DBA (fixed)
0	1	Increment DBA+SIX
1	0	Increment DBA-SIX

- DCA mode
Address calculation is performed as follows according to the CCTL register bits DAU1 and DAU0.

DAU1	DAU0	Address Calculation
0	0	Hold DBA (fixed)
0	1	Increment DBA+PIX
1	0	Decrement DBA-PIX

- SCI mode
Address is fixed to DBA. Address calculation value=DBA (fixed)

- 2D mode

Horizontal direction address calculation is performed according to DAU as shown in table A below. Furthermore, when the number of bytes indicated by BCL is transferred, the vertical direction address calculation is performed according to DAU as shown in table B below.

Table A horizontal direction

DAU1	DAU0	Address Calculation
0	0	Hold DBA (fixed)
0	1	Increment DBA+PIX
1	0	Decrement DBA-PIX

Table B vertical direction

DAU1	DAU0	Address Calculation
0	0	Hold DBA (fixed)
0	1	Increment DBA+AP
1	0	Increment DBA-AP

25.3.7 Primary Index Registers (PIX0 to PIX3)

These are 32-bit registers that store the address of the next source side data to be transferred in the form of offset from base address. They are available for each channel.

Figure 25.3-7 shows the bit configuration of the primary index registers (PIX0 to PIX3).

Figure 25.3-7 Bit Configuration of Primary Index Registers (PIX0 to PIX3)

bit	31		0
	<div>PIX</div>		
Attribute	R/W		
Initial value	0		
R/W: Read/write allowed			

[bit 31 to bit 0]: PIX (primary index)

The function of these bits changes as follows according to the channel mode.

Set "0" in this register by software before starting transfer.

After the DMA controller starts transfer, this register may not contain the transferred bytes (read value is not guaranteed) because this register is used for control transfer.

• DA mode

Stores the source side transferred bytes.

The size of the transferred data is added after each transaction.

The valid value for PIX is from "00000000_H" (0 byte) to "01000000_H" (2²⁴ bytes) because the maximum transfer size of the DMA controller is 2²⁴ bytes.

• SCA mode

Contains the next data read position in the form of byte offset from beginning of the circular buffer.

The size of the transferred data is added after each transaction. After updating the value, if the value of the offset of the end of the circular buffer, which is stored in BCL, is exceeded, the value is cleared to "0" and the read position is returned to the beginning of the buffer.

The valid value for PIX is from "00000000_H" (0 byte) to "01000000_H" (2²⁴ bytes) because the maximum transfer size of the DMA controller is 2²⁴ bytes.

• DCA mode

Contains the next data store position in the form of byte offset from beginning of the circular buffer.

The size of the transferred data is added after each transaction. After updating the value, it is cleared to "0" if the value matches the value of BCL and the write position is returned to the beginning of the buffer.

The valid value for PIX is from "00000000_H" (0 byte) to "01000000_H" (2²⁴ bytes) because the maximum transfer size of the DMA controller is 2²⁴ bytes.

• 2D mode

Stores the transferred data bytes in the scan line.

The size of the transferred data is added after each transaction. After updating the value, it is cleared to "0" if the value matches the value of BCL.

The valid value for PIX is from "00000000_H" (0 byte) to "01000000_H" (2²⁴ bytes) because the maximum transfer size of the DMA controller is 2²⁴ bytes.

25.3.8 Secondary Index Registers (SIX0 to SIX3)

These are 32-bit registers that store the address of the next destination side data to be transferred in the form of offset from base address. They are available for each channel.
In 2D mode, they are used to store the remaining scan lines. In SCA and DCA mode, they are used to store the address to suspend transfer when the addresses match.

Figure 25.3-8 shows the bit configuration of the secondary index registers (SIX0 to SIX3).

Figure 25.3-8 Bit Configuration of Secondary Index Registers (SIX0 to SIX3)

bit	31	0
	SIX	
Attribute	R/W	
Initial value	0	
R/W: Read/write allowed		

[bit 31 to bit 0]: SIX (secondary index)

The function of these bits changes as follows according to the channel mode.

When using this register in DA mode, set "0" by software before starting transfer.

After the DMA controller starts transfer, this register may not contain the transferred bytes or scan lines (read value is not guaranteed) because this register is used for control transfer.

- DA mode
Stores the destination side transferred bytes.
The size of the transferred data is added after each transaction.
The valid value for SIX is from "00000000_H" (0 byte) to "01000000_H" (2^{24} bytes) because the maximum transfer size of the DMA controller is 2^{24} bytes.
- SCA, DCA mode
The value of SIX is compared after each transfer.
When ICE=1 in the CCTR register, if both match, the ACT bit in the CCTR register is cleared to "0" and the channel is stopped after completing the transfer at the time of match.
- 2D mode
Stores the remaining scan lines.
It is decremented (-1) after completing transfer of each scan line. Specifying "0" is equivalent to specifying 4G (2^{32}).
The maximum scan line of the DMA controller is 2^{24} . Therefore, a value between "00000001_H" (1 line remaining) and "01000000_H" (2^{24} lines remaining) can be set in SIX. Other values are prohibited.

25.3.9 Byte Count Limit Registers (BCL0 to BCL3)

These are 32-bit registers that store the total number of data bytes to be transferred. They are available for each channel.

Figure 25.3-9 shows the bit configuration of the byte count limit registers (BCL0 to BCL3).

Figure 25.3-9 Bit Configuration of Byte Count Limit Registers (BCL0 to BCL3)

bit	31		0
	<div>BCL</div>		
Attribute	R/W		
Initial value	0		
R/W: Read/write allowed			

[bit 31 to bit 0]: BCL (byte count limit)

Stores the number of data bytes to be transferred.

The maximum transfer byte of the DMA controller is 2²⁴ bytes. Therefore, a value between "00000001_H" (1 byte) and "01000000_H" (2²⁴ bytes) can be set in BCL. Other values are prohibited.

The function changes as follows according to the channel mode.

Furthermore, this register value must be aligned with the size specified by CCTR register bits SSIZ/DSIZ. See "■ Settings specific to each channel" for the alignment method.

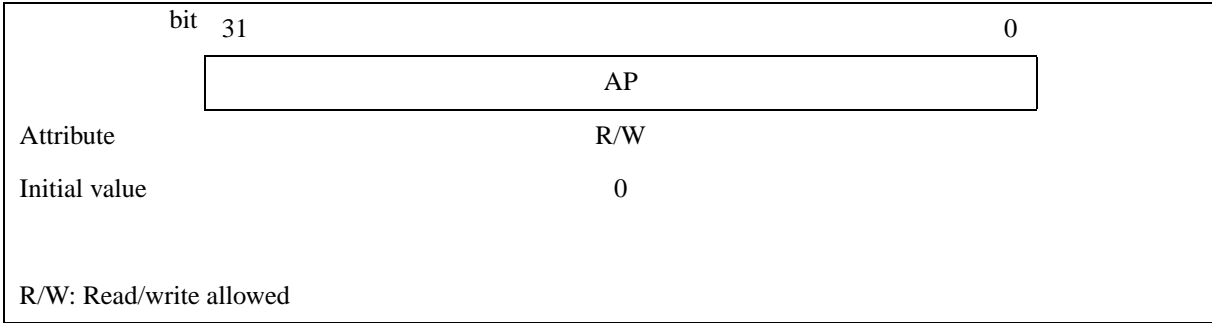
- DA mode
End of transfer is assumed when the number of bytes specified by BCL has been transferred.
- SCA, DCA mode
PIX is cleared to "0" when the number of bytes specified by BCL has been transferred. Channel operation continues.
- 2D mode
The following operations are performed when the number of bytes specified by BCL has been transferred. Channel operation ends if the value of SIX becomes "0" as the result of decrement. Operation continues in all other cases.
 - SBA and DBA are updated using the value of AP.
 - PIX is cleared to "0".
 - Value of SIX is decremented.

25.3.10 Alternate Pointer Registers (APR0 to APR3)

These are 32-bit registers used to store the frame buffer width when the channel mode is 2D. They are available for each channel.

Figure 25.3-10 shows the bit configuration of the alternate pointer registers (APR0 to APR3).

Figure 25.3-10 Bit Configuration of Alternate Pointer Registers (APR0 to APR3)



[bit 31 to bit 0]: AP (alternate pointer)

These bits contain the offset to the transfer start address in the next scan line.

The maximum offset that can be set for DMA controller is $2^{24}-1$. Therefore, a value between "00000000_H" (0 byte) and "00FFFFFF_H" (offset: $2^{24}-1$) can be set in AP. Other values are prohibited.

When transfer of one scan line worth of valid data completes, the value of AP is added to or subtracted from the base register. Furthermore, this register value must be aligned with the size specified by the CCTR register bits SSIZ/DSIZ. See "■ Settings specific to each channel" for the alignment method.

25.3.11 DMA-halt by NMI Flag Register (DNMIR)

This register controls the halt of DMA transfer by user NMI. This register is used as an interrupt request flag because NMI request has no interrupt request flag, unlike other peripheral functions. However, this register controls only DMA transfer halt function and does not affect the NMI request itself.

Figure 25.3-11 shows the bit configuration of the DMI-halt by NMI flag register (DNMIR).

Figure 25.3-11 Bit Configuration of DMA-halt by NMI Flag Register (DNMIR)

bit	7	6	0
	NMIH	Reserved	
Attribute	R/W		
Initial value	0		
R/W: Read/write allowed			

[bit7]: NMIH (NMI halt control flag)

This is a flag that indicates a user NMI request has occurred.
When this bit is set, a DMA transfer halt request occurs. To resume DMA transfer, clear this bit when user NMI handler processing completes.

Written Value	DMA transfer halt by user NMI
0	DMA transfer is not halted
1	User NMI occurs and DMA transfer is halted

[bit 6 to bit 0]: Reserved bits

Write	Always write "0".
Read	"0" is read.

25.3.12 DMA-Halt by Interrupt Level Register (DILVR)

A reference interrupt level is set in order to control halting of DMA transfer by peripheral function's interrupt request. Because this register is initialized to 1F_H when reset, DMA transfer is halted for all interrupt requests until the setting is changed.

Figure 25.3-12 shows the bit configuration of the DMI-Halt by interrupt level register (DILVR).

Figure 25.3-12 Bit Configuration of 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/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	1	1	1	1
R/W: Read/write allowed								

[bit 7 to bit 5]: Reserved bits

Write	Always write "0".
Read	"0" is read.

[bit 4 to bit 0]: LVL4 to LVL0 (DMA-halt interrupt level)

Set the interrupt level to halt DMA transfer.

A DMA transfer halt request is issued when a peripheral function interrupt higher than the level set in this register occurs. Bit 4 is fixed to "1" and bits 3 to 0 are settable.

LVL4	LVL3	LVL2	LVL1	LVL0	DMA-Halt by Interrupt Level Setting
1	1	1	1	1	Halt DMA transfer for all peripheral function interrupt requests
1	1	1	1	0	Halt DMA transfer for peripheral function interrupt requests with level higher than 1E _H
1	1	1	0	1	Halt DMA transfer for peripheral function interrupt requests with level higher than 1D _H
and so on					and so on
1	0	0	0	1	Halt DMA transfer for peripheral function interrupt requests with level higher than 10 _H
1	0	0	0	0	Do not halt DMA transfer for peripheral function interrupt request

25.4 Explanation of Operations

This section describes the interrupt controller operation.

25.4.1 Settings

This section describes the settings common to all channels and specific to each channel.

■ Settings common to all channels

The following settings are common to all channels:

- Channel priority settings
- DMA halt by interrupt settings

Details of each setting are described below.

Channel priority settings

How to determine priority between channels is set with the AT bit of the GCFR register.

Channel priority is determined by one of the following methods:

- Round-robin priority [AT=0]
- Fixed priority [AT=1]

DMA halt by interrupt settings

DILVR register bits LVL3 to LVL0 are used to set the level of interrupts that have priority over DMA transfer.

Levels between 1F_H and 10_H can be set.

■ Settings specific to each channel

The following settings are specific to each channel:

- DMA transfer request setting
- Transfer mode setting
- Acknowledge timing setting
- Transfer unit setting
- Transfer address update setting
- Channel mode and transfer address/transfer data settings
- Interrupt enable setting
- DMA Transfer buffer enabled access setting
- SCA/DCA mode address compare detection setting

Details of each setting are described below.

DMA transfer request setting

CCFR register bits RS3 to RS0 are used to set the transfer request source that accepts transfer request.

The following two transfer requests are available:

- Internal (constant) transfer request [RS3 to RS0=0000]
- Transfer request from ICH [RS3 to RS0=0001]

Transfer mode setting

The transfer mode of DMA transfer is set with the TM1 and TM0 bits of the CCFR register.

The following two transfer modes are available:

- Block transfer [TM1, TM0=00]
- Burst transfer [TM1, TM0=01]

Transfer unit setting

The transfer size of DMA transfer from the source side is set with the SSIZ3 and SSIZ0 bits of the CCTR register.

The transfer size of DMA transfer from the destination side is set with the DSIZ3 and DSIZ0 bits of the CCTR register.

The following four transfer units are available for both the SSIZ and DSIZ bit:

- 1 byte [SSIZ3 to SSIZ0=000, DSIZ3 to DSIZ0=000]
- 2 bytes [SSIZ3 to SSIZ0=001, DSIZ3 to DSIZ0=001]
- 4 bytes [SSIZ3 to SSIZ0=010, DSIZ3 to DSIZ0=010]
- 32 bytes [SSIZ3 to SSIZ0=101, DSIZ3 to DSIZ0=101]

Transfer address update setting

The source side address update for DMA transfer is set with the SAU1 and SAU0 bits of the CCTR register.

The destination side address update for DMA transfer is set with the DAU bit of the CCTR register.

The following three address updates are available for both the SAU and DAU bit:

- Hold [SAU1, SAU0=00, DAU=00]
- Increment [SAU1, SAU0=01, DAU=01]
- Decrement [SAU1, SAU0=10, DAU=10]

Channel mode and transfer address/transfer data settings

The DMA transfer channel mode is set with the CM1 and CM0 bits of the CCFR register.

The following four channel modes are available:

- DA (Dual Addressing) mode [CM1, CM0=00]
- SCA (Source Circular Addressing) mode [CM1, CM0=01]
- DCA (Destination Circular Addressing) mode [CM1, CM0=10]
- 2D (2-Dimensional Addressing) mode [CM1, CM0=11]

The function of SBA/DBA/PIX/SIZ/BCL/APR transfer address/transfer data setting registers change according to the channel mode.

See "■ Channel Modes" for the details of each channel mode.

Interrupt enable setting

DMA controller interrupt is enabled with the IE bit of the CCTR register.

DMA transfer buffer enabled access setting

Buffering of DMA transfer access is enabled with the BUF bit of the CCTR register.

SCA/DCA mode address compare detection setting

Address comparison detection in SCA/DCA mode is enabled with the ICE bit of the CCTR register.

■ Channel Modes

The relationship between the setting of each register and transfer address/transfer data for each mode and the alignment of settings are described below.

The initial value of the transfer address/transfer data setting registers is undefined. Initialize these register with software regardless of whether the channel is used or not.

● Relationship between each register setting and transfer address/transfer data for each mode

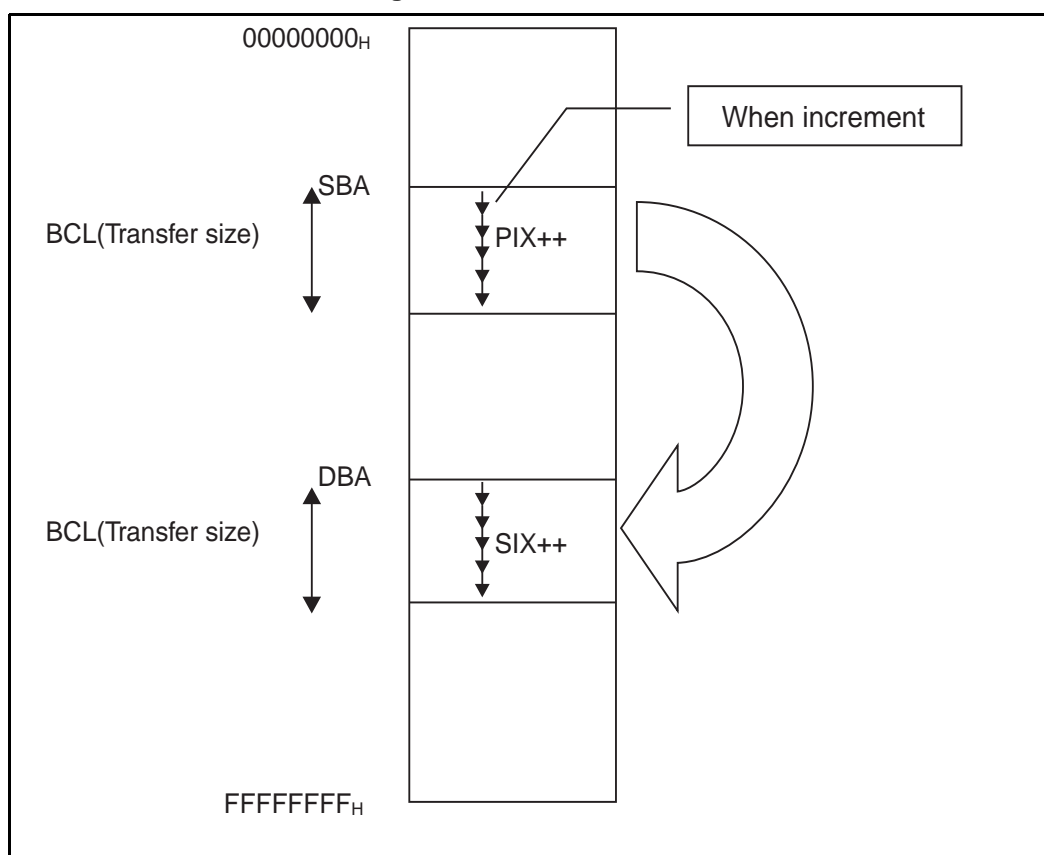
DA mode

This is the most basic transfer mode.

The number of bytes specified by BCL are transferred while repeating the process of transferring from address SBA+PIX to address DBA+SIX and updating PIX and SIX by the specified amount.

- SBA :FBeginning address of transfer source data area
- DBA :FBeginning address of transfer destination data area
- PIX :FSet "0"
- SIX :FSet "0"
- BCL :FTransfer size
- APR :FSet "0"

Figure 25.4-1 DA Mode



SCA mode

This mode performs transfer from circular buffer in memory to fixed address.

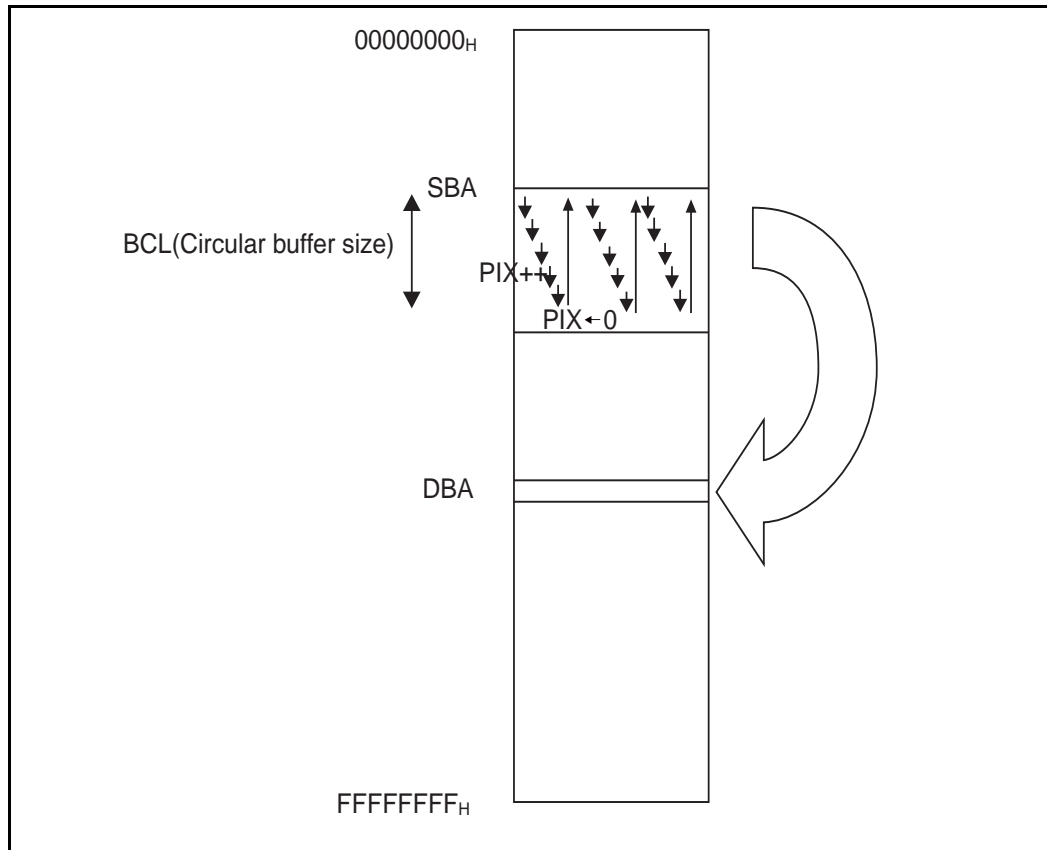
This mode cannot be used together with internal (constant) request setting.

Transfer is performed from address specified by SBA+PIX to fixed address specified by DBA. The updating of PIX by a specified amount after transfer is similar to DA mode, but if the value of PIX reaches BCL, PIX is returned to "0" and operation is continued without stopping.

In SCA mode, transfer can be stopped when SBA+PIX=SIX by setting the ICE bit of CCTR register to "1".

- SBA :Beginning address of circular buffer
- DBA :Transfer destination I/O address
- PIX :FSet "0"
- SIX :FAddress within circular buffer to stop transfer
- BCL :FCircular buffer size
- APR :FSet "0"

Figure 25.4-2 SCA Mode



DCA mode

This mode performs transfer from fixed address to circular buffer in memory.

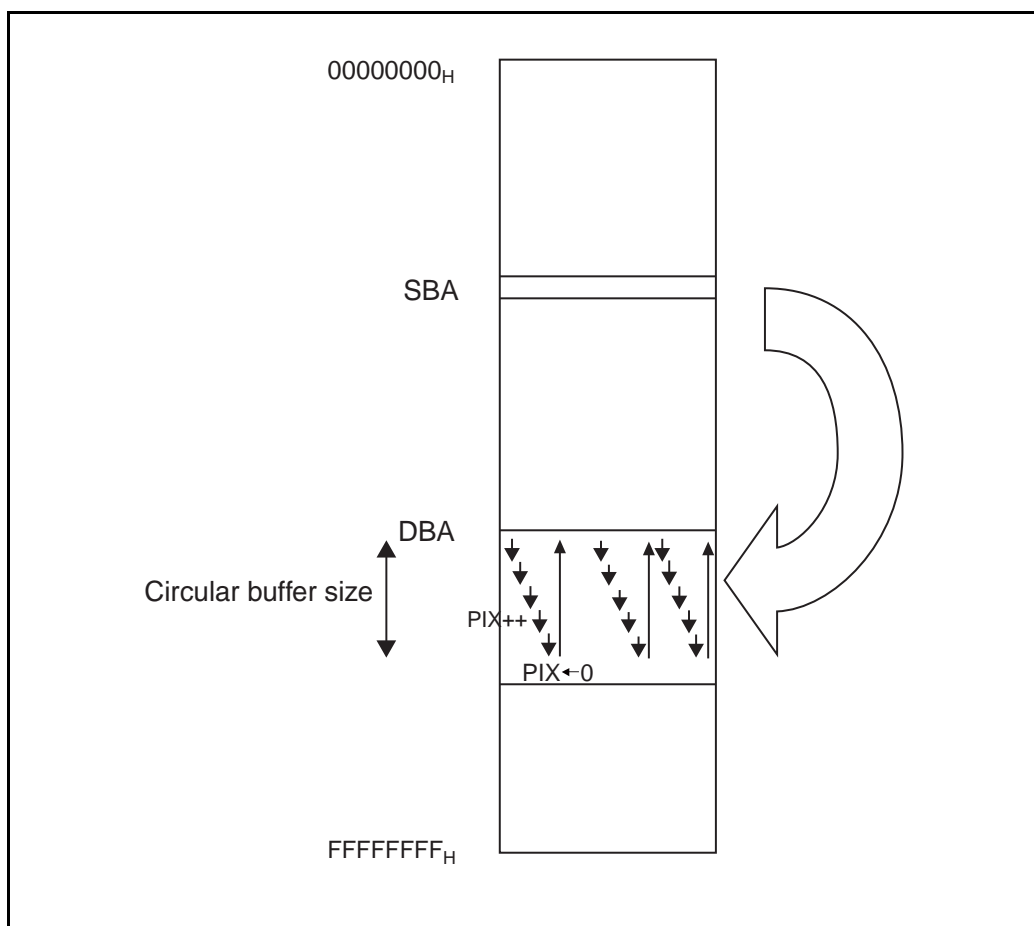
This mode cannot be used together with internal (constant) request setting.

Transfer is performed from fixed address specified by SBA to address specified by DBA+PIX. Similar to SCA mode, PIX is returned to "0" and transfer is continued when the value of PIX reaches BCL.

In this mode also, transfer can be stopped when DBA+PIX=SIX by setting the ICE bit of CCTR register to "1".

- SBA :FTransfer source I/O address
- DBA :FBeginning address of circular buffer
- PIX :FSet "0"
- SIX :FAddress within circular buffer to stop transfer
- BCL :FCircular buffer size
- APR :FSet "0"

Figure 25.4-3 DCA Mode



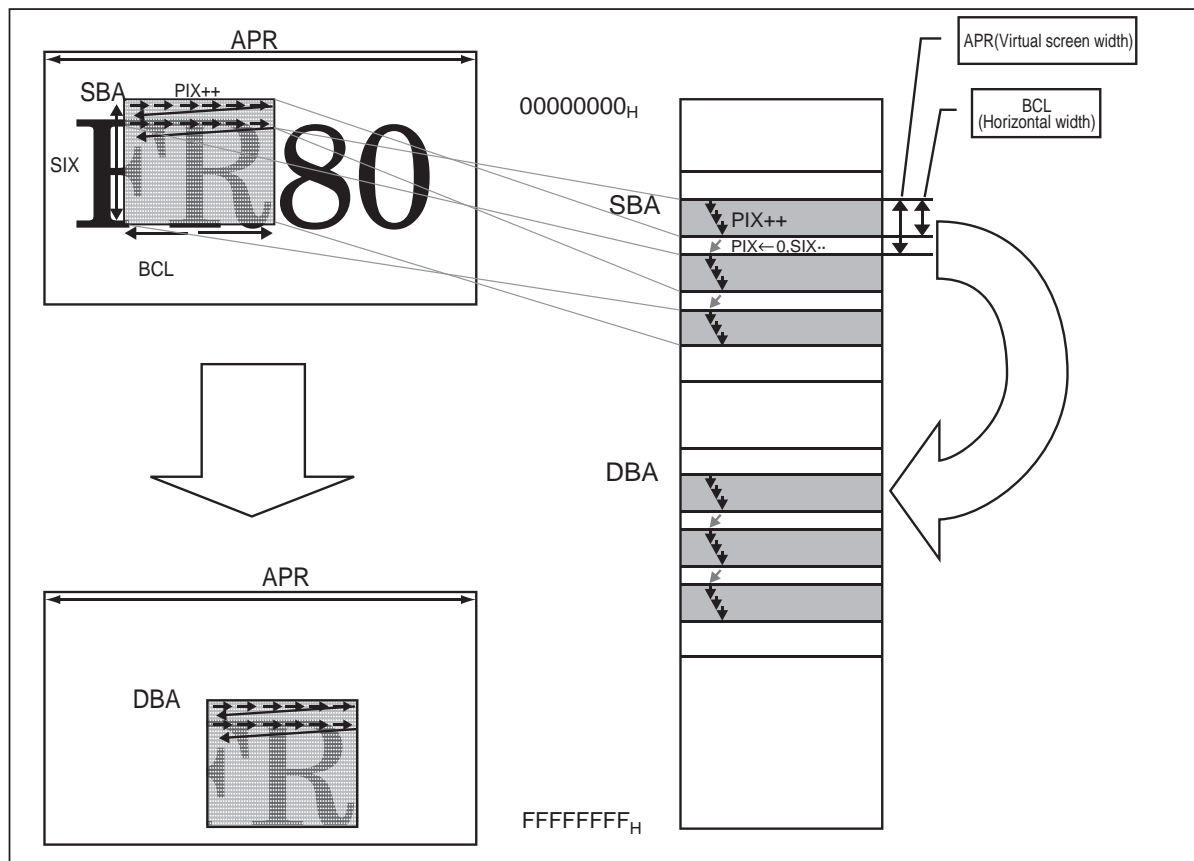
2D mode

In this mode, a rectangular area inside the frame buffer in memory can be transferred.

To use this mode, start transfer by setting the channel control registers as follows:

- SBA :Reference address of the transfer source rectangular area (Example: Address of the lower left corner pixel)
- DBA :Reference address of the transfer destination rectangular area (Example: Address of the lower left corner pixel)
- PIX :FSet "0"
- SIX :FNumber of scan lines in the rectangular area
- BCL :FNumber of transfer bytes per line of rectangular area
- APR :FNumber of bytes per line of virtual screen

Figure 25.4-4 2D Mode



● Transfer address/transfer data setting alignment

The SBA/DBA/PIX/SSIZ/BCL/APR transfer address/transfer data setting registers must be aligned on the size shown in the following table.

Transfer Mode	Registers				
	SBA	DBA	BCL	APR	SIX
SCA or transfer destination address is fixed	Larger of SSIZ and DSIZ	DSIZ(*1)	DSIZ(*2)	Larger of SSIZ and DSIZ	Larger of SSIZ and DSIZ
DCA or transfer source address is fixed	SSIZ(*1)	Larger of SSIZ and DSIZ	SSIZ(*3)	Larger of SSIZ and DSIZ	Larger of SSIZ and DSIZ
Both source and destination are not fixed	Larger of SSIZ and DSIZ	Larger of SSIZ and DSIZ	Larger of SSIZ and DSIZ	Larger of SSIZ and DSIZ	-

- (*1) If decrement is specified for the source side or destination side, align to the multiple of the size in SSIA2 to SSIZ0 or DSIZ2 to DSIZ0 of CCTR register bits, whichever is larger.
- (*2) More data than that is written to the transfer destination may be read from the transfer source because read is performed in the size indicated by the SSIZ bits of the CCTR register. To match the size read with the size sent to the transfer destination, align the SSIZ bits of the CCTR register in addition to the DSIZ bits. Also align to the multiple of SSIZ bits when decrement is specified for the source side.
- (*3) Write to transfer destination may not be performed with the DSIZ bits of the CCTR register. If write must be performed with the size in the DSIZ bits, also align with DSIZ bits. Also, align to the multiple of DSIZ bits when decrement is specified for the destination side. In addition, align to the multiple of DSIZ bits when destination side transfer is started with DREQ (ATS2 to ATS0=010 in CCFR register).

25.4.2 Activation

The DMA controller is activated by setting the CCTR register's ACT bit of each channel to "1". The DMA controller will have one of the following statuses according to the DMA transfer request set with the RS3 to RS0 of CCFR register bits:

- Internal (constant) transfer request [RS3 to RS0=0000]
DMA transfer starts immediately when DMA controller is activated.
- Transfer request from each resource [RS3 to RS0=0001]
CCTR register becomes ready to accept transfer request when DMA controller is activated. Transfer does not start until there is a transfer request.

25.4.3 Operation

This section describes the operation of the DMA controller.

■ Checking DMA controller activation

The DMA controller activation status for each channel is checked as follows:

Status check bit

If the DMA controller is active, the BUSY bit of the CSTR register is set to "1" to indicate that the channel is busy.

When the DMA controller becomes inactive because transfer has completed or has stopped, the BUSY bit is set to "0" to indicate that the channel is halted.

■ Transfer operations

The settings and operations of DMA controller transfer operation are as follows:

Transfer Mode

Transfer mode determines the type of transfer DMA controller performs for a transfer request.

The DMA controller supports the following transfer modes:

- Block transfer

The DMA controller transfers one transfer unit for a transfer request and then stops transfer until the next transfer request is received. A transfer unit is as follows:

- Determined by CCTR register bits SSIZ2 to SSIZ0, if the transfer request is made to the source side (ATS2 to ATs0 of CCFR register bits are 001).
- Determined by CCTR register bits DSIZ2 to DSIZ0, if the transfer request is made to the destination side (ATS2 to ATs0 of CCFR register bits are 010).

- Burst transfer

For a single transfer request, transfer request is made continuously within the channel for each transfer unit until the number of bytes set in the BCL register is transferred. For a single transfer request, transfer continues until transfer completes.

Transfer request

Transfer request determines from which transfer request source a transfer request is to be accepted.

The DMA controller supports the following transfer request sources:

- Internal (constant) transfer request

When internal (constant) transfer request is set, DMA transfer starts simply by setting the ACT bit of the CCTR register to "1" (channel activation). Therefore, perform all settings necessary for DMA transfer before setting the ACT bit to "1" of CCTR register.

- Transfer request from ICH

When transfer request from ICH is set, DMA transfer starts when a transfer request from ICH is detected while the channel is active.

Transfer unit

The following transfer units can be set independently for source side/destination side access.

- 1 byte
- 2 bytes
- 4 bytes
- 32 bytes

When the transfer unit is set to 1, 2, or 4 bytes, single access is performed on on-chip bus. When the transfer unit is set to 32 bytes, 8-bit wrap burst access is performed on on-chip bus.

Transfer address update

Transfer address update can be set to one of the following:

- Hold
- Increment
- Decrement

Depending on the above update setting, the transfer address index (PIX/SIX) is updated as follows after transfer of each transfer unit. However, the update method of some indexes (PIX/SIX) is determined by the channel mode. In which case, the transfer address is updated by the channel mode update method.

- Hold

The transfer address index is fixed and SIX/PIX is not updated.

- Increment

Source side: The transfer unit (SSIZ2 to SSIZ0 of CCTR register bits) is added.

Destination side: The transfer unit (DSIZ2 to DSIZ0 of CCTR register bits) is added.

- Decrement

Source side: The transfer unit (SSIZ2 to SSIZ0 of CCTR register bits) is subtracted.

Destination side: The transfer unit (DSIZ2 to DSIZ0 of CCTR register bits) is subtracted.

Acknowledge timing

The following timings can be set for the acknowledge of transfer request for each DMA transfer unit:

- No acknowledge
- Acknowledge at source side access
- Acknowledge at destination side access
- Acknowledge at both source side and destination side access

The transfer request and acknowledge operations at the above four timings are as follows:

- No acknowledge

For transfer request of each transfer size, only the larger size of SSIZ or DSIZ of CCTR register bits is read to the source side. Then, write is performed to the same size destination side. No acknowledge is output to source side or destination side.

- Acknowledge at source side access

For transfer request of each transfer size, only the size of SSIZ of CCTR register bits is read to the source side and acknowledge is output. Then write is performed to destination side, but no acknowledge is output.

- Acknowledge at destination side access

For transfer request of each transfer size, only the size of DSIZ of CCTR register bits is written to the destination side. Read to source side is performed automatically regardless of the transfer request and acknowledge is output. Therefore, the transfer source data must be prepared before activating the channel.

- Acknowledge at both source side and destination side access

For transfer request of each transfer size, only the size of the SSIZ of CCTR register bits is read to the source side and write is performed to the destination side only for the size of the DSIZ of CCTR register bits. Acknowledge is output for both source side and destination side.

Channel Modes

The DMA controller supports the following four channel modes:

- DA mode
- SCA mode
- DCA mode
- 2D mode

See "■ Channel Modes" for the details of each channel mode.

Access buffer enable

The DMA controller supports buffer enabled access to the DMA transfer target. Buffer enabled access is possible for target supporting buffer enabled access. However, when performing DMA transfer to target not supporting buffer enabled access, specification of buffer enabled access with the DMA controller is ignored.

Inter-channel transfer priority

The following two inter-channel transfer priorities are supported:

- Round-robin priority
The channel starting DMA transfer is assigned the lowest priority and priority of the channels that had priority less than that channel is increased by one. The priority is updated each time transfer unit transfer is performed.

- Fixed priority

The transfer priority is fixed as follows:

Channel 0 (highest) > channel 1 > channel 2 > channel 3 (lowest)

For either setting, if transfer requests occur simultaneously, transfer is performed for the channel with the highest priority among the ready channels.

For fixed priority, the following operation is performed when two or more channels are activated simultaneously.

Lower priority channels perform DMA transfer when higher priority channels are not performing DMA transfer. Therefore, there is a time lag from the time of transfer request until the transfer operation.

To keep this time lag within a certain value, use channel 0 with fixed priority or use round-robin priority.

DMA transfer data movement (normal transfer)

Figure 25.4-4 to Figure 25.4-10 show the transfer examples.

Figure 25.4-5 External Area to External Area Transfer

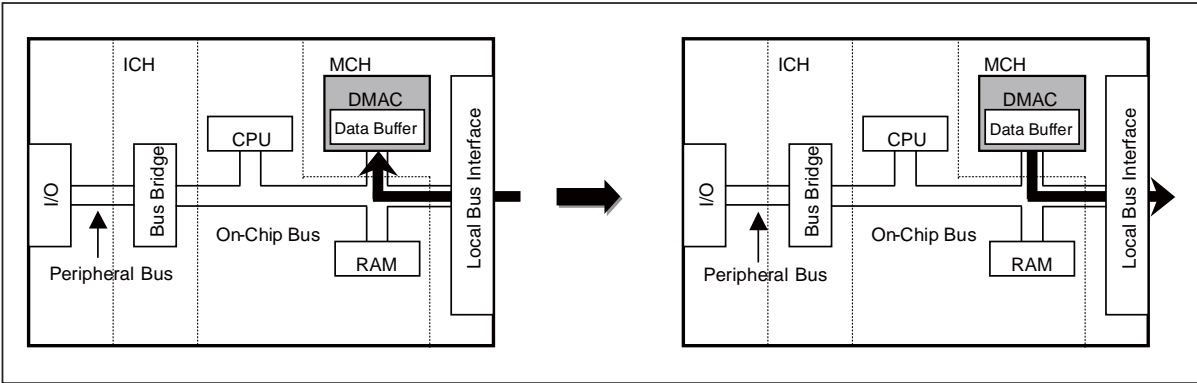


Figure 25.4-6 External Area to Internal RAM Area Transfer

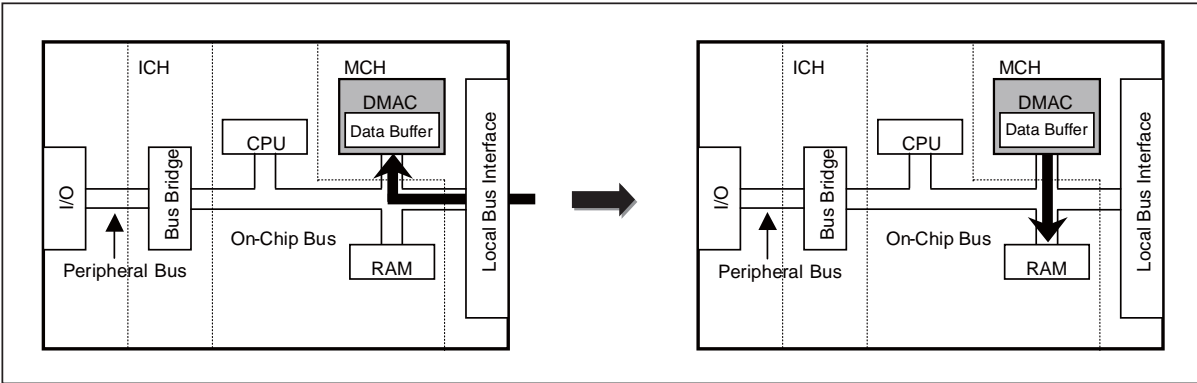


Figure 25.4-7 External Area to Internal IO Area Transfer

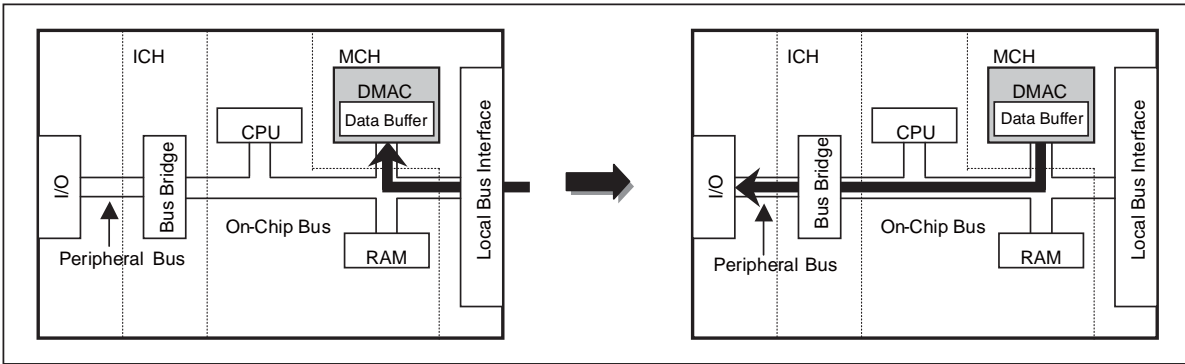


Figure 25.4-8 Internal RAM Area to External Area Transfer

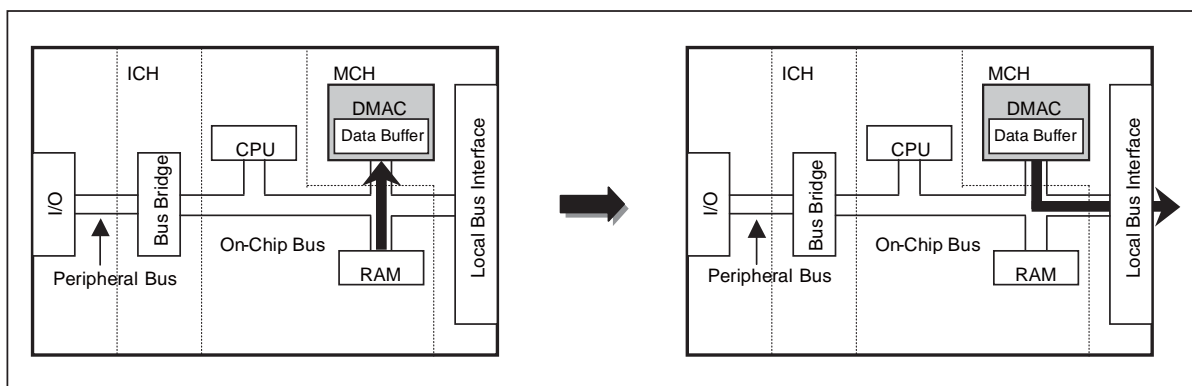


Figure 25.4-9 Internal RAM Area to Internal IO Area Transfer

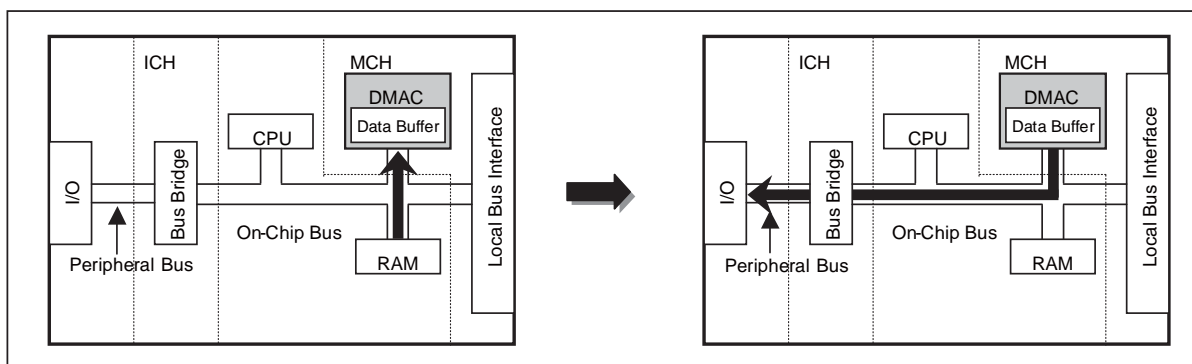
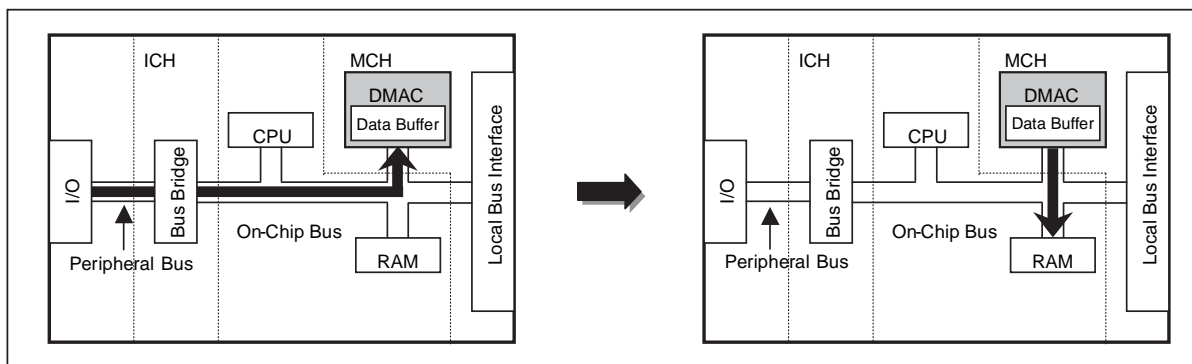


Figure 25.4-10 Internal IO Area to Internal RAM Area Transfer



■ Transfer completion and interrupts

This section describes the normal transfer completion and interrupts.

Status after normal transfer completion

When transfer completes normally, the ACT bit of the CCTR register becomes "0" because the DMA controller enters the halt state. In addition, the following CSTR register flag bits also change.

- BUSY bit
Becomes "0" to indicate that the DMA controller channel has entered halt state.
- INT bit
Becomes "1" to indicate that an interrupt request (DMA transfer normal completion) was detected.
- NE bit
Becomes "1" to indicate that the configured transfer operation has completed normally.

Interrupt

When transfer completes normally, the INT bit of the CSTR register becomes "1". If interrupt of the channel is enabled with the IE bit of the CCTR register, the DMA controller outputs an interrupt request to the interrupt controller.

An interrupt request is cleared by writing "0" to the INT bit of the CSTR register.

Processing after transfer completion

When transfer completes normally, be sure to perform the following before performing transfer once more.

- Write "0" in the INT bit of the CSTR register (initialize the interrupt flag)
- Write "0" in the NE bit of the CSTR register (initialize the normal transfer completion flag)

■ Suspending transfer and interrupts

This section describes the suspension of transfer and interrupts.

Transfer operation suspension request

The DMA controller suspends transfer for the following requests:

- When channel mode is SCA or DCA, transfer suspension caused by the transfer halt function due to address match detection
- Transfer suspension due to error response detection during on-chip bus read access
- Transfer suspension due to error response detection during on-chip bus write access
- Transfer suspension due to transfer halt request from transfer request source
- Forced halt due to clearing of ACT bit of the CCTR for the active channel

Transfer operation suspension

The suspension of transfer operation is described for each request.

- When channel mode is SCA or DCA, transfer suspension caused by the transfer halt function due to address match detection

When the channel mode is SCA/DCA, transfer to any address can be suspended by setting the ICE bit of the CCTR register to "1". When the DMA controller suspends transfer, transfer can be continued by setting the ACT bit of the CCTR register to "1" because it transfers the data at the transfer address matching the value in SIX before suspending transfer.

Also, if the source side transfer size (SSIZ2 to SSIZ0 of CCTR register bits) and the destination side transfer size (DSIZ2 to DSIZ0 of CCTR register bits) differ, the value in SIX must be aligned with the larger transfer size. In that case, transfer is suspended after transferring up to the address where the total of source side transfer size and destination side transfer size match.

Figure 25.4-11 shows the example when the transfer mode is set to SCA.

Figure 25.4-11 When the Transfer Mode is set to SCA

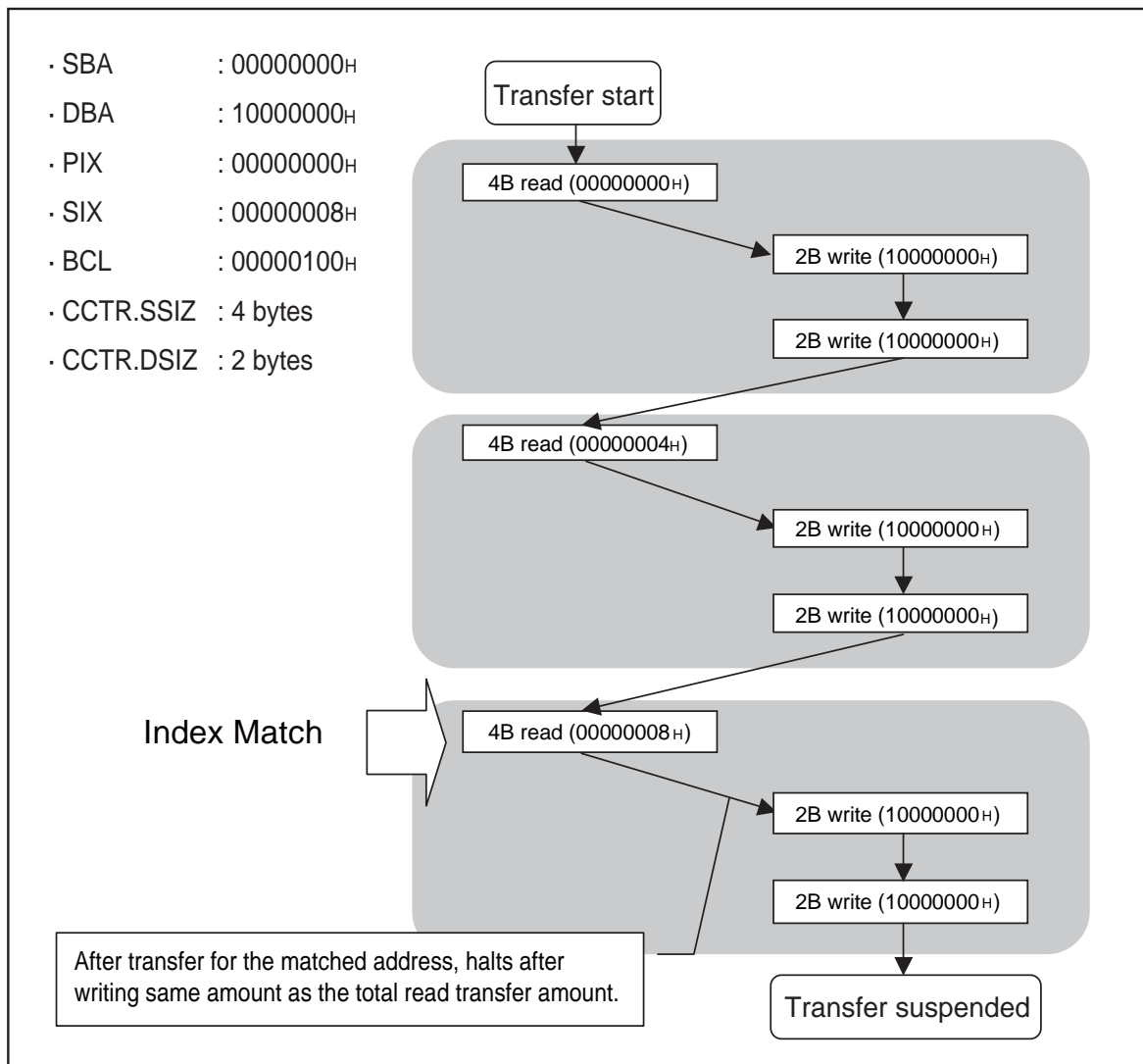


Figure 25.4-12 shows the example when the transfer mode is set to DCA.

Figure 25.4-12 When the Transfer Mode is set to DCA

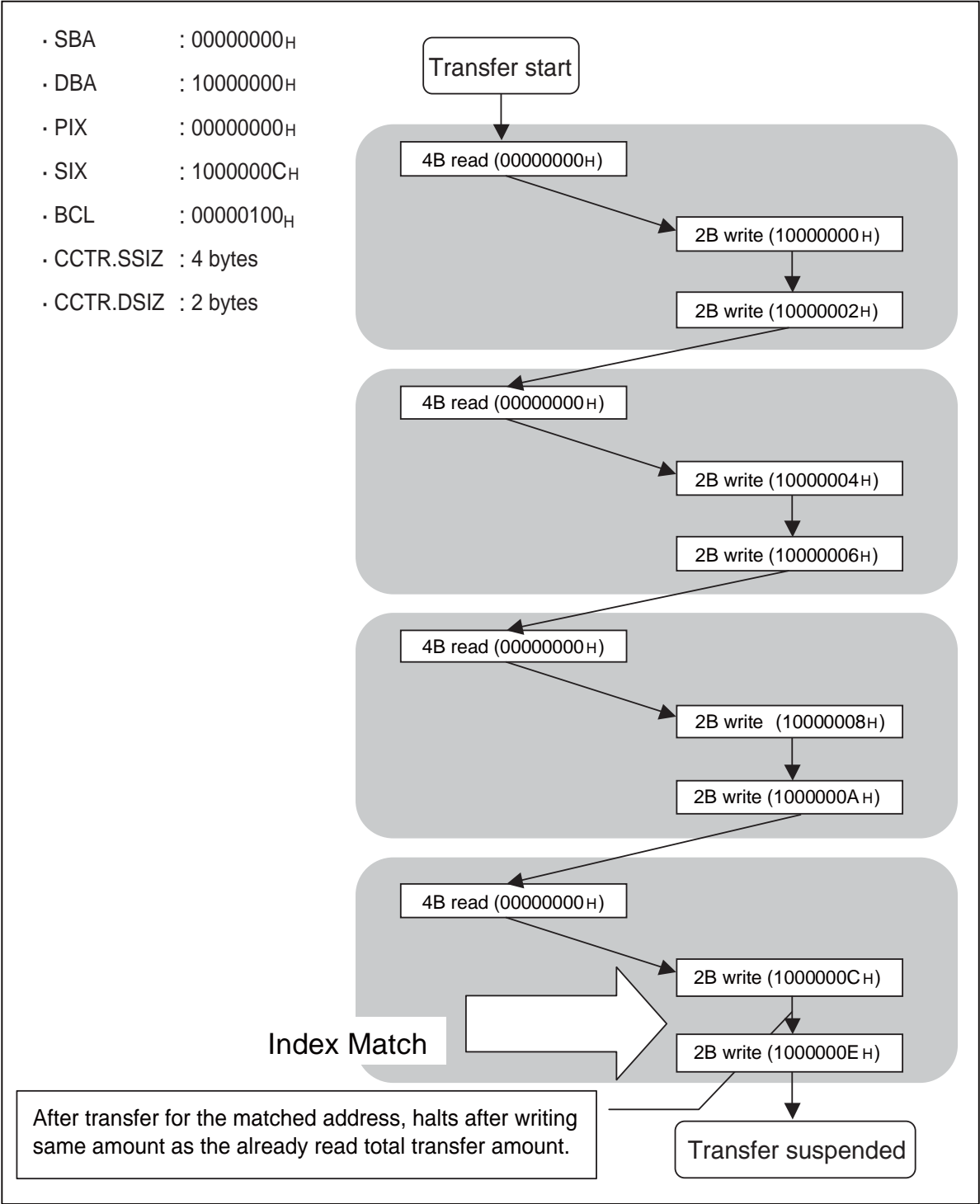


Figure 25.4-13 shows the example when the transfer mode is set to SCA.

Figure 25.4-13 When the Transfer Mode is set to SCA

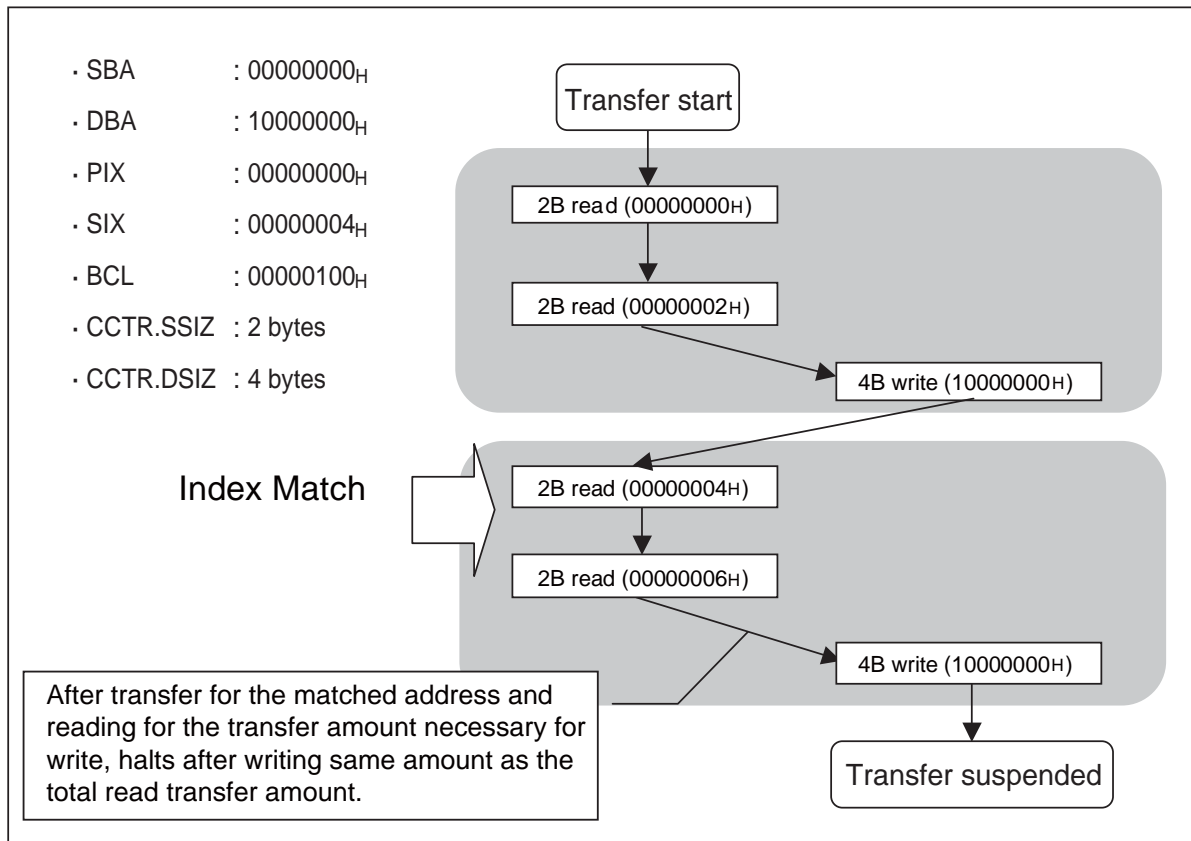
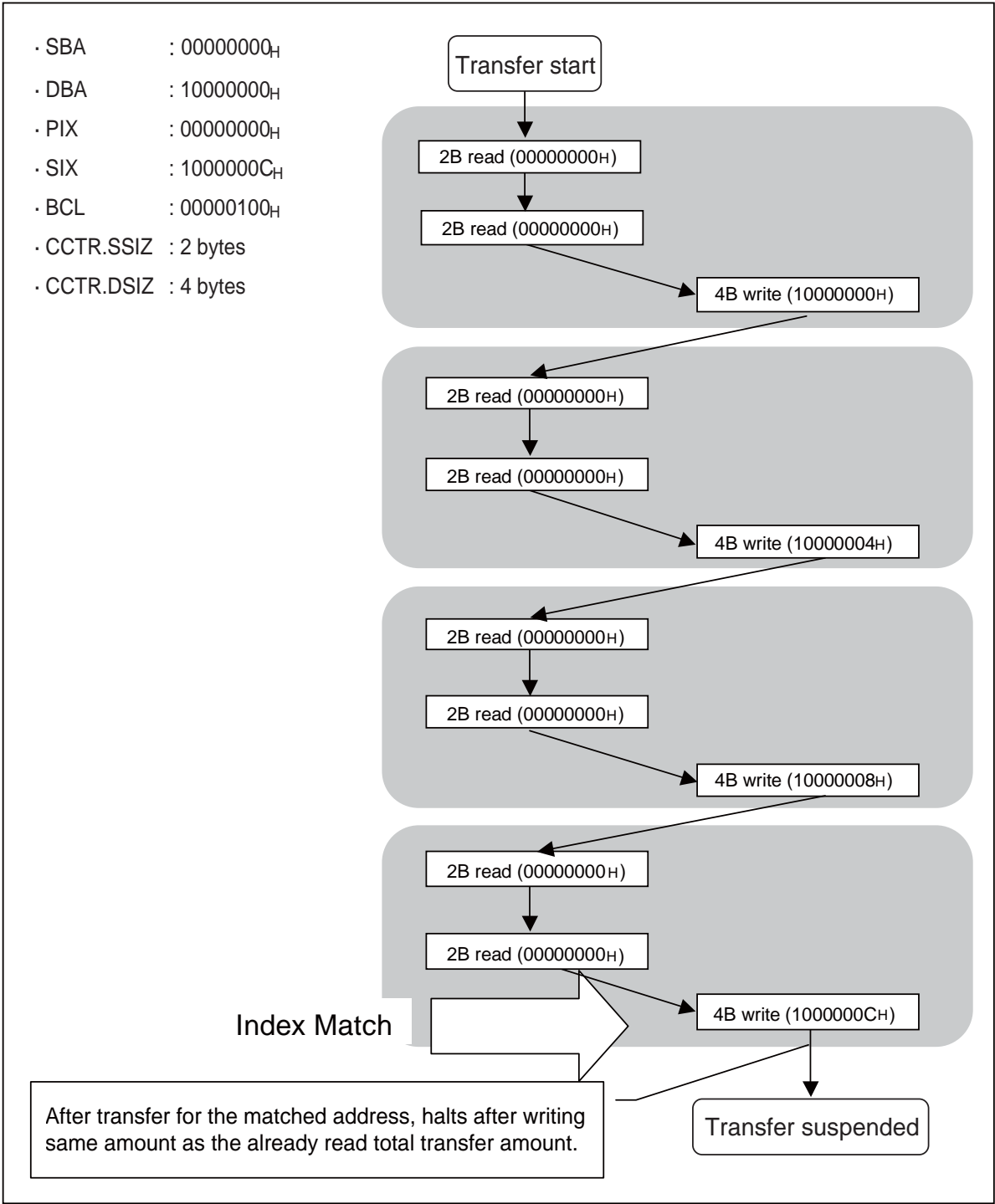


Figure 25.4-14 shows the example when the transfer mode is set to DCA.

Figure 25.4-14 When the Transfer Mode is set to DCA



- Transfer suspension due to error response detected during on-chip bus read/write access
When transfer is suspended due to an error, do not use data at the location of error. When an error occurs, the index register is not updated. Therefore, the address indicated by the base address and index is where the error occurred.
When an error occurs, that transfer cannot be resumed. Clear FIFO with the FC bit of the CCTR register and then set each register once again to start transfer.
- Transfer suspension due to transfer halt request from transfer request source
When a halt transfer request is detected from the transfer request source set with RS3 to RS0 of CCFR register bits, transfer is suspended after completing transfer of already accepted transfer requests. When "1" is written in the ACT bit of the CCTR register in that state, the DMA controller resumes transfer. When writing "0" or "1" in the ACT bit to temporarily suspend or resume transfer, read-modify-write must be performed in order to prevent updating other settings.
- Forced halt due to clearing of ACT bit of the CCTR register for the active channel
When "0" is written in the ACT bit of the CCTR register for the active channel, transfer is suspended after completing transfer of the already accepted transfer requests. When "1" is written in the ACT bit in that state, the DMA controller resumes transfer. When writing "0" or "1" in the ACT bit to temporarily suspend or resume transfer, read-modify-write must be performed in order to prevent updating other settings.

Status after transfer suspension

When transfer is suspended, the ACT bit of the CCTR register becomes "0" because the DMA controller enters the halt state. In addition, the CSTR register flag bits shown in Table 25.4-1 also change regardless of the transfer operation suspend request.

Table 25.4-1 CSTR Register Flag Bits that Change Regardless of Transfer Operation Suspend Request

Bit	Change
BUSY	Becomes "0" to indicate that the DMA controller channel has entered the halt state.
INT	Becomes "1" to indicate that an interrupt request (DMA transfer suspension) was detected.

In addition, the CSTR register flag bits shown in Table 25.4-2 change due to the transfer operation suspend request.

Table 25.4-2 CSTR Register Flag Bits that Change Due to Transfer Operation Suspend Request

Bit	Change
CE	When the channel mode is SCA or DCA, becomes "1" if transfer operation is suspended by the transfer halt function due to address match. Otherwise, it does not change.
RER	Becomes "1" when transfer operation is suspended because error was detected during on-chip bus read access. Otherwise, it does not change.
WER	Becomes "1" when transfer operation is suspended because error was detected during on-chip bus write access. Otherwise, it does not change.
STP	Becomes "1" when transfer operation was suspended due to transfer halt request from the transfer request source. Otherwise, it does not change.
FED	Becomes "1" when halting of transfer operation is forced by writing "0" in the ACT bit of the CCTR register for the active channel. Otherwise, it does not change.

Interrupt

When transfer is suspended, the INT bit of the CSTR register becomes "1". If interrupt of the channel is enabled with the IE bit of the CCTR register, the DMA controller outputs an interrupt request to the interrupt controller.

An interrupt request is cleared by writing "0" in the INT bit.

Processing while transfer is suspended

After transfer is suspended, be sure to perform the following before performing transfer once more.

- Write "0" in the INT bit of the CSTR register (initialize the interrupt flag)
- Write "0" in the CE/RER/WER/STP/FED bits of the CSTR register (initialize the transfer suspend request flag)

Processing of data not transferred due to transfer suspension

If transfer is suspended due to transfer halt request from the transfer request source or forced to halt due to clearing of the ACT bit of the CCTR register for the active channel, data may be remaining in the DMA controller data buffer for which write has not completed. In such case, FS6 to FS0 of the CSTR register can be read when the BUSY bit of the CSTR register is "0" and checked against Table 25.4-3 and Table 25.4-4.

Table 25.4-3 Processing of Data Not Transferred due to Transfer Suspension (1/2)

Status	SSIZ = DSIZ		SSIZ < DSIZ		
	FS = 0	FS != 0	FS = 0	FS != 0 and FS[6] = 0	FS != 0 and FS[6] = 1
Un-transferred data	None	Remaining	None	Remaining	Remaining
Valid data length after write	IX	IX + SSIZ	IX	IX + FS	BCL

Table 25.4-4 Processing of Data Not Transferred due to Transfer Suspension (2/2)

Status	SSIZ > DSIZ		
	FS = SSIZ or (IX = 0 and FS != 0)	FS != SSIZ and (BCL % SSIZ <= BCL - IX) and not(IX = 0 and FS != 0)	FS != SSIZ and (BCL % SSIZ > BCL - IX) and not(IX = 0 and FS != 0)
Un-transferred data	None	Remaining	Remaining
Valid data length after write	IX	IX + SSIZ - FS	BCL

Also, if there is un-transferred data remaining in the data buffer of the DMA controller, that data can be written to the destination side. To do that, set the ACT and FC bits of the CCTR register to "1" at the same time without changing any other settings. In this case, transfer is performed with transfer size indicated by DSIZ2 to DSIZ0 of CCTR registers and the data starting from DBA with the length shown in the following table is valid. Also, the value of IX depends on the mode and is the value of SIX before write in DA mode and the value of PIX before write in SCA/DCA/2D mode.

Mode	Value of IX
DA	Value of SIX before write
SCA	Value of PIX before write
DCA	Value of PIX before write
2D	Value of PIX before write

■ Halting DMA transfer when interrupt occurs

This section describes the operation of the function to halt DMA transfer when interrupt occurs.

Purpose of halting DMA transfer

Response to interrupt is an important factor for applications that depends on real time processing.

During DMA transfer, the DMA controller which has higher bus right assignment priority has the bus right rather than the CPU. While performing DMA transfer, the on-chip bus system grants bus right to the CPU only between DMA controller's bus accesses. Therefore, when an interrupt occurs during DMA transfer, the following situation occurs:

- Bus right is not immediately granted to CPU
- CPU executes the interrupt handler between DMA controller bus accesses

Consequently, interrupt response and processing are delayed significantly. In order to prevent this situation, a function to halt DMA transfer when an interrupt occurs is necessary.

Relationship between DMA transfer halt level and interrupt level

Depending on the relationship between the DMA transfer halt level set with LVL4 to LVL0 of the DILVR register and the level of the occurred interrupt, the CPU core behaves as follows when an interrupt occurs.

- Generated interrupt level > Level set in DILVR register
DMA transfer is halted and the CPU interrupt handler is executed with priority.
- Generated interrupt level ≤ Level set in DILVR register
DMA transfer is not halted. DMA transfer is executed with priority.

Interrupt handler processing

With regard to DMA transfer suspend function due to interrupt, the interrupt handler should process interrupts according to the following flow:

- Interrupts other than those due to user NMI
 1. Interrupt handler main processing
 2. Clear interrupt flag
 3. Return from interrupt handler with RETI instruction
- Interrupts due to user NMI
 1. Interrupt handler main processing
 2. Clear the NMIH bit of the DNMIR register
 3. Return from interrupt handler with RETI instruction

In the above flows, DMA transfer may resume before executing the RETI instruction. This can occur in the following cases.

When the interrupt level just after step 2 is below DMA transfer suspension level

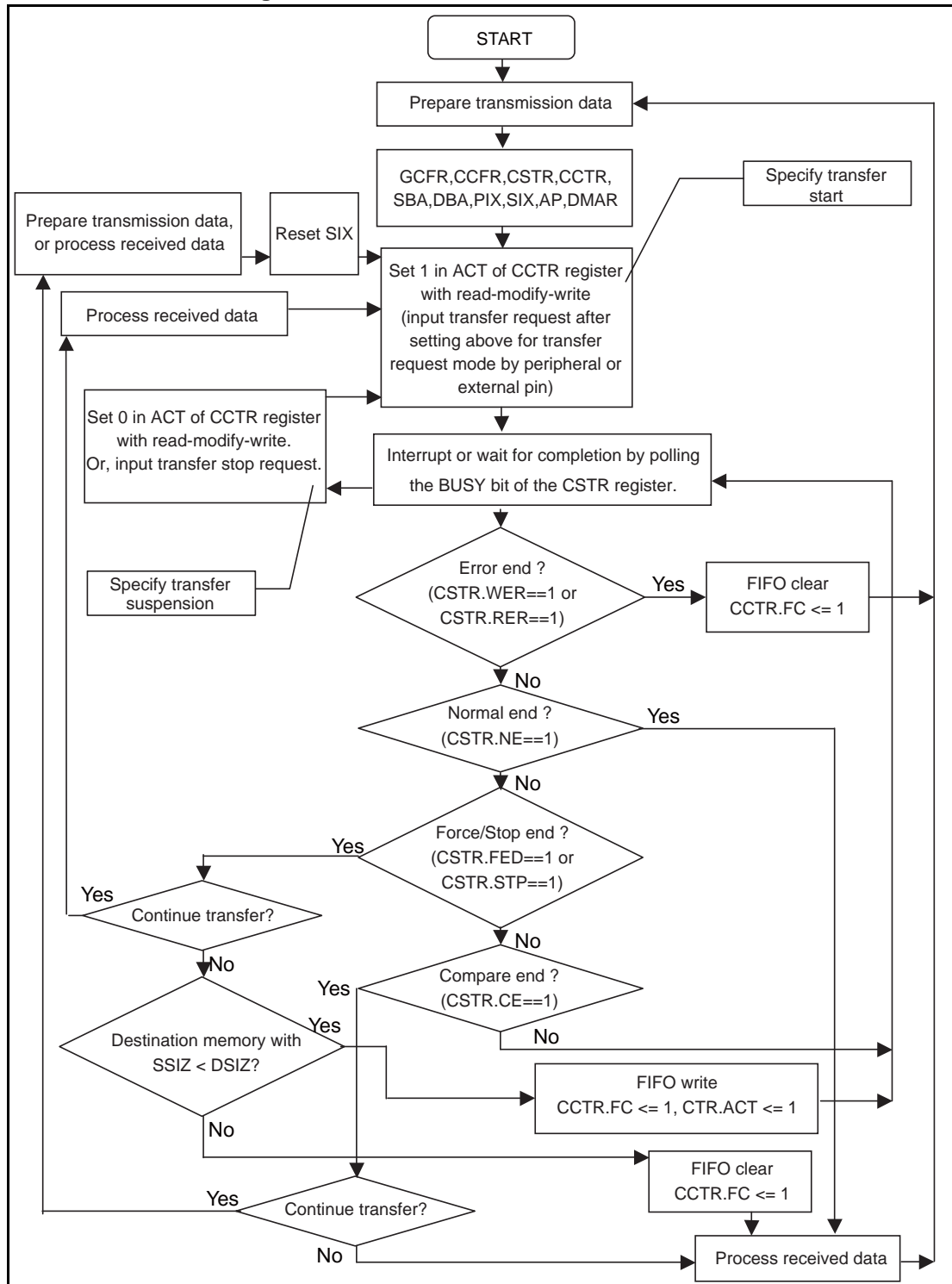
DMA transfer suspension in emulator mode

In emulator mode, DMA transfer is suspended regardless of the DMA transfer suspension level. During ICE connection, DMA transfer can be suspended even during execution of RETI instruction to transfer from emulation state to trace state with the DSU's DMA transfer suspend function (DMST bit of the ECTL3 register).

25.5 Control Flow

This section describes the DMA controller control flow.

Figure 25.5-1 DMA Controller Control Flow



25.6 Notes on Use

Note the following points when using the DMA controller.

Operation when interrupt of peripheral function not supporting the DMA transfer's transfer activation request is set as the transfer activation request

If the interrupt of peripheral function not supporting the DMA transfer's transfer activation request (such as DMA controller interrupt or delayed interrupt) is set as the transfer activation request with ICH of the IORR register, the following operation is performed when an interrupt occurs.

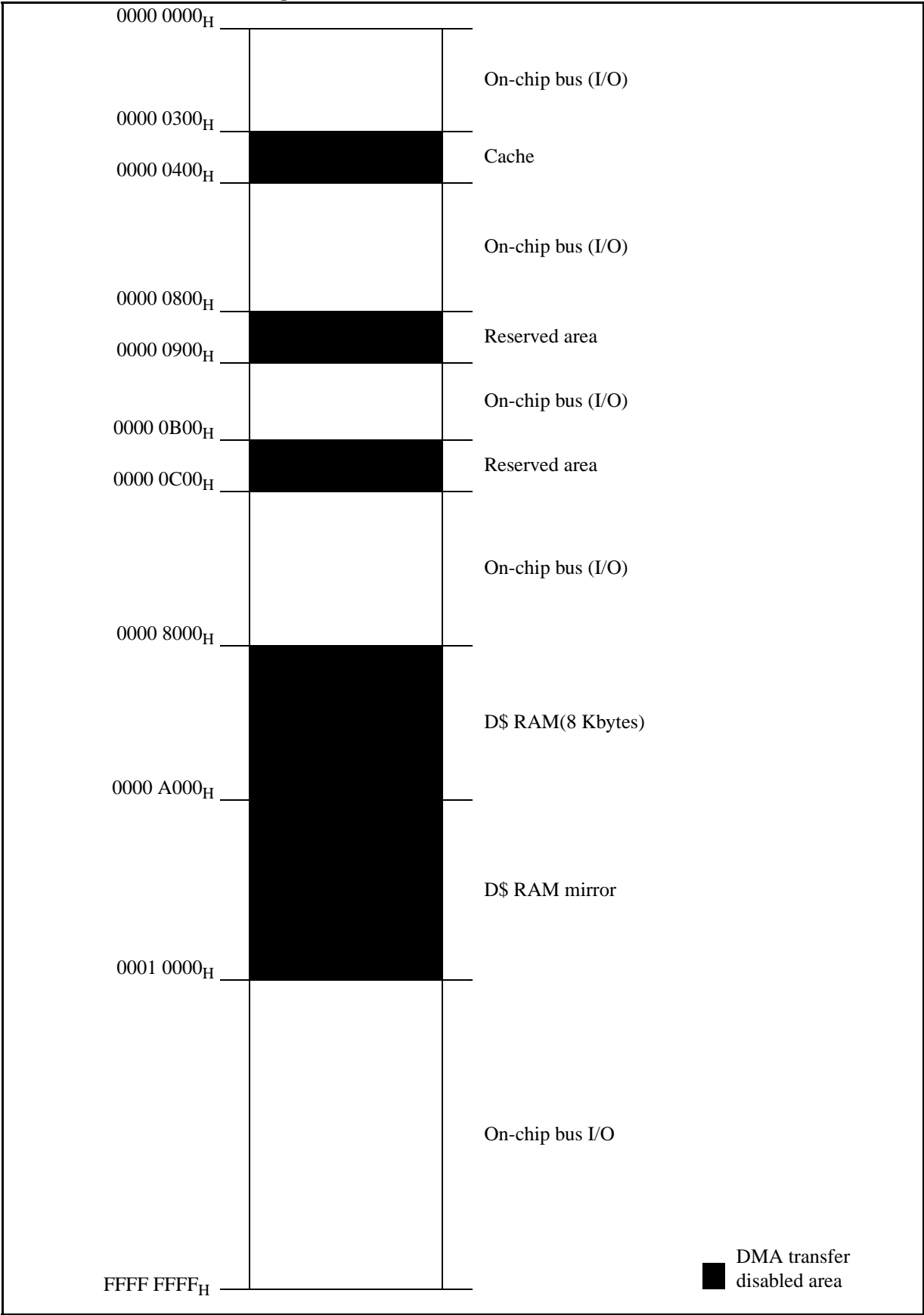
1. The transfer activation request and transfer stop request become active.
2. The DMA controller activates transfer with the transfer activation request. The channel stops without performing transfer because transfer is stopped with the transfer stop request at the same time.
3. The DMA controller channel becomes stop ended.

DMA transfer transfer prohibited area

With this product type, DMA transfer is prohibited for the following areas:

- Reserved area 00000800_H to 000008FF_H
- Reserved area 00000B00_H to 00000BFF_H
- Cache area 00000300_H to 000003FF_H
- Data cache RAM area 00008000_H to 00009FFF_H
- Data cache RAM mirror area 0000A000_H to 0000FFFF_H

Figure 25.6-1 DMA Transfer Disabled Area



CHAPTER 26 **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).

- 26.1 Overview
- 26.2 Configuration
- 26.3 Registers
- 26.4 An Explanation of Operations and Setting Procedure Examples

26.1 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.

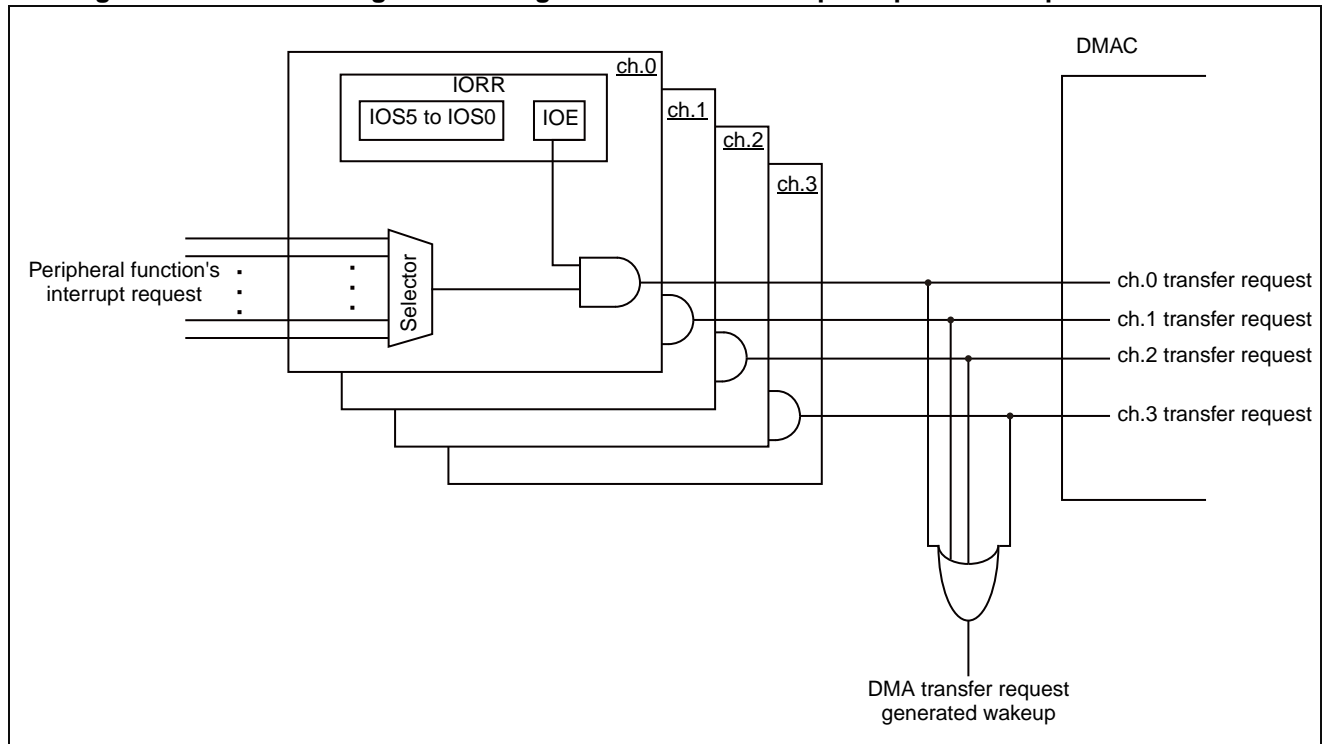
26.2 Configuration

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 26.2-1 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.

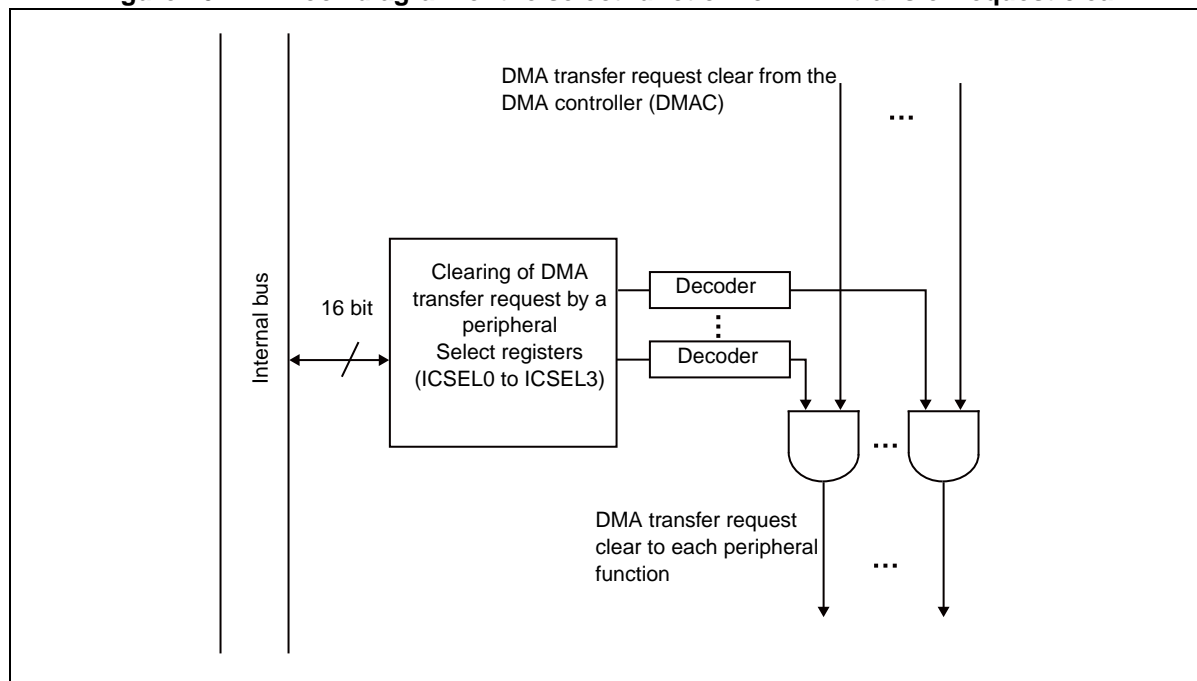
Figure 26.2-1 Block Diagram showing the Use of an Interrupt Request of Peripheral Functions



■ Block diagram of the select function for DMA transfer request clear

Figure 26.2-2 is a block diagram of the select function for DMA transfer request clear.

Figure 26.2-2 Block diagram of the select function for DMA transfer request clear



- Select Register for DMA transfer request clear by a peripheral function (ICSEL0 to ICSEL3)
This register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

26.3 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 26.3-1 is a list of the registers of the part at which a DMA transfer request is generated.

Table 26.3-1 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	26.3.1
1	IORR1	IO-data request register 1	26.3.1
2	IORR2	IO-data request register 2	26.3.1
3	IORR3	IO-data request register 3	26.3.1

■ List of registers of the select function for DMA transfer request clear

Table 26.3-2 shows a list of the registers of the select function for DMA transfer request clear.

Table 26.3-2 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	26.3.2
	ICSEL1	Select register 1 for DMA transfer request clear by a peripheral function	26.3.3
	ICSEL2	Select register 2 for DMA transfer request clear by a peripheral function	26.3.4
	ICSEL3	Select register 3 for DMA transfer request clear by a peripheral function	26.3.5

26.3.1 IO-Data Request Registers (IORR0 to IORR3)

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 26.3-1 shows the bit configuration of the IO-data request registers (IORR0 to IORR3).

Figure 26.3-1 Bit configuration of the IO-data request registers (IORR0 to IORR3)

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 is set as the IORR registers (RS3 to RS0 = 0001) in the RS3 to RS0 bit of the channel configuration registers (CCFR0 to CCFR3).

[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 ch.0
000001	17	11	External interrupt ch.1
000010	18	12	External interrupt ch.2
000011	19	13	External interrupt ch.3
000100	20	14	External interrupt ch.4
000101	21	15	External interrupt ch.5
000110	22	16	External interrupt ch.6
000111	23	17	External interrupt ch.7
001000	24	18	16-bit reload timer ch.0
001001	25	19	16-bit reload timer ch.1
001010	26	1A	16-bit reload timer ch.2
001011	27	1B	Multi-function serial interface ch.0 RX
001100	28	1C	Multi-function serial interface ch.0 TX
001101	29	1D	-
001110	30	1E	Multi-function serial interface ch.1 RX
001111	31	1F	Multi-function serial interface ch.1 TX
010000	32	20	-
010001	33	21	Multi-function serial interface ch.2 RX
010010	34	22	Multi-function serial interface ch.2 TX
010011	35	23	-

IOS5 to IOS0	Interrupt Vector Number		Peripheral Function
	Decimal	Hexadecimal	
010100	36	24	Multi-function serial interface ch.3 RX/TX
010101	37	25	Multi-function serial interface ch.4 RX/TX
010110	38	26	Multi-function serial interface ch.5 RX/TX
010111	39	27	A/D converter
011000	40	28	-
011001	41	29	External interrupt ch.8 to ch.15
011010	42	2A	External interrupt ch.16 to ch.23
011011	43	2B	Multi-function serial interface ch.6 RX/TX
011100	44	2C	Multi-function serial interface ch.7 RX/TX
011101	45	2D	Multi-function serial interface ch.8 RX/TX
011110	46	2E	Multi-function serial interface ch.9 RX/TX
011111	47	2F	Multi-function serial interface ch.10 RX/TX
100000	48	30	Multi-function serial interface ch.11 RX/TX
100001	49	31	Base timer ch.0 IRQ0/IRQ1
100010	50	32	Base timer ch.1 IRQ0/IRQ1
100011	51	33	Base timer ch.2 IRQ0/IRQ1
100100	52	34	Base timer ch.3 IRQ0/IRQ1
100101	53	35	Base timer ch.4 IRQ0/IRQ1
100110	54	36	Base timer ch.5 IRQ0/IRQ1
100111	55	37	Base timer ch.6 IRQ0/IRQ1
101000	56	38	Base timer ch.7 IRQ0/IRQ1
101001	57	39	Base timer ch.8/ch.9/ch.10/ch.11 IRQ0/IRQ1
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 ICSEL3).
- 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 \triangleleft ICR

26.3.2 Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0)

The external interrupt request ch.8 to ch.15 is assigned to interrupt vector number 41 (decimal). From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 26.3-2 shows the bit configuration of select register 0 for DMA transfer request clear by a peripheral function (ICSEL0).

Figure 26.3-2 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	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 flag bit to be cleared 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 bit selected here will be cleared.

EISEL12	EISEL11	EISEL10	Description
0	0	0	External interrupt ch.8
0	0	1	External interrupt ch.9
0	1	0	External interrupt ch.10
0	1	1	External interrupt ch.11
1	0	0	External interrupt ch.12
1	0	1	External interrupt ch.13
1	1	0	External interrupt ch.14
1	1	1	External interrupt ch.15

26.3.3 Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1)

The external interrupt request ch.16 to ch.23 is 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 26.3-3 shows the bit configuration of select register 1 for DMA transfer request clear by a peripheral function (ICSEL1).

Figure 26.3-3 The 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	EISEL22	EISEL21	EISEL20
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]: EISEL22 to EISEL20 (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 42 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 42 (decimal), the bit selected here will be cleared.

EISEL22	EISEL21	EISEL20	Description
0	0	0	External interrupt ch.16
0	0	1	External interrupt ch.17
0	1	0	External interrupt ch.18
0	1	1	External interrupt ch.19
1	0	0	External interrupt ch.20
1	0	1	External interrupt ch.21
1	1	0	External interrupt ch.22
1	1	1	External interrupt ch.23

26.3.4 Select Register 2 for DMA Transfer Request Clear by a Peripheral Function (ICSEL2)

The each interrupt request of base timer ch.0 to ch.7 is assigned to interrupt vector numbers 49 to 56 (decimal).

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 26.3-4 shows the bit configuration of select register 2 for DMA transfer request clear by a peripheral function (ICSEL2).

Figure 26.3-4 Bit Configuration of Select Register 2 for DMA Transfer Request Clear by a Peripheral Function (ICSEL2)

bit	7	6	5	4	3	2	1	0
	BTSEL07	BTSEL06	BTSEL05	BTSEL04	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 bit0]: BTSEL07 to BTSEL00 (Interrupt request selection bit)

From the interrupt requests assigned to interrupt vector numbers 49 to 56 (decimal), these bits select an interrupt request that clears the flag bit with the DMA controller (DMAC) from interrupt request 0 and interrupt request 1.

The interrupt requests assigned to interrupt request 0 and interrupt request 1 depend on the usage of the base timer.

Base Timer Usage	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

The flag bit of the interrupt request selected with this bit is cleared when the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector numbers 49 to 56 (decimal).

Bit Name	Interrupt Vector Number (Decimal)	Written Value	Description
BTSEL07	56	0	Base timer ch.7 interrupt request 0
		1	Base timer ch.7 interrupt request 1
BTSEL06	55	0	Base timer ch.6 interrupt request 0
		1	Base timer ch.6 interrupt request 1
BTSEL05	54	0	Base timer ch.5 interrupt request 0
		1	Base timer ch.5 interrupt request 1
BTSEL04	53	0	Base timer ch.4 interrupt request 0
		1	Base timer ch.4 interrupt request 1
BTSEL03	52	0	Base timer ch.3 interrupt request 0
		1	Base timer ch.3 interrupt request 1
BTSEL02	51	0	Base timer ch.2 interrupt request 0
		1	Base timer ch.2 interrupt request 1
BTSEL01	50	0	Base timer ch.1 interrupt request 0
		1	Base timer ch.1 interrupt request 1
BTSEL00	49	0	Base timer ch.0 interrupt request 0
		1	Base timer ch.0 interrupt request 1

26.3.5 Select Register 3 for DMA Transfer Request Clear by a Peripheral Function (ICSEL3)

The interrupt request of base timer ch.8 to ch.11 is assigned to interrupt vector number 57 (decimal). From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 26.3-5 shows the bit configuration of select register 3 for DMA transfer request clear by a peripheral function (ICSEL3).

Figure 26.3-5 The Bit Configuration of Select Register 3 for DMA Transfer Request Clear by a Peripheral Function (ICSEL3)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	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 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]: BTSEL12 to BTSEL10 (Interrupt request selection bits)

From the interrupt requests assigned to interrupt vector number 57 (decimal), these bits select an interrupt request that clears the flag bit with the DMA controller (DMAC) from interrupt request 0 and interrupt request 1.

The interrupt requests assigned to interrupt request 0 and interrupt request 1 depend on the usage of the base timer.

Base Timer Usage	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

The flag bit of the interrupt request selected with this bit is cleared when the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 57 (decimal).

BTSEL12	BTSEL11	BTSEL10	Description
0	0	0	Base timer ch.8 interrupt request 0
0	0	1	Base timer ch.8 interrupt request 1
0	1	0	Base timer ch.9 interrupt request 0
0	1	1	Base timer ch.9 interrupt request 1
1	0	0	Base timer ch.10 interrupt request 0
1	0	1	Base timer ch.10 interrupt request 1
1	1	0	Base timer ch.11 interrupt request 0
1	1	1	Base timer ch.11 interrupt request 1

26.4 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.

26.4.1 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 IORR3]).
 - Write values corresponding to the interrupt vector numbers to the bits ranging from IOS5 to IOS0.
 - 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 ICSEL3]).
3. Set the DMA controller (DMAC).

For details, see "CHAPTER 25 DMA Controller (DMAC)".

- Set a transfer request source of DMA transfer to an interrupt request of the peripheral functions.
 - 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.

- Clear the flag of an interrupt request to be used for a DMA transfer.
 - 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 "CHAPTER 12 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. DMA transfer is finished.

An interrupt request flag of the peripheral functions is cleared with the DMA controller (DMAC).

<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 \leq 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.

CHAPTER 27 Remote Control Reception

This chapter describes the functions and operations of HDMI-CEC receive/ACK auto response, and remote control reception.

27.1 Overview

27.2 Registers

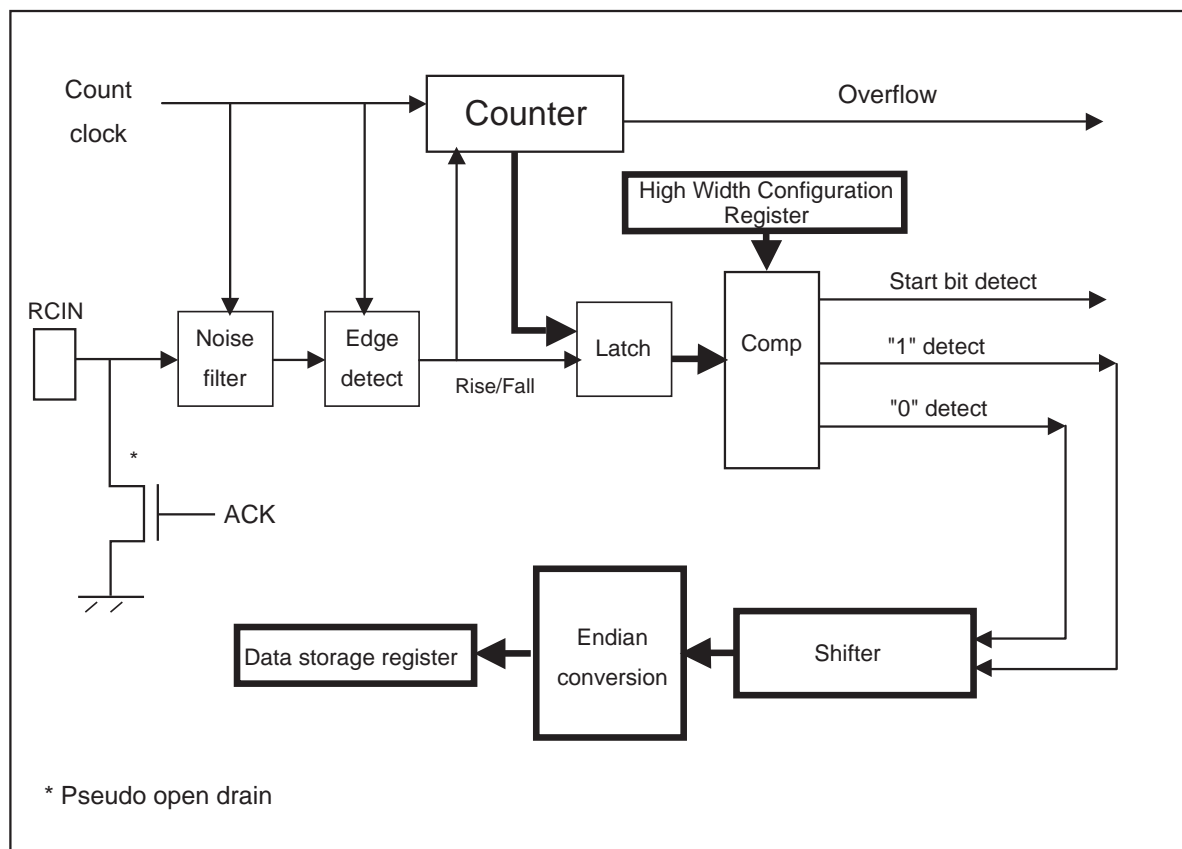
27.3 Explanation of Operations and Setting Procedure
Examples

27.1 Overview

This product is equipped with HDMI-CEC receive/ACK auto response, and remote control reception function.

■ Block diagram of Remote control reception

Figure 27.1-1 Block Diagram of Remote Control Reception



27.2 Registers

This section describes the configurations and functions of the registers used for remote control reception.

■ List of Registers

Table 27.2-1 shows a list of remote control reception registers.

Table 27.2-1 Registers of Remote Control Reception

Register Abbreviation	Register Name	See
RCCR	Remote control reception control register	27.2.1
RCST	Remote control reception interrupt control register	27.2.2
RCSHW	Start bit high width configuration register	27.2.4
RCDAHW	High width configuration register A	27.2.5
RCDBHW	High width configuration register B	27.2.6
RCADR1	Device address configuration register 1	27.2.3
RCADR2	Device address configuration register 2	27.2.3
RCDTHH	Data storage register HH	27.2.7
RCDTHL	Data storage register HL	27.2.7
RCDTLH	Data storage register LH	27.2.7
RCDTLL	Data storage register LL	27.2.7
RCCKD	Clock division register	27.2.8

27.2.1 Remote Control Reception Control Register (RCCR)

Figure 27.2-1 shows the bit configuration of the remote control reception control register (RCCR).

Figure 27.2-1 Bit Configuration of 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: Read/write allowed								

<Note>

This register allows only 8-bit access.

[bit 7]: THSEL

This is the threshold select bit.

Initial value is "0".

Set the criteria to check for "0" and "1" in the high width setting register A/B.

Status	THSEL	
	0	1
W > width A W < width B	Logic "0"	Logic "1"
W > width A W ≥ width B	Logic "1"	Logic "0"

[bit 6 to bit 4]: Reserved bits

Write	Ignored.
Read	The read value is "0".

[bit 3]: ADRCE

Enable address comparison bit.

The initial value is "0" (disable comparison). Comparison of reception address and device address is enabled when set to "1".

When comparison is enabled, an ACK/OVF interrupt occurs only when the addresses match.

In CEC mode, an ACK is returned when the address match is detected. In the case of broadcast address, match is assumed, but no ACK is returned.

Set "0" if the mode is other than SIRCS mode or HDMI-CEC mode.

[bit 2,bit 1]: MOD1,MOD0

These bits are used to set the remote control reception operation mode.

MOD1	MOD0	Function
0	0	SIRCS mode
0	1	Forbidden to set
1	0	NEC/AEHA mode (repeat signal not supported)
1	1	HDMI-CEC mode

Except for SIRCS mode (MOD1=1), the input signal is internally reversed.

High width comparison is applied to Low width.

[bit 0]: EN

This is the operation enable bit.

Remote control reception operation starts when this bit is set to "1".

Initial value is "0" (stopped).

Do not change the following registers and bits when this bit is "1" (operating).

RCCR register's THSEL bit, ADRCE bit, and MOD bit

RCST register's OVFSEL

RCSHW, RCDAHW, RCDBHW, RCADR1, RCADR2, and RCCKD register

27.2.2 Remote Control Reception Interrupt Control Register (RCST)

Figure 27.2-2 shows the bit configuration of the remote control reception interrupt control register (RCST).

Figure 27.2-2 Bit Configuration of 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: Read/write allowed								

<Note>

This register allows only 8-bit access.

[bit 7]: STIE

This bit enables start bit interrupt.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

[bit 6]: ACKIE

This bit enables ACK interrupt.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

It is valid only in CEC mode.

[bit 5]: OVFIE

This bit enables counter overflow interrupt.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

This interrupt occurs only when there is an overflow after a start bit is detected.
No interrupt occurs if start bit is undetected.

[bit 4]: OVFSEL

This bit is used to set the overflow detection condition.

Value	Description
0	Overflow occurs when the counter reaches 128 clock counts.
1	Overflow occurs when the counter reaches 256 clock counts.

[bit 3]: ST

This bit indicates the detection of start bit.

Value	Description
0	Start bit undetected
1	Start bit detected

Cleared when "0" is written.

An interrupt occurs when a start bit is detected while the STIE bit is "1".

[bit 2]: ACK

This bit indicates the detection of ACK.

Value	Description
0	ACK undetected
1	ACK detected

Cleared when "0" is written.

An interrupt occurs when an ACK is detected while the ACKIE bit is "1".

When address comparison is enabled, an interrupt occurs only when the addresses match.

It is valid only in CEC mode.

[bit 1]: EOM

This bit indicates the detection of EOM.

Value	Description
0	EOM undetected
1	EOM detected

Cleared when "0" is written.

It is valid only in CEC mode.

[bit 0]: OVF

This bit indicates the detection of counter overflow.

Value	Description
0	Counter overflow undetected
1	Counter overflow detected

When address comparison is enabled, an interrupt occurs only when the addresses match.

Cleared when "0" is written.

In SIRCS mode, the OVF flag is not set until the second byte is received.

27.2.3 Device Address Configuration Registers 1, 2
(RCADR1, RCADR2)

Figure 27.2-3 shows the bit configuration of the device address configuration registers 1, 2 (RCADR1, RCADR2).

Figure 27.2-3 Bit Configuration of Device Address Configuration Registers 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: Read/write allowed								

<Note>

This register allows only 8-bit access.

[bit 7 to bit 5]: Reserved bits

Write	Ignored.
Read	The read value is "0".

[bit 4 to bit 0]: RCADR1, 2

This is the register to set the device side (reception side) address.
The address set in this register is compared with the remote control reception device address and HDMI-CEC destination.
Do not set 0F_H (broadcast address) in this register in HDMI-CEC mode.

27.2.4 Start Bit High Width Configuration Register (RCSHW)

Figure 27.2-4 shows the bit configuration of the start bit high width configuration register (RCSHW).

Figure 27.2-4 Bit Configuration of Start Bit High Width Configuration Register (RCSHW)

bit	7	0
	RCSHW	
Attribute	R/W	
Initial value	0	
R/W: Read/write allowed		

<Note>

This register allows only 8-bit access.

This register is used to set 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, no start bit is detected and start bit High detection wait state is reentered.

Set RCSHW ≤ 127 (value not exceeding overflow detection) if OVFSSEL=0.

27.2.5 High Width Configuration Register A(RCDAHW)

Figure 27.2-5 shows the bit configuration of the High Width Configuration register A (RCDAHW).

Figure 27.2-5 Bit Configuration of High Width Configuration A (RCDAHW)

bit	7	0
	RCDAHW	
Attribute	R/W	
Initial value	0	
R/W: Read/write allowed		

<Note>

This register allows only 8-bit access.

This is register A used to set the High duration.

The value set in this register must be such that $2 \leq \text{RCDAHW} < \text{RCDBHW}$.

Also, in CEC mode, set so that $\text{RCDAHW} < 46$ (less than ACK response pulse width).

27.2.6 High Width Configuration Register B(RCDAHW)

Figure 27.2-6 shows the bit configuration of the High Width Configuration register B (RCDBHW).

Figure 27.2-6 Bit Configuration of High Width Configuration Register B (RCDBHW)

bit	7	0
	RCDBHW	
Attribute	R/W	
Initial value	0	
R/W: Read/write allowed		

<Note>

This register allows only 8-bit access.

This is register B used to set the High duration.
Do not set value less than RCCDAHW.
Make sure $RCCDAHW < RCCDBHW < RCSHW$ is satisfied.

27.2.7 Data Storage Registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL)

Figure 27.2-7 shows the bit configuration of the data storage registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL).

Figure 27.2-7 Bit Configuration of Data Storage Registers (RCDTHH, RCDTHL, RCDTLH, RCDTLL)

	bit	31	24	23	16	15	8	7	0		
		RCDTHH				RCDTHL		RCDTLH		RCDTLL	
Attribute						R					
Initial value						0					
R/W: Read/write allowed											

These are the registers to store the received data.
In CEC mode, the received data is stored in RCDTHH.
In remote control mode, data is stored starting from RCDTHH as each 8 bits are received.
When a counter overflow interrupt occurs, the bits received up to that point are stored MSB justified.
When the EN bit of RCCR register is "0", the value read from these registers is undefined.
If a signal exceeding 4 bytes is received, the excess data is ignored and not reflected in the registers.

27.2.8 Clock Division Configuration Register (RCCKD)

Figure 27.2-8 shows the bit configuration of the clock division configuration register (RCCKD).

Figure 27.2-8 Bit Configuration of 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: Read/write allowed					

<Note>

This register allows only 16-bit access.

[bit 15 to bit 13]: Reserved bits

Write	Ignored.
Read	The read value is "0".

[bit 12]: CKSEL

This bit selects the operating clock.

Value	Description
0	Clock dividing the peripheral clock is selected.
1	Clock dividing the source oscillation clock is selected.

[bit 11 to bit 0]: CKDIV

These bits are used to set the clock division rate.

Division rate is CKDIV+1.

From 1 division (no division) to 4096 divisions can be set.

Set so that the frequency after division is close to 32.768KHz.

Setting examples are shown below.

Operating Clock Frequency	CKDIV value	Frequency after Division
16MHz (source oscillation)	487	32.787KHz
32MHz (peripheral clock)	976	32.753KHz
40MHz (peripheral clock)	1220	32.760KHz

27.3 Explanation of Operations and Setting Procedure Examples

This section describes the remote control reception operation. A setting examples for each operation status are also shown.

■ Operation flowchart

Figure 27.3-1 Flowchart of HDMI-CEC Receive Operation

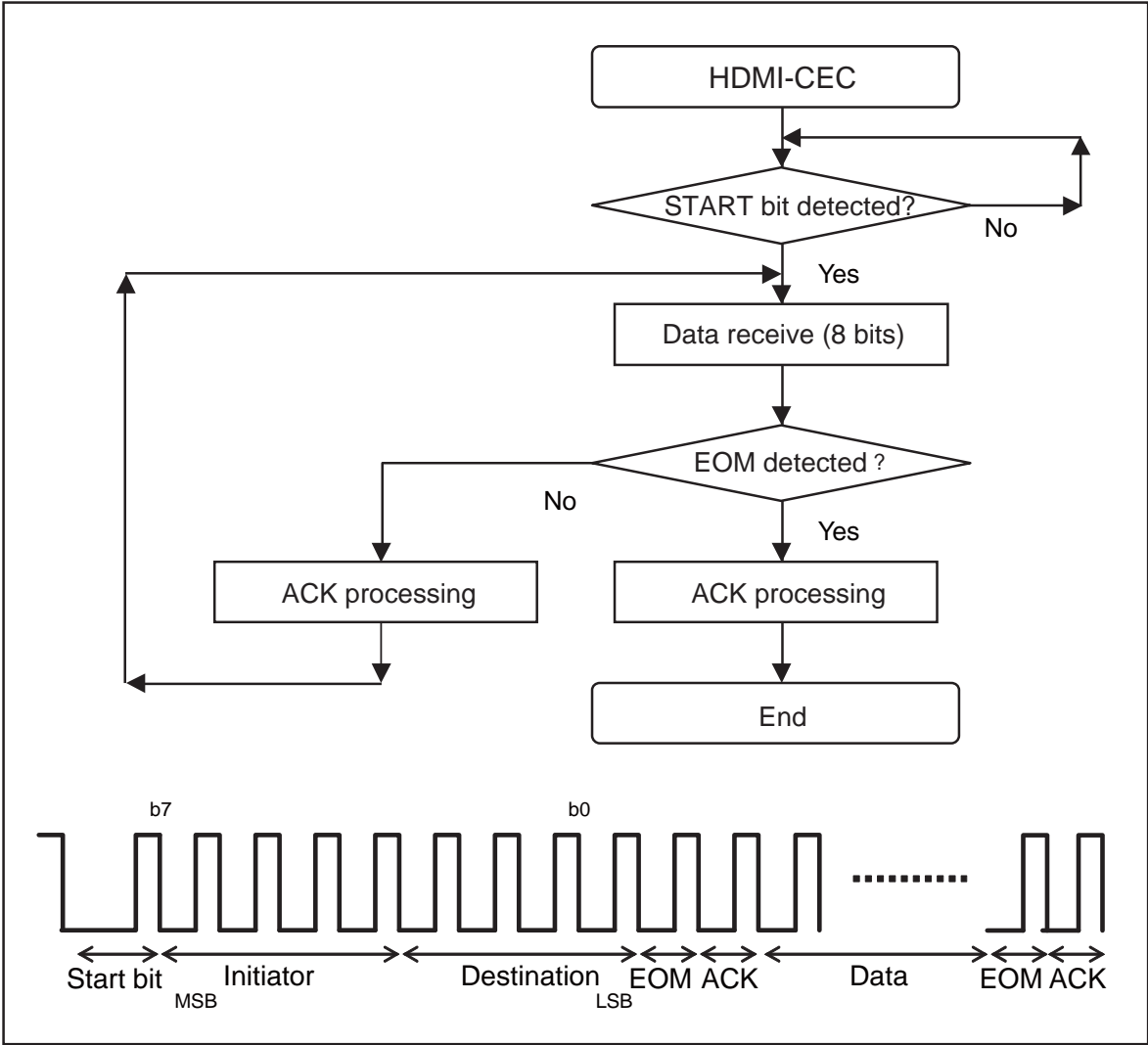
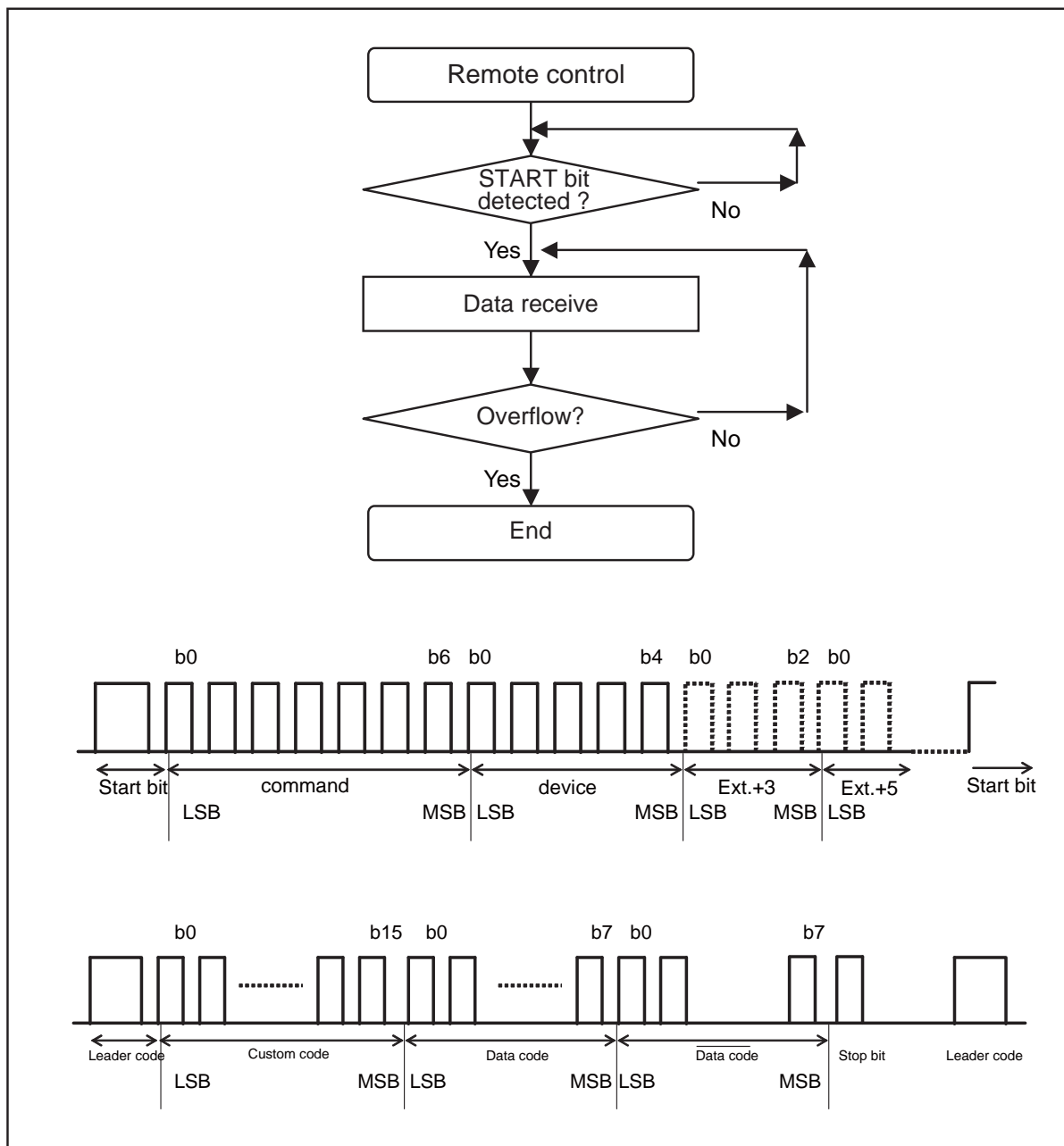


Figure 27.3-2 Flowchart of Remote Control Reception Operation



■ **Setting Example**

HDMI-CEC

Registers	Settings	
Remote control reception control register	MOD=11,THSEL=1,ADRCE=1	
Remote control reception interrupt control register	ACKIE=1,OVFSEL=1,OVFIE=1	(7.8ms)
Start bit high width configuration register	114	3.5ms
High width configuration register A	13	0.4ms
High width configuration register B	42	1.3ms

Remote Control (SIRCS)

Registers	Settings	
Remote control reception control register	MOD=00,THSEL=0,ADRCE=1	
Remote control reception interrupt control register	ACKIE=0,OVFSEL=0,OVFIE=1	3.9ms
Start bit high width configuration register	76	2.3ms
High width configuration register A	17	0.52ms
High width configuration register B	37	1.1ms

Remote Control (NEC)

Registers	Settings	
Remote control reception control register	MOD=10,THSEL=0	
Remote control reception interrupt control register	ACKIE=0,OVFSEL=1,OVFIE=1	7.8ms
Start bit high width configuration register	144	4.4ms
High width configuration register A	15	0.46ms
High width configuration register B	52	1.6ms

CHAPTER 28 Serial Programming Connection Example

The MB91605A supports serial onboard write to flash memory connected through an external bus. This chapter describes its specification.

28.1 Pins Used

28.2 Serial Programming Connection Example

28.1 Pins Used

This section describes the pins used for serial onboard write.

Table 28.1-1 Pins Used for Serial Onboard Write

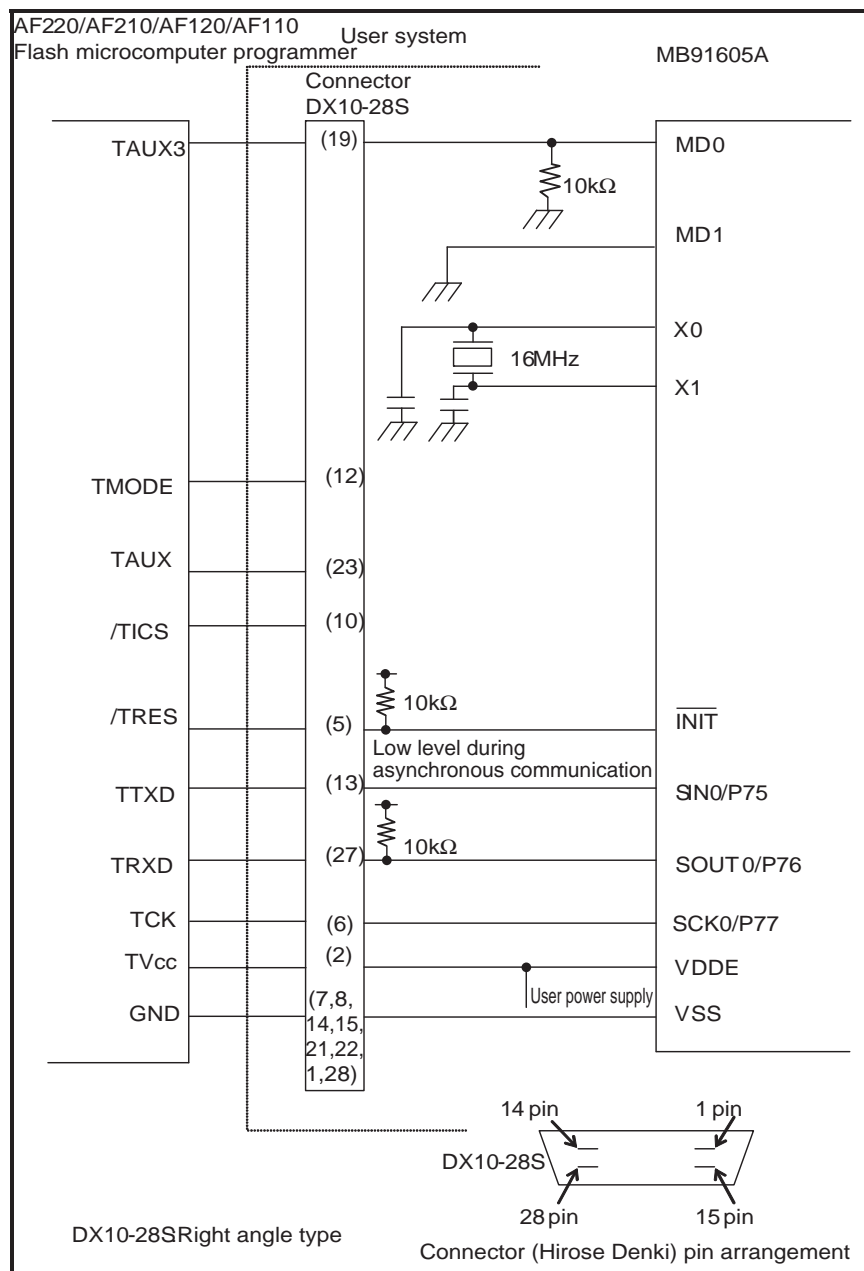
Pins	Function	Supplement
MD1, MD0	Mod pin	MB91605A enters serial write mode when it is reset with SOUT0=1 after setting MD1=0 and MD0=1. When attaching a pull-up/pull-down resistor, short wiring is recommended.
X0,X1	Oscillation pin	The source frequency clock available in serial write mode is 16MHz.
SOUT0/P76	Serial write mode activation pin/UART serial data output pin	This pin becomes the serial write program activation pin when the level at reset is set to "1" with external pull-up resistor added. If communication mode is set to UART, this pin becomes the serial data output pin when program is activated and communication is started.
SIN0/P75	UART synchronous/asynchronous select pin/UART serial data input pin	Setting the input level of this pin to "1" until the start of communication enables the asynchronous UART communication mode and setting it to "0" enables the synchronous UART communication mode. This pin is used as UART serial data input pin when program is activated and communication starts.
SCK0/P77	Serial clock input pin	This pin becomes the serial clock input pin when the communication mode is set to UART synchronous communication.
$\overline{\text{INIT}}$	Reset pin	-
VDDE	Supply voltage pin	Supply the right voltage from the user system. When connecting, do not short-circuit with the user side power supply.
VSS	GND pin	Shared with flash microcomputer programmer GND.

28.2 Serial Programming Connection Example

This section shows the serial programming connection example.

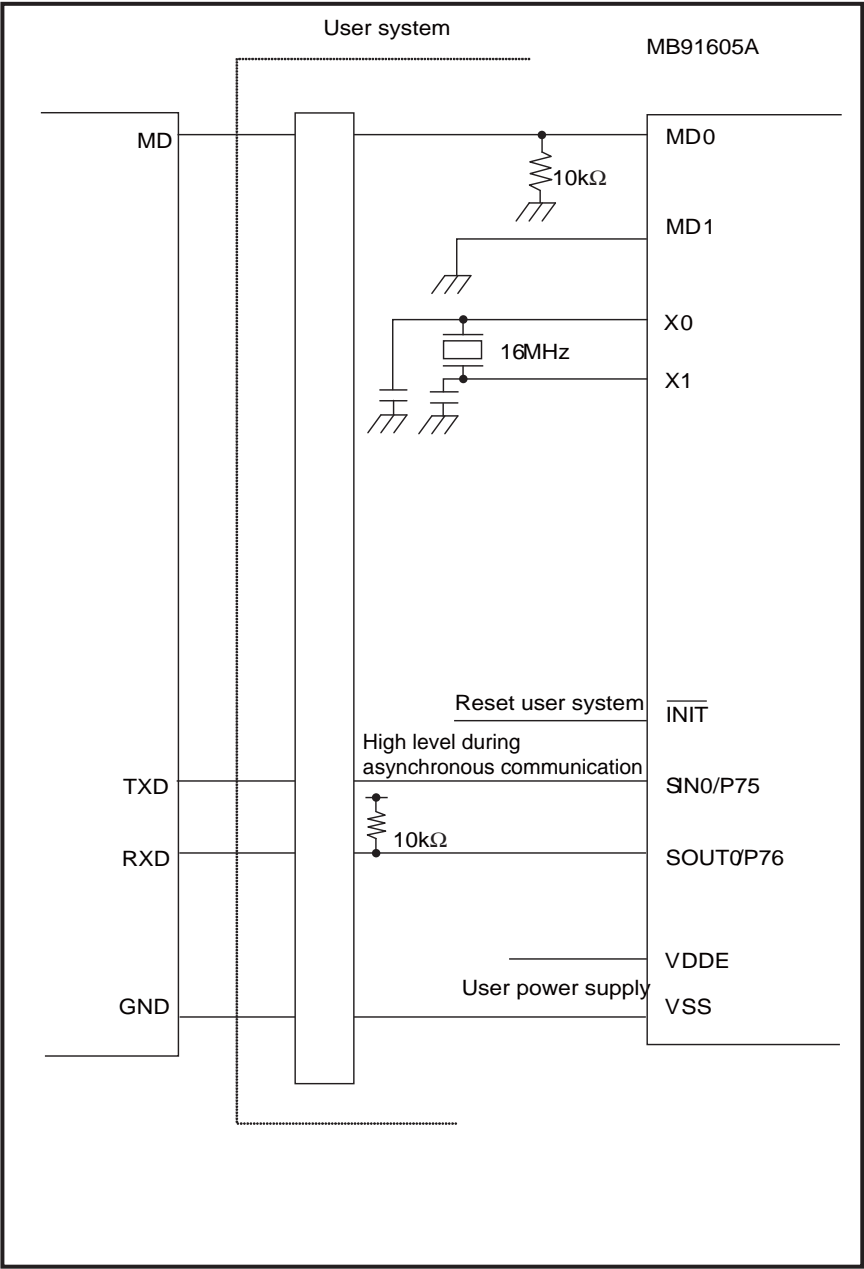
28.2.1 Synchronous Serial Programming Connection Example

Figure 28.2-1 Example of Serial Programming Connection (Synchronous Communication Mode)



28.2.2 Asynchronous Serial Programming Connection Example

Figure 28.2-2 Example of Serial Programming Connection (Asynchronous Communication Mode)



CHAPTER 29 Handling the Device

This chapter provides notes on using this series.

29.1 Notes on Handling the Device

29.1 Notes on Handling the Device

Note the following points on using this series.

■ HANDLING DEVICES

● Preventing latch-up

Latch-up may occur in a CMOS IC if a voltage higher than VDDE or VDDI, or less than VSS is applied to an input or output pin or if a voltage exceeding the rated value is applied between VDDE and VSS, or VDDI and VSS. If latch-up occurs, the power supply current increases rapidly, sometimes resulting in thermal breakdown of the device. Therefore, be very careful not to apply voltages in excess of the absolute maximum ratings.

● Handling of unused pins

Leaving unused input pins unconnected can cause misoperation or permanent damage to the device due to latch-up. Always pull-up or pull-down unused pins using a 2 k Ω or larger resistor.

If some I/O pins are unused, either set as outputs and leave open circuit or set as inputs and treat in the same way as input pins.

● Power supply pins

The MB91605A series has multiple VDDE, VDDI, and VSS pins. respective pins at the same potential are interconnected in order to prevent latch-up and other malfunctions. 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 VDDE pins, VDDI pins and VSS pins of the MB91605A series must be connected to the current supply source at a low impedance.

It is also recommended to connect a ceramic capacitor of approximately 0.1 mF as a bypass capacitor between the VDDE, VDDI and VSS pins near the device.

● Crystal oscillator circuit

Noise in proximity to the X0 and X1 pins can cause the device to malfunction. Printed circuit boards should be designed so that the X0 and X1 pins, crystal oscillator, and bypass capacitors connected to ground are located near the device and ground.

It is recommended that the printed circuit board artwork be designed such that the X0 and X1 pins are surrounded by ground plane for the stable operation.

Please request the oscillator manufacturer to evaluate the oscillational characteristics of the crystal and this device.

● Mode pins (MD0, MD1)

Connect them directly to power supply pins or GND pins. To prevent the device from entering test mode accidentally due to noise, minimize the lengths of the patterns between each mode pin and power supply pin or GND pin on the printed circuit board as much as possible and connect them at a low impedance.

● Operation at power-on

Ensure that a settings initialization reset (INIT) is performed using the $\overline{\text{INIT}}$ pin immediately after the power is turned on.

Maintain the "L" level input to the $\overline{\text{INIT}}$ pin for the duration of the stabilization wait time immediately after the power on to ensure the stabilization wait time as required by the oscillator circuit (the stabilization wait time is reset to the minimum value when INIT is asserted using the $\overline{\text{INIT}}$ pin).

● Note on oscillator input at power-on

At power-on, ensure that the clock is input until the oscillator stabilization wait time has elapsed.

● Notes on the turning on and off the power to the VDDI pin (1.8 V internal power supply) and VDDE pin (3.3 V external pin power supply).

- Turn on/off the power in the following procedure.

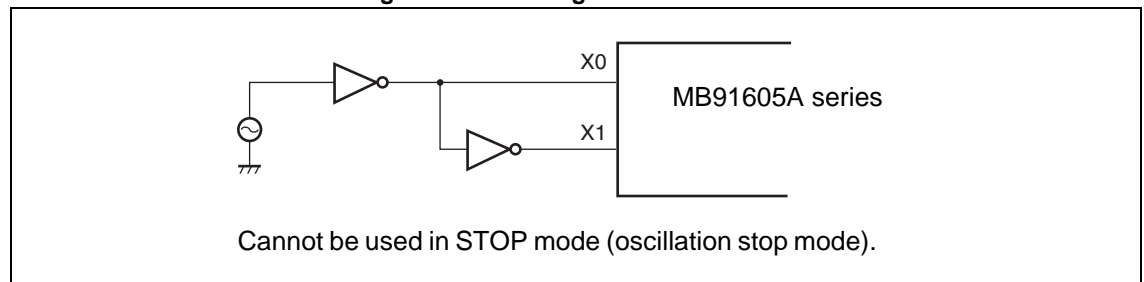
Power on	VDDI pin (internal power supply) → VDDE pin (external power supply) → Analog → Signal
Power off	Signal → Analog → VDDE pin (external power supply) → VDDI pin (internal power supply)

- Do not continuously (more than one minute) apply power to the VDDE pin (external power supply) while the VDDI pin (internal power supply) is disconnected as this will adversely affect the reliability of the LSI.
- When the VDDE pin (external power supply) returns from the off state to the on state, the internal state of the circuit might not be able to be maintained due to power supply noise and other effects.
- When the power is turned on, the states of the output pins may remain undefined until the internal power supply becomes stable.
- There is no problem for turning on and off the power (VDDI/VDDE/analog) simultaneously.

● Notes on using an external clock

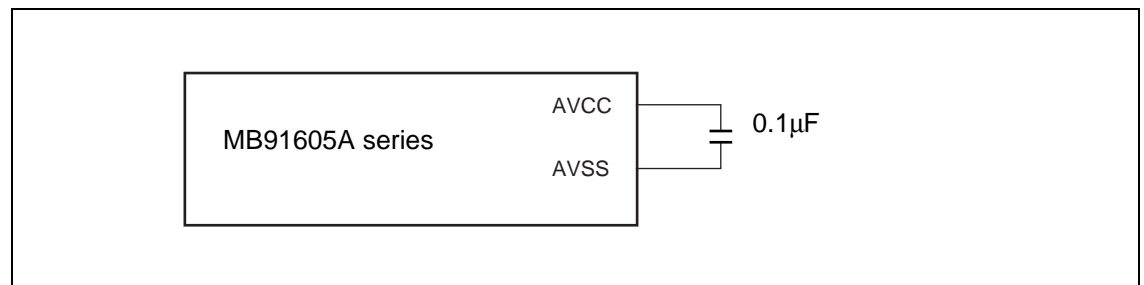
When using an external clock, in principal, the clock signal should be supplied simultaneously to the X0 and X1 pins, with the phase-inverted clock signal of X0 supplied to the X1 pin. However, the external clock must not be used while the microcontroller is in stop mode (oscillator stop mode). (This is because the X1 pin stops at "H" output in STOP mode.)

Figure 29.1-1 Using an External Clock



● AVCC pin

The MB91605A series has a built-in A/D converter. A capacitor of approximately 0.1mF must be connected between the AVCC pin and AVSS pin.



● 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 29.1-1 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

● Precautions when the PLL clock is selected

If the crystal oscillator is disconnected or the clock input stops while the PLL clock is selected, the microcontroller may continue to operate at the free-running frequency of the self-oscillator within the PLL. However, this operation is not guaranteed.

■ RESTRICTIONS

1. Notes on the PS register

Some instructions write to the PS register in advance before executing. When a debugger is being used, execution may break within an interrupt handler routine, or the values of the flags within the PS register may be updated due to exception processing. However, the microcontroller is designed to reprocess correctly after returning from the EIT, and to execute before and after the EIT proceeds according to the specifications.

If any of the following situations occur in the instruction immediately before a DIV0U or DIV0S instruction, the processing in (1) to (3) will be performed.

- A user interrupt or NMI is accepted
- Step execution is performed
- A break occurs due to a data event or by being selected from the emulator menu

1. The D0 and D1 flags are updated in advance.
2. The EIT handling routine (user interrupt/NMI or emulator) is executed.
3. Upon returning from the EIT, the DIV0U or DIV0S instruction is executed and the D0/D1 flags are updated back to the same value as in step (1).

If any of the OR CCR, ST ILM, or MOV Ri, PS instructions are executed to enable a user interrupt or NMI interrupt source when that interrupt has occurred, the following operation will be performed

1. The PS register is updated in advance.
2. The EIT handling routine (user interrupt/NMI or emulator) is executed.
3. Upon returning from the EIT, the above instructions are executed and the PS registers are updated

2. Watchdog timer

The watchdog timer is a function that monitors the program to check that it delays a reset within a certain period of time, and resets the CPU if the program runs out of control and fails to delay the reset. Once the watchdog timer has been enabled, it keeps running until reset. As an exception, the reset is automatically delayed in conditions where the execution of the CPU program stops. It is possible that the watchdog timer will not be triggered if these conditions arise as a result of the system running out of control. In that case, please reset (INIT) using the external INIT pin.

3. Notes on debugger

- Step execution of RETI instruction

If stepped execution is used in an environment where interrupts occur frequently, the interrupt processing routines corresponding to those interrupts are executed repeatedly, and the programs for the main routine and low-level interrupt routines are not able to execute as a result. (For example, if a reload timer is enabled, execution will always break at the beginning of the reload timer interrupt routine when the RETI instruction is step-executed.) Disable the corresponding interrupts when the interrupt processing routines no longer need to be debugged.

- Break function

If the target address of a hardware break (including event breaks) is set to the address of the current system stack pointer or to an area that contains the stack pointer, execution will break after each instruction regardless of whether the user program actually contains data access instructions.

To prevent this, do not set (word) access to the area containing the address of the system stack pointer as the target of a hardware break (including an event break).

- Operand break

Malfunctions may occur if the stack pointer is within an area which is set as the DSU operand break. Do not set the access to the areas containing the address of system stack pointer as a target of data event break.

APPENDIXES

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.

APPENDIX A I/O Map

APPENDIX B List of Registers

APPENDIX C Interrupt Vectors

APPENDIX D Pin State in Each CPU State

APPENDIX E Lists of Instructions

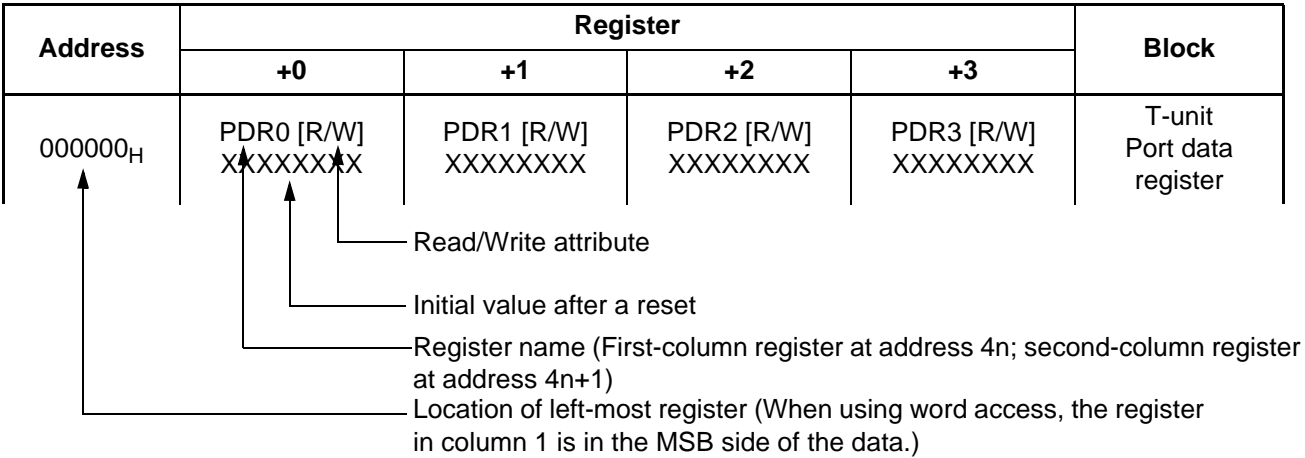
APPENDIX F Program Loader Mode

APPENDIX A 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 How to read the table



<Notes>

- The bit values in the register represent the following initial values :
 - "1": Initial value "1"
 - "0": Initial value "0"
 - "X": Initial value "Undefined"
 - "-": No physical register at this location
 - "**": Uninitialized bit
- Read/write attribute is as follows.
 - "R": Indicates that there is a read only bit.
 - "R/W": Indicates that there is a read/write bit.
 - "W": Indicates that there is a write only bit.
- Access is prohibited for data access attributes that are not listed.

MB91605A Series**Table A-1 I/O map (1 / 15)**

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000000 _H	Reserved		PDR2 [R/W] ----XXXX	PDR3 [R/W] XXXXXXXXXX	Port data register
000004 _H	PDR4 [R/W] --XXXXXX	PDR5 [R/W] XXXXXXXXXX	PDR6 [R/W] XXXXXXXXXX	PDR7[R/W] XXXXXXXXXX	
000008 _H	PDR8 [R/W] --XXXXXX	PDR9 [R/W] --XXXXXX	PDRA [R/W] --XXXXXX	PDRB[R/W] XXXXXXXXXX	
00000C _H	PDRC [R/W] XXXXXXXXXX	PDRD [R/W] XXXXXXXXXX	PDRE [R/W] XXXXXXXXXX	Reserved	
000010 _H to 00001C _H	Reserved				
000020 _H	ADCTH [R/W] 00000000	ADCTL [R/W] 00000000	ADCH [R/W] 00000000 00000000		10-bit A/D converter
000024 _H	ADAT0 [R] 00000000 00000000		ADAT1 [R] 00000000 00000000		
000028 _H	ADAT2 [R] 00000000 00000000		ADAT3 [R] 00000000 00000000		
00002C _H	ADAT4 [R] 00000000 00000000		ADAT5 [R] 00000000 00000000		
000030 _H	ADAT6 [R] 00000000 00000000		ADAT7 [R] 00000000 00000000		
000034 _H	ADAT8 [R] 00000000 00000000		ADAT9 [R] 00000000 00000000		
000038 _H	ADAT10 [R] 00000000 00000000		ADAT11 [R] 00000000 00000000		
00003C _H	WDTCR0 [R/W] 00000000	WDTCPR0 [R/W] 00000000	Reserved		
000040 _H	EIRR0 [R/W] 00000000	ENIR0 [R/W] 00000000	ELVR0 [R/W] 00000000 00000000		External interrupt 0 to 7
000044 _H	DICR [R/W] 11111110	Reserved			Delay interrupt
000048 _H	TMRLRA0 [R/W] XXXXXXXXXX XXXXXXXXXX		TMR0 [R] XXXXXXXXXX XXXXXXXXXX		16-bit reload timer ch.0
00004C _H	Reserved		TMCSR0 [R/W] 00000000 XX000000		

Table A-1 I/O map (2 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000050 _H	TMRLRA1 [R/W] XXXXXXXXXX XXXXXXXXXX		TMR1 [R] XXXXXXXXXX XXXXXXXXXX		16-bit reload timer ch.1
000054 _H	Reserved		TMCSR1 [R/W] 00000000 XX000000		
000058 _H	TMRLRA2 [R/W] XXXXXXXXXX XXXXXXXXXX		TMR2 [R] XXXXXXXXXX XXXXXXXXXX		16-bit reload timer ch.2
00005C _H	Reserved		TMCSR2 [R/W] 00000000 XX000000		
000060 _H	SCR0/IBCR0 [R, R/W] 0--00000	SMR0 [R/W] 00000000	SSR0 [R, R/W] 0-000011	ESCR0 [R/W], IBSR0 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.0
000064 _H	RDR0[R]/TDR0[W] -----0 00000000		BGR10 [R/W] 00000000	BGR00 [R/W] 00000000	
000068 _H	ISMK0 [R/W] 01111111	ISBA0 [R/W] 00000000	Reserved		
00006C _H	FCR10 [R/W] 00-00100	FCR00 [R, R/W] 00000000	FBYTE20 [R/W] 00000000	FBYTE10 [R/W] 00000000	
000070 _H	SCR1/IBCR1 [R, R/W] 0--00000	SMR1 [R/W] 00000000	SSR1 [R, R/W] 0-000011	ESCR1 [R/W], IBSR1 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.1
000074 _H	RDR1[R]/TDR1[W] -----0 00000000		BGR11 [R/W] 00000000	BGR01 [R/W] 00000000	
000078 _H	ISMK1 [R/W] 01111111	ISBA1 [R/W] 00000000	Reserved		
00007C _H	FCR11 [R/W] 00-00100	FCR01 [R, R/W] 00000000	FBYTE21 [R/W] 00000000	FBYTE11 [R/W] 00000000	
000080 _H	SCR2/IBCR2 [R, R/W] 0--00000	SMR2 [R/W] 00000000	SSR2 [R, R/W] 0-000011	ESCR2 [R/W], IBSR2 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.2
000084 _H	RDR2[R]/TDR2[W] -----0 00000000		BGR12 [R/W] 00000000	BGR02 [R/W] 00000000	
000088 _H	ISMK2 [R/W] 01111111	ISBA2 [R/W] 00000000	Reserved		
00008C _H	FCR12 [R/W] 00-00100	FCR02 [R, R/W] 00000000	FBYTE22 [R/W] 00000000	FBYTE12 [R/W] 00000000	

MB91605A Series**Table A-1 I/O map (3 / 15)**

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000090 _H	SCR3/IBCR3 [R, R/W] 0--00000	SMR3 [R/W] 00000000	SSR3 [R, R/W] 0-000011	ESCR3 [R/W], IBSR3 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.3
000094 _H	RDR3[R]/TDR3[W] -----0 00000000		BGR13 [R/W] 00000000	BGR03 [R/W] 00000000	
000098 _H	ISMK3 [R/W] 01111111	ISBA3 [R/W] 00000000	Reserved		
00009C _H	FCR13 [R/W] 00-00100	FCR03 [R, R/W] 00000000	FBYTE23 [R/W] 00000000	FBYTE13 [R/W] 00000000	
0000A0 _H	SCR4/IBCR4 [R, R/W] 0--00000	SMR4 [R/W] 00000000	SSR4 [R, R/W] 0-000011	ESCR4 [R/W], IBSR4 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.4
0000A4 _H	RDR4[R]/TDR4[W] -----0 00000000		BGR14 [R/W] 00000000	BGR04 [R/W] 00000000	
0000A8 _H	ISMK4 [R/W] 01111111	ISBA4 [R/W] 00000000	Reserved		
0000AC _H	FCR14 [R/W] 00-00100	FCR04 [R, R/W] 00000000	FBYTE24 [R/W] 00000000	FBYTE14 [R/W] 00000000	
0000B0 _H	SCR5/IBCR5 [R, R/W] 0--00000	SMR5 [R/W] 00000000	SSR5 [R, R/W] 0-000011	ESCR5 [R/W], IBSR5 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.5
0000B4 _H	RDR5[R]/TDR5[W] -----0 00000000		BGR15 [R/W] 00000000	BGR05 [R/W] 00000000	
0000B8 _H	ISMK5 [R/W] 01111111	ISBA5 [R/W] 00000000	Reserved		
0000BC _H	FCR15 [R/W] 00-00100	FCR05 [R, R/W] 00000000	FBYTE25 [R/W] 00000000	FBYTE15 [R/W] 00000000	
0000C0 _H	SCR6/IBCR6 [R, R/W] 0--00000	SMR6 [R/W] 00000000	SSR6 [R, R/W] 0-000011	ESCR6 [R/W], IBSR6 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.6
0000C4 _H	RDR6[R]/TDR6[W] -----0 00000000		BGR16 [R/W] 00000000	BGR06 [R/W] 00000000	
0000C8 _H	ISMK6 [R/W] 01111111	ISBA6 [R/W] 00000000	Reserved		
0000CC _H	FCR16 [R/W] 00-00100	FCR06 [R, R/W] 00000000	FBYTE26 [R/W] 00000000	FBYTE16 [R/W] 00000000	

Table A-1 I/O map (4 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
0000D0 _H	SCR7/IBCR7 [R, R/W] 0--00000	SMR7 [R/W] 00000000	SSR7 [R, R/W] 0-000011	ESCR7 [R/W], IBSR7 [R,R/W] -0000000	Multi-function serial interface (with FIFO) ch.7
0000D4 _H	RDR7[R]/TDR7[W] -----0 00000000		BGR17 [R/W] 00000000	BGR07 [R/W] 00000000	
0000D8 _H	ISMK7 [R/W] 01111111	ISBA7 [R/W] 00000000	Reserved		
0000DC _H	FCR17 [R/W] 00-00100	FCR07 [R, R/W] 00000000	FBYTE27 [R/W] 00000000	FBYTE17 [R/W] 00000000	
0000E0 _H	EIRR1 [R/W] 00000000	ENIR1 [R/W] 00000000	ELVR1 [R/W] 00000000 00000000		External interrupt 8 to 15
0000E4 _H	EIRR2 [R/W] 00000000	ENIR2 [R/W] 00000000	ELVR2 [R/W] 00000000 00000000		External interrupt 16 to 23
0000E8 _H	IRPROH [R] 00000000	Reserved			Interrupt request batch read function
0000EC _H	Reserved				Reserved
0000F0 _H	RCCR [R/W] 0---0000	RCST [R/W] 00000000	RCSHW [R/W] 00000000	RCDAHW [R/W] 00000000	Remote control
0000F4 _H	RCDBHW [R/W] 00000000	Reserved	RCADR1 [R/W] ---00000	RCADR2 [R/W] ---00000	
0000F8 _H	RCDTHH [R] 00000000	RCDTHL [R] 00000000	RCDTLH [R] 00000000	RCDTLL[R] 00000000	
0000FC _H	RCCKD [R/W] ---00000 00000000		Reserved		
000100 _H to 0001FC _H	Reserved				Reserved
000200 _H	SCR8/IBCR8 [R, R/W] 0--00000	SMR8 [R/W] 00000000	SSR8 [R, R/W] 0-000011	ESCR8 [R/W], IBSR8 [R,R/W] -0000000	Multi-function serial interface (without FIFO) ch.8
000204 _H	RDR8[R]/TDR8[W] -----0 00000000		BGR18 [R/W] 00000000	BGR08 [R/W] 00000000	
000208 _H	ISMK8 [R/W] 01111111	ISBA8 [R/W] 00000000	Reserved		
00020C _H	Reserved				

MB91605A Series**Table A-1 I/O map (5 / 15)**

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000210 _H	SCR9/IBCR9 [R, R/W] 0--00000	SMR9 [R/W] 00000000	SSR9 [R, R/W] 0-000011	ESCR9 [R/W], IBSR9 [R,R/W] -0000000	Multi-function serial interface (without FIFO) ch.9
000214 _H	RDR9[R]/TDR9[W] -----0 00000000		BGR19 [R/W] 00000000	BGR09 [R/W] 00000000	
000218 _H	ISMK9 [R/W] 01111111	ISBA9 [R/W] 00000000	Reserved		
00021C _H	Reserved				
000220 _H	SCRA/IBCRA [R, R/W] 0--00000	SMRA [R/W] 00000000	SSRA [R, R/W] 0-000011	ESCRA [R/W], IBSRA [R,R/W] -0000000	Multi-function serial interface (without FIFO) ch.10
000224 _H	RDRA[R]/TDRA[W] -----0 00000000		BGR1A [R/W] 00000000	BGR0A [R/W] 00000000	
000228 _H	ISMKA [R/W] 01111111	ISBAA [R/W] 00000000	Reserved		
00022C _H	Reserved				
000230 _H	SCRB/IBCRB [R, R/W] 0--00000	SMRB [R/W] 00000000	SSRB [R, R/W] 0-000011	ESCRB [R/W], IBSRB [R,R/W] -0000000	Multi-function serial interface (without FIFO) ch.11
000234 _H	RDRB[R]/TDRB[W] -----0 00000000		BGR1B [R/W] 00000000	BGR0B [R/W] 00000000	
000238 _H	ISMKB [R/W] 01111111	ISBAB [R/W] 00000000	Reserved		
00023C _H	Reserved				
000240 _H	RDRM8/TDRM8 [R/W] 00000000	RDRM9/TDRM9 [R/W] 00000000	RDRMA/TDRMA [R/W] 00000000	RDRMB/TDRMB [R/W] 00000000	Multi-function serial interface data register (mirror)
000244 _H	SSEL89AB [R/W] -----00	Reserved			Multi-function serial interface serial clock selection
000248 _H to 00027C _H	Reserved				Reserved

Table A-1 I/O map (6 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000280 _H	BT0TMR [R] 00000000 00000000		BT0TMCR [R/W] 00000000 00000000		Base timer ch.0
000284 _H	Reserved	BT0STC [R/W] 00000000	Reserved		
000288 _H	BT0PCSR/BT0PRLL [R/W] XXXXXXXX XXXXXXXX		BT0PDUT/BT0PRLH [R/W], BT0DTBF [R] XXXXXXXX XXXXXXXX		
00028C _H	Reserved				
000290 _H	BT1TMR [R] 00000000 00000000		BT1TMCR [R/W] 00000000 00000000		Base timer ch.1
000294 _H	Reserved	BT1STC [R/W] 00000000	Reserved		
000298 _H	BT1PCSR/BT1PRLL [R/W] XXXXXXXX XXXXXXXX		BT1PDUT/BT1PRLH [R/W], BT1DTBF [R] XXXXXXXX XXXXXXXX		
00029C _H	Reserved				
0002A0 _H	BT2TMR [R] 00000000 00000000		BT2TMCR [R/W] 00000000 00000000		Base timer ch.2
0002A4 _H	Reserved	BT2STC [R/W] 00000000	Reserved		
0002A8 _H	BT2PCSR/BT2PRLL [R/W] XXXXXXXX XXXXXXXX		BT2PDUT/BT2PRLH [R/W], BT2DTBF [R] XXXXXXXX XXXXXXXX		
0002AC _H	Reserved				
0002B0 _H	BT3TMR [R] 00000000 00000000		BT3TMCR [R/W] 00000000 00000000		Base timer ch.3
0002B4 _H	Reserved	BT3STC [R/W] 00000000	Reserved		
0002B8 _H	BT3PCSR/BT3PRLL [R/W] XXXXXXXX XXXXXXXX		BT3PDUT/BT3PRLH [R/W], BT3DTBF [R] XXXXXXXX XXXXXXXX		
0002BC _H	BTSEL0123 [R/ W] 00000000	Reserved			

MB91605A Series**Table A-1 I/O map (7 / 15)**

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
0002C0 _H	BT4TMR [R] 00000000 00000000		BT4TMCR [R/W] 00000000 00000000		Base timer ch.4
0002C4 _H	Reserved	BT4STC [R/W] 00000000	Reserved		
0002C8 _H	BT4PCSR/BT4PRL [R/W] XXXXXXXX XXXXXXXX		BT4PDUT/BT4PRLH [R/W], BT4DTBF [R] XXXXXXXX XXXXXXXX		
0002CC _H	Reserved				
0002D0 _H	BT5TMR [R] 00000000 00000000		BT5TMCR [R/W] 00000000 00000000		Base timer ch.5
0002D4 _H	Reserved	BT5STC [R/W] 00000000	Reserved		
0002D8 _H	BT5PCSR/BT5PRL [R/W] XXXXXXXX XXXXXXXX		BT5PDUT/BT5PRLH [R/W], BT5DTBF [R] XXXXXXXX XXXXXXXX		
0002DC _H	Reserved				
0002E0 _H	BT6TMR [R] 00000000 00000000		BT6TMCR [R/W] 00000000 00000000		Base timer ch.6
0002E4 _H	Reserved	BT6STC [R/W] 00000000	Reserved		
0002E8 _H	BT6PCSR/BT6PRL [R/W] XXXXXXXX XXXXXXXX		BT6PDUT/BT6PRLH [R/W], BT6DTBF [R] XXXXXXXX XXXXXXXX		
0002EC _H	Reserved				
0002F0 _H	BT7TMR [R] 00000000 00000000		BT7TMCR [R/W] 00000000 00000000		Base timer ch.7
0002F4 _H	Reserved	BT7STC [R/W] 00000000	Reserved		
0002F8 _H	BT7PCSR/BT7PRL [R/W] XXXXXXXX XXXXXXXX		BT7PDUT/BT7PRLH [R/W], BT7DTBF [R] XXXXXXXX XXXXXXXX		
0002FC _H	BTSEL4567 [R/ W] 00000000	Reserved			
000300 _H to 0003CC _H	Reserved				Reserved

Table A-1 I/O map (8 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
0003D0 _H to 0003DC _H	Reserved				Instruction/data cache
0003E0 _H	CARR [R/W] 00000000	Reserved		DCHCR [R/W] XXXXXX00	
0003E4 _H	Reserved			ICHCR [R/W] XXXX0000	
0003E8 _H	DSIZE [R] 00100000 00000000		DFUNC [R] 00000001 01010010		
0003EC _H	ISIZE [R] 00100000 00000000		IFUNC [R] 00000000 00010110		
0003F0 _H to 0003FC _H	Reserved				Reserved
000400 _H	Reserved		DDR2 [R/W] ----0000	DDR3 [R/W] 00000000	Data direction register
000404 _H	DDR4 [R/W] --000000	DDR5 [R/W] 00000000	DDR6 [R/W] 00000000	DDR7[R/W] 00000000	
000408 _H	DDR8 [R/W] --000000	DDR9 [R/W] --000000	DDRA [R/W] --000000	DDRB[R/W] 00000000	
00040C _H	DDRC [R/W] 00000000	DDRD [R/W] 00000000	DDRE [R/W] 00000000	Reserved	
000410 _H to 00041C _H	Reserved				
000420 _H , 000424 _H	Reserved				Port pull-up control register
000428 _H	Reserved			PCRB [R/W] 00000000	
00042C _H	PCRC [R/W] 00000000	Reserved	PCRE [R/W] 00000000	Reserved	
000430 _H to 00043C _H	Reserved				

Table A-1 I/O map (9 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000440 _H	ICR00 [R/W] 11111111	ICR01 [R/W] 11111111	ICR02 [R/W] 11111111	ICR03 [R/W] 11111111	Interrupt controller
000444 _H	ICR04 [R/W] 11111111	ICR05 [R/W] 11111111	ICR06 [R/W] 11111111	ICR07 [R/W] 11111111	
000448 _H	ICR08 [R/W] 11111111	ICR09 [R/W] 11111111	ICR10 [R/W] 11111111	ICR11 [R/W] 11111111	
00044C _H	ICR12 [R/W] 11111111	ICR13 [R/W] 11111111	ICR14 [R/W] 11111111	ICR15 [R/W] 11111111	
000450 _H	ICR16 [R/W] 11111111	ICR17 [R/W] 11111111	ICR18 [R/W] 11111111	ICR19 [R/W] 11111111	
000454 _H	ICR20 [R/W] 11111111	ICR21 [R/W] 11111111	ICR22 [R/W] 11111111	ICR23 [R/W] 11111111	
000458 _H	ICR24 [R/W] 11111111	ICR25 [R/W] 11111111	ICR26 [R/W] 11111111	ICR27 [R/W] 11111111	
00045C _H	ICR28 [R/W] 11111111	ICR29 [R/W] 11111111	ICR30 [R/W] 11111111	ICR31 [R/W] 11111111	
000460 _H	ICR32 [R/W] 11111111	ICR33 [R/W] 11111111	ICR34 [R/W] 11111111	ICR35 [R/W] 11111111	
000464 _H	ICR36 [R/W] 11111111	ICR37 [R/W] 11111111	ICR38 [R/W] 11111111	ICR39 [R/W] 11111111	
000468 _H	ICR40 [R/W] 11111111	ICR41 [R/W] 11111111	ICR42 [R/W] 11111111	ICR43 [R/W] 11111111	
00046C _H	ICR44 [R/W] 11111111	ICR45 [R/W] 11111111	ICR46 [R/W] 11111111	ICR47 [R/W] 11111111	
000470 _H to 00047C _H	Reserved				
000480 _H	RSTRR [R] XXXXXXXX	RSTCR [R/W] 00000000	STBCR [R/W] 00000011	SLPRR [R/W] 00000000	Clock control unit
000484 _H	Reserved				
000488 _H	DIVR0 [R/W] 00000000	DIVR1 [R/W] 00010000	DIVR2 [R/W] 00110000	Reserved	
00048C _H	Reserved				
000490 _H	IORR0 [R/W] 00000000	IORR1 [R/W] 00000000	IORR2 [R/W] 00000000	IORR3 [R/W] 00000000	Peripheral DMA transfer request
000494 _H to 00049C _H	Reserved				Reserved

Table A-1 I/O map (10 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
0004A0 _H	Reserved		PFR2 [R/W] ----1110	PFR3 [R/W] 10001111	Port function register
0004A4 _H	PFR4 [R/W] --110000	PFR5 [R/W] 11111111	PFR6 [R/W] 00000000	PFR7 [R/W] 00000000	
0004A8 _H	PFR8 [R/W] --000000	PFR9 [R/W] --000000	PFRA [R/W] --000000	PFRB [R/W] 00000000	
0004AC _H	PFRC [R/W] 00000000	PFRD [R/W] 00000000	PFRE [R/W] 00000000	Reserved	
0004B0 _H to 0004DC _H	Reserved				
0004E0 _H	ADER [R/W] 00001111 11111111		Reserved		A/D input enable
0004E4 _H to 0004EC _H	Reserved				Reserved
0004F0 _H	ICSEL0[R/W] 00000000	ICSEL1[R/W] 00000000	ICSEL2[R/W] 00000000	ICSEL3[R/W] 00000000	DMA start request clear select function
0004F4 _H to 00050C _H	Reserved				Reserved
000510 _H	CSELR [R/W] -01---00 (at INIT) _**_*** (at RST)	CMONR [R] -01---00 (at INIT) _**_*** (at RST)	MTMCR [R/W] 00001111	Reserved	Clock generation
000514 _H	PLLCR [R/W] --000000 11110000 (at INIT) --***** (at RST)		CSTBR [R/W] ----0000(INIT pin = "L" level) ---- *****(at INIT) 0*** *****(at RST)	Reserved	
000518 _H to 0007DC _H	Reserved				Reserved
0007E0 _H to 0007E8 _H	Reserved				DMAC
0007EC _H	Reserved		DNMIR [R/W] 00000000	DILVR [R/W] 00011111	

Table A-1 I/O map (11 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
0007F0 _H to 0007F8 _H	Reserved				Reserved
0007FC _H	BMODR [R] XXXXXXXX (Varied by operation modes)	MODR [R] 000XXXXX (Varied by operation modes)	Reserved		Operation mode control
000800 _H to 000BFC _H	Reserved				Reserved
000C00 _H	GCFR [R/W] 00000000 00000000 00000000 00000000				DMAC
000C04 _H to 000CFC _H	Reserved				Reserved
000D00 _H	CCFR0 [R/W] 00000000 00000000		CSTR0 [R/W] 00000000 00000000		DMAC
000D04 _H	CCTR0 [R/W] 00000000 00000000		Reserved		
000D08 _H	SBA0 [R/W] 00000000 00000000 00000000 00000000				
000D0C _H	DBA0 [R/W] 00000000 00000000 00000000 00000000				
000D10 _H	PIX0 [R/W] 00000000 00000000 00000000 00000000				
000D14 _H	SIX0 [R/W] 00000000 00000000 00000000 00000000				
000D18 _H	BCL0 [R/W] 00000000 00000000 00000000 00000000				
000D1C _H	APR0 [R/W] 00000000 00000000 00000000 00000000				
000D20 _H	CCFR1 [R/W] 00000000 00000000		CSTR1 [R/W] 00000000 00000000		
000D24 _H	CCTR1 [R/W] 00000000 00000000		Reserved		
000D28 _H	SBA1 [R/W] 00000000 00000000 00000000 00000000				
000D2C _H	DBA1 [R/W] 00000000 00000000 00000000 00000000				

Table A-1 I/O map (12 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000D30 _H	PIX1 [R/W] 00000000 00000000 00000000 00000000				DMAC
000D34 _H	SIX1 [R/W] 00000000 00000000 00000000 00000000				
000D38 _H	BCL1 [R/W] 00000000 00000000 00000000 00000000				
000D3C _H	APR1 [R/W] 00000000 00000000 00000000 00000000				
000D40 _H	CCFR2 [R/W] 00000000 00000000		CSTR2 [R/W] 00000000 00000000		
000D44 _H	CCTR2 [R/W] 00000000 00000000		Reserved		
000D48 _H	SBA2 [R/W] 00000000 00000000 00000000 00000000				
000D4C _H	DBA2 [R/W] 00000000 00000000 00000000 00000000				
000D50 _H	PIX2 [R/W] 00000000 00000000 00000000 00000000				
000D54 _H	SIX2 [R/W] 00000000 00000000 00000000 00000000				
000D58 _H	BCL2 [R/W] 00000000 00000000 00000000 00000000				
000D5C _H	APR2 [R/W] 00000000 00000000 00000000 00000000				
000D60 _H	CCFR3 [R/W] 00000000 00000000		CSTR3 [R/W] 00000000 00000000		
000D64 _H	CCTR3 [R/W] 00000000 00000000		Reserved		
000D68 _H	SBA3 [R/W] 00000000 00000000 00000000 00000000				
000D6C _H	DBA3 [R/W] 00000000 00000000 00000000 00000000				
000D70 _H	PIX3 [R/W] 00000000 00000000 00000000 00000000				
000D74 _H	SIX3 [R/W] 00000000 00000000 00000000 00000000				
000D78 _H	BCL3 [R/W] 00000000 00000000 00000000 00000000				

MB91605A Series**Table A-1 I/O map (13 / 15)**

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000D7C _H	APR3 [R/W] 00000000 00000000 00000000 00000000				DMAC
000D80 _H to 000EFC _H	Reserved				Reserved
000F00 _H	BT8TMR [R] 00000000 00000000		BT8TMCR [R/W] 00000000 00000000		Base timer ch.8
000F04 _H	Reserved	BT8STC [R/W] 00000000	Reserved		
000F08 _H	BT8PCSR/BT8PRL [R/W] XXXXXXXX XXXXXXXX		BT8PDUT/BT8PRLH [R/W], BT8DTBF [R] XXXXXXXX XXXXXXXX		
000F0C _H	Reserved				
000F10 _H	BT9TMR [R] 00000000 00000000		BT9TMCR [R/W] 00000000 00000000		Base timer ch.9
000F14 _H	Reserved	BT9STC [R/W] 00000000	Reserved		
000F18 _H	BT9PCSR/BT9PRL [R/W] XXXXXXXX XXXXXXXX		BT9PDUT/BT9PRLH [R/W], BT9DTBF [R] XXXXXXXX XXXXXXXX		
000F1C _H	Reserved				
000F20 _H	BTATMR [R] 00000000 00000000		BTATMCR [R/W] 00000000 00000000		Base timer ch.10
000F24 _H	Reserved	BTASTC [R/W] 00000000	Reserved		
000F28 _H	BTAPCSR/BTAPRL [R/W] XXXXXXXX XXXXXXXX		BTAPDUT/BTAPRLH [R/W], BTADTBF [R] XXXXXXXX XXXXXXXX		
000F2C _H	Reserved				
000F30 _H	BTBTMR [R] 00000000 00000000		BTBTMCR [R/W] 00000000 00000000		Base timer ch.11
000F34 _H	Reserved	BTBSTC [R/W] 00000000	Reserved		
000F38 _H	BTBPCSR/BTBPRLL [R/W] XXXXXXXX XXXXXXXX		BTBPDUT/BTBPRLLH [R/W], BTBDTBF [R] XXXXXXXX XXXXXXXX		
000F3C _H	BTSEL89AB [R/W] 00000000	Reserved	BTSSSR [W] -----		

Table A-1 I/O map (14 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
000F40 _H to 001FFC _H	Reserved				Reserved
002000 _H	MCMR0 [R/W] ----- -0000000				External bus interface
002004 _H	MCMR1 [R/W] ----- -0000000				
002008 _H	MCMR2 [R/W] ----- -0000000				
00200C _H	MCMR3 [R/W] ----- -0000000				
002010 _H	MCMR4 [R/W] ----- -0000001				
002014 _H	MCMR5 [R/W] ----- -0000000				
002018 _H	MCMR6 [R/W] ----- -0000000				
00201C _H	MCMR7 [R/W] ----- -0000000				
002020 _H	MCTR0 [R/W] 00000101 01011111 11110000 00001111				
002024 _H	MCTR1 [R/W] 00000101 01011111 11110000 00001111				
002028 _H	MCTR2 [R/W] 00000101 01011111 11110000 00001111				
00202C _H	MCTR3 [R/W] 00000101 01011111 11110000 00001111				
002030 _H	MCTR4 [R/W] 00000101 01011111 11110000 00001111				
002034 _H	MCTR5 [R/W] 00000101 01011111 11110000 00001111				
002038 _H	MCTR6 [R/W] 00000101 01011111 11110000 00001111				
00203C _H	MCTR7 [R/W] 00000101 01011111 11110000 00001111				
002040 _H	MCAR0 [R/W] ----- -0001111 ----- 01000000				
002044 _H	MCAR1 [R/W] ----- -0001111 ----- 00010000				

Table A-1 I/O map (15 / 15)

Address	Registers				Peripheral Function
	+0	+1	+2	+3	
002048 _H	MCAR2 [R/W] ----- -0001111 ----- 00100000				External bus interface
00204C _H	MCAR3 [R/W] ----- -0001111 ----- 00110000				
002050 _H	MCAR4 [R/W] ----- -0001111 ----- 00000000				
002054 _H	MCAR5 [R/W] ----- -0001111 ----- 01010000				
002058 _H	MCAR6 [R/W] ----- -0001111 ----- 01100000				
00205C _H	MCAR7 [R/W] ----- -0001111 ----- 01110000				
002000 _H to 0020FC _H	Reserved				Reserved
002100 _H	SDMR [R/W] ----- ----- 00010011 --00-000				SDRAM interface
002104 _H	SDRTR [R/W] -----0 00000000 00000000 00101000				
002108 _H	SDPDR [R/W] ----- ----- 00000000 00000000				
00210C _H	SDTR [R/W] -----00 01000010 00010001 0100--01				
002110 _H	SDCMR [R/W] 0----- ---00000 00000000 00000000				
002114 _H to 0022FC _H	Reserved				Reserved
002300 _H	CLKCTL [R/W] -----00	Reserved			External bus interface
002304 _H to 007FFC _H	Reserved				Reserved

APPENDIX B 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.

Abbreviation	Register Name	Address	See
A			
ADAT0	A/D conversion result register ch.0	000024 _H	23.4.3
ADAT1	A/D conversion result register ch.1	000026 _H	23.4.3
ADAT2	A/D conversion result register ch.2	000028 _H	23.4.3
ADAT3	A/D conversion result register ch.3	00002A _H	23.4.3
ADAT4	A/D conversion result register ch.4	00002C _H	23.4.3
ADAT5	A/D conversion result register ch.5	00002E _H	23.4.3
ADAT6	A/D conversion result register ch.6	000030 _H	23.4.3
ADAT7	A/D conversion result register ch.7	000032 _H	23.4.3
ADAT8	A/D conversion result register ch.8	000034 _H	23.4.3
ADAT9	A/D conversion result register ch.9	000036 _H	23.4.3
ADAT10	A/D conversion result register ch.10	000038 _H	23.4.3
ADAT11	A/D conversion result register ch.11	00003A _H	23.4.3
ADCH	Software conversion analog input select register	000022 _H	23.4.2
ADCTH	A/DC control register	000020 _H	23.4.1
ADCTL	A/DC control register	000021 _H	23.4.1
ADER	ADER control register	0004E0 _H	17.4.5
APR0	Alternate pointer register 0	000D1C _H	25.3.10
APR1	Alternate pointer register 1	000D3C _H	25.3.10
APR2	Alternate pointer register 2	000D5C _H	25.3.10
APR3	Alternate pointer register 3	000D7C _H	25.3.10

MB91605A Series**B**

BCL0	Byte count limit register 0	000D18 _H	25.3.9
BCL1	Byte count limit register 1	000D38 _H	25.3.9
BCL2	Byte count limit register 2	000D58 _H	25.3.9
BCL3	Byte count limit register 3	000D78 _H	25.3.9
BGR00	Baud rate generator register 00	000067 _H	24.4.6 ,24.13.6
BGR01	Baud rate generator register 01	000077 _H	24.4.6 ,24.13.6 ,24.21.8
BGR02	Baud rate generator register 02	000087 _H	24.4.6 ,24.13.6 ,24.21.8
BGR03	Baud rate generator register 03	000097 _H	24.4.6 ,24.13.6 ,24.21.8
BGR04	Baud rate generator register 04	0000A7 _H	24.4.6 ,24.13.6 ,24.21.8
BGR05	Baud rate generator register 05	0000B7 _H	24.4.6 ,24.13.6 ,24.21.8
BGR06	Baud rate generator register 06	0000C7 _H	24.4.6 ,24.13.6 ,24.21.8
BGR07	Baud rate generator register 07	0000D7 _H	24.4.6 ,24.13.6 ,24.21.8
BGR08	Baud rate generator register 08	000207 _H	24.4.6 ,24.13.6 ,24.21.8
BGR09	Baud rate generator register 09	000217 _H	24.4.6 ,24.13.6 ,24.21.8
BGR0A	Baud rate generator register 0A	000227 _H	24.4.6 ,24.13.6 ,24.21.8
BGR0B	Baud rate generator register 0B	000237 _H	24.4.6 ,24.13.6 ,24.21.8
BGR10	Baud rate generator register 10	000066 _H	24.4.6 ,24.13.6
BGR11	Baud rate generator register 11	000076 _H	24.4.6 ,24.13.6 ,24.21.8
BGR12	Baud rate generator register 12	000086 _H	24.4.6 ,24.13.6 ,24.21.8
BGR13	Baud rate generator register 13	000096 _H	24.4.6 ,24.13.6 ,24.21.8
BGR14	Baud rate generator register 14	0000A6 _H	24.4.6 ,24.13.6 ,24.21.8
BGR15	Baud rate generator register 15	0000B6 _H	24.4.6 ,24.13.6 ,24.21.8
BGR16	Baud rate generator register 16	0000C6 _H	24.4.6 ,24.13.6 ,24.21.8
BGR17	Baud rate generator register 17	0000D6 _H	24.4.6 ,24.13.6 ,24.21.8
BGR18	Baud rate generator register 18	000206 _H	24.4.6 ,24.13.6 ,24.21.8
BGR19	Baud rate generator register 19	000216 _H	24.4.6 ,24.13.6 ,24.21.8
BGR1A	Baud rate generator register 1A	000226 _H	24.4.6 ,24.13.6 ,24.21.8
BGR1B	Baud rate generator register 1B	000236 _H	24.4.6 ,24.13.6 ,24.21.8
BMODR	Bus mode data register	0007FC _H	6.3.1

BT0PCSR	Base timer 0 cycle configuration register	000288 _H	22.8.1.2 , 22.8.3.2
BT0PDUT	Base timer 0 duty configuration register	00028A _H	22.8.1.3
BT0PRLH	Base timer 0 H width setting reload register	00028A _H	22.8.2.3
BT0DTBF	Base timer 0 data buffer register	00028A _H	22.8.4.2
BT0PRLL	Base timer 0 L width configuration reload register	000288 _H	22.8.2.2
BT0STC	Base timer 0 status control register	000285 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT0TMCR	Base timer 0 timer control register	000282 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT0TMR	Base timer 0 timer register	000280 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT1PCSR	Base timer 1 cycle configuration register	000298 _H	22.8.1.2 , 22.8.3.2
BT1PDUT	Base timer 1 duty configuration register	00029A _H	22.8.1.3
BT1PRLH	Base timer 1 H width configuration reload register	00029A _H	22.8.2.3
BT1DTBF	Base timer 1 data buffer register	00029A _H	22.8.4.2
BT1PRLL	Base timer 1 L width configuration reload register	000298 _H	22.8.2.2
BT1STC	Base timer 1 status control register	000295 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT1TMCR	Base timer 1 timer control register	000292 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT1TMR	Base timer 1 timer register	000290 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT2PCSR	Base timer 2 cycle configuration register	0002A8 _H	22.8.1.2 , 22.8.3.2
BT2PDUT	Base timer 2 duty configuration register	0002AA _H	22.8.1.3
BT2PRLH	Base timer 2 H width configuration reload register	0002AA _H	22.8.2.3
BT2DTBF	Base timer 2 data buffer register	0002AA _H	22.8.4.2
BT2PRLL	Base timer 2 L width configuration reload register	0002A8 _H	22.8.2.2
BT2STC	Base timer 2 status control register	0002A5 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT2TMCR	Base timer 2 timer control register	0002A2 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT2TMR	Base timer 2 timer register	0002A0 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT3PCSR	Base timer 3 cycle configuration register	0002B8 _H	22.8.1.2 , 22.8.3.2
BT3PDUT	Base timer 3 duty configuration register	0002BA _H	22.8.1.3
BT3PRLH	Base timer 3 H width configuration reload register	0002BA _H	22.8.2.3
BT3DTBF	Base timer 3 data buffer register	0002BA _H	22.8.4.2

MB91605A Series

BT3PRL	Base timer 3 L width configuration reload register	0002B8 _H	22.8.2.2
BT3STC	Base timer 3 status control register	0002B5 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT3TMCR	Base timer 3 timer control register	0002B2 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT3TMR	Base timer 3 timer register	0002B0 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT4PCSR	Base timer 4 cycle configuration register	0002C8 _H	22.8.1.2 , 22.8.3.2
BT4PDUT	Base timer 4 duty configuration register	0002CA _H	22.8.1.3
BT4PRLH	Base timer 4 H width configuration reload register	0002CA _H	22.8.2.3
BT4DTBF	Base timer 4 data buffer register	0002CA _H	22.8.4.2
BT4PRL	Base timer 4 L width configuration reload register	0002C8 _H	22.8.2.2
BT4STC	Base timer 4 status control register	0002C5 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT4TMCR	Base timer 4 timer control register	0002C2 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT4TMR	Base timer 4 timer register	0002C0 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT5PCSR	Base timer 5 cycle configuration register	0002D8 _H	22.8.1.2 , 22.8.3.2
BT5PDUT	Base timer 5 duty configuration register	0002DA _H	22.8.1.3
BT5PRLH	Base timer 5 H width configuration reload register	0002DA _H	22.8.2.3
BT5DTBF	Base timer 5 data buffer register	0002DA _H	22.8.4.2
BT5PRL	Base timer 5 L width configuration reload register	0002D8 _H	22.8.2.2
BT5STC	Base timer 5 status control register	0002D5 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT5TMCR	Base timer 5 timer control register	0002D2 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT5TMR	Base timer 5 timer register	0002D0 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT6PCSR	Base timer 6 cycle configuration register	0002E8 _H	22.8.1.2 , 22.8.3.2
BT6PDUT	Base timer 6 duty configuration register	0002EA _H	22.8.1.3
BT6PRLH	Base timer 6 H width configuration reload register	0002EA _H	22.8.2.3
BT6DTBF	Base timer 6 data buffer register	0002EA _H	22.8.4.2
BT6PRL	Base timer 6 L width configuration reload register	0002E8 _H	22.8.2.2
BT6STC	Base timer 6 status control register	0002E5 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT6TMCR	Base timer 6 timer control register	0002E2 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1

BT6TMR	Base timer 6 timer register	0002E0 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT7PCSR	Base timer 7 cycle configuration register	0002F8 _H	22.8.1.2 , 22.8.3.2
BT7PDUT	Base timer 7 duty configuration register	0002FA _H	22.8.1.3
BT7PRLH	Base timer 7 H width configuration reload register	0002FA _H	22.8.2.3
BT7DTBF	Base timer 7 data buffer register	0002FA _H	22.8.4.2
BT7PRLL	Base timer 7 L width configuration reload register	0002F8 _H	22.8.2.2
BT7STC	Base timer 7 status control register	0002F5 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT7TMCR	Base timer 7 timer control register	0002F2 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT7TMR	Base timer 7 timer register	0002F0 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT8PCSR	Base timer 8 cycle configuration register	000F08 _H	22.8.1.2 , 22.8.3.2
BT8PDUT	Base timer 8 duty configuration register	000F0A _H	22.8.1.3
BT8PRLH	Base timer 8 H width configuration reload register	000F0A _H	22.8.2.3
BT8DTBF	Base timer 8 data buffer register	000F0A _H	22.8.4.2
BT8PRLL	Base timer 8 L width configuration reload register	000F08 _H	22.8.2.2
BT8STC	Base timer 8 status control register	000F05 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT8TMCR	Base timer 8 timer control register	000F02 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT8TMR	Base timer 8 timer register	000F00 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BT9PCSR	Base timer 9 cycle configuration register	000F18 _H	22.8.1.2 , 22.8.3.2
BT9PDUT	Base timer 9 duty configuration register	000F1A _H	22.8.1.3
BT9PRLH	Base timer 9 H width configuration reload register	000F1A _H	22.8.2.3
BT9DTBF	Base timer 9 data buffer register	000F1A _H	22.8.4.2
BT9PRLL	Base timer 9 L width configuration reload register	000F18 _H	22.8.2.2
BT9STC	Base timer 9 status control register	000F15 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT9TMCR	Base timer 9 timer control register	000F12 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BT9TMR	Base timer 9 timer register	000F10 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BTAPCSR	Base timer 10 cycle configuration register	000F28 _H	22.8.1.2 , 22.8.3.2
BTAPDUT	Base timer 10 duty configuration register	000F2A _H	22.8.1.3
BTAPRLH	Base timer 10 H width configuration reload register	000F2A _H	22.8.2.3

MB91605A Series

BTADTBF	Base timer 10 data buffer register	000F2A _H	22.8.4.2
BTAPRLL	Base timer 10 L width configuration reload register	000F28 _H	22.8.2.2
BTASTC	Base timer 10 status control register	000F25 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BTATMCR	Base timer 10 timer control register	000F22 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BTATMR	Base timer 10 timer register	000F20 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BTBPCSR	Base timer 11 cycle configuration register	000F38 _H	22.8.1.2 , 22.8.3.2
BTBPDUT	Base timer 11 duty configuration register	000F3A _H	22.8.1.3
BTBPRLL	Base timer 11 H width configuration reload register	000F3A _H	22.8.2.3
BTBDTBF	Base timer 11 data buffer register	000F3A _H	22.8.4.2
BTBPRLL	Base timer 11 L width configuration reload register	000F38 _H	22.8.2.2
BTBSTC	Base timer 11 status control register	000F35 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BTBTMCR	Base timer 11 timer control register	000F32 _H	22.8.1.1 , 22.8.2.1 , 22.8.3.1 , 22.8.4.1
BTBTMR	Base timer 11 timer register	000F30 _H	22.8.1.4 , 22.8.2.4 , 22.8.3.3
BTSEL0123	IO select register for ch. 0/1/2/3	0002BC _H	21.4.1
BTSEL4567	IO select register for ch.4/5/6/7	0002FC _H	21.4.2
BTSEL89AB	IO select register for ch.8/9/A/B	000F3C _H	21.4.3
BTSSSR	Simultaneous software start register	000F3E _H	21.4.4

C

CARR	Instruction/data cache area configuration register	0003E0 _H	4.3.1
CCFR0	Channel configuration register 0	000D00 _H	25.3.2
CCFR1	Channel configuration register 1	000D20 _H	25.3.2
CCFR2	Channel configuration register 2	000D40 _H	25.3.2
CCFR3	Channel configuration register 3	000D60 _H	25.3.2
CCTR0	Channel control register 0	000D04 _H	25.3.4
CCTR1	Channel control register 1	000D24 _H	25.3.4
CCTR2	Channel control register 2	000D44 _H	25.3.4
CCTR3	Channel control register 3	000D64 _H	25.3.4

CLKCTL	Clock control register	002300 _H	16.3.9
CMONR	Clock source monitor register	000511 _H	7.4.2
CSELR	Clock source select register	000510 _H	7.4.1
CSTBR	Clock stabilization time select register	000516 _H	7.4.3
CSTR0	Channel status register 0	000D02 _H	25.3.3
CSTR1	Channel status register 1	000D22 _H	25.3.3
CSTR2	Channel status register 2	000D42 _H	25.3.3
CSTR3	Channel status register 3	000D62 _H	25.3.3

D

DBA0	Destination base address register 0	000D0C _H	25.3.6
DBA1	Destination base address register 1	000D2C _H	25.3.6
DBA2	Destination base address register 2	000D4C _H	25.3.6
DBA3	Destination base address register 3	000D6C _H	25.3.6
DCHCR	Data cache control register	0003E3 _H	5.3.1
DDR2	Data direction register 2	000402 _H	17.4.2
DDR3	Data direction register 3	000403 _H	17.4.2
DDR4	Data direction register 4	000404 _H	17.4.2
DDR5	Data direction register 5	000405 _H	17.4.2
DDR6	Data direction register 6	000406 _H	17.4.2
DDR7	Data direction register 7	000407 _H	17.4.2
DDR8	Data direction register 8	000408 _H	17.4.2
DDR9	Data direction register 9	000409 _H	17.4.2
DDRA	Data direction register A	00040A _H	17.4.2
DDRB	Data direction register B	00040B _H	17.4.2
DDRC	Data direction register C	00040C _H	17.4.2
DDRD	Data direction register D	00040D _H	17.4.2
DDRE	Data direction register E	00040E _H	17.4.2
DFUNC	Data cache function register	0003EA _H	5.3.3
DICR	Delayed interrupt control register	000044 _H	15.3.1

MB91605A Series

DILVR	DMA-halt by interrupt level register	0007EF _H	25.3.12
DIVR0	Divide clock configuration register 0	000488 _H	8.4.1
DIVR1	Divide clock configuration register 1	000489 _H	8.4.2
DIVR2	Divide clock configuration register 2	00048A _H	8.4.3
DNMIR	DMA –halt by NMI flag register	0007EE _H	25.3.11
DSIZE	Data cache size register	0003E8 _H	5.3.2

E

EIRR0	External interrupt trigger register 0	000040 _H	18.4.2
EIRR1	External interrupt trigger register 1	0000E0 _H	18.4.2
EIRR2	External interrupt trigger register 2	0000E4 _H	18.4.2
ELVR0	External interrupt request level registers 0	000042 _H	18.4.1
ELVR1	External interrupt request level registers 1	0000E2 _H	18.4.1
ELVR2	External interrupt request level registers 2	0000E6 _H	18.4.1
ENIR0	Enable interrupt register 0	000041 _H	18.4.3
ENIR1	Enable interrupt register 1	0000E1 _H	18.4.3
ENIR2	Enable interrupt register 2	0000E5 _H	18.4.3
ESCR0	Extended communication register 0	000063 _H	24.4.4 , 24.13.4
ESCR1	Extended communication control register 1	000073 _H	24.4.4 , 24.13.4
ESCR2	Extended communication control register 2	000083 _H	24.4.4 , 24.13.4
ESCR3	Extended communication control register 3	000093 _H	24.4.4 , 24.13.4
ESCR4	Extended communication control register 4	0000A3 _H	24.4.4 , 24.13.4
ESCR5	Extended communication control register 5	0000B3 _H	24.4.4 , 24.13.4
ESCR6	Extended communication control register 6	0000C3 _H	24.4.4 , 24.13.4
ESCR7	Extended communication control register 7	0000D3 _H	24.4.4 , 24.13.4
ESCR8	Extended communication control register 8	000203 _H	24.4.4 , 24.13.4
ESCR9	Extended communication control register 9	000213 _H	24.4.4 , 24.13.4
ESCRA	Extended communication control register A	000223 _H	24.4.4 , 24.13.4
ESCRB	Extended communication control register B	000233 _H	24.4.4 , 24.13.4

F

FBYTE10	FIFO1 byte register 0	00006F _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE11	FIFO1 byte register 1	00007F _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE12	FIFO1 byte register 2	00008F _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE13	FIFO1 byte register 3	00009F _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE14	FIFO1 byte register 4	0000AF _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE15	FIFO1 byte register 5	0000BF _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE16	FIFO1 byte register 6	0000CF _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE17	FIFO1 byte register 7	0000DF _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE20	FIFO2 byte register 0	00006E _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE21	FIFO2 byte register 1	00007E _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE22	FIFO2 byte register 2	00008E _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE23	FIFO2 byte register 3	00009E _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE24	FIFO2 byte register 4	0000AE _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE25	FIFO2 byte register 5	0000BE _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE26	FIFO2 byte register 6	0000CE _H	24.4.9 , 24.13.9 , 24.21.11
FBYTE27	FIFO2 byte register 7	0000DE _H	24.4.9 , 24.13.9 , 24.21.11
FCR00	FIFO control register 00	00006D _H	24.4.8 , 24.13.8 , 24.21.10
FCR01	FIFO control register 01	00007D _H	24.4.8 , 24.13.8 , 24.21.10
FCR02	FIFO control register 02	00008D _H	24.4.8 , 24.13.8 , 24.21.10
FCR03	FIFO control register 03	00009D _H	24.4.8 , 24.13.8 , 24.21.10
FCR04	FIFO control register 04	0000AD _H	24.4.8 , 24.13.8 , 24.21.10
FCR05	FIFO control register 05	0000BD _H	24.4.8 , 24.13.8 , 24.21.10
FCR06	FIFO control register 06	0000CD _H	24.4.8 , 24.13.8 , 24.21.10
FCR07	FIFO control register 07	0000DD _H	24.4.8 , 24.13.8 , 24.21.10
FCR10	FIFO control register 10	00006C _H	24.4.7 , 24.13.7 , 24.21.9
FCR11	FIFO control register 11	00007C _H	24.4.7 , 24.13.7 , 24.21.9
FCR12	FIFO control register 12	00008C _H	24.4.7 , 24.13.7 , 24.21.9
FCR13	FIFO control register 13	00009C _H	24.4.7 , 24.13.7 , 24.21.9
FCR14	FIFO control register 14	0000AC _H	24.4.7 , 24.13.7 , 24.21.9

MB91605A Series

FCR15	FIFO control register 15	0000BC _H	24.4.7 , 24.13.7 , 24.21.9
FCR16	FIFO control register 16	0000CC _H	24.4.7 , 24.13.7 , 24.21.9
FCR17	FIFO control register 17	0000DC _H	24.4.7 , 24.13.7 , 24.21.9

G

GCFR	Global configuration register	000C00 _H	25.3.1
------	-------------------------------	---------------------	---------------

I

IBCR0	I ² C bus control register 0	000060 _H	24.21.1
IBCR1	I ² C bus control register 1	000070 _H	24.21.1
IBCR2	I ² C bus control register 2	000080 _H	24.21.1
IBCR3	I ² C bus control register 3	000090 _H	24.21.1
IBCR4	I ² C bus control register 4	0000A0 _H	24.21.1
IBCR5	I ² C bus control register 5	0000B0 _H	24.21.1
IBCR6	I ² C bus control register 6	0000C0 _H	24.21.1
IBCR7	I ² C bus control register 7	0000D0 _H	24.21.1
IBCR8	I ² C bus control register 8	000200 _H	24.21.1
IBCR9	I ² C bus control register 9	000210 _H	24.21.1
IBCR A	I ² C bus control register A	000220 _H	24.21.1
IBCR B	I ² C bus control register B	000230 _H	24.21.1
IBSR0	I ² C bus status register 0	000063 _H	24.21.3
IBSR1	I ² C bus status register 1	000073 _H	24.21.3
IBSR2	I ² C bus status register 2	000083 _H	24.21.3
IBSR3	I ² C bus status register 3	000093 _H	24.21.3
IBSR4	I ² C bus status register 4	0000A3 _H	24.21.3
IBSR5	I ² C bus status register 5	0000B3 _H	24.21.3
IBSR6	I ² C bus status register 6	0000C3 _H	24.21.3

IBSR7	I ² C bus status register 7	0000D3 _H	24.21.3
IBSR8	I ² C bus status register 8	000203 _H	24.21.3
IBSR9	I ² C bus status register 9	000213 _H	24.21.3
IBSRA	I ² C bus status register A	000223 _H	24.21.3
IBSRB	I ² C bus status register B	000233 _H	24.21.3
ICHCR	Instruction cache control register	0003E7 _H	4.3.2
ICR00	Interrupt control register 00	000440 _H	12.3.1
ICR01	Interrupt control register 01	000441 _H	12.3.1
ICR02	Interrupt control register 02	000442 _H	12.3.1
ICR03	Interrupt control register 03	000443 _H	12.3.1
ICR04	Interrupt control register 04	000444 _H	12.3.1
ICR05	Interrupt control register 05	000445 _H	12.3.1
ICR06	Interrupt control register 06	000446 _H	12.3.1
ICR07	Interrupt control register 07	000447 _H	12.3.1
ICR08	Interrupt control register 08	000448 _H	12.3.1
ICR09	Interrupt control register 09	000449 _H	12.3.1
ICR10	Interrupt control register 10	00044A _H	12.3.1
ICR11	Interrupt control register 11	00044B _H	12.3.1
ICR12	Interrupt control register 12	00044C _H	12.3.1
ICR13	Interrupt control register 13	00044D _H	12.3.1
ICR14	Interrupt control register 14	00044E _H	12.3.1
ICR15	Interrupt control register 15	00044F _H	12.3.1
ICR16	Interrupt control register 16	000450 _H	12.3.1
ICR17	Interrupt control register 17	000451 _H	12.3.1
ICR18	Interrupt control register 18	000452 _H	12.3.1
ICR19	Interrupt control register 19	000453 _H	12.3.1
ICR20	Interrupt control register 20	000454 _H	12.3.1
ICR21	Interrupt control register 21	000455 _H	12.3.1
ICR22	Interrupt control register 22	000456 _H	12.3.1
ICR23	Interrupt control register 23	000457 _H	12.3.1

MB91605A Series

ICR24	Interrupt control register 24	000458 _H	12.3.1
ICR25	Interrupt control register 25	000459 _H	12.3.1
ICR26	Interrupt control register 26	00045A _H	12.3.1
ICR27	Interrupt control register 27	00045B _H	12.3.1
ICR28	Interrupt control register 28	00045C _H	12.3.1
ICR29	Interrupt control register 29	00045D _H	12.3.1
ICR30	Interrupt control register 30	00045E _H	12.3.1
ICR31	Interrupt control register 31	00045F _H	12.3.1
ICR32	Interrupt control register 32	000460 _H	12.3.1
ICR33	Interrupt control register 33	000461 _H	12.3.1
ICR34	Interrupt control register 34	000462 _H	12.3.1
ICR35	Interrupt control register 35	000463 _H	12.3.1
ICR36	Interrupt control register 36	000464 _H	12.3.1
ICR37	Interrupt control register 37	000465 _H	12.3.1
ICR38	Interrupt control register 38	000466 _H	12.3.1
ICR39	Interrupt control register 39	000467 _H	12.3.1
ICR40	Interrupt control register 40	000468 _H	12.3.1
ICR41	Interrupt control register 41	000469 _H	12.3.1
ICR42	Interrupt control register 42	00046A _H	12.3.1
ICR43	Interrupt control register 43	00046B _H	12.3.1
ICR44	Interrupt control register 44	00046C _H	12.3.1
ICR45	Interrupt control register 45	00046D _H	12.3.1
ICR46	Interrupt control register 46	00046E _H	12.3.1
ICR47	Interrupt control register 47	00046F _H	12.3.1
ICSEL0	Select register 0 for DMA transfer request clear by a peripheral	0004F0 _H	26.3.2
ICSEL1	Select register 1 for DMA transfer request clear by a peripheral	0004F1 _H	26.3.3
ICSEL2	Select register 2 for DMA transfer request clear by a peripheral	0004F2 _H	26.3.4
ICSEL3	Select register 3 for DMA transfer request clear by a peripheral	0004F3 _H	26.3.5
IFUNC	Instruction cache function register	0003EE _H	4.3.4

IORR0	IO transfer request register 0	000490 _H	26.3.1
IORR1	IO transfer request register 1	000491 _H	26.3.1
IORR2	IO transfer request register 2	000492 _H	26.3.1
IORR3	IO transfer request register 3	000493 _H	26.3.1
IRPR0H	Interrupt request batch-read register 0 upper	0000E8 _H	14.3.1
ISBA0	7 bit slave address register 0	000069 _H	24.21.7
ISBA1	7 bit slave address register 1	000079 _H	24.21.7
ISBA2	7 bit slave address register 2	000089 _H	24.21.7
ISBA3	7 bit slave address register 3	000099 _H	24.21.7
ISBA4	7 bit slave address register 4	0000A9 _H	24.21.7
ISBA5	7 bit slave address register 5	0000B9 _H	24.21.7
ISBA6	7 bit slave address register 6	0000C9 _H	24.21.7
ISBA7	7 bit slave address register 7	0000D9 _H	24.21.7
ISBA8	7 bit slave address register 8	000209 _H	24.21.7
ISBA9	7 bit slave address register 9	000219 _H	24.21.7
ISBAA	7 bit slave address register A	000229 _H	24.21.7
ISBAB	7 bit slave address register B	000239 _H	24.21.7
ISIZE	Instruction cache size register	0003EC _H	4.3.3
ISMK0	7 bit slave address mask register 0	000068 _H	24.21.6
ISMK1	7 bit slave address mask register 1	000078 _H	24.21.6
ISMK2	7 bit slave address mask register 2	000088 _H	24.21.6
ISMK3	7 bit slave address mask register 3	000098 _H	24.21.6
ISMK4	7 bit slave address mask register 4	0000A8 _H	24.21.6
ISMK5	7 bit slave address mask register 5	0000B8 _H	24.21.6
ISMK6	7 bit slave address mask register 6	0000C8 _H	24.21.6
ISMK7	7 bit slave address mask register 7	0000D8 _H	24.21.6
ISMK8	7 bit slave address mask register 8	000208 _H	24.21.6
ISMK9	7 bit slave address mask register 9	000218 _H	24.21.6
ISMKA	7 bit slave address mask register A	000228 _H	24.21.6
ISMKB	7 bit slave address mask register B	000238 _H	24.21.6

MB91605A Series**M**

MCAR0	SRAM/FLASH area register 0	002040 _H	16.3.3
MCAR1	SRAM/FLASH area register 1	002044 _H	16.3.3
MCAR2	SRAM/FLASH area register 2	002048 _H	16.3.3
MCAR3	SRAM/FLASH area register 3	00204C _H	16.3.3
MCAR4	SRAM/FLASH area register 4	002050 _H	16.3.3
MCAR5	SRAM/FLASH area register 5	002054 _H	16.3.3
MCAR6	SRAM/FLASH area register 6	002058 _H	16.3.3
MCAR7	SRAM/FLASH area register 7	00205C _H	16.3.3
MCMR0	SRAM/FLASH mode register 0	002000 _H	16.3.1
MCMR1	SRAM/FLASH mode register 1	002004 _H	16.3.1
MCMR2	SRAM/FLASH mode register 2	002008 _H	16.3.1
MCMR3	SRAM/FLASH mode register 3	00200C _H	16.3.1
MCMR4	SRAM/FLASH mode register 4	002010 _H	16.3.1
MCMR5	SRAM/FLASH mode register 5	002014 _H	16.3.1
MCMR6	SRAM/FLASH mode register 6	002018 _H	16.3.1
MCMR7	SRAM/FLASH mode register 7	00201C _H	16.3.1
MCTR0	SRAM/FLASH timing register 0	002020 _H	16.3.2
MCTR1	SRAM/FLASH timing register 1	002024 _H	16.3.2
MCTR2	SRAM/FLASH timing register 2	002028 _H	16.3.2
MCTR3	SRAM/FLASH timing register 3	00202C _H	16.3.2
MCTR4	SRAM/FLASH timing register 4	002030 _H	16.3.2
MCTR5	SRAM/FLASH timing register 5	002034 _H	16.3.2
MCTR6	SRAM/FLASH timing register 6	002038 _H	16.3.2
MCTR7	SRAM/FLASH timing register 7	00203C _H	16.3.2
MODR	Mode register	0007FD _H	6.3.2
MTMCR	Main timer control register	000512 _H	9.3.1

P

PCRB	Pull-up control register B	00042B _H	17.4.4
PCRC	Pull-up control register C	00042C _H	17.4.4
PCRE	Pull-up control register E	00042E _H	17.4.4
PDR2	Port data register 2	000002 _H	17.4.1
PDR3	Port data register 3	000003 _H	17.4.1
PDR4	Port data register 4	000004 _H	17.4.1
PDR5	Port data register 5	000005 _H	17.4.1
PDR6	Port data register 6	000006 _H	17.4.1
PDR7	Port data register 7	000007 _H	17.4.1
PDR8	Port data register 8	000008 _H	17.4.1
PDR9	Port data register 9	000009 _H	17.4.1
PDRA	Port data register A	00000A _H	17.4.1
PDRB	Port data register B	00000B _H	17.4.1
PDRC	Port data register C	00000C _H	17.4.1
PDRD	Port data register D	00000D _H	17.4.1
PDRE	Port data register E	00000E _H	17.4.1
PFR2	Port function register 2	0004A2 _H	17.4.3
PFR3	Port function register 3	0004A3 _H	17.4.3
PFR4	Port function register 4	0004A4 _H	17.4.3
PFR5	Port function register 5	0004A5 _H	17.4.3
PFR6	Port function register 6	0004A6 _H	17.4.3
PFR7	Port function register 7	0004A7 _H	17.4.3
PFR8	Port function register 8	0004A8 _H	17.4.3
PFR9	Port function register 9	0004A9 _H	17.4.3
PFRA	Port function register A	0004AA _H	17.4.3
PFRB	Port function register B	0004AB _H	17.4.3
PFRC	Port function register C	0004AC _H	17.4.3
PFRD	Port function register D	0004AD _H	17.4.3
PFRE	Port function register E	0004AE _H	17.4.3

MB91605A Series

PIX0	Primary index register 0	000D10 _H	25.3.7
PIX1	Primary index register 1	000D30 _H	25.3.7
PIX2	Primary index register 2	000D50 _H	25.3.7
PIX3	Primary index register 3	000D70 _H	25.3.7
PLLCR	PLL configuration register	000514 _H	7.4.4

R

RCADR1	Device address configuration register 1	0000F6 _H	27.2.3
RCADR2	Device address configuration register 2	0000F7 _H	27.2.3
RCCKD	Clock division register	0000FC _H	27.2.8
RCCR	Remote control reception control register	0000F0 _H	27.2.1
RCDAHW	High width configuration register A	0000F3 _H	27.2.5
RCDBHW	High width configuration register B	0000F4 _H	27.2.6
RCDTHH	Data storage register HH	0000F8 _H	27.2.7
RCDTHL	Data storage register HL	0000F9 _H	27.2.7
RCDTLH	Data storage register LH	0000FA _H	27.2.7
RCDTLL	Data storage register LL	0000FB _H	27.2.7
RCSHW	Start bit high width configuration register	0000F2 _H	27.2.4
RCST	Remote control reception interrupt control register	0000F1 _H	27.2.2
RDR0	Receive data register 0	000064 _H	24.4.5 , 24.13.5 , 24.21.5
RDR1	Receive data register 1	000074 _H	24.4.5 , 24.13.5 , 24.21.5
RDR2	Receive data register 2	000084 _H	24.4.5 , 24.13.5 , 24.21.5
RDR3	Receive data register 3	000094 _H	24.4.5 , 24.13.5 , 24.21.5
RDR4	Receive data register 4	0000A4 _H	24.4.5 , 24.13.5 , 24.21.5
RDR5	Receive data register 5	0000B4 _H	24.4.5 , 24.13.5 , 24.21.5
RDR6	Receive data register 6	0000C4 _H	24.4.5 , 24.13.5 , 24.21.5
RDR7	Receive data register 7	0000D4 _H	24.4.5 , 24.13.5 , 24.21.5
RDR8	Receive data register 8	000204 _H	24.4.5 , 24.13.5 , 24.21.5
RDR9	Receive data register 9	000214 _H	24.4.5 , 24.13.5 , 24.21.5
RDRA	Receive data register A	000224 _H	24.4.5 , 24.13.5 , 24.21.5

RDRB	Receive data register B	000234 _H	24.4.5 ,24.13.5 ,24.21.5
RDRM8	Receive data mirror register 8	000240 _H	24.13.11
RDRM9	Receive data mirror register 9	000241 _H	24.13.11
RDRMA	Receive data mirror register A	000242 _H	24.13.11
RDRMB	Receive data mirror register B	000243 _H	24.13.11
RSTCR	Reset control register	000481 _H	11.4.2
RSTRR	Reset trigger register	000480 _H	11.4.1

S

SBA0	Source base address register 0	000D08 _H	25.3.5
SBA1	Source base address register 1	000D28 _H	25.3.5
SBA2	Source base address register 2	000D48 _H	25.3.5
SBA3	Source base address register 3	000D68 _H	25.3.5
SCR0	Serial control register 0	000060 _H	24.4.1 , 24.13.1
SCR1	Serial control register 1	000070 _H	24.4.1 , 24.13.1
SCR2	Serial control register 2	000080 _H	24.4.1 , 24.13.1
SCR3	Serial control register 3	000090 _H	24.4.1 , 24.13.1
SCR4	Serial control register 4	0000A0 _H	24.4.1 , 24.13.1
SCR5	Serial control register 5	0000B0 _H	24.4.1 , 24.13.1
SCR6	Serial control register 6	0000C0 _H	24.4.1 , 24.13.1
SCR7	Serial control register 7	0000D0 _H	24.4.1 , 24.13.1
SCR8	Serial control register 8	000200 _H	24.4.1 , 24.13.1
SCR9	Serial control register 9	000210 _H	24.4.1 , 24.13.1
SCRA	Serial control register A	000220 _H	24.4.1 , 24.13.1
SCRB	Serial control register B	000230 _H	24.4.1 , 24.13.1
SDCMR	SDRAM command register	002110 _H	16.3.8
SDMR	SDRAM mode register	002100 _H	16.3.4
SDPDR	SDRAM power down counter register	002108 _H	16.3.6
SDRTR	SDRAM refresh timer register	002104 _H	16.3.5
SDTR	SDRAM timing register	00210C _H	16.3.7

MB91605A Series

SIX0	Secondary index register 0	000D14 _H	25.3.8
SIX1	Secondary index register 1	000D34 _H	25.3.8
SIX2	Secondary index register 2	000D54 _H	25.3.8
SIX3	Secondary index register 3	000D74 _H	25.3.8
SLPRR	Sleep rate configuration register	000483 _H	10.3.2
SMR0	Serial mode register 0	000061 _H	24.4.2 ,24.13.2 ,24.21.2
SMR1	Serial mode register 1	000071 _H	24.4.2 ,24.13.2 ,24.21.2
SMR2	Serial mode register 2	000081 _H	24.4.2 ,24.13.2 ,24.21.2
SMR3	Serial mode register 3	000091 _H	24.4.2 ,24.13.2 ,24.21.2
SMR4	Serial mode register 4	0000A1 _H	24.4.2 ,24.13.2 ,24.21.2
SMR5	Serial mode register 5	0000B1 _H	24.4.2 ,24.13.2 ,24.21.2
SMR6	Serial mode register 6	0000C1 _H	24.4.2 ,24.13.2 ,24.21.2
SMR7	Serial mode register 7	0000D1 _H	24.4.2 ,24.13.2 ,24.21.2
SMR8	Serial mode register 8	000201 _H	24.4.2 ,24.13.2 ,24.21.2
SMR9	Serial mode register 9	000211 _H	24.4.2 ,24.13.2 ,24.21.2
SMRA	Serial mode register A	000221 _H	24.4.2 ,24.13.2 ,24.21.2
SMRB	Serial mode register B	000231 _H	24.4.2 ,24.13.2 ,24.21.2
SSEL89AB	Serial mode select register for ch. 8/9/A/B	000244 _H	24.13.10
SSR0	Serial status register 0	000062 _H	24.4.3 ,24.13.3 ,24.21.4
SSR1	Serial status register 1	000072 _H	24.4.3 ,24.13.3 ,24.21.4
SSR2	Serial status register 2	000082 _H	24.4.3 ,24.13.3 ,24.21.4
SSR3	Serial status register 3	000092 _H	24.4.3 ,24.13.3 ,24.21.4
SSR4	Serial status register 4	0000A2 _H	24.4.3 ,24.13.3 ,24.21.4
SSR5	Serial status register 5	0000B2 _H	24.4.3 ,24.13.3 ,24.21.4
SSR6	Serial status register 6	0000C2 _H	24.4.3 ,24.13.3 ,24.21.4
SSR7	Serial status register 7	0000D2 _H	24.4.3 ,24.13.3 ,24.21.4
SSR8	Serial status register 8	000202 _H	24.4.3 ,24.13.3 ,24.21.4
SSR9	Serial status register 9	000212 _H	24.4.3 ,24.13.3 ,24.21.4
SSRA	Serial status register A	000222 _H	24.4.3 ,24.13.3 ,24.21.4
SSRB	Serial status register B	000232 _H	24.4.3 ,24.13.3 ,24.21.4
STBCR	Standby control register	000482 _H	10.3.1

T

TDR0	Send data register 0	000064 _H	24.4.5 ,24.13.5 ,24.21.5
TDR1	Send data register 1	000074 _H	24.4.5 ,24.13.5 ,24.21.5
TDR2	Send data register 2	000084 _H	24.4.5 ,24.13.5 ,24.21.5
TDR3	Send data register 3	000094 _H	24.4.5 ,24.13.5 ,24.21.5
TDR4	Send data register 4	0000A4 _H	24.4.5 ,24.13.5 ,24.21.5
TDR5	Send data register 5	0000B4 _H	24.4.5 ,24.13.5 ,24.21.5
TDR6	Send data register 6	0000C4 _H	24.4.5 ,24.13.5 ,24.21.5
TDR7	Send data register 7	0000D4 _H	24.4.5 ,24.13.5 ,24.21.5
TDR8	Send data register 8	000204 _H	24.4.5 ,24.13.5 ,24.21.5
TDR9	Send data register 9	000214 _H	24.4.5 ,24.13.5 ,24.21.5
TDRA	Send data register A	000224 _H	24.4.5 ,24.13.5 ,24.21.5
TDRB	Send data register B	000234 _H	24.4.5 ,24.13.5 ,24.21.5
TDRM8	Send data mirror register 8	000240 _H	24.13.11
TDRM9	Send data mirror register 9	000241 _H	24.13.11
TDRMA	Send data mirror register A	000242 _H	24.13.11
TDRMB	Send data mirror register B	000243 _H	24.13.11
TMCSR0	Control status register 0	00004E _H	20.4.1
TMCSR1	Control status register 1	000056 _H	20.4.1
TMCSR2	Control status register 2	00005E _H	20.4.1
TMR0	16 bit timer register 0	00004A _H	20.4.3
TMR1	17 bit timer register 0	000052 _H	20.4.3
TMR2	18 bit timer register 0	00005A _H	20.4.3
TMRLRA0	16 bit timer reload register A0	000048 _H	20.4.2
TMRLRA1	17 bit timer reload register A0	000050 _H	20.4.2
TMRLRA2	18 bit timer reload register A0	000058 _H	20.4.2

W

WDTCPR0	Watchdog timer 0 clear register	00003D _H	19.3.2
WDTCR0	Watchdog timer 0 control register	00003C _H	19.3.1

MB91605A Series

APPENDIX C 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	Interrupt number		Interrupt level	Offset	TBR default address	DMA transfer request
	Decimal	Hexadecimal				
Reset	0	00	—	3FC _H	000FFFFC _H	—
System reserved	1	01	—	3F8 _H	000FFFF8 _H	—
System reserved	2	02	—	3F4 _H	000FFFF4 _H	—
System reserved	3	03	—	3F0 _H	000FFFF0 _H	—
System reserved	4	04	—	3EC _H	000FFFE _C	—
System reserved	5	05	—	3E8 _H	000FFFE8 _H	—
System reserved	6	06	—	3E4 _H	000FFFE4 _H	—
System reserved	7	07	—	3E0 _H	000FFFE0 _H	—
System reserved	8	08	—	3DC _H	000FFFD _C	—
INTE instruction	9	09	—	3D8 _H	000FFFD8 _H	—
Instruction break exception	10	0A	—	3D4 _H	000FFFD4 _H	—
Operand break	11	0B	—	3D0 _H	000FFFD0 _H	—
Step trace trap	12	0C	—	3CC _H	000FFFC _C	—
NMI request (tool)	13	0D	—	3C8 _H	000FFFC8 _H	—
Undefined instruction exception	14	0E	—	3C4 _H	000FFFC4 _H	—
NMI request	15	0F	15 (F _H) fixed	3C0 _H	000FFFC0 _H	—
External interrupt ch.0	16	10	ICR00	3BC _H	000FFFB _C	○
External interrupt ch.1	17	11	ICR01	3B8 _H	000FFFB8 _H	○
External interrupt ch.2	18	12	ICR02	3B4 _H	000FFFB4 _H	○
External interrupt ch.3	19	13	ICR03	3B0 _H	000FFFB0 _H	○
External interrupt ch.4	20	14	ICR04	3AC _H	000FFFA _C	○
External interrupt ch.5	21	15	ICR05	3A8 _H	000FFFA8 _H	○

Interrupt source	Interrupt number		Interrupt level	Offset	TBR default address	DMA transfer request
	Decimal	Hexadecimal				
External interrupt ch.6	22	16	ICR06	3A4 _H	000FFFA4 _H	○
External interrupt ch.7	23	17	ICR07	3A0 _H	000FFFA0 _H	○
16-bit reload timer ch.0	24	18	ICR08	39C _H	000FFF9C _H	○
16-bit reload timer ch.1	25	19	ICR09	398 _H	000FFF98 _H	○
16-bit reload timer ch.2	26	1A	ICR10	394 _H	000FFF94 _H	○
Multi-function serial interface ch.0 RX	27	1B	ICR11	390 _H	000FFF90 _H	○
Multi-function serial interface ch.0 TX	28	1C	ICR12	38C _H	000FFF8C _H	○
Multi-function serial interface ch.0 I ² C status	29	1D	ICR13	388 _H	000FFF88 _H	—
Multi-function serial interface ch.1 RX	30	1E	ICR14	384 _H	000FFF84 _H	○
Multi-function serial interface ch.1 TX	31	1F	ICR15	380 _H	000FFF80 _H	○
Multi-function serial interface ch.1 I ² C status	32	20	ICR16	37C _H	000FFF7C _H	—
Multi-function serial interface ch.2 RX	33	21	ICR17	378 _H	000FFF78 _H	○
Multi-function serial interface ch.2 TX	34	22	ICR18	374 _H	000FFF74 _H	○
Multi-function serial interface ch.2 I ² C status	35	23	ICR19	370 _H	000FFF70 _H	—
Multi-function serial interface ch.3 RX/TX/I ² C status	36	24	ICR20	36C _H	000FFF6C _H	○*
Multi-function serial interface ch.4 RX/TX/I ² C status	37	25	ICR21	368 _H	000FFF68 _H	○*
Multi-function serial interface ch.5 RX/TX/I ² C status	38	26	ICR22	364 _H	000FFF64 _H	○*
A/D converter	39	27	ICR23	360 _H	000FFF60 _H	○
HDMI-CEC/Remote control	40	28	ICR24	35C _H	000FFF5C _H	—

MB91605A Series

Interrupt source	Interrupt number		Interrupt level	Offset	TBR default address	DMA transfer request
	Decimal	Hexadecimal				
External interrupt 8-15	41	29	ICR25	358 _H	000FFF58 _H	○
External interrupt 16-23	42	2A	ICR26	354 _H	000FFF54 _H	○
Multi-function serial interface ch.6 RX/TX/I ² C status	43	2B	ICR27	350 _H	000FFF50 _H	○*
Multi-function serial interface ch.7 RX/TX/I ² C status	44	2C	ICR28	34C _H	000FFF4C _H	○*
Multi-function serial interface ch.8 RX/TX/I ² C status	45	2D	ICR29	348 _H	000FFF48 _H	○*
Multi-function serial interface ch.9 RX/TX/I ² C status	46	2E	ICR30	344 _H	000FFF44 _H	○*
Multi-function serial interface ch.10 RX/TX/I ² C status	47	2F	ICR31	340 _H	000FFF40 _H	○*
Multi-function serial interface ch.11 RX/TX/I ² C status	48	30	ICR32	33C _H	000FFF3C _H	○*
Base timer ch.0	49	31	ICR33	338 _H	000FFF38 _H	○
Base timer ch.1	50	32	ICR34	334 _H	000FFF34 _H	○
Base timer ch.2	51	33	ICR35	330 _H	000FFF30 _H	○
Base timer ch.3	52	34	ICR36	32C _H	000FFF2C _H	○
Base timer ch.4	53	35	ICR37	328 _H	000FFF28 _H	○
Base timer ch.5	54	36	ICR38	324 _H	000FFF24 _H	○
Base timer ch.6	55	37	ICR39	320 _H	000FFF20 _H	○
Base timer ch.7	56	38	ICR40	31C _H	000FFF1C _H	○
Base timer ch.8/ch.9/ch.10/ch.11	57	39	ICR41	318 _H	000FFF18 _H	○
DMAC ch.0	58	3A	ICR42	314 _H	000FFF14 _H	—
DMAC ch.1	59	3B	ICR43	310 _H	000FFF10 _H	—
DMAC ch.2	60	3C	ICR44	30C _H	000FFF0C _H	—
DMAC ch.3	61	3D	ICR45	308 _H	000FFF08 _H	—
Timebase timer	62	3E	ICR46	304 _H	000FFF04 _H	—

Interrupt source	Interrupt number		Interrupt level	Offset	TBR default address	DMA transfer request
	Decimal	Hexadecimal				
Delay interrupt	63	3F	ICR47	300 _H	000FFF00 _H	—
System reserved (Used by REALOS)	64	40	—	2FC _H	000FFEFC _H	—
System reserved (Used by REALOS)	65	41	—	2F8 _H	000FEF8 _H	—
Used by INT instruction	66 to 255	42 to FF	—	2F4 _H to 000 _H	000FEF4 _H to 000FFC00 _H	—

* : The I²C status interrupt cannot be used for DMAC transfer requests.

APPENDIX D 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.

Table D-1 Pin state in external bus 16-bit mode (1 / 6)

Pin Number	Port Name	Specified Function Name	When initialized (INIT=0)		SLEEP	STOP	
			Function Name	Initial Value		HIZ=0	HIZ=1
3	P45	MCLK	MCLK	"L" output	P: Maintain last state F: "L" output	P: Maintain last state F: "L" output	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
4	--	A0	A0	Output undefined	Address output	Address output	Output Hi-Z
5	--	A1	A1				
6	--	A2	A2				
7	--	A3	A3				
8	--	A4	A4				
9	--	A5	A5				
10	--	A6	A6				
11	--	A7	A7				
14	--	A8	A8	Output undefined	Address output	Address output	Output Hi-Z
15	--	A9	A9				
16	--	A10	A10				
17	--	A11	A11				
18	--	A12	A12				
19	--	A13	A13				
20	--	A14	A14				
21	--	A15	A15				
25	P50	A16	A16	Output undefined	P: Maintain last state F: Address output	P: Maintain last state F: Address output	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
26	P51	A17	A17				
27	P52	A18	A18				
28	P53	A19	A19				
29	P54	A20	A20				
30	P55	A21	A21				
31	P56	A22	A22				
32	P57	A23	A23				

P: In the case of port connecting F: In the case of specified function using

MB91605A Series**Table D-1 Pin state in external bus 16-bit mode (2 / 6)**

Pin Number	Port Name	Specified Function Name	When initialized (INIT=0)		SLEEP	STOP	
			Function Name	Initial Value		HIZ=0	HIZ=1
57	P60	AN0	AN0	Analog input disabled	P: Maintain last state F: Normal operation	P: Maintain last state F: Maintain last state	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
58	P61	AN1	AN1				
59	P62	AN2/SIN10	AN2				
60	P63	AN3/ SOUT10/ SDA10	AN3				
61	P64	AN4/SCK10/ SCL10	AN4				
62	P65	AN5/SIN11	AN5				
63	P66	AN6/ SOUT11/ SDA11	AN6				
64	P67	AN7/SCK11/ SCL11	AN7				
65	P70	AN8/TIOA10	AN8				
66	P71	AN9/TIOB10	AN9				
67	P72	AN10/ TIOA11	AN10				
68	P73	AN11/ TIOB11	AN11				
71	P74	RCIN	P74	Output Hi-Z Input enabled	P: Maintain last state F: Normal operation	P: Maintain last state F: Maintain last state	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
72	P75	SIN0	P75				
73	P76	SOUT0/SDA0	P76				
74	P77	SCK0/SCL0	P77				
75	P80	SIN1	P80				
76	P81	SOUT1/SDA1	P81				
77	P82	SCK1/SCL1	P82				
78	P83	SIN2	P83				
79	P84	SOUT2/SDA2	P84				
80	P85	SCK2/SCL2	P85				

* Only SIN0, SOUT0, and SCK0 pins are different in FLASH serial write mode.

P: In the case of port connecting F: In the case of specified function using

Table D-1 Pin state in external bus 16-bit mode (3 / 6)

Pin Number	Port Name	Specified Function Name	When initialized ($\overline{\text{INIT}}=0$)		SLEEP	STOP	
			Function Name	Initial Value		HIZ=0	HIZ=1
81	P90	SIN3	P90	Output Hi-Z Input enabled	P: Maintain last state F: Normal operation	P: Maintain last state F: Maintain last state	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
82	P91	SOUT3/SDA3	P91				
83	P92	SCK3/SCL3	P92				
84	P93	SIN4	P93				
85	P94	SOUT4/SDA4	P94				
86	P95	SCK4/SCL4	P95				
87	--	$\overline{\text{NMI}}$	$\overline{\text{NMI}}$	Input enabled	Input enabled	Input enabled	Input enabled
91	PA0	SIN5	PA0	Output Hi-Z Input enabled	P: Maintain last state F: Normal operation	P: Maintain last state F: Maintain last state	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
92	PA1	SOUT5/SDA5	PA1				
93	PA2	SCK5/SCL5	PA2				
94	PA3	SIN6/ATR $\overline{\text{G}}$	PA3				
95	PA4	SOUT6/SDA6	PA4				
96	PA5	SCK6/SCL6	PA5				
97	PB0	INT0/SIN7	PB0				
98	PB1	INT1/SOUT7/ SDA7	PB1				
99	PB2	INT2/SCK7/ SCL7	PB2				
100	PB3	INT3/TOUT0	PB3				
101	PB4	INT4/TOUT1	PB4				
102	PB5	INT5/TOUT2	PB5				
103	PB6	INT6/TIOA8	PB6				
104	PB7	INT7/TIOB8	PB7				
105	PC0	INT8/TIOA9	PC0				P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0" (Input enabled if external interrupt trigger is enabled)
106	PC1	INT9/TIOB9	PC1				
107	PC2	INT10/SIN8	PC2				
108	PC3	INT11/ SOUT8/SDA8	PC3				

P: In the case of port connecting F: In the case of specified function using

MB91605A Series**Table D-1 Pin state in external bus 16-bit mode (4 / 6)**

Pin Number	Port Name	Specified Function Name	When initialized (INIT=0)		SLEEP	STOP	
			Function Name	Initial Value		HIZ=0	HIZ=1
109	PC4	INT12/SCK8/SCL8	PC4	Output Hi-Z Input enabled	P: Maintain last state F: Normal operation	P: Maintain last state F: Maintain last state	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0" (Input enabled if external interrupt trigger is enabled)
110	PC5	INT13/SIN9	PC5				
111	PC6	INT14/SOUT9/SDA9	PC6				
112	PC7	INT15/SCK9/SCL9	PC7				
116	PD0	TIOA0	PD0	Output Hi-Z Input enabled	P: Maintain last state F: Normal operation	P: Maintain last state F: Maintain last state	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
117	PD1	TIOB0	PD1				
118	PD2	TIOA1	PD2				
119	PD3	TIOB1	PD3				
120	PD4	TIOA2	PD4				
121	PD5	TIOB2	PD5				
122	PD6	TIOA3	PD6				
123	PD7	TIOB3	PD7				
124	PE0	INT16/TIOA4	PE0				P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0" (Input enabled if external interrupt trigger is enabled)
125	PE1	INT17/TIOB4	PE1				
126	PE2	INT18/TIOA5	PE2				
127	PE3	INT19/TIOB5	PE3				
128	PE4	INT20/TIOA6	PE4				
129	PE5	INT21/TIOB6	PE5				
130	PE6	INT22/TIOA7	PE6				
131	PE7	INT23/TIOB7	PE7				

P: In the case of port connecting F: In the case of specified function using

Table D-1 Pin state in external bus 16-bit mode (5 / 6)

Pin Number	Port Name	Specified Function Name	When initialized (INIT=0)		SLEEP	STOP	
			Function Name	Initial Value		HIZ=0	HIZ=1
135	--	D16	D16	Output Hi-Z Input enabled	Maintain output or Hi-Z	Maintain output or Hi-Z	Output Hi-Z/ input fixed to "0"
136	--	D17	D17				
137	--	D18	D18				
138	--	D19	D19				
139	--	D20	D20				
140	--	D21	D21				
141	--	D22	D22				
142	--	D23	D23	Output Hi-Z Input enabled	Maintain output or Hi-Z	Maintain output or Hi-Z	Output Hi-Z/ input fixed to "0"
145	--	D24	D24				
146	--	D25	D25				
147	--	D26	D26				
148	--	D27	D27				
149	--	D28	D28				
150	--	D29	D29				
151	--	D30	D30	Output Hi-Z Input enabled	P: Maintain last state F: SYSCLK output	P: Maintain last state F: "H" or "L" output	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
152	--	D31	D31				
155	P20	SYSCLK	P20	Output Hi-Z Input enabled	P: Maintain last state F: SYSCLK output	P: Maintain last state F: "H" or "L" output	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
156	--	\overline{RD}	\overline{RD}	"H" output	"H" output	"H" output	Output Hi-Z
157	P21	\overline{WE}	\overline{WE}		P: Maintain last state F: "H" output	P: Maintain last state F: "H" output	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
158	P22	MDQM3	MDQM3				
159	P23	MDQM2	MDQM2				
160	P30	$\overline{CS0}$	$\overline{CS0}$				
161	P31	$\overline{CS1}$	$\overline{CS1}$				
162	P32	$\overline{CS2}$	$\overline{CS2}$				
163	P33	$\overline{CS3}$	$\overline{CS3}$				
164	--	$\overline{CS4}$	$\overline{CS4}$	"H" output	"H" output	"H" output	Output Hi-Z

P: In the case of port connecting F: In the case of specified function using

MB91605A Series**Table D-1 Pin state in external bus 16-bit mode (6 / 6)**

Pin Number	Port Name	Specified Function Name	When initialized (INIT=0)		SLEEP	STOP	
			Function Name	Initial Value		HIZ=0	HIZ=1
167	P34	$\overline{AS}/TIN0$	P34	Output Hi-Z Input enabled	P: Maintain last state F: "H" output	P: Maintain last state F: "H" output	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
168	P35	$\overline{CS5}/TIN1$	P35				
169	P36	$\overline{CS6}/TIN2$	P36				
170	P37	$\overline{CS8}$	$\overline{CS8}$	"H" output	P: Maintain last state F: "H" output	P: Maintain last state F: "H" output	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
171	P40	RDY	P40	Output Hi-Z Input enabled	P: Maintain last state F: RDY input	P: Maintain last state F: Maintain last state	P: Output Hi-Z/ input fixed to "0" F: Output Hi-Z/ input fixed to "0"
172	P41	\overline{MRAS}	P41		P: Maintain last state F: "H" output	P: Maintain last state F: "H" output	
173	P42	\overline{MCAS}	P42				
174	P43	\overline{MDWE}	P43				
175	P44	MCLKE	MCLKE	"H" output	P: Maintain last state F: Maintain last state	P: Maintain last state F: Maintain last state	

P: In the case of port connecting F: In the case of specified function using

APPENDIX E Lists of Instructions

This section lists and maps instructions for the FR80 family CPUs.

E.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.

Flag change	Meaning of flag
C : Change	N : Negative flag
- : No change	Z : Zero flag
0 : Clear	V : Overflow flag
1 : Set	C : Carry flag

(7) O is applied for RMW type of instructions.

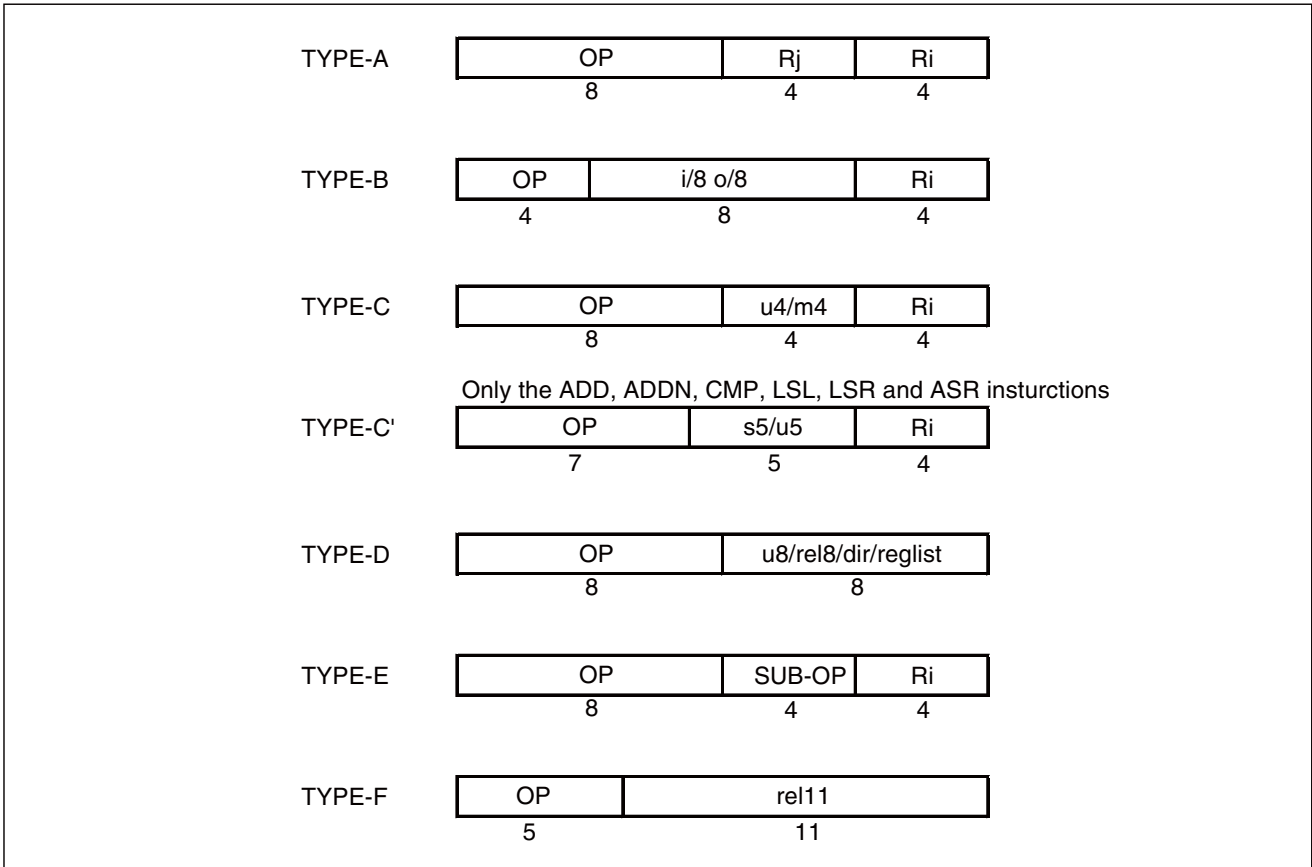
(8) The instruction operation is written.

■ 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 0XFFFFFF) Attention: -0X7FFFF to -1 is treated as 0X7FFFF to 0XFFFFFF.
#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.

E.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 E-1 Add-subtract instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ADD Rj, Ri	A	A6	1	CCCC	-	$Ri + Rj \rightarrow Ri$	
*ADD #s5, Ri	C'	-	1	CCCC	-	$Ri + s5 \rightarrow Ri$	The upper 1 bit of s5 is considered as a sign in the assembler.
ADD #i4, Ri	C	A4	1	CCCC	-	$Ri + \text{extu}(i4) \rightarrow Ri$	i4 is a zero extension.
ADD2 #i4, Ri	C	A5	1	CCCC	-	$Ri + \text{extn}(i4) \rightarrow Ri$	i4 is a negative extension.
ADDC Rj, Ri	A	A7	1	CCCC	-	$Ri + Rj + C \rightarrow Ri$	Addition with a carry
ADDN Rj, Ri	A	A2	1	----	-	$Ri + Rj \rightarrow Ri$	
*ADDN #s5, Ri	C'	-	1	----	-	$Ri + s5 \rightarrow Ri$	The upper 1 bit of s5 is considered as a sign in the assembler.
ADDN #i4, Ri	C	A0	1	----	-	$Ri + \text{extu}(i4) \rightarrow Ri$	i4 is a zero extension.
ADDN2 #i4, Ri	C	A1	1	----	-	$Ri + \text{extn}(i4) \rightarrow Ri$	i4 is a negative extension.
SUB Rj, Ri	A	AC	1	CCCC	-	$Ri - Rj \rightarrow Ri$	
SUBC Rj, Ri	A	AD	1	CCCC	-	$Ri - Rj - C \rightarrow Ri$	Subtraction with a carry
SUBN Rj, Ri	A	AE	1	----	-	$Ri - Rj \rightarrow Ri$	

Table E-2 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	-	$Ri - \text{extu}(i4)$	i4 is a zero extension.
CMP2 #i4, Ri	C	A9	1	CCCC	-	$Ri - \text{extn}(i4)$	i4 is a negative extension.

Table E-3 Logical operation instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
AND Rj, Ri	A	82	1	CC--	-	$Ri \& Rj \rightarrow Ri$	Word
AND Rj, @Ri	A	84	1 + 2a	CC--	O	$(Ri) \& Rj \rightarrow (Ri)$	Word
ANDH Rj, @Ri	A	85	1 + 2a	CC--	O	$(Ri) \& Rj \rightarrow (Ri)$	Half word
ANDB Rj, @Ri	A	86	1 + 2a	CC--	O	$(Ri) \& Rj \rightarrow (Ri)$	Byte
OR Rj, Ri	A	92	1	CC--	-	$Ri Rj \rightarrow Ri$	Word
OR Rj, @Ri	A	94	1 + 2a	CC--	O	$(Ri) Rj \rightarrow (Ri)$	Word
ORH Rj, @Ri	A	95	1 + 2a	CC--	O	$(Ri) Rj \rightarrow (Ri)$	Half word
ORB Rj, @Ri	A	96	1 + 2a	CC--	O	$(Ri) Rj \rightarrow (Ri)$	Byte
EOR Rj, Ri	A	9A	1	CC--	-	$Ri \wedge Rj \rightarrow Ri$	Word
EOR Rj, @Ri	A	9C	1 + 2a	CC--	O	$(Ri) \wedge Rj \rightarrow (Ri)$	Word
EORH Rj, @Ri	A	9D	1 + 2a	CC--	O	$(Ri) \wedge Rj \rightarrow (Ri)$	Half word
EORB Rj, @Ri	A	9E	1 + 2a	CC--	O	$(Ri) \wedge Rj \rightarrow (Ri)$	Byte

Table E-4 Bit operation instructions (1 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
BANDL #u4, @Ri	C	80	1 + 2a	----	O	$(Ri) \& \{F0_H + u4\} \rightarrow (Ri)$	Lower 4 bits
BANDH #u4, @Ri	C	81	1 + 2a	----	O	$(Ri) \& \{u4 << 4 + 0F_H\} \rightarrow (Ri)$	Higher 4 bits
*BAND #u8, @Ri ^{*1}	-	-	-	----	O	$(Ri) \&= u8$	
BORL #u4, @Ri	C	90	1 + 2a	----	O	$(Ri) u4 \rightarrow (Ri)$	Lower 4 bits
BORH #u4, @Ri	C	91	1 + 2a	----	O	$(Ri) \{u4 << 4\} \rightarrow (Ri)$	Higher 4 bits
*BOR #u8, @Ri ^{*2}	-	-	-	----	O	$(Ri) = u8$	
BEORL #u4, @Ri	C	98	1 + 2a	----	O	$(Ri) \wedge u4 \rightarrow (Ri)$	Lower 4 bits
BEORH #u4, @Ri	C	99	1 + 2a	----	O	$(Ri) \wedge \{u4 << 4\} \rightarrow (Ri)$	Higher 4 bits

Table E-4 Bit operation instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
*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 E-5 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 E-6 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 E-7 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 E-8 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
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 E-9 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 E-10 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 E-11 Non-delayed branch instructions (1 / 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
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

Table E-11 Non-delayed branch instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
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/TYPER-F rel11 fields and label9/label12 in assembler code is as follows:

$$\text{rel8} = (\text{label9} - \text{PC} - 2) / 2$$

$$\text{rel11} = (\text{label12} - \text{PC} - 2) / 2$$

Table E-12 Delayed branch instructions (1 / 2)

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

Table E-12 Delayed branch instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
BGE:D label9	D	FB	1	----	-	if ($V \wedge N == 0$) then $PC + 2 + \text{exts}(\text{rel8} \times 2) \rightarrow PC$
BLE:D label9	D	FC	1	----	-	if ($\{V \wedge N\} \mid Z == 1$) then $PC + 2 + \text{exts}(\text{rel8} \times 2) \rightarrow PC$
BGT:D label9	D	FD	1	----	-	if ($\{V \wedge N\} \mid Z == 0$) then $PC + 2 + \text{exts}(\text{rel8} \times 2) \rightarrow PC$
BLS:D label9	D	FE	1	----	-	if ($C \text{ or } Z == 1$) then $PC + 2 + \text{exts}(\text{rel8} \times 2) \rightarrow PC$
BHI:D label9	D	FF	1	----	-	if ($C \text{ or } Z == 0$) then $PC + 2 + \text{exts}(\text{rel8} \times 2) \rightarrow PC$

- 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:

$$\text{rel8} = (\text{label9} - PC - 2) / 2$$

$$\text{rel11} = (\text{label12} - PC - 2) / 2$$

Table E-13 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 E-14 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 E-15 Other instructions (1 / 2)

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

Table E-15 Other instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ENTER #u10	D	0F	1 + a	----	-	R14 \rightarrow (R15-4), R15 - 4 \rightarrow R14, R15-extu (u8 x 4) \rightarrow R15	Function entry processing
LEAVE	E	9F - 9	b	----	-	R14 + 4 \rightarrow R15, (R15-4) \rightarrow R14	Function exit processing
XCHB @Rj, Ri	A	8A	2a	----	O	Ri \rightarrow TEMP, extu((Rj)) \rightarrow Ri, TEMP \rightarrow (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:

$$s8 = s10 \gg 2$$

- In the ENTER instruction, the relationship between the instruction format TYPE-D u8 field and u10 in assembler code is as follows:

$$u8 = u10 \gg 2$$

E.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 #i4, 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		

APPENDIX F Program Loader Mode

This mode operates as follows;

- (1) The program loader that is stored in built-in ROM communicates with the serial to outside by multifunction serial interface ch.0.
- (2) The program is loaded from the outside into built-in RAM (128K byte).
- (3) The loaded program is started.

Serial communications can select the asynchronous communication or synchronous communications by the state of MD1, MD0, SIN0/P75 and SOUT0/P76 when initialized by $\overline{\text{INIT}}$. Please refer to "■Setting Method" for the pin setting.

■ Setting Method

The program loader stored in built-in ROM starts when the state of MD1, MD0, SIN0/P75 and SOUT0/P76 pin is set as shown in the following table at $\overline{\text{INIT}}$ initializing.

The program loader is determined by MD1, MD0, and SOUT0/P76.

Whether it is an asynchronous communication or synchronous communications of multifunction serial interface ch.0 used for serial communications to the outside are determined by SIN0/P75.

The source oscillation frequency is 16 MHz.

● Pin setting (source oscillation frequency: 16MHz)

Specification	Pin name			
	MD1	MD0	SIN0/P75	SOUT0/P76
Asynchronous communication	0	1	1	1
Synchronous communication	0	1	0	1

■ Detail

- Asynchronous communication to Source oscillation frequency (16 MHz)

Serial communication is doing in the asynchronous mode of multifunction serial interface ch.0 (mode 0).

The baud rate becomes 9600bps.

The serial setting is data length: 8-bit, the stop bit lengths: 1-bit, with no parity, and LSB first.

- Synchronous communication

Serial communication is done in the synchronous mode of multifunction serial interface ch.0 (mode 2).

The selection of the baud rate can be set by clock input (SCK0) without restraint. (The frequency of clock input SCK0 is set to baud rate.)

The maximum frequency of clock input is 2MHz.

The serial setting is data length: 8-bit, with no parity, and LSB first.

Each mode also sequentially gives the following 3 download information data to the FR side from upper byte by one byte. In addition, it enters the download routine to RAM by giving these SUM check data (lower eight bits are taken out after adding all data). Next, the data that should be downloaded to built-in RAM is sequentially given to the FR side from upper byte by one byte, and the SUM check data is given. After transferring ends, the program jumped to RAM and was downloaded is executed.

- Command data (00_H)
- Download destination RAM address 4 byte (00020000_H to 0003FEFF_H)
(using prohibited: 0003FF00_H to 0003FFFF_H)
- Download byte number: 4 bytes (Max 0001FEFF_H)

Example: at transferring the "0000005B_H" data to RAM address "00020000_H"

Table F-1 Handshake at downloading

		PC etc.		FR
Command data	(1)	00 _H	→	
Download destination Address	(2)	00 _H	→	
	(3)	02 _H	→	
	(4)	00 _H	→	
	(5)	00 _H	→	
Download byte number (91 bytes)	(6)	00 _H	→	
	(7)	00 _H	→	
	(8)	00 _H	→	
	(9)	5B _H	→	
SUM check data	(10)	5E _H	→	
Acknowledge from FR	(11)		←	01 _H
Data sending	(12)	DATA	→	
SUM check data	(13)	*	→	
Acknowledge from FR	(14)		←	01 _H

*: Lower eight bits are taken out after adding all sending data

Table F-2 Handshake at jumping to RAM

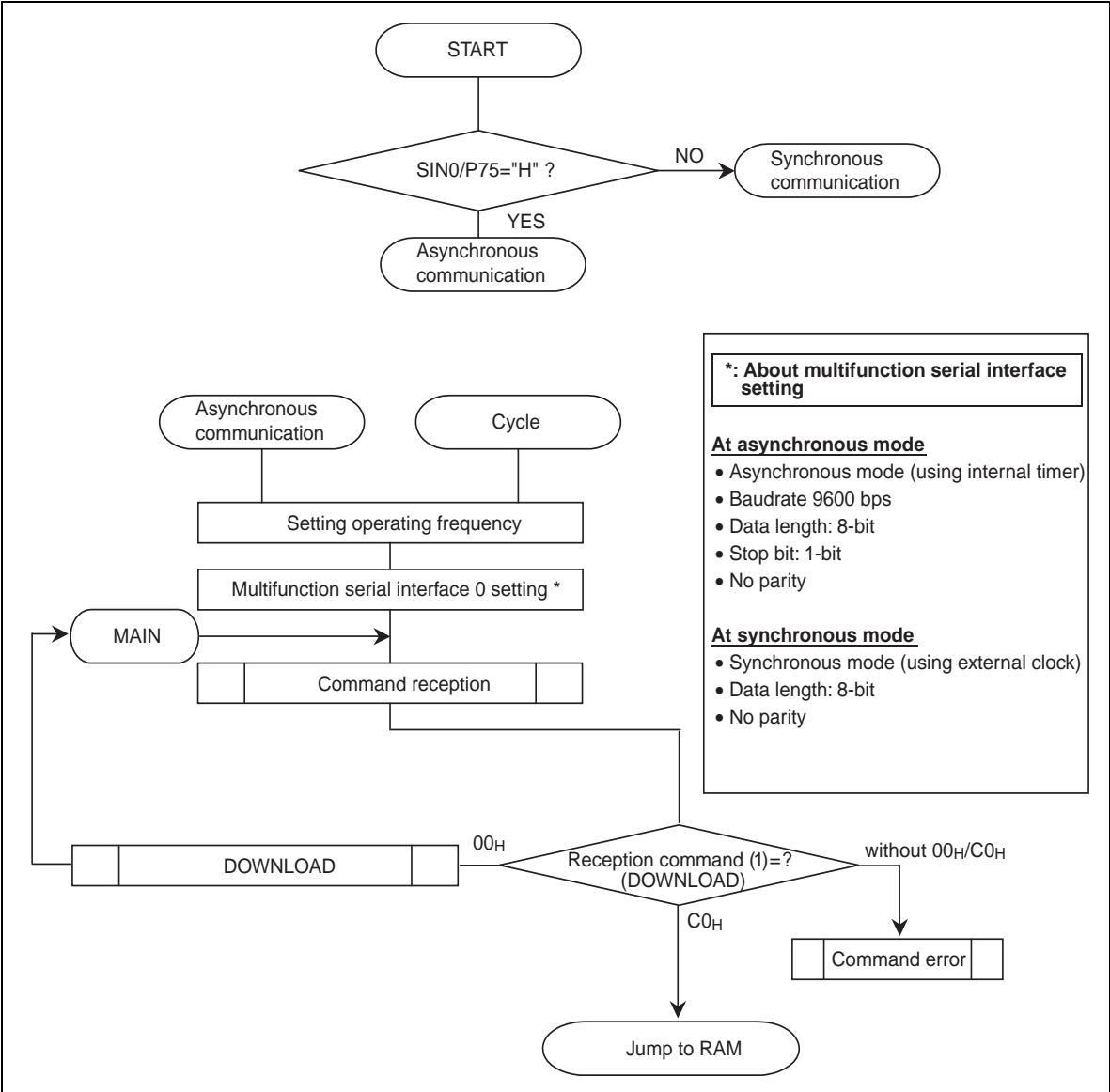
		PC etc.		FR
Command data	(1)	C0 _H	→	
Dummy data	(2)	00 _H	→	
	(3)	00 _H	→	
	(4)	00 _H	→	
	(5)	00 _H	→	
Dummy data	(6)	00 _H	→	
	(7)	00 _H	→	
	(8)	00 _H	→	
	(9)	00 _H	→	
SUM check data	(10)	C0 _H	→	
Jump to RAM	(11)			

Please refer to the flowchart of the program in special ROM since it the next paragraphs for a detailed operating.

For detailed operating and the all pins state etc. of multifunction serial interface, please refer to the column of "When initialized ($\overline{\text{INIT}}=0$)" of Table D-1 or "CHAPTER 24 Multi-function Serial Interface".

- Main program flowchart

Figure F-1 Main Program Flowchart

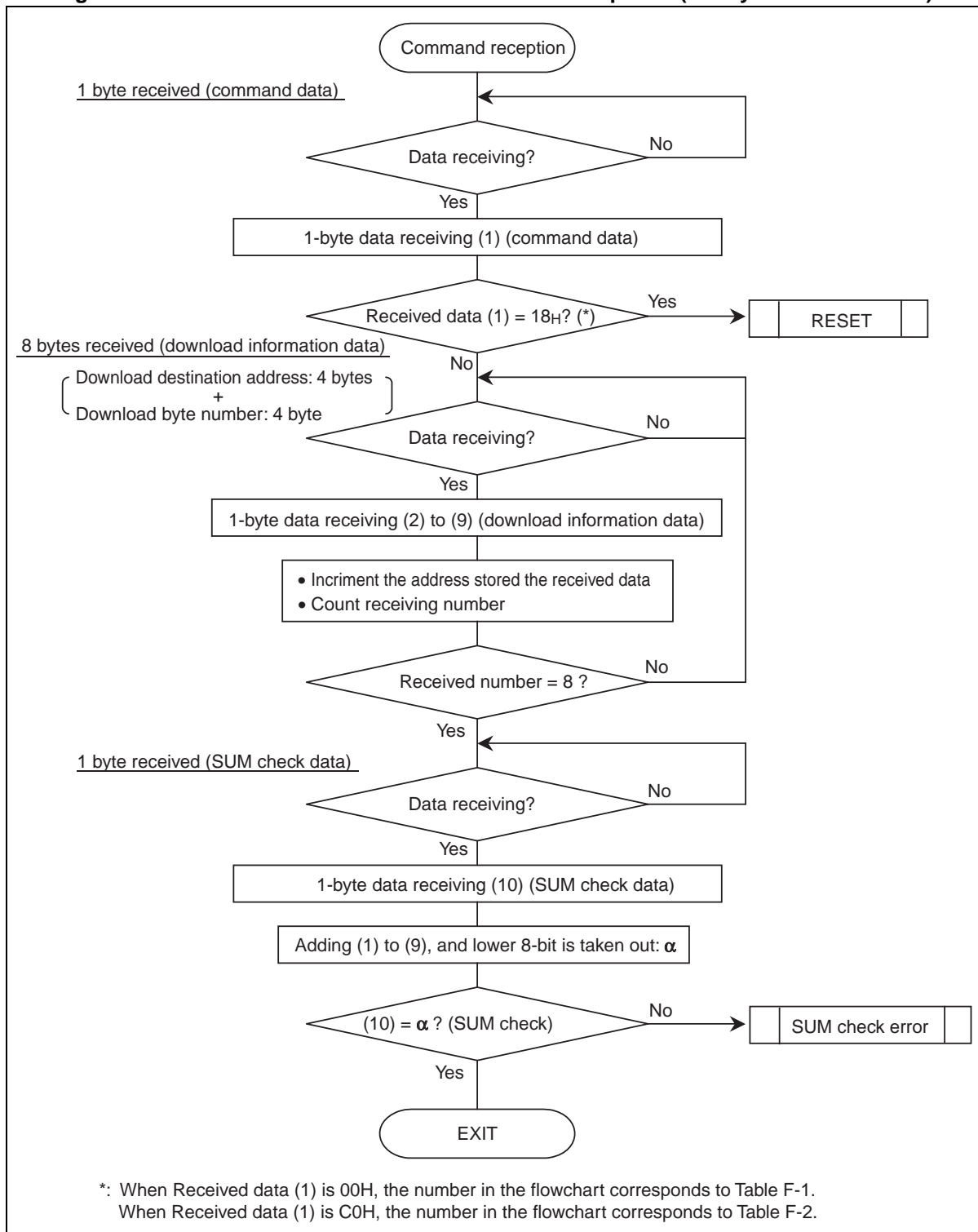


The list of the command issued to the FR side and the reply signal from the FR side is shown below.

		FR		PC etc.
Command	Download	-	←	00 _H
	Reset	-	←	18 _H
	RAM jump	-	←	C0 _H
Command reception response	Command error	(Received command & F0 _H) 04 _H	→	-
	SUM check error	(Received command (00 _H) & F0 _H)	→	-
	RESET command reception	11 _H	→	-
	DOWNLOAD command reception	01 _H	→	-

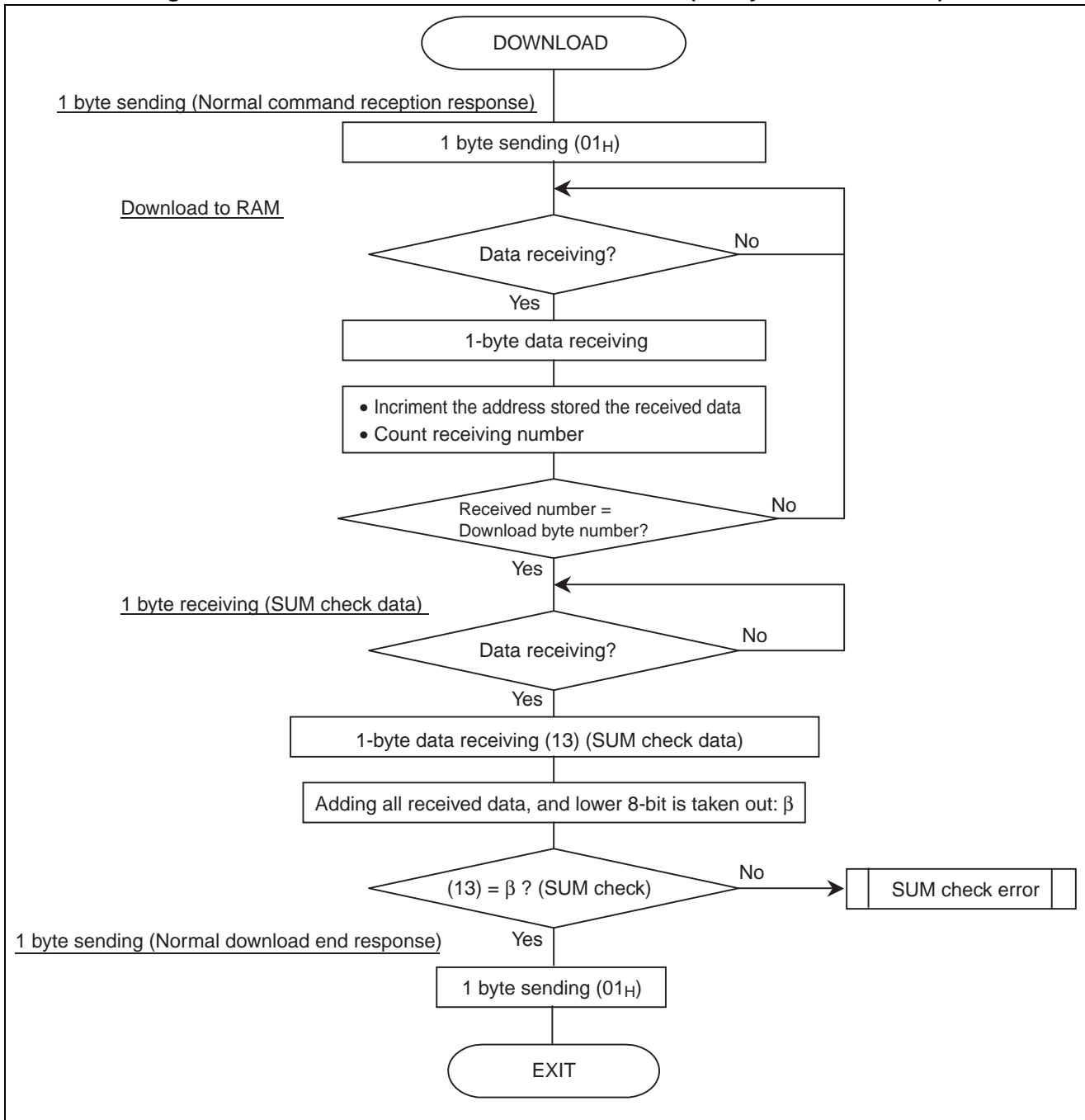
- Flowchart of sub-routine "Command reception" (at asynchronous mode)

Figure F-2 Flowchart of Sub-routine "Command Reception" (at Asynchronous Mode)



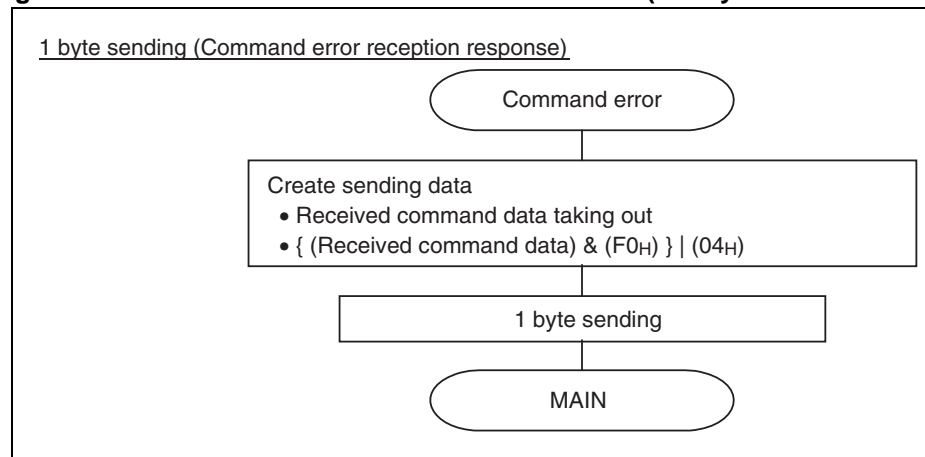
- Flowchart of sub-routine "DOWNLOAD" (at asynchronous mode)

Figure F-3 Flowchart of Sub-routine "DOWNLOAD" (at Asynchronous Mode)



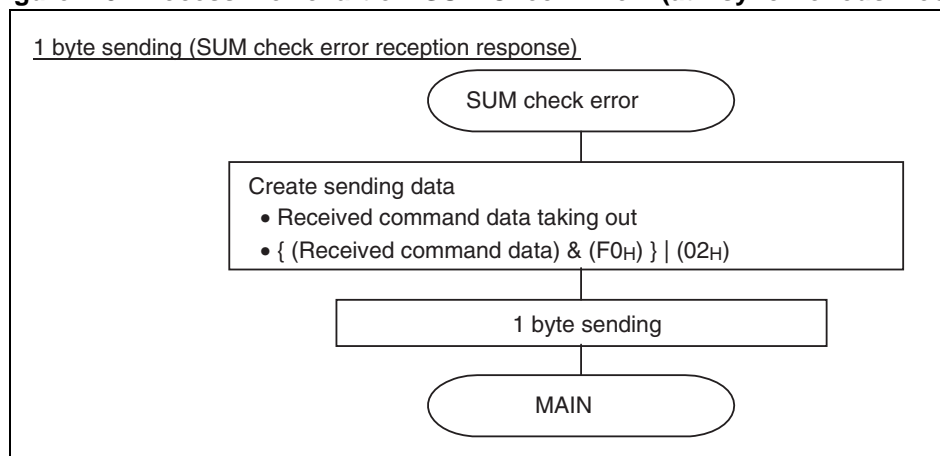
- Process flowchart of "Command error" (at asynchronous mode)

Figure F-4 Process Flowchart of "Command Error" (at Asynchronous Mode)



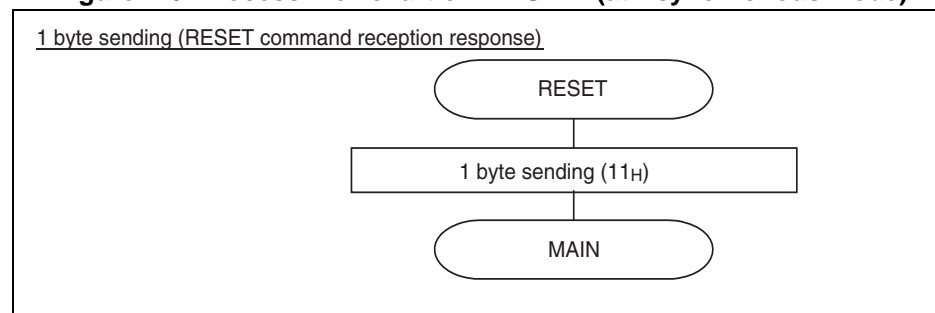
- Process flowchart of "SUM check error" (at asynchronous mode)

Figure F-5 Process Flowchart of "SUM Check Error" (at Asynchronous Mode)



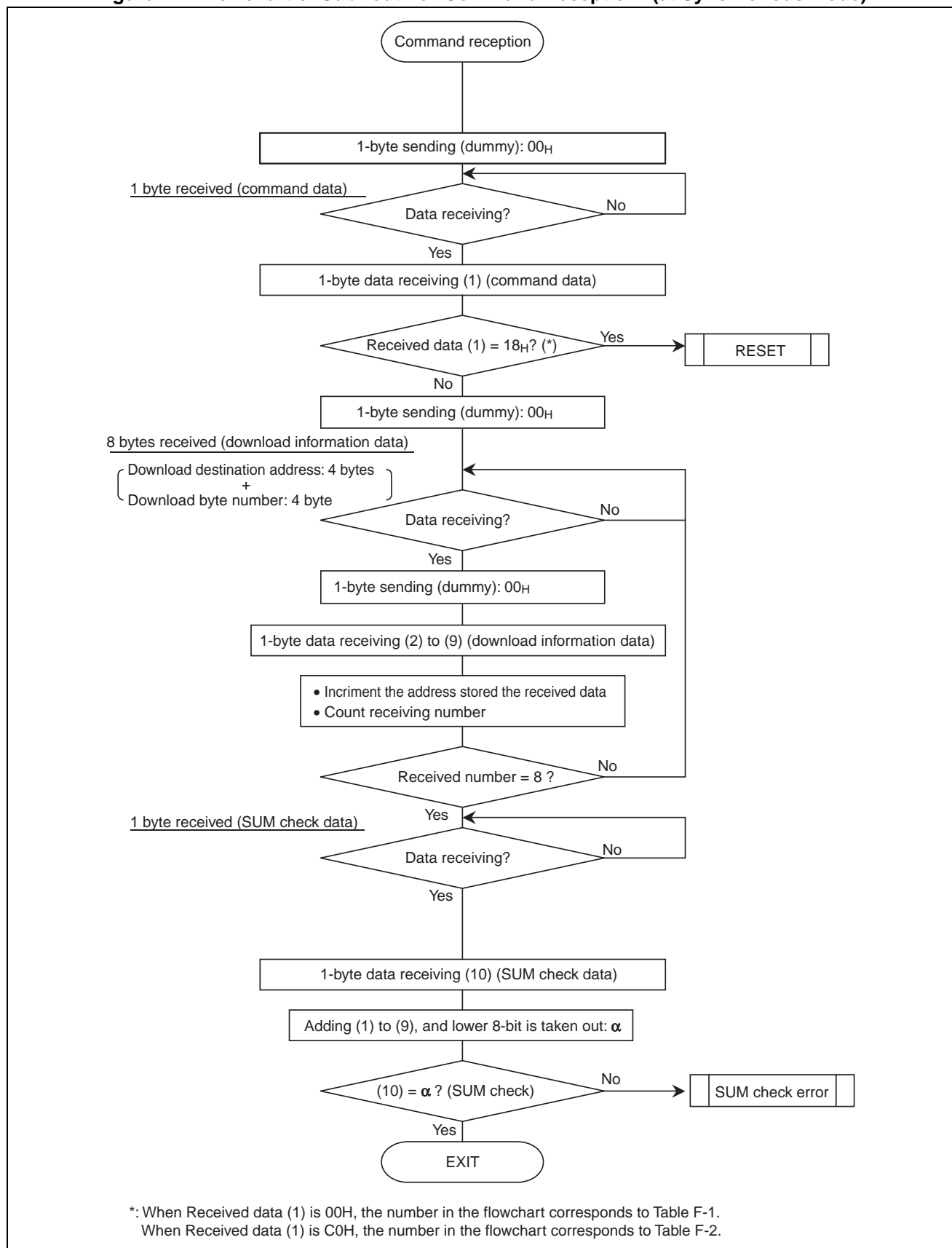
- Process flowchart of "RESET" (at asynchronous mode)

Figure F-6 Process Flowchart of "RESET" (at Asynchronous Mode)



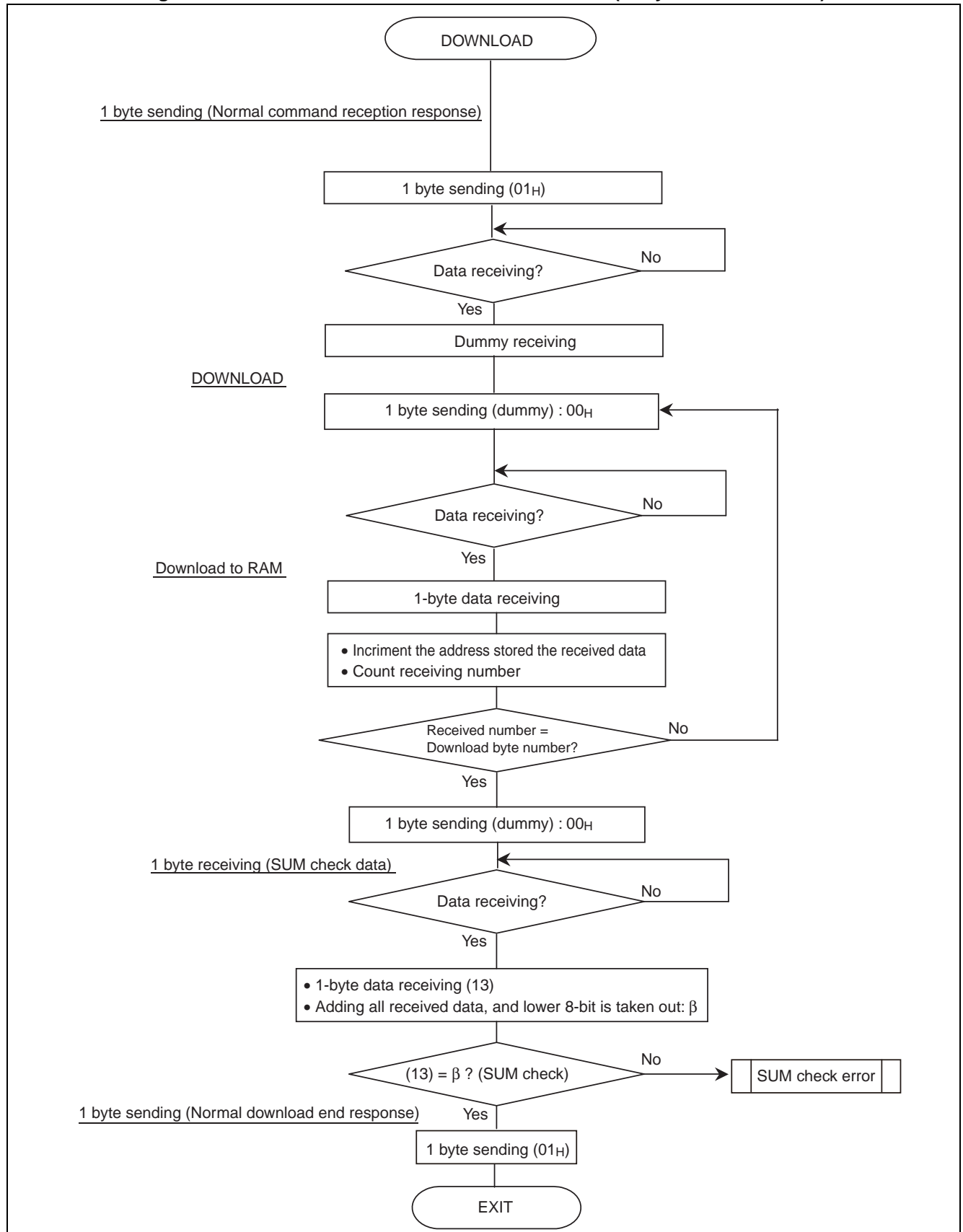
- Flowchart of sub-routine "Command reception" (at synchronous mode)

Figure F-7 Flowchart of Sub-routine "Command Reception" (at Synchronous Mode)



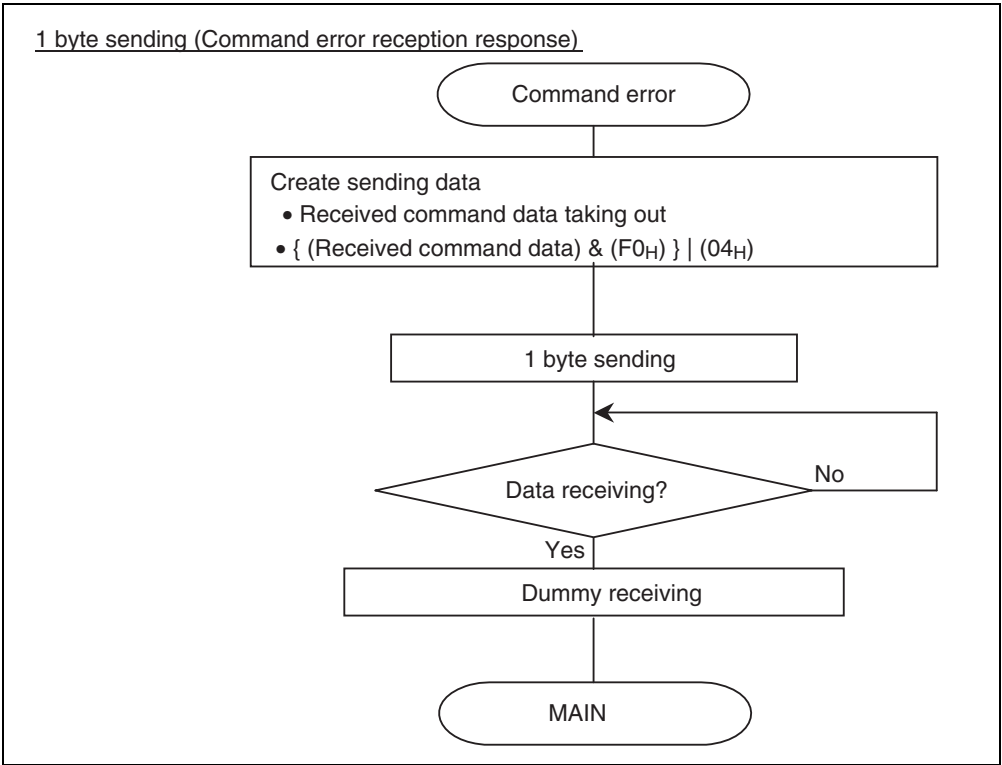
- Flowchart of sub-routine "DOWNLOAD" (at synchronous mode)

Figure F-8 Flowchart of Sub-routine "DOWNLOAD" (at Synchronous Mode)



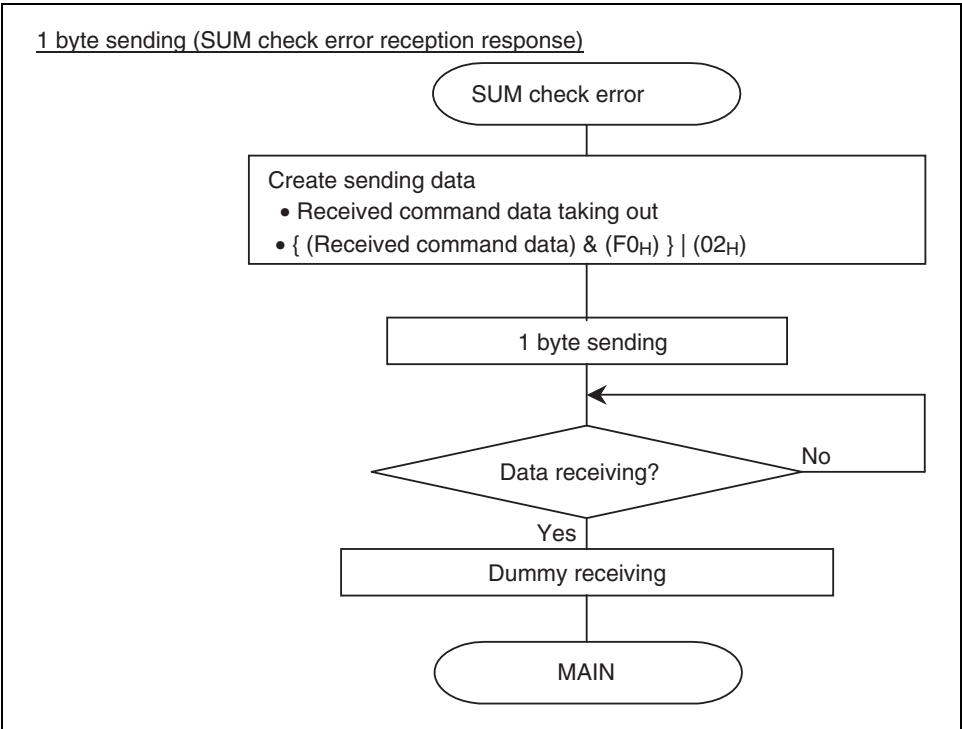
- Process flowchart of "Command error" (at synchronous mode)

Figure F-9 Process Flowchart of "Command Error" (at Synchronous Mode)



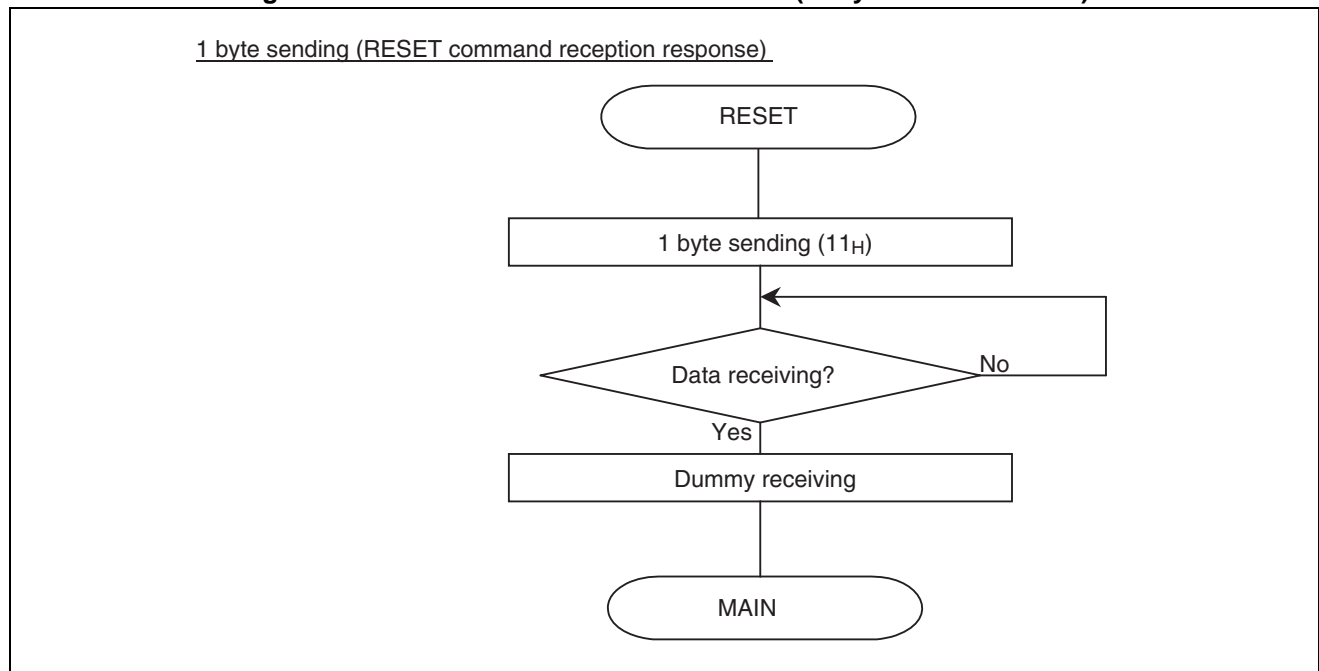
- Process flowchart of "SUM check error" (at synchronous mode)

Figure F-10 Process Flowchart of "SUM Check Error" (at Synchronous Mode)



- Process flowchart of "RESET" (at synchronous mode)

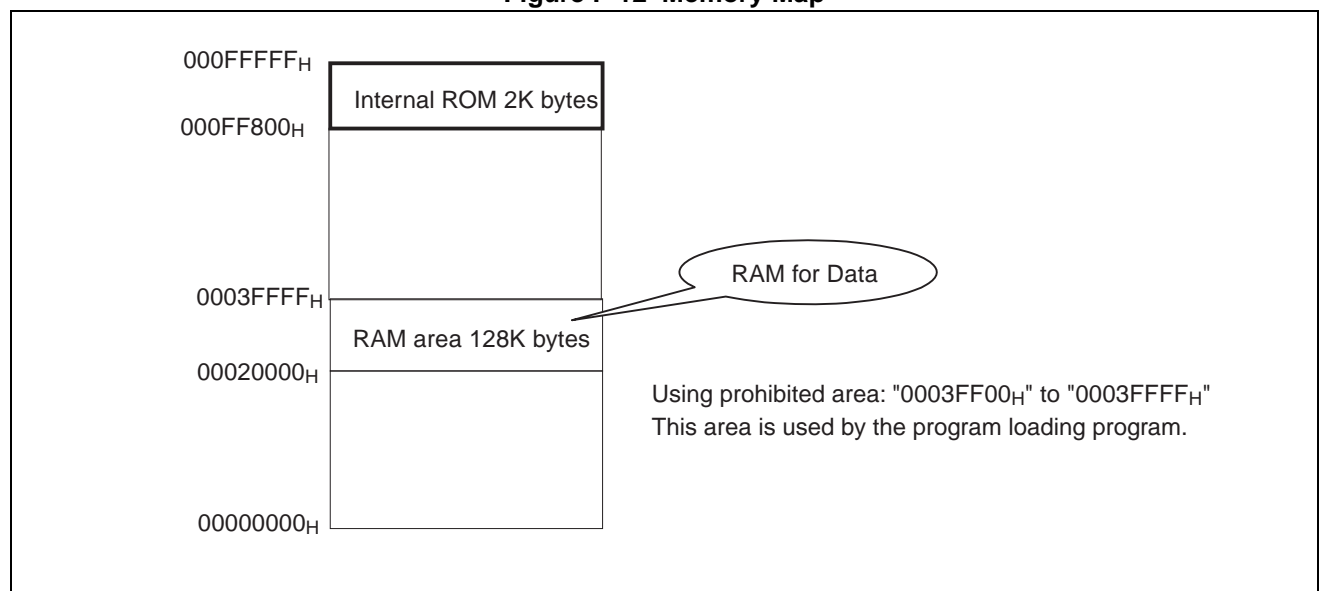
Figure F-11 Process Flowchart of "RESET" (at Synchronous Mode)



■ Memory Map

The loader program is executed in the serial writer mode. The memory map after execution as follows. The program can be stored in the following RAM area. Please set the register by the downloaded program, when you access to external area etc.

Figure F-12 Memory Map



■ Example of Write to FLASH Memory by Using Loading Program

The connection example for "Write the FLASH memory connected with CS4" by the program loaded (Develop with RAM) is indicated in the following.

To access the entire flash, it is necessary to store the flash memory in the area that doesn't overlap with internal areas such as built-in resources and RAM. In this example, it is assumed to access the Flash memory of 1Mbyte connected with the CS4 area as "0X10 0000_H" to "0X1F FFFF_H". However, the reset vector is fixed to 0XF FFFC_H, and the mode vector is fixed in 0XF FFF8_H in the FR family. Therefore, it is necessary to cover this area to normally execute the program in written Flash. The upper address signal from A20 is not connected with Flash in case of Flash of 1Mbyte. Therefore, "0X0_H" to "0XF FFFF_H" and "0X10 0000_H" to "0X1F FFFF_H" can be accessed as a mirror area by setting CS4's address range to "0X0_H" to "0X1F FFFF_H".

When the program written in Flash is executed, it becomes "External ROM external bus mode". Therefore, built-in ROM area where the loader was stored need not be considered.

(Refer to Figure F-14 .)

- Memory access that added offset address

Figure F-13 Connection Example of 1M Bytes Flash

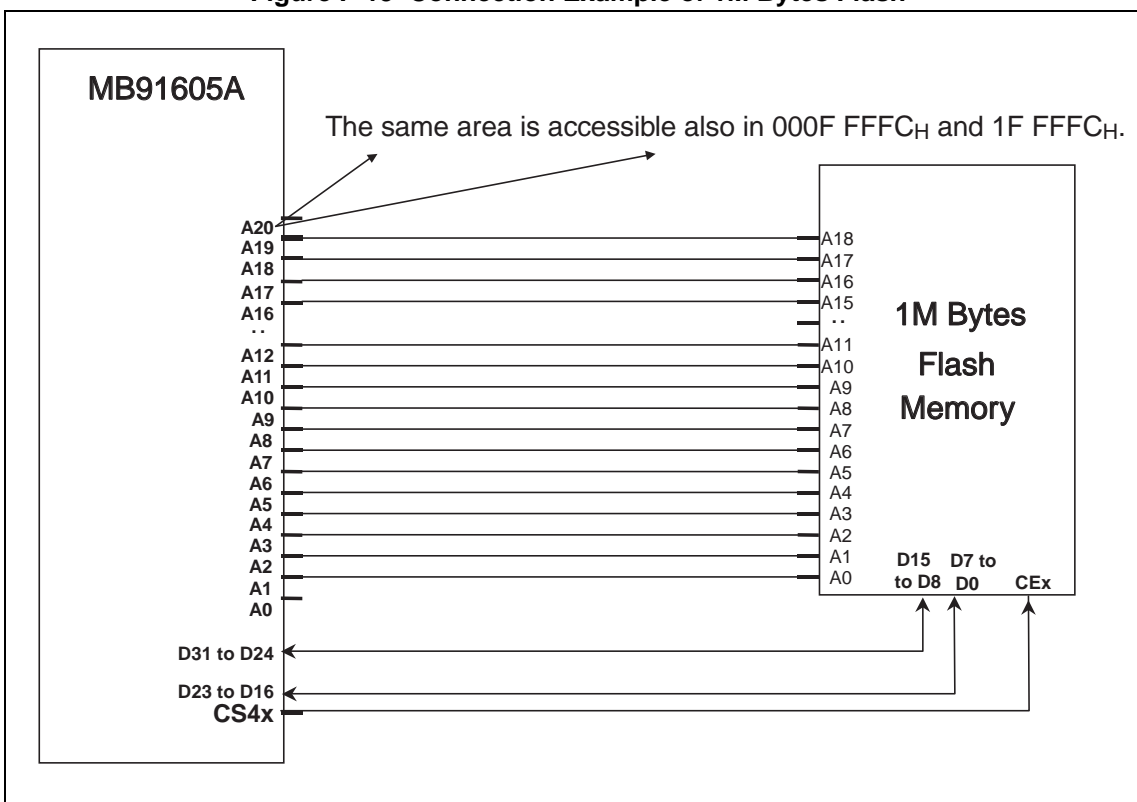
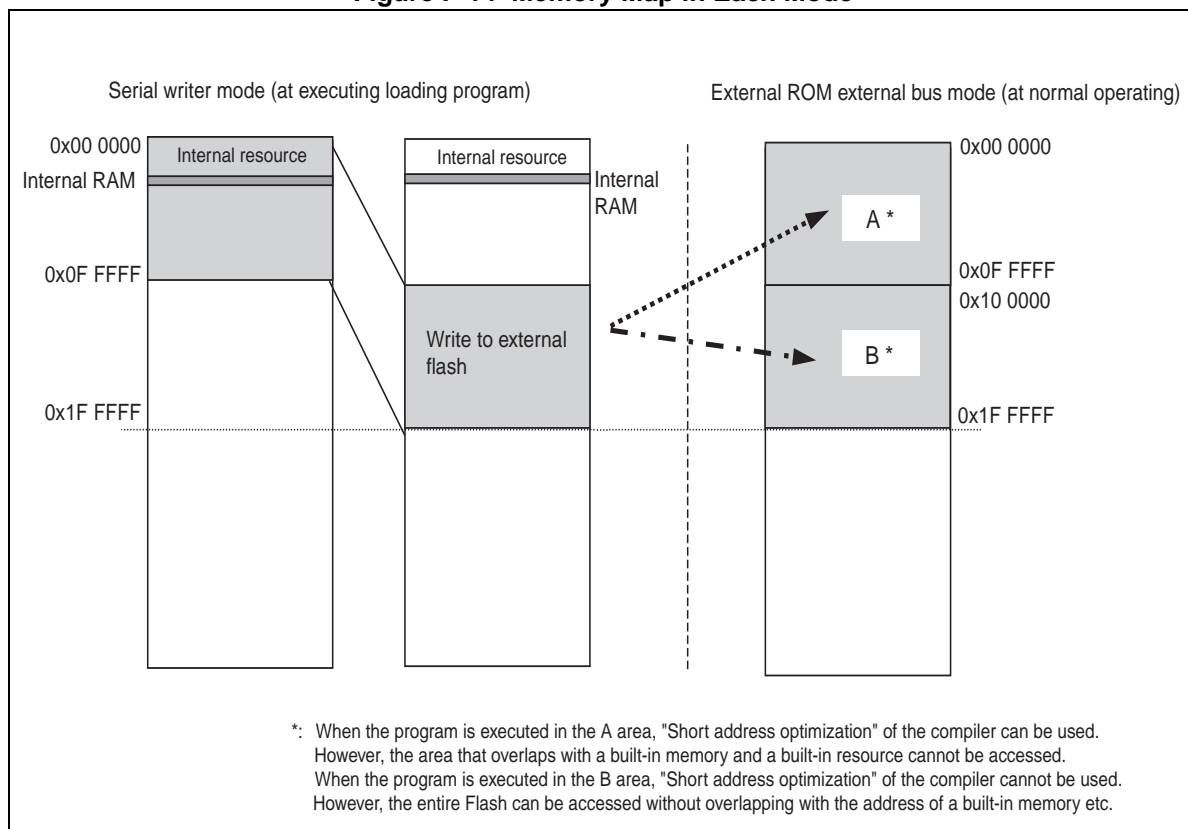


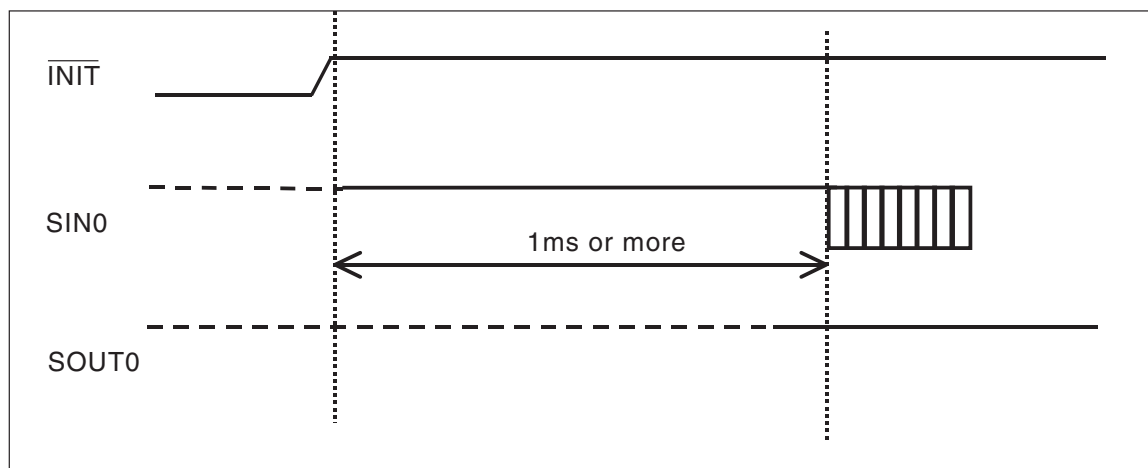
Figure F-14 Memory Map in Each Mode



■ Timing Chart of Reset

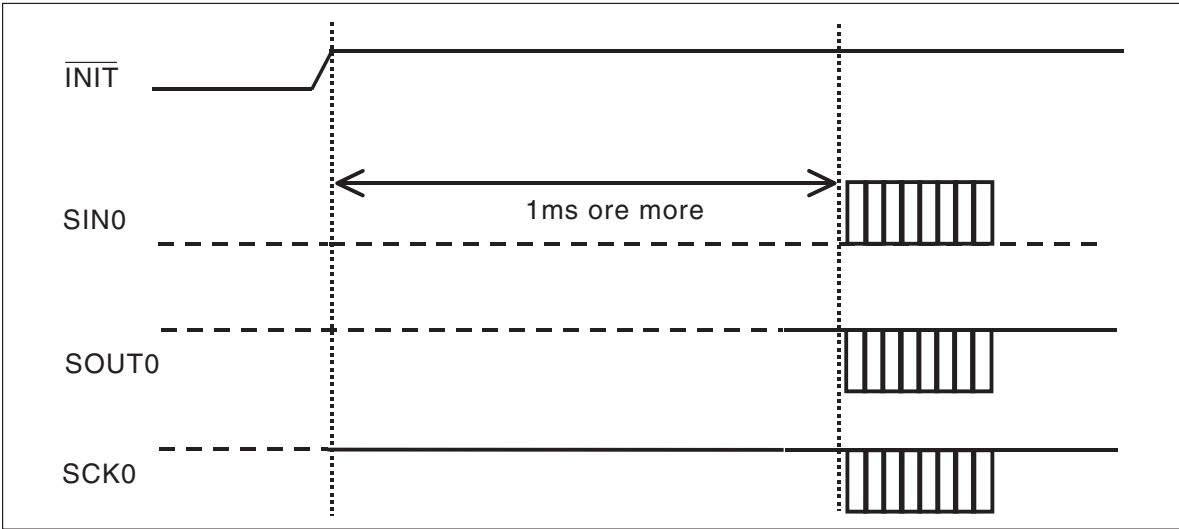
- Reset timing (asynchronous communication)

Figure F-15 Reset Timing (Asynchronous Communication)



- Reset timing (synchronous communication)

Figure F-16 Reset Timing (Synchronous Communication)



INDEX

**The index follows on the next page.
This is listed in alphabetic order.**

Index

Numerics

10-bit A/D converter	
configuration	481
explanation of operations	489
interrupt	488
pins	482
registers	483
16/32-bit PWC	
16/32-bit PWC Timer	397
Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)	
.....	403
16/32-bit Reload Timer	
16/32-bit Reload Timer	397
Block Diagram of 16/32-bit Reload Timer	
(ch.1, ch.0)	400
16-bit PPG	
16-bit PPG Timer	396
Block Diagram of 16-bit PPG Timer	399
16-bit PWM Timer	
16-bit PWM Timer	396
Block Diagram of 16-bit PWM Timer	398
16-bit PWM/PPG/Reload Timer	
Notes on Using the 16-bit PWM/PPG/Reload	
Timer	419
16-bit reload timer	
configuration	321
interruption	333
notes on use	355
operation in cascade mode	353
operation in event counter mode	348
operation in interval mode	335
pin	323
register	324
32-bit	
Functions of 32-bit Mode	417
Operations in 32-bit Mode	417
Setting the 32-bit Mode	417
7-bit Slave Address Mask Register	
7-bit Slave Address Mask Register (ISMK)	637
7-bit Slave Address Register	
7-bit Slave Address Register (ISBA)	638

A

A/D converter	
configuration	481
explanation of operations	489
interrupt	488
pins	482
registers	483
Acceptable Baud Rate Range	
Acceptable Baud Rate Range for Reception	539
Acknowledge	
Receiving Acknowledge after Transmitting	
1st Byte	654
Arbitration Lost	
Arbitration Lost Condition	667
Asynchronous	
Functions of UART	
(Asynchronous Serial Interface)	495
List of Registers of UART	
(Asynchronous Serial Interface)	497

B

Base Timer	
Base Timer Function	422
base timer i/o select function	
configuration	359
I/O mode	376
pin	360
register	362
Baud Rate	
Acceptable Baud Rate Range for Reception	539
Baud Rate Selection	673
Baud Rate Selection for CSIO	
(Clock Synchronous Serial	
Interface)	606
Bit Structure of the Baud Rate Generator Registers 1, 0	
(BGR1, BGR0)	516, 569, 639
Calculating the Baud Rate	537, 607, 673
Reload Values and Baud Rates for Different Machine	
Clock Frequencies	538, 608, 674
UART Baud Rate Selection	536
Baud Rate Generator Registers	
Bit Structure of the Baud Rate Generator Registers 1, 0	
(BGR1, BGR0)	516, 569, 639
BGR	
Bit Structure of the Baud Rate Generator Registers 1, 0	
(BGR1, BGR0)	516, 569, 639
Bit	
Data Direction Bit	669
Function Mode Bit Settings and Timer Function Modes	
Assigned	396

Bit Configuration

Bit Configuration of the Data Buffer Register (BTnDTBF)	471
Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)	445
Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	444
Bit Configuration of the Period Setting Register (BTnPCSR)	459
Bit Configuration of the PWM Duty Setting Register (BTnPDUT)	431
Bit Configuration of the PWM Period Setting Register (BTnPCSR)	430
Bit Configuration of the Timer Register (BTnTMR)	432, 446, 460

Bit Structure

Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)	523, 575, 646
Bit Structure of FIFO Control Register 0 (FCR0)	520, 572, 642
Bit Structure of FIFO Control Register 1 (FCR1)	518, 570, 640
Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	516, 569, 639
Bit Structure of the Extended Communication Control Register (ESCR)	511, 564

Block Diagram

Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)	403
Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)	400
Block Diagram of 16-bit PPG Timer	399
Block Diagram of 16-bit PWM Timer	398

BTnDTBF

Bit Configuration of the Data Buffer Register (BTnDTBF)	471
---	-----

BTnPCSR

Bit Configuration of the Period Setting Register (BTnPCSR)	459
Bit Configuration of the PWM Period Setting Register (BTnPCSR)	430

BTnPDUT

Bit Configuration of the PWM Duty Setting Register (BTnPDUT)	431
--	-----

BTnPRLH

Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)	445
--	-----

BTnPRLL

Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	444
--	-----

BTnSTC

Status Control Register (BTnSTC)	428, 442, 457, 469
--	--------------------

BTnTMCR

Timer Control Register (BTnTMCR Lower Byte)	426, 440, 455, 467
---	--------------------

Timer Control Register (BTnTMCR Upper Byte)	453, 465
Timer Control Register (BTnTMCR Upper Byte)	424, 438

BTnTMR

Bit Configuration of the Timer Register (BTnTMR)	432, 446, 460
--	---------------

Bus Error

Bus Error Operation	672
Conditions for the Occurrence of Bus Errors	672

Byte

Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)	523, 575, 646
Receiving Acknowledge after Transmitting 1st Byte	654
Timer Control Register (BTnTMCR Lower Byte)	426, 440, 455, 467
Timer Control Register (BTnTMCR Upper Byte)	424, 438

C

Calculating the Baud Rate

Calculating the Baud Rate	537, 607, 673
---------------------------------	---------------

Clear

Clearing the Counters and Their Initial Values	475
--	-----

Clock

Baud Rate Selection for CSIO (Clock Synchronous Serial Interface)	606
Counting with the Internal Clock Selected	461
External Clock	540
Functions of CSIO (Clock Synchronous Serial Interface)	549
List of Registers of CSIO (Clock Synchronous Serial Interface)	550
Operation of CSIO (Clock Synchronous Serial Interface)	587
Reload Values and Baud Rates for Different Machine Clock Frequencies	538, 608, 674
Selecting the Count Clock	473

clock division control part

configuration	129
division rate	137
internal clock	127
register	130

Clock Frequency

Reload Values and Baud Rates for Different Machine Clock Frequencies	538, 608, 674
--	---------------

clock generating part

configuration	103
explanation of operation	118
multiple rate for generating PLL clock (PLLCLK)	122
switching source clock (SRCCLK)	120

clock generating unit	
pin	106
register	107
Common	
Common Notes on Using Each Type	
of Timer	419
Communication	
Bit Structure of the Extended Communication Control Register (ESCR)	511, 564
Conditions	
Conditions for the Occurrence of Bus Errors	672
Configuration	
Bit Configuration of the Data Buffer Register (BTnDTBF)	471
Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)	445
Bit Configuration of the "L"-width Setting Reload Register (BTnPRL)	444
Bit Configuration of the Period Setting Register (BTnPCSR)	459
Bit Configuration of the PWM Duty Setting Register (BTnPDUT)	431
Bit Configuration of the PWM Period Setting Register (BTnPCSR)	430
Bit Configuration of the Timer Register (BTnTMR)	432, 446, 460
Connection	
Connection among CPUs	543
Connection Between CPUs	541, 610
Continuous	
Continuous Operation	433, 448
Control Register	
Timer Control Register (BTnTMCR Upper Byte)	453
Count Clock	
Selecting the Count Clock	473
Counter	
Clearing the Counters and Their Initial Values	475
Counting	
Counting with the Internal Clock Selected	461
Measurement Mode and Counting	476
CPU	
Addressing	50
Basic Programming Model	34
Branch Instructions	52
Data Configuration	47
EIT (Exception, Interrupt, Trap)	55
Features of the Internal Architecture	28
Memory Space	26
Overview of Instructions	32
Pipeline	30
Registers	35
CPUs	
Connection among CPUs	543
Connection Between CPUs	541, 610

CSIO

Baud Rate Selection for CSIO	
(Clock Synchronous Serial Interface)	606
Functions of CSIO (Clock Synchronous Serial Interface)	549
Interrupts of CSIO	580
List of Registers of CSIO (Clock Synchronous Serial Interface)	550
operation in 4-channel simultaneous communication mode	603

D

Data Buffer Register

Bit Configuration of the Data Buffer Register (BTnDTBF)	471
---	-----

data cache memory

cache content in each cache operation state	92
configuration	85
data cache area	91
enabling data cache operation	92
initial state	91
invalidating data cache	92
RAM operation mode when cache is off	93
registers	87

Data Direction Bit

Data Direction Bit	669
--------------------------	-----

delay interruption

configuration	219
explanation of operation	222
notes on use	223
register	220

Details

Details of Pulse Width Measurement	
Operation	475

DMA controller (DMAC)

activation	720
configuration	691
control flow	735
notes on use	736
operation	721
registers	692
settings	714

doze mode

operation in doze mode	164
------------------------------	-----

E

EIT

EIT (Exception, Interrupt, Trap)	55
--	----

Error

Bus Error Operation	672
Conditions for the Occurrence of Bus Errors	672

ESCR

Bit Structure of the Extended Communication Control Register (ESCR)	511, 564
---	----------

Extended Communication Control Register

- Bit Structure of the Extended Communication Control Register (ESCR) 511, 564

external bus interface

- configuration 227
- notes on use 259
- register setting examples 251
- registers 229

External Clock

- External Clock 540

external interrupt controller

- configuration 289
- explanation of operation 300
- pin 291
- register 292
- return from sleep mode 305
- return from standby mode 303

F

FBYTE

- Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2) 523, 575, 646

FCR

- Bit Structure of FIFO Control Register 0 (FCR0) 520, 572, 642
- Bit Structure of FIFO Control Register 1 (FCR1) 518, 570, 640

Feature

- Pulse Width Measurement Feature 472

FIFO

- Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2) 523, 575, 646
- Bit Structure of FIFO Control Register 0 (FCR0) 520, 572, 642
- Bit Structure of FIFO Control Register 1 (FCR1) 518, 570, 640
- Functions of FIFO 614
- Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing 528, 583
- Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing 530, 586
- Transmission/Reception FIFO 493

FIFO Byte Register

- Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2) 523, 575, 646

FIFO Control Register

- Bit Structure of FIFO Control Register 0 (FCR0) 520, 572, 642
- Bit Structure of FIFO Control Register 1 (FCR1) 518, 570, 640

Flag

- Occurrence of Reception Interrupts and Flag Set Timing 527, 582

Occurrence of Reception Interrupts when Reception

- FIFO is Used and Flag Set Timing 528, 583

Occurrence of Transmission Interrupts and Flag Set

- Timing 529, 585

Occurrence of Transmission Interrupts when

- Transmission FIFO is Used and Flag Set

- Timing 530, 586

FLASH Memory

Example of Write to FLASH Memory by Using

- Loading Program 860

Flow

- Pulse Width Measurement Operation Flow 478

Flowchart

- Flowchart 541, 610
- Flowcharts 545

Function

- Base Timer Function 422
- Function Mode Bit Settings and Timer Function Modes Assigned 396
- Function Selection 543
- Functions of 32-bit Mode 417
- Functions of CSIO (Clock Synchronous Serial Interface) 549
- Functions of FIFO 614
- Functions of I²C Interface 614
- Functions of Reload Counters 540, 609, 674
- Functions of UART (Asynchronous Serial Interface) 495
- Interrupt Control Bits and Interrupt Factors by Timer Function Mode 421

G

Generating

- Generating a Start Condition 651

H

handling the device

- notes on handling the device 778

"H"-width Setting Reload Register

- Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH) 445

I**I**

- Normal Transfer (I) 587
- SPI Transfer (I) 595

I/O circuit type

- I/O circuit types 20

I/O ports

- configuration 264
- pins 265
- registers 266

I²C

- Functions of I²C Interface 614

Interrupt	
Interrupt Control Bits and Interrupt Factors by Timer Function Mode	421
Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)	451
Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	435
Interrupts of CSIO	580
Interrupts of I ² C Interface	648
Interrupts of UART	525
Occurrence of Reception Interrupts and Flag Set Timing	527, 582
Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing	528, 583
Occurrence of Transmission Interrupts and Flag Set Timing	529, 585
Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing	530, 586
Interrupt Control	
Interrupt Control Bits and Interrupt Factors by Timer Function Mode	421
interrupt controller	
configuration	197
explanation of operation	201
notes on use	203
register	198
interrupt request batch-read function	
configuration	211
notes on use	215
register	212
ISBA	
7-bit Slave Address Register (ISBA)	638
ISMK	
7-bit Slave Address Mask Register (ISMK)	637
L	
List	
List of Base Timer's Registers	406
List of Registers of CSIO (Clock Synchronous Serial Interface)	550
List of Registers of I ² C Interface	615
List of Registers of UART (Asynchronous Serial Interface)	497
Loading Program	
Example of Write to FLASH Memory by Using Loading Program	860
low-power dissipation mode	
configuration	153
notes on use	172
operation in doze mode	164
operation in sleep mode	165
operation in stop mode	170
operation when clock control is set	162
register	155

"L"-width Setting Reload Register	
Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	444

M

Machine Clock

Reload Values and Baud Rates for Different Machine Clock Frequencies.....	538, 608, 674
---	---------------

main timer

configuration	141
interruption	147
main timer operation	148
register	142
transition to stop mode	150

Master

Master Data Reception	664
Master Data Transmission	658
Wait in Master Mode.....	667

MB91605A

block diagram	6
overview	2
package dimensions	7
product configuration	5

Measurement

Measurement Mode and Counting	476
-------------------------------------	-----

Mode

Function Mode Bit Settings and Timer Function Modes	
Assigned	396
Functions of 32-bit Mode	417
Interface Mode.....	493
Interrupt Control Bits and Interrupt Factors by Timer	
Function Mode	421
Measurement Mode and Counting	476
Operation Mode	502
Operations in 32-bit Mode.....	417
Reset Mode	396
Selecting the Operation Mode	473
Serial Mode Register (SMR).....	505, 559, 627
Setting the 32-bit Mode	417
Switching the Interface Mode	493
Wait in Master Mode.....	667

multifunction serial interface (CSIO)

operation in 4-channel simultaneous communication mode	603
--	-----

N

NMI input

explanation of operations	207
---------------------------------	-----

Normal

Normal Transfer (I).....	587
Normal Transfer (II).....	591
Normal Transfer (II) Timing Chart.....	592

Normal Polarity

Interrupt Factors and Timing Chart (PPG Output: Normal Polarity).....	451
---	-----

Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	435
--	-----

Note

Common Notes on Using Each Type of Timer	419
Notes on Using the 16-bit PWM/PPG/Reload Timer	419
Notes on Using the PWC Timer	420
Number of Channels	493

O

Occurrence

Conditions for the Occurrence of Bus Errors	672
Occurrence of Reception Interrupts and Flag Set Timing	527, 582
Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing	528, 583
Occurrence of Transmission Interrupts and Flag Set Timing	529, 585
Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing	530, 586

One-shot Operation

One-shot Operation	434, 449
--------------------------	----------

Operation

Bus Error Operation	672
Continuous Operation	433, 448
Details of Pulse Width Measurement Operation.....	475
Input Pin Operation	462
One-shot Operation	434, 449
Operation after being Started	475
Operation Mode	502
Operation of CSIO (Clock Synchronous Serial Interface)	587
Operation of UART	531
Operations in 32-bit Mode	417
Operations of the Base Timer	415
Output Pin Operation	462
Principles of Operation.....	447
Pulse Width Measurement Operation Flow	478
Selecting the Operation Mode.....	473
Underflow Operation	461

Operation Flow

Pulse Width Measurement Operation Flow	478
--	-----

Operation Mode

Operation Mode	502
Selecting the Operation Mode.....	473

operation mode

configuration	97
details of each mode	100
how to set the operation mode	100
registers	98

Output

Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)	451
Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	435
Output Pin Operation	462
Outputting a Slave Address	652
PWM Output at All "L" or All "H" Level.....	436

Output Pin

Output Pin Operation	462
----------------------------	-----

P**Period Setting Register**

Bit Configuration of the Period Setting Register (BTnPCSR).....	459
--	-----

pin

I/O circuit types	20
pin assignment diagram	10
pin functions	11

PPG

16-bit PPG Timer	396
Block Diagram of 16-bit PPG Timer	399
Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)	451
Notes on Using the 16-bit PWM/PPG/Reload Timer	419

PPG Timer

16-bit PPG Timer	396
Block Diagram of 16-bit PPG Timer	399

Principles

Principles of Operation.....	447
------------------------------	-----

Pulse Width

Details of Pulse Width Measurement Operation.....	475
Pulse Width Measurement Feature	472
Pulse Width Measurement Operation Flow	478
Relationship Between Reload Value and Pulse Width.....	450
Starting and Stopping Pulse Width Measurement.....	474

PWC

16/32-bit PWC Timer	397
Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)	403
Notes on Using the PWC Timer	420

PWM

16-bit PWM Timer	396
Bit Configuration of the PWM Duty Setting Register (BTnPDUT)	431
Bit Configuration of the PWM Period Setting Register (BTnPCSR).....	430
Block Diagram of 16-bit PWM Timer	398
Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	435
Notes on Using the 16-bit PWM/PPG/Reload Timer	419

PWM Output at All "L" or All "H" Level	436
PWM Duty Setting Register Bit Configuration of the PWM Duty Setting Register (BTnPDUT).....	431

PWM Output

Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)	435
PWM Output at All "L" or All "H" Level	436

PWM Period Setting Register

Bit Configuration of the PWM Period Setting Register (BTnPCSR)	430
---	-----

PWM Timer

16-bit PWM Timer	396
Block Diagram of 16-bit PWM Timer.....	398

R**Range**

Acceptable Baud Rate Range for Reception	539
--	-----

RDR

Reception Data Register (RDR)	635
Reception Data Register (RDR0/RDR1)	513, 566

Receiving

Receiving Acknowledge after Transmitting 1st Byte	654
--	-----

Reception

Acceptable Baud Rate Range for Reception	539
Master Data Reception	664
Occurrence of Reception Interrupts and Flag Set Timing.....	527, 582
Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing.....	528, 583
Reception Data Register (RDR)	635
Reception Data Register (RDR0/RDR1)	513, 566
Slave Reception	669
Transmission/Reception FIFO	493

Reception Data Register

Reception Data Register (RDR)	635
Reception Data Register (RDR0/RDR1)	513, 566

Register

7-bit Slave Address Mask Register (ISMK)	637
7-bit Slave Address Register (ISBA)	638
Bit Configuration of the Data Buffer Register (BTnDTBF).....	471
Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH).....	445
Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	444
Bit Configuration of the Period Setting Register (BTnPCSR)	459
Bit Configuration of the PWM Duty Setting Register (BTnPDUT).....	431

Bit Configuration of the PWM Period Setting Register (BTnPCSR)	430	Reload Register Write Timing	447
Bit Configuration of the Timer Register (BTnTMR)	432, 446, 460	Reload Values and Baud Rates for Different Machine Clock Frequencies	538, 608, 674
Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)	523, 575	Reload Register	
Bit Structure of FIFO Control Register 0 (FCR0)	520, 572, 642	Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)	445
Bit Structure of FIFO Control Register 1 (FCR1)	518, 570, 640	Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	444
Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)	516, 569, 639	Reload Register Write Timing	447
Bit Structure of the Extended Communication Control Register (ESCR)	511, 564	Reload Timer	
I ² C Bus Control Register (IBCR)	622	16/32-bit Reload Timer	397
I ² C Bus Status Register (IBSR)	629	Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)	400
List of Base Timer's Registers	406	Notes on Using the 16-bit PWM/PPG/Reload Timer	419
List of Registers of CSIO (Clock Synchronous Serial Interface)	550	reload timer (16-bit)	
List of Registers of I ² C Interface	615	configuration	321
List of Registers of UART (Asynchronous Serial Interface)	497	interruption	333
Reception Data Register (RDR)	635	notes on use	355
Reception Data Register (RDR0/RDR1)	513, 566	operation in cascade mode	353
Reload Register Write Timing	447	operation in event counter mode	348
Serial Control Register (SCR)	503, 556	operation in interval mode	335
Serial Mode Register (SMR)	505, 559, 627	pin	323
Serial Status Register (SSR)	508, 562, 633	register	324
Status Control Register (BTnSTC)	428, 442, 457, 469	Reload Value	
Timer Control Register (BTnTMCR Lower Byte)	426, 440, 455, 467	Relationship Between Reload Value and Pulse Width	450
Timer Control Register (BTnTMCR Upper Byte)	453, 465	Reload Values and Baud Rates for Different Machine Clock Frequencies	538, 608, 674
Timer Control Register (BTnTMCR Upper Byte)	424, 438	remote control receive	
Transmission Data Register (TDR)	636	explanation of operations and setting procedure	770
Transmission Data Register (TDR0/TDR1)	514, 567	examples	770
Relationship		registers	761
Relationship Between Reload Value and Pulse Width	450	Repeated	
Reload		Repeated Start Condition for I ² C Bus	650
16/32-bit Reload Timer	397	Reset	
Bit Configuration of the "H"-width Setting Reload Register (BTnPRLH)	445	Reset Mode	396
Bit Configuration of the "L"-width Setting Reload Register (BTnPRLL)	444	Timing Chart of Reset	861
Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)	400	reset	
Functions of Reload Counters	540, 609, 674	configuration	175
Notes on Using the 16-bit PWM/PPG/Reload Timer	419	operating state and transition	191
Relationship Between Reload Value and Pulse Width	450	operation of reset	186
		pin	177
		register	178
		reset resource	184
		reset type	183
		Restart	
		Restart	540, 609
		Restarting	475
		S	
		SCR	
		Serial Control Register (SCR)	503, 556
		Select	
		Selecting the Count Clock	473
		Selecting the Operation Mode	473

select function for DMA transfer request generation/
clear by a peripheral function

- configuration 741
- operation upon DMA transfer 757
- register 743

Selection

- Baud Rate Selection 673
- Baud Rate Selection for CSIO
(Clock Synchronous Serial Interface) 606
- Function Selection 543
- UART Baud Rate Selection 536

Serial Control Register

- Serial Control Register (SCR) 503, 556

Serial Mode Register

- Serial Mode Register (SMR) 505, 559, 627

serial programming connection example

- asynchronous serial programming connection example
..... 776
- pins used 774
- synchronous serial programming connection example
..... 775

Serial Status Register

- Serial Status Register (SSR) 508, 562, 633

Set

- Bit Configuration of the "H"-width Setting Reload
Register (BTnPRLH) 445
- Bit Configuration of the "L"-width Setting Reload
Register (BTnPRL) 444
- Bit Configuration of the Period Setting Register
(BTnPCSR) 459
- Bit Configuration of the PWM Duty Setting Register
(BTnPDUT) 431
- Bit Configuration of the PWM Period Setting Register
(BTnPCSR) 430
- Function Mode Bit Settings and Timer Function Modes
Assigned 396
- Setting the 32-bit Mode 417

Slave

- 7-bit Slave Address Mask Register (ISMK) 637
- 7-bit Slave Address Register (ISBA) 638
- Outputting a Slave Address 652
- Slave Address Match Detection 668
- Slave Reception 669
- Slave Transmission 671

Slave Address

- 7-bit Slave Address Mask Register (ISMK) 637
- 7-bit Slave Address Register (ISBA) 638
- Outputting a Slave Address 652
- Slave Address Match Detection 668

Slave Address Mask Register

- 7-bit Slave Address Mask Register (ISMK) 637

Slave Address Register

- 7-bit Slave Address Register (ISBA) 638

sleep mode

- operation in sleep mode 165

SMR

- Serial Mode Register (SMR) 505, 559, 627

source clock

- switch of source clock (SRCCLK) 120

SPI

- SPI Transfer (I) 595
- SPI Transfer (II) 599

SSR

- Serial Status Register (SSR) 508, 562, 633

Start

- Generating a Start Condition 651
- Repeated Start Condition for I²C Bus 650
- Start Condition for I²C Bus 650
- Starting a Count 540, 609, 674
- Starting and Stopping Pulse Width
Measurement 474

Status

- I²C Bus Status Register (IBSR) 629
- Serial Status Register (SSR) 508, 562, 633

Status Control Register

- Status Control Register
(BTnSTC) 428, 442, 457, 469

Stop

- Starting and Stopping Pulse Width
Measurement 474
- Stop Condition for I²C Bus 650
- Stopping 475

stop mode

- operation in stop mode 170

Structure

- Bit Structure of FIFO Byte Register
(FBYTE1/FBYTE2) 523, 575, 646
- Bit Structure of FIFO Control Register 0
(FCR0) 520, 572, 642
- Bit Structure of FIFO Control Register 1
(FCR1) 518, 570, 640
- Bit Structure of the Baud Rate Generator Registers 1, 0
(BGR1, BGR0) 516, 569, 639
- Bit Structure of the Extended Communication Control
Register (ESCR) 511, 564

Switching

- Switching the Interface Mode 493

Synchronous

- Baud Rate Selection for CSIO
(Clock Synchronous Serial
Interface) 606
- Functions of CSIO (Clock Synchronous Serial
Interface) 549
- List of Registers of CSIO (Clock Synchronous Serial
Interface) 550
- Operation of CSIO (Clock Synchronous Serial
Interface) 587

T**TDR**

- Transmission Data Register (TDR) 636

Transmission Data Register (TDR0/TDR1).....	514, 567
Timer	
16/32-bit PWC Timer	397
16/32-bit Reload Timer	397
16-bit PPG Timer	396
16-bit PWM Timer	396
Base Timer Function	422
Bit Configuration of the Timer Register (BTnTMR)	432, 446, 460
Block Diagram of 16-bit PPG Timer	399
Block Diagram of 16-bit PWM Timer	398
Common Notes on Using Each Type of Timer	419
Function Mode Bit Settings and Timer Function Modes Assigned	396
Interrupt Control Bits and Interrupt Factors by Timer Function Mode	421
List of Base Timer's Registers	406
Notes on Using the 16-bit PWM/PPG/Reload Timer	419
Notes on Using the PWC Timer	420
Operations of the Base Timer	415
Timer Control Register (BTnTMCR Lower Byte)	426, 440, 455, 467
Timer Control Register (BTnTMCR Upper Byte)	453, 465
Timer Control Register (BTnTMCR Upper Byte).....	424, 438
Timer Control Register	
Timer Control Register (BTnTMCR Lower Byte)	426, 440, 455, 467
Timer Control Register (BTnTMCR Upper Byte)	453, 465
Timer Control Register (BTnTMCR Upper Byte).....	424, 438
Timer Register	
Bit Configuration of the Timer Register (BTnTMR)	432, 446, 460
Timing	
Interrupt Factors and Timing Chart (PPG Output: Normal Polarity).....	451
Interrupt Factors and Timing Chart (PWM Output: Normal Polarity).....	435
Normal Transfer (II) Timing Chart.....	592
Occurrence of Reception Interrupts and Flag Set Timing.....	527, 582
Occurrence of Transmission Interrupts and Flag Set Timing.....	529, 585
Reload Register Write Timing.....	447
Timing Chart	
Interrupt Factors and Timing Chart (PPG Output: Normal Polarity).....	451
Interrupt Factors and Timing Chart (PWM Output: Normal Polarity).....	435
Normal Transfer (II) Timing Chart.....	592
Transfer	
Normal Transfer (I)	587
Normal Transfer (II)	591
Normal Transfer (II) Timing Chart	592
SPI Transfer (I).....	595
SPI Transfer (II)	599
Transmission	
Master Data Transmission.....	658
Occurrence of Transmission Interrupts and Flag Set Timing	529, 585
Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing	530, 586
Slave Transmission	671
Transmission Data Register (TDR)	636
Transmission Data Register (TDR0/TDR1)	514, 567
Transmission Data Register	
Transmission Data Register (TDR)	636
Transmission Data Register (TDR0/TDR1)	514, 567
U	
UART	
Functions of UART (Asynchronous Serial Interface)	495
Interrupts of UART	525
List of Registers of UART (Asynchronous Serial Interface)	497
Operation of UART	531
UART Baud Rate Selection.....	536
Underflow Operation	
Underflow Operation	461
Use	
Common Notes on Using Each Type of Timer	419
Notes on Using the 16-bit PWM/PPG/Reload Timer	419
Notes on Using the PWC Timer	420
W	
Wait	
Wait in Master Mode	667
watchdog timer	
configuration	309
explanation of operation	316
register	311
Write	
Reload Register Write Timing	447

Index of Pins

Pin Name	Related Function	Page
A		
A0	External bus interface	225
A1	External bus interface	225
A2	External bus interface	225
A3	External bus interface	225
A4	External bus interface	225
A5	External bus interface	225
A6	External bus interface	225
A7	External bus interface	225
A8	External bus interface	225
A9	External bus interface	225
A10	External bus interface	225
A11	External bus interface	225
A12	External bus interface	225
A13	External bus interface	225
A14	External bus interface	225
A15	External bus interface	225
A16	External bus interface	225
A17	External bus interface	225
A18	External bus interface	225
A19	External bus interface	225
A20	External bus interface	225
A21	External bus interface	225
A22	External bus interface	225
A23	External bus interface	225
AN0	10-bit A/D converter	482
AN1	10-bit A/D converter	482
AN2	10-bit A/D converter	482
AN3	10-bit A/D converter	482
AN4	10-bit A/D converter	482
AN5	10-bit A/D converter	482
AN6	10-bit A/D converter	482
AN7	10-bit A/D converter	482
AN8	10-bit A/D converter	482
AN9	10-bit A/D converter	482
AN10	10-bit A/D converter	482
AN11	10-bit A/D converter	482
$\overline{\text{AS}}$	External bus interface	225
ATRG	10-bit A/D converter	482
AVCC	10-bit A/D converter	482
AVRH	10-bit A/D converter	482

Pin Name	Related Function	Page
AVSS	10-bit A/D converter	482
C		
$\overline{\text{CS4}}$	External bus interface	225
$\overline{\text{CS5}}$	External bus interface	225
$\overline{\text{CS6}}$	External bus interface	225
$\overline{\text{CS8}}$	SDRAM interface	225
D		
D16	External bus interface	225
D17	External bus interface	225
D18	External bus interface	225
D19	External bus interface	225
D20	External bus interface	225
D21	External bus interface	225
D22	External bus interface	225
D23	External bus interface	225
D24	External bus interface	225
D25	External bus interface	225
D26	External bus interface	225
D27	External bus interface	225
D28	External bus interface	225
D29	External bus interface	225
D30	External bus interface	225
D31	External bus interface	225
I		
$\overline{\text{INIT}}$	Reset	177
INT0	External interrupt request	291
INT1	External interrupt request	291
INT2	External interrupt request	291
INT3	External interrupt request	291
INT4	External interrupt request	291
INT5	External interrupt request	291
INT6	External interrupt request	291
INT7	External interrupt request	291
INT8	External interrupt request	291
INT9	External interrupt request	291
INT10	External interrupt request	291
INT11	External interrupt request	291
INT12	External interrupt request	291
INT13	External interrupt request	291
INT14	External interrupt request	291
INT15	External interrupt request	291
INT16	External interrupt request	291
INT17	External interrupt request	291
INT18	External interrupt request	291
INT19	External interrupt request	291
INT20	External interrupt request	291

Pin Name	Related Function	Page
INT21	External interrupt request	291
INT22	External interrupt request	291
INT23	External interrupt request	291
M		
$\overline{\text{MCAS}}$	SDRAM interface	225
MCLK	External bus interface	225
MCLKE	SDRAM interface	225
MD0	Mode pin	11
MD1	Mode pin	11
MDQM2	External bus interface	225
MDQM3	External bus interface	225
$\overline{\text{MDWE}}$	SDRAM interface	225
$\overline{\text{MRAS}}$	SDRAM interface	225
N		
$\overline{\text{NMI}}$	NMI input	205
P		
P20	I/O port	265
P21	I/O port	265
P22	I/O port	265
P23	I/O port	265
P30	I/O port	265
P31	I/O port	265
P32	I/O port	265
P33	I/O port	265
P34	I/O port	265
P35	I/O port	265
P36	I/O port	265
P37	I/O port	265
P40	I/O port	265
P41	I/O port	265
P42	I/O port	265
P43	I/O port	265
P44	I/O port	265
P45	I/O port	265
P50	I/O port	265
P51	I/O port	265
P52	I/O port	265
P53	I/O port	265
P54	I/O port	265
P55	I/O port	265
P56	I/O port	265
P57	I/O port	265
P60	I/O port	265
P61	I/O port	265
P62	I/O port	265
P63	I/O port	265

Pin Name	Related Function	Page
P64	I/O port	265
P65	I/O port	265
P66	I/O port	265
P67	I/O port	265
P70	I/O port	265
P71	I/O port	265
P72	I/O port	265
P73	I/O port	265
P74	I/O port	265
P75	I/O port	265
P76	I/O port	265
P77	I/O port	265
P80	I/O port	265
P81	I/O port	265
P82	I/O port	265
P83	I/O port	265
P84	I/O port	265
P85	I/O port	265
P90	I/O port	265
P91	I/O port	265
P92	I/O port	265
P93	I/O port	265
P94	I/O port	265
P95	I/O port	265
PA0	I/O port	265
PA1	I/O port	265
PA2	I/O port	265
PA3	I/O port	265
PA4	I/O port	265
PA5	I/O port	265
PB0	I/O port	265
PB1	I/O port	265
PB2	I/O port	265
PB3	I/O port	265
PB4	I/O port	265
PB5	I/O port	265
PB6	I/O port	265
PB7	I/O port	265
PC0	I/O port	265
PC1	I/O port	265
PC2	I/O port	265
PC3	I/O port	265
PC4	I/O port	265
PC5	I/O port	265
PC6	I/O port	265
PC7	I/O port	265
PD0	I/O port	265

Pin Name	Related Function	Page
PD1	I/O port	265
PD2	I/O port	265
PD3	I/O port	265
PD4	I/O port	265
PD5	I/O port	265
PD6	I/O port	265
PD7	I/O port	265
PE0	I/O port	265
PE1	I/O port	265
PE2	I/O port	265
PE3	I/O port	265
PE4	I/O port	265
PE5	I/O port	265
PE6	I/O port	265
PE7	I/O port	265
R		
RCIN	Remote control	759
$\overline{\text{RD}}$	External bus interface	225
RDY	External bus interface	225
S		
SCK0	Multifunction serial interface	491
SCK1	Multifunction serial interface	491
SCK2	Multifunction serial interface	491
SCK3	Multifunction serial interface	491
SCK4	Multifunction serial interface	491
SCK5	Multifunction serial interface	491
SCK6	Multifunction serial interface	491
SCK7	Multifunction serial interface	491
SCK8	Multifunction serial interface	491
SCK9	Multifunction serial interface	491
SCK10	Multifunction serial interface	491
SCK11	Multifunction serial interface	491
SIN0	Multifunction serial interface	491
SIN1	Multifunction serial interface	491
SIN2	Multifunction serial interface	491
SIN3	Multifunction serial interface	491
SIN4	Multifunction serial interface	491
SIN5	Multifunction serial interface	491
SIN6	Multifunction serial interface	491
SIN7	Multifunction serial interface	491
SIN8	Multifunction serial interface	491
SIN9	Multifunction serial interface	491
SIN10	Multifunction serial interface	491
SIN11	Multifunction serial interface	491
SOUT0	Multifunction serial interface	491
SOUT1	Multifunction serial interface	491

Pin Name	Related Function	Page
SOUT2	Multifunction serial interface	491
SOUT3	Multifunction serial interface	491
SOUT4	Multifunction serial interface	491
SOUT5	Multifunction serial interface	491
SOUT6	Multifunction serial interface	491
SOUT7	Multifunction serial interface	491
SOUT8	Multifunction serial interface	491
SOUT9	Multifunction serial interface	491
SOUT10	Multifunction serial interface	491
SOUT11	Multifunction serial interface	491
T		
TIOA0	Base timer	360
TIOA1	Base timer	360
TIOA2	Base timer	360
TIOA3	Base timer	360
TIOA4	Base timer	360
TIOA5	Base timer	360
TIOA6	Base timer	360
TIOA7	Base timer	360
TIOA8	Base timer	360
TIOA9	Base timer	360
TIOA10	Base timer	360
TIOA11	Base timer	360
TIOB0	Base timer	360
TIOB1	Base timer	360
TIOB2	Base timer	360
TIOB3	Base timer	360
TIOB4	Base timer	360
TIOB5	Base timer	360
TIOB6	Base timer	360
TIOB7	Base timer	360
TIOB8	Base timer	360
TIOB9	Base timer	360
TIOB10	Base timer	360
TIOB11	Base timer	360
TIN0	16-bit reload timer	323
TIN1	16-bit reload timer	323
TIN2	16-bit reload timer	323
TOUT0	16-bit reload timer	323
TOUT1	16-bit reload timer	323
TOUT2	16-bit reload timer	323
V		
VDDE	Power pin	19
VDDI	Power pin	19
VSS	Power pin	19

Pin Name	Related Function	Page
W		
$\overline{\text{WE}}$	External bus interface	225
X		
X0	Clock control part	106
X1	Clock control part	106

CM71-10147-2E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

FR80

32-BIT MICROCONTROLLER

MB91605A Series

HARDWARE MANUAL

May 2010 the second edition

Published **FUJITSU SEMICONDUCTOR LIMITED**

Edited Sales Promotion Dept.
