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

How to Check the Ordering Part Number

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

For More Information

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

About Cypress

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



MB91550 Series

FR81S MB91F552 Hardware Manual

Doc. No. 002-05582 Rev. *A

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

© Cypress Semiconductor Corporation, 2014-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

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

REALOS, SOFTUNE are trademarks of Cypress.

Preface

Thank you for your continued use of Cypress semiconductor products.

Read this manual and "Data Sheet" thoroughly before using products in the MB91550 series.

Purpose of this manual and intended readers

This series is Cypress 32-bit microcontroller designed for automotive and industrial control. It contains the FR81S CPU that is compatible with the FR family. The FR81S CPU has a high level performance among the FR family by enhancing instruction pipeline and load store processing, and improving internal bus transfer.

This manual explains the function, operation, and the usage for the engineer who develops the product by actually using this series.

Sample programs and development environment

Cypress offers sample programs free of charge for using the peripheral functions of the FR81S family. Cypress also makes available descriptions of the development environment required for the MB91550 series. Feel free to use them to verify the operational specifications and usage of this Cypress microcontroller.

- Microcontroller support information:

  www.cypress.com/support

Note:

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

How to Use This Manual

Finding a function

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

- Search from the table of the contents

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

- Search from the register

  The register list for this device has been described. You can look up the name of a desired register on the list to find the address of its location or the page that explains it.
  The address where each register is located is not described in the text. To verify the address of a register, see "A.2. I/O Map" of "Appendix".

- Search from the index

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

About the chapters

Basically, this manual explains 1 peripheral function per chapter.

Terminology

This manual uses the following terminology.

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

How to Read This Manual

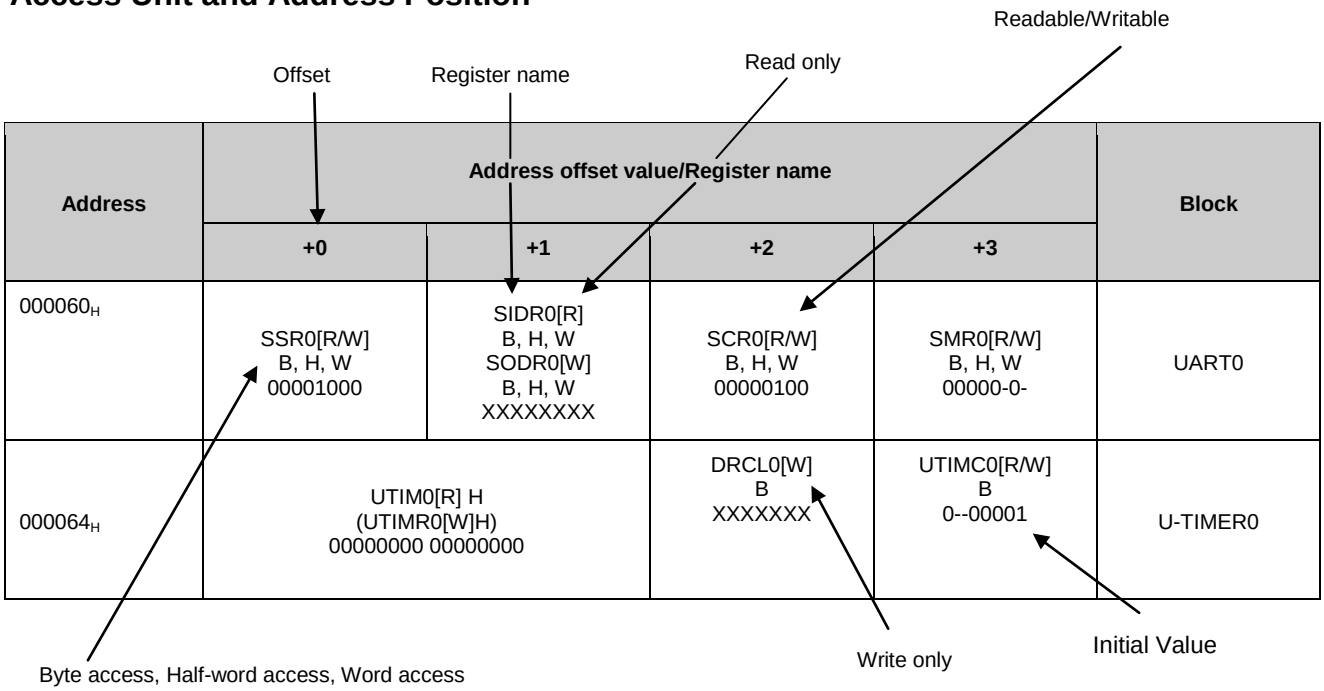
- Primary Terms

The following explains the primary terms used in this series

Term	Explanation
XBS	A 32-bit width, high-speed internal bus. The bus master is used for access from the CPU (for instruction fetch), the CPU (for data reading or writing), or the on-chip bus. The bus slave is used to access to the on-chip bus, RAM (via the XBS built-in wild register), and flash memory. The bus has a crossbar switch configuration, and a circuit from each bus master to each bus slave can operate simultaneously.
On-chip bus	A 32-bit width, high-speed internal bus. It has a 2-layer structure for XBS and DMA, and they can operate simultaneously. The bus master of the XBS layer is accessed from the XBS. The bus master of the DMA layer is accessed from the DMA. The bus slave of both layers has CAN, 16/32-bit peripheral bus bridge and others. The bus slave of only DMA layer has an access to the XBS.
32-bit peripheral bus	A 32-bit width, low-speed internal bus. It connects to various types of peripherals.
16-bit peripheral bus (R-bus)	A 16-bit width, low-speed internal bus. It connects to various types of peripherals. The 32-bit width access to this bus is divided into 16 bits x 2.
Main clock (MCLK)	This is the reference clock for LSI operation, and it is supplied from the high-speed system oscillator. It is connected to the timer for main oscillation stabilization wait, the clock generator (PLL) and others.
CR oscillation	The clock for watchdog timer 1 (hardware watchdog)
PLL clock (PLLCLK)	The main clock is multiplied by PLL.
CPU clock (CCLK)	The clock for peripherals operating under the XBS.
On-chip bus clock (HCLK)	The clock for peripherals operating under the on-chip bus.
Peripheral clock (PCLK)	The clock for peripherals operating under the 32-bit peripheral bus and 16-bit peripheral bus.
PWM clock (PWMCLK)	The operating clock for the PWM
Main clock mode	The operation mode based on the main clock. The main clock mode has the main RUN, main sleep, main stop, oscillation stabilization wait RUN, oscillation stabilization wait reset, and program reset state.
Main RUN	The main clock mode is selected, and all circuits are operable.
Oscillation stabilization wait time	When the clock is switched from the stop state to the oscillation state, the clock takes the oscillation stabilization time. During the oscillation stabilization wait time, the clock is not supplied.
OCD	The on-chip debugger for this series
OCDU	The OCD interface built in this product.
OCD tool	The OCD tool can be connected to the DEBUG I/F pin of this device.
Chip reset sequence	In the chip reset sequence, the connection of OCD tool is checked. It takes (1114+3) PCLK cycles.

Term	Explanation
SSCG	SSCG mean "Spread Spectrum Clock Generator." When the clock in electronic equipment generates a single frequency, the radiation because of the frequency and the higher harmonics wave grows. SSCG is a technology that does working that suppresses the peak of EMI to low especially depending that makes the clock frequency change slightly and oscillates it (= frequency modulation).

Access Unit and Address Position



Although three types of access (Byte, Half-word, and Word access) are enabled, some registers have access restrictions. For details, see "Appendix", or section "4. Detailed Register Description" of each chapter.

B, H, W	: Byte access, Half-word access, and Word access are enabled.
B	: Byte access (Use the Byte access only.)
H	: Half-word access (Use the Half-word access only.)
W	: Word access (Use the Word access only.)
B, H	: Byte access and Half-word access only (The Word access is not allowed.)
H, W	: Half-word access and Word access only (The Byte access is not allowed.)

(Reference)

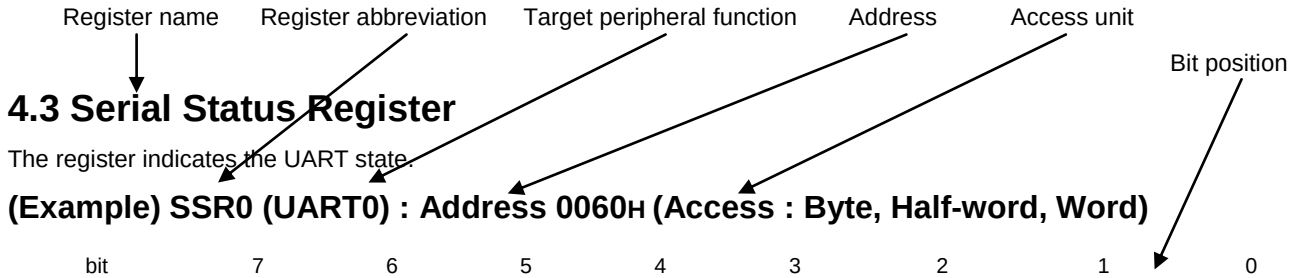
The following explains the address position during access.

- During Word access, the address is a multiple of 4 (the lowest order 2 bits are forcibly set to "00").
- During Half-word access, the address is a multiple of 2 (the lowest order 1 bit is forcibly set to "0").
- During Byte access, the address remains unchanged.

Therefore, if the SSR0 register is to be accessed by Half-word, for example, SSR0 + SIDR0 (SODR0) register at address 060H needs to be accessed.

(If the address offsets are +1 and +2 (for example, SIDR0+SCR0), the Half-word access is not allowed.)

Access Unit and Bit Position



4.3 Serial Status Register

The register indicates the UART state.

(Example) SSR0 (UART0) : Address 0060H (Access : Byte, Half-word, Word)

bit	7	6	5	4	3	2	1	0
	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE
Initial value	0	0	0	0	1	0	0	0
Attribute	R/W	R/WX	R/WX	R/WX	R/WX	R/W	R/W	R/W

If the access unit is changed, the bit position changes.

If the address offset is +0: (Example of SSR0 register)

Access size	Address	Bit position							
Word	060H+0H	7	6	5	4	3	2	1	0
Half-word	060H+0H	15	14	13	12	11	10	9	8
Word	060H+0H	31	30	29	28	27	26	25	24
Bit name		PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE

If the address offset is +1: (Example of SIDR0 register)

Access size	Address	Bit position							
Word	060H+1H	7	6	5	4	3	2	1	0
Half-word	060H+0H	7	6	5	4	3	2	1	0
Word	060H+0H	23	22	21	20	19	18	17	16
Bit name		D7	D6	D5	D4	D3	D2	D1	D0

If the address offset is +2: (Example of SCR0 register)

Access size	Address	Bit position							
Word	060H+2H	7	6	5	4	3	2	1	0
Half-word	060H+2H	15	14	13	12	11	10	9	8
Word	060H+0H	15	14	13	12	11	10	9	8
Bit name		PEN	P	SBL	CL	A/D	REC	RXE	TXE

If the address offset is +3: (Example of SMR0 register)

Access size	Address	Bit position							
Word	060H+3H	7	6	5	4	3	2	1	0
Half-word	060H+2H	7	6	5	4	3	2	1	0
Word	060H+0H	7	6	5	4	3	2	1	0
Bit name		MD1	MD0	CS2	CS1	CS0	-	SCKE	-

Meaning of Bit Attribute Symbols

R	: Read enabled
W	: Write enabled
RM	: Reading operation during read-modify-write(RMW) operation
"/" (slash) R/W	: Read and write enabled. (The read value is the written value.)
"," (comma) R, W	: The read and written values differ from each other. (The read value is different from the written value.)
R0	: The read value is "0".
R1	: The read value is "1".
W0	: This bit must always be written to "0".
W1	: This bit must always be written to "1".
(RM0)	: "0" is read by read-modify-write(RMW) operation.
(RM1)	: "1" is read by read-modify-write(RMW) operation.
RX	: The read value is undefined. (A reserved bit or an undefined bit)
WX	: Writing does not affect on the operation. (Undefined bit)

- R/W writing examples
  - ☐ R/W : Read and write enabled (The read value is the written value.)
  - ☐ R,W : Read and write enabled (The read value is different from the written value.)
  - ☐ R,RM/W : Read and write enabled (The read value is different from the written value. The written value is read by read-modify-write (RMW) instruction.) An example is a port data register.
  - ☐ R(RM1),W : Read and write enabled (The read value is different from the written value. For read-modify-write (RMW) instructions, "1" will be read out.) An example is an interrupt request flag.
  - ☐ R,WX : Read only (Read enabled. Writing has no effect on operation.)
  - ☐ R1,W : Write only (Write enabled. The read value is "1".)
  - ☐ R0,W : Write only (Write enabled. The read value is "0".)
  - ☐ RX,W : Write only (Write enabled. The read value is undefined.)
  - ☐ R0,W0 : Reserved bit (The written value is "0". The read value is the written value.)
  - ☐ R0,W0 : Reserved bit (The written value is "0". The read value is "0".)
  - ☐ R1,W0 : Reserved bit (The written value is "0". The read value is "1".)
  - ☐ RX,W0 : Reserved bit (The written value is "0". The read value is undefined.)
  - ☐ R/W1 : Reserved bit (The written value is "1". The read value is the written value.)
  - ☐ R1,W1 : Reserved bit (The written value is "1". The read value is "1".)
  - ☐ R0,W1 : Reserved bit (The written value is "1". The read value is "0".)
  - ☐ RX,W1 : Reserved bit (The written value is "1". The read value is undefined.)
  - ☐ RX,WX : Undefined bit (The read value is undefined. Writing has no effect on operation.)
  - ☐ R0,WX : Undefined bit (The read value is "0". Writing has no effect on operation.)

Contents



Contents **10**

1. Overview **29**

1.1 Overview 30
1.2 Features 31
1.2.1 FR81S CPU Core 32
1.2.2 Peripheral Functions 33
1.3 Product Line-up 36
1.4 Function Overview 37
1.5 Block Diagram 39
1.6 Memory Map 40
1.7 Pin Assignment 41
1.8 Device Package 42
1.9 Explanation of Pin Functions 43
1.9.1 Pins of Each Function 46
1.10 I/O Circuit Types 51

2. Handling the Device **55**

2.1 Handling Precautions 56
2.2 Handling Device 60
2.3 Application Notes 62
2.3.1 Function Switching of a Multiplexed Port 63
2.3.2 Low-power Consumption Mode 64
2.3.3 Notes When Writing Data in a Register that Includes the Status Flag 65

3. CPU **67**

3.1 Overview 68
3.2 Features 69
3.3 CPU Operating Description 71
3.3.1 CPU Operating Status 72
3.4 Pipeline Operation 77
3.5 Floating Point Operation Processing 78
3.6 Data Structure 79
3.7 Addressing 80
3.8 Programming Model 81
3.8.1 General-purpose Registers, Dedicated Registers, and Floating Point Registers 82
3.8.2 System Register 83
3.9 Reset and EIT Processing 84
3.9.1 Reset 85
3.9.2 EIT Processing 86

3.9.3 Vector Table 87
3.10 Memory Protection Function (MPU) 90
3.10.1 Overview 91
3.10.2 List of Registers 92
3.10.3 Description of Registers 93
3.10.4 Operations of Memory Protection Function 111

4. Operation Mode 119

4.1 Overview 120
4.2 Features 121
4.3 Configuration 122
4.4 Register 123
4.4.1 Bus Mode Register: BMODR (Bus MODe Register) 124
4.5 Operation 125
4.5.1 MD0, MD1, P044 Pins Settings 126
4.5.2 Fetching the Operation Mode 127
4.5.3 Explanation of Each Operation Mode 128

5. Clock 133

5.1 Overview 134
5.2 Features 136
5.3 Configuration 137
5.4 Registers 142
5.4.1 Frequency Dividing Setting Register 0: DIVR0 (DIVide clock configuration Register 0) 143
5.4.2 Frequency Dividing Setting Register 2: DIVR2 (DIVide clock configuration Register 2) 144
5.4.3 Clock Source Setting Register: CSELR (Clock source SELect Register) 145
5.4.4 Clock Source Monitor Register: CMONR (Clock source MONitor Register) 147
5.4.5 Main Timer Control Register: MTMCR (Main clock TiMer Control Register) 149
5.4.6 PLL Setting Register: PLLCR (PLL Configuration Register) 151
5.4.7 Oscillation Stabilization Wait Setting Register: CSTBR (Clock STaBilization select Register) 154
5.4.8 PLL Oscillation Stabilization Wait Timer Setting Register: PTMCR (PLL clock osc TiMer Control Register) 156
5.4.9 PLL/SSCG Clock Selection Register: CCPSSELR (CCtl Pll/Sscg clock SELect Register) 157
5.4.10 PLL/SSCG Output Clock Frequency Dividing Setting Register: CCPSDIVR (CCtl Pll/Sscg clock DIVide Register) 158
5.4.11 PLL Feedback Frequency Dividing Setting Register: CCPLLFBR (CCtl PLL FB clock divide Register) 160
5.4.12 SSCG Feedback Frequency Dividing Setting Register 0: CCSSFBR0 (CCtl SScg FB clock divide Register 0) 161
5.4.13 SSCG Feedback Frequency Dividing Setting Register 1: CCSSFBR1 (CCtl SScg FB clock divide Register 1) 162
5.4.14 SSCG Configuration Setting Register 0: CCSSCCR0 (CCtl SSCg Config. Register 0) 163
5.4.15 SSCG Configuration Setting Register 1: CCSSCCR1 (CCtl SSCg Config. Register 1) 165
5.4.16 Clock Gear Configuration Setting Register 0: CCCGRCR0 (CCtl Clock Gear Config. Register 0) 166
5.4.17 Clock Gear Configuration Setting Register 1: CCCGRCR1 (CCtl Clock Gear Config. Register 1) 168
5.4.18 Clock Gear Configuration Setting Register 2: CCCGRCR2 (CCtl Clock Gear Config. Register 2) 169
5.4.19 Sync/Async Control Register 170
5.4.20 Peripheral Interface Clock Divider 171
5.5 Operation 172

5.5.1 Oscillation Control ... 173
5.5.2 Oscillation Stabilization Wait ... 178
5.5.3 Selecting the Source Clock (SRCCLK) ... 182
5.5.4 Timer ... 188
5.5.5 Notes when Clocks Conflict ... 195
5.5.6 The Clock Gear Circuit ... 196
5.5.7 Operations during MDI Communications ... 199

6. Clock Reset State Transitions ... 201

6.1 Overview ... 202
6.2 Device States and Transitions ... 203
6.2.1 Diagram of State Transitions ... 204
6.2.2 Explanation of Each States ... 205
6.2.3 Priority of State Transition Requests ... 206
6.3 Device State and Regulator Mode Corresponding to those States ... 207

7. Reset ... 209

7.1 Overview ... 210
7.2 Features ... 211
7.3 Configuration ... 212
7.4 Registers ... 215
7.4.1 Reset Source Register: RSTRR (ReSeT Result Register) ... 216
7.4.2 Reset Control Register: RSTCR (ReSeT Control Register) ... 218
7.4.3 CPU Abnormal Operation Register: CPUAR (CPU Abnormal operation Register) ... 219
7.5 Operation ... 221
7.5.1 Reset Level ... 222
7.5.2 Reset Factor ... 225
7.5.3 Reset Acceptance ... 235
7.5.4 Reset Issue ... 240
7.5.5 Reset Sequence ... 245

8. DMA Controller (DMAC) ... 253

8.1 Overview ... 254
8.2 Features ... 255
8.3 Configuration ... 256
8.4 Registers ... 257
8.4.1 DMA Control Register: DMACR (DMA Control Register) ... 259
8.4.2 DMA Channel Control Register 0 to 7: DCCR0 to 7 (DMA Channel Control Register 0 to 7) ... 261
8.4.3 DMA Channel Status Register 0 to 7: DCSR0 to 7 (DMA Channel Status Register 0 to 7) ... 267
8.4.4 DMA Transfer Count Register 0 to 7: DTCR0 to 7 (DMA Transfer Count Register 0 to 7) ... 269
8.4.5 DMA Transfer Source Register 0 to 7: DSAR0 to 7 (DMA Source Address Register 0 to 7) ... 270
8.4.6 DMA Transfer Destination Register 0 to 7: DDAR0 to 7 (DMA Destination Address Register 0 to 7) ... 271
8.4.7 DMA Transfer Suppression NMI Flag Register: DNMIR (DMA-halt by NMI Register) ... 273
8.4.8 DMA Transfer Suppression Level Register: DILVR (DMA-halt by Interrupt Level Register) ... 274
8.5 Operation ... 275
8.5.1 Configuration ... 276
8.5.2 Table for On-chip Bus IPs and Corresponding DMAC Channels ... 293
8.6 DMA Usage Examples ... 294

9. Generation and Clearing of DMA Transfer Requests 297
9.1 Overview 298
9.2 Features 299
9.2.1 Transfer Request Generation Setting 300
9.2.2 Interrupt Clearing Setting 301
9.3 Configuration 302
9.4 Registers 303
9.4.1 DMA Request Clear Register 0: ICSEL0 (Interrupt Clear SELect register 0) 304
9.4.2 DMA Request Clear Register 2: ICSEL2 (Interrupt Clear SELect register 2) 305
9.4.3 DMA Request Clear Register 3: ICSEL3 (Interrupt Clear SELect register 3) 306
9.4.4 DMA Request Clear Register 5: ICSEL5 (Interrupt Clear SELect register 5) 307
9.4.5 DMA Request Clear Register 11: ICSEL11 (Interrupt Clear SELect register 11) 308
9.4.6 DMA Request Clear Register 13: ICSEL13 (Interrupt Clear SELect register 13) 309
9.4.7 DMA Request Clear Register 14: ICSEL14 (Interrupt Clear SELect register 14) 310
9.4.8 DMA Request Clear Register 15: ICSEL15 (Interrupt Clear SELect register 15) 311
9.4.9 DMA Request Clear Register 16: ICSEL16 (Interrupt Clear SELect register 16) 312
9.4.10 DMA Request Clear Register 17: ICSEL17 (Interrupt Clear SELect register 17) 313
9.4.11 DMA Request Clear Register 18: ICSEL18 (Interrupt Clear SELect register 18) 314
9.4.12 DMA Request Clear Register 19: ICSEL19 (Interrupt Clear SELect register 19) 315
9.4.13 DMA Request Clear Register 20: ICSEL20 (Interrupt Clear SELect register 20) 316
9.4.14 DMA Request Clear Register 21: ICSEL21 (Interrupt Clear SELect register 21) 317
9.4.15 DMA Request Clear Register 22: ICSEL22 (Interrupt Clear SELect register 22) 318
9.4.16 DMA Request Clear Register 23: ICSEL23 (Interrupt Clear SELect register 23) 319
9.4.17 DMA Request Clear Register 25: ICSEL25 (Interrupt Clear SELect register 25) 320
9.4.18 IO Transfer Request Setting Register 0 to 7: IORR0 to 7 (IO triggered DMA Request Register for ch.0 to 7) 321
9.5 Operation 323
9.5.1 Configuration 324
9.5.2 Notes 325

10. FixedVector Function 327
10.1 Overview 328
10.2 Features 329
10.3 Configuration 330
10.4 Registers 331
10.5 Operation 332
10.5.1 Operation After Reset Released 333
10.5.2 Usage 334
10.6 Notes 335

11. I/O Ports 337
11.1 Overview 338
11.2 Features 339
11.3 Configuration 340
11.4 Registers 341
11.4.1 Port Data Register 00, 02 to 04: PDR00, 02 to 04 (Port Data Register 00, 02 to 04) 342
11.4.2 Data Direction Register 02 to 04: DDR02 to 04 (Data Direction Register 02 to 04) 343

11.4.3 Port Function Register 00, 02 to 04: PFR00, 02 to 04 (Port Function Register 00, 02 to 04) 344
11.4.4 Input Data Direct Register 00, 02 to 04: PDDR00, 02 to 04 (Port Data Direct Register 00, 02 to 04) ... 345
11.4.5 Port Pull-up/down Enable Register 00, 02 to 04: PPER00, 02 to 04 (Port Pull-up/down Enable Register 00, 02 to 04) 346
11.4.6 Extended Port Function Register 02, 16, 26, 35, 65, 86, 88: EPFR02, 16, 26, 35, 65, 86, 88 (Extended Port Function Register 02, 16, 26, 35, 65, 86, 88) 347
11.4.7 Port Input Enable Register: PORTEN (PORT ENable register) 355
11.4.8 KEY CoDe Register: KEYCDR (KEY CoDe register) 356
11.5 Operation 358
11.5.1 Pin I/O Assignment 359
11.5.2 EPFR Setting Priority 366
11.5.3 Noise Filter 367
11.5.4 Input Blocked by GPORTEN 368
11.5.5 Notes on Pins with the A/D Converter Function 369
11.5.6 Settings When Using the Base Timer TIOA1 and TIOA3 Pins 370
11.5.7 Key Code Register Function Settings 371
11.5.8 Notes on Switching the I/O Port Function 372
11.5.9 Input blocked when specific peripheral functions are used 373

12. Interrupt Control (Interrupt Controller) 375

12.1 Overview 376
12.2 Features 377
12.3 Configuration 378
12.4 Registers 379
12.4.1 Interrupt Control Registers 00 to 47: ICR00 to ICR47 (Interrupt Control Register 00 to 47) 380
12.5 Operation 381
12.5.1 Setting 382
12.5.2 Starting 383
12.5.3 Determining Priorities 384
12.5.4 Recovering From Stop Mode 385

13. External Interrupt Input 387

13.1 Overview 388
13.2 Features 389
13.3 Configuration 390
13.4 Registers 391
13.4.1 External Interrupt Factor Register 0: EIRR0 (External Interrupt Request Register 0) 392
13.4.2 External Interrupt Enable Register 0: ENIR0 (ENable Interrupt request Register 0) 393
13.4.3 External Interrupt Request Level Register 0: ELVR0 (External interrupt LeVel Register 0) 394
13.5 Operation 395
13.6 Setting 397
13.7 Q&A 398
13.8 Notes 399

13. External Interrupt Input 387

13.1 Overview 388
13.2 Features 389
13.3 Configuration 390

13.4 Registers 391
13.4.1 External Interrupt Factor Register 0: EIRR0 (External Interrupt Request Register 0) 392
13.4.2 External Interrupt Enable Register 0: ENIR0 (ENable Interrupt request Register 0) 393
13.4.3 External Interrupt Request Level Register 0: ELVR0 (External interrupt LeVel Register 0) 394
13.5 Operation 395
13.6 Setting 397
13.7 Q&A 398
13.8 Notes 399

14. NMI Input 401

14.1 Overview 402
14.2 Features 403
14.3 Configuration 404
14.4 Register 405
14.5 Operation 406
14.6 Usage Example 407

15. Delay Interrupt 409

15.1 Overview 410
15.2 Features 411
15.3 Configuration 412
15.4 Registers 413
15.5 Operation 414
15.6 Restrictions 415

16. Interrupt Request Batch Read 417

16.1 Overview 418
16.2 Features 419
16.3 Configuration 420
16.4 Registers 421
16.4.1 Interrupt Request Batch Read Register 0 Upper-order: IRPR0H (Interrupt Request Peripheral Read register 0H) 422
16.4.2 Interrupt Request Batch Read Register 0 Lower-order: IRPR0L (Interrupt Request Peripheral Read register 0L) 423
16.4.3 Interrupt Request Batch Read Register 1 Upper-order: IRPR1H (Interrupt Request Peripheral Read register 1H) 424
16.4.4 Interrupt Request Batch Read Register 1 Lower-order: IRPR1L (Interrupt Request Peripheral Read register 1L) 425
16.4.5 Interrupt Request Batch Read Register 2 Upper-order: IRPR2H (Interrupt Request Peripheral Read register 2H) 426
16.4.6 Interrupt Request Batch Read Register 2 Lower-order: IRPR2L (Interrupt Request Peripheral Read register 2L) 427
16.4.7 Interrupt Request Batch Read Register 3 Upper-order: IRPR3H (Interrupt Request Peripheral Read register 3H) 428
16.4.8 Interrupt Request Batch Read Register 3 Lower-order: IRPR3L (Interrupt Request Peripheral Read register 3L) 429
16.4.9 Interrupt Request Batch Read Register 4 Upper-order: IRPR4H (Interrupt Request Peripheral Read register 4H) 430
16.4.10 Interrupt Request Batch Read Register 4 Lower-order: IRPR4L (Interrupt Request Peripheral Read register 4L) 431

16.4.11 Interrupt Request Batch Read Register 5 Upper-order: IRPR5H (Interrupt Request Peripheral Read register 5H) 432
16.4.12 Interrupt Request Batch Read Register 5 Lower-order IRPR5L (Interrupt Request Peripheral Read register 5L) 433
16.4.13 Interrupt Request Batch Read Register 6 Upper-order: IRPR6H (Interrupt Request Peripheral Read register 6H) 434
16.4.14 Interrupt Request Batch Read Register 6 Lower-order: IRPR6L (Interrupt Request Peripheral Read register 6L) 435
16.4.15 Interrupt Request Batch Read Register 7 Upper-order: IRPR7H (Interrupt Request Peripheral Read register 7H) 436
16.4.16 Interrupt Request Batch Read Register 7 Lower-order: IRPR7L (Interrupt Request Peripheral Read register 7L) 437
16.4.17 Interrupt Request Batch Read Register 8 Upper-order: IRPR8H (Interrupt Request Peripheral Read register 8H) 438
16.4.18 Interrupt Request Batch Read Register 8 Lower-order: IRPR8L (Interrupt Request Peripheral Read register 8L) 439
16.4.19 Interrupt Request Batch Read Register 9 Upper-order: IRPR9H (Interrupt Request Peripheral Read register 9H) 440
16.4.20 Interrupt Request Batch Read Register 9 Lower-order: IRPR9L (Interrupt Request Peripheral Read register 9L) 441
16.4.21 Interrupt Request Batch Read Register 10 Upper-order: IRPR10H (Interrupt Request Peripheral Read register 10H) 442
16.4.22 Interrupt Request Batch Read Register 10 Lower-order: IRPR10L (Interrupt Request Peripheral Read register 10L) 443
16.4.23 Interrupt Request Batch Read Register 11 Upper-order: IRPR11H (Interrupt Request Peripheral Read register 11H) 444
16.4.24 Interrupt Request Batch Read Register 11 Lower-order: IRPR11L (Interrupt Request Peripheral Read register 11L) 445
16.4.25 Interrupt Request Batch Read Register 12 Upper-order: IRPR12H (Interrupt Request Peripheral Read register 12H) 446
16.4.26 Interrupt Request Batch Read Register 12 Lower-order: IRPR12L (Interrupt Request Peripheral Read register 12L) 447
16.4.27 Interrupt Request Batch Read Register 13 Upper-order:IRPR13H (Interrupt Request Peripheral Read register 13H) 448
16.4.28 Interrupt Request Batch Read Register 13 Lower-order: IRPR13L (Interrupt Request Peripheral Read register 13L) 449
16.4.29 Interrupt Request Batch Read Register 14 Upper-order: IRPR14H (Interrupt Request Peripheral Read register 14H) 450
16.4.30 Interrupt Request Batch Read Register 15 Upper-order: IRPR15H (Interrupt Request Peripheral Read register 15H) 451
16.4.31 Interrupt Request Batch Read Register 15 Lower-order: IRPR15L (Interrupt Request Peripheral Read register 15L) 452
16.5 Operation 453

17. Watchdog Timer 455

17.1 Overview 456
17.2 Features 457
17.2.1 Watchdog Timer 0 (Software Watchdog) 458
17.2.2 Watchdog Timer 1 (Hardware Watchdog) 459
17.3 Configuration 460
17.4 Registers 461
17.4.1 Watchdog Timer 0 Control Register: WDTCR0 (WatchDog Timer Configuration Register 0) 462

17.4.2 Watchdog Timer 0 Clear Register: WDTCPR0 (WatchDog Timer Clear Pattern Register 0) 464
17.4.3 Watchdog Timer 0 Extended Control Register: WDTECR0 (Watchdog Timer Extended Configuration Register 0) 465
17.4.4 Watchdog Timer 1 Cycle Information Register: WDTCR1 (WatchDog Timer Cycle information Register 1) 467
17.4.5 Watchdog Timer 1 Clear Register: WDTCPR1 (WatchDog Timer Clear Pattern Register 1) 468
17.5 Operation 469
17.5.1 Software Watchdog Function 470
17.5.2 Hardware Watchdog Function 474
17.6 Usage Example 478

18. Base Timer 479

18.1 Overview 480
18.2 Features 481
18.2.1 16/32-bit Reload Timer 482
18.2.2 16-bit PWM Timer 483
18.2.3 16/32-bit PWC Timer 484
18.2.4 16-bit PPG Timer 485
18.3 Configuration 486
18.4 Registers 488
18.4.1 Common Registers 490
18.4.2 Registers for 16/32-bit Reload Timer 501
18.4.3 Registers for 16-bit PWM Timer 504
18.4.4 Registers for 16-bit PPG Timer 509
18.4.5 16/32-bit PWC Timer Register 513
18.5 Operation 517
18.5.1 Selection of Timer Function 518
18.5.2 I/O Allocation 519
18.5.3 32-bit Mode Operation 522
18.5.4 16/32-bit Reload Timer Operation 526
18.5.5 16-bit PWM Timer Operation 538
18.5.6 16-bit PPG Timer Operation 548
18.5.7 16/32-bit PWC Timer Operation 561

19. Reload Timer 575

19.1 Overview 576
19.2 Features 577
19.3 Configuration 579
19.4 Registers 580
19.4.1 Control Status Register: TMCSR (TiMer Control and Status Register) 581
19.4.2 16-bit Timer Register: TMR (16bit TiMer Register) 585
19.4.3 16-bit Timer Reload Register A, 16-bit Timer Reload Register B: TMRLRA, TMRLRB (16bit TiMer ReLoad Register A/B) 586
19.5 Operation 589
19.5.1 Setting 590
19.5.2 Operation Procedure 598
19.5.3 Operations of Each Counter 606
19.5.4 Cascade Input 624

19.5.5 Priority of Concurrent Operations 625
19.6 Application Note 626
19.6.1 Single One-shot Timer 628
19.6.2 Reload Timer 631
19.6.3 PPG 634
19.6.4 PWM 638
19.6.5 PWC 642

20. 16-bit Free-Run Timer 645

20.1 Overview 646
20.2 Features 647
20.3 Configuration 648
20.4 Registers 650
20.4.1 Registers for the 16-bit Free-run Timer 651
20.5 Operation 662
20.5.1 Interrupt for the 16-bit Free-run Timer 663
20.5.2 Operation of the 16-bit Free-run Timer 664
20.5.3 Notes on Operating Specifications 673

21. 16-bit Input Capture 677

21.1 Overview 678
21.2 Features 679
21.3 Configuration 680
21.4 Registers 681
21.4.1 16-bit Input Capture Registers 682
21.5 Operation 687
21.5.1 Interrupts for 16-bit Input Capture 688
21.5.2 Operation of 16-bit Input Capture 689
21.5.3 Notes on Using the 16-bit Input Capture 692

22. WDT1 Calibration 693

22.1 Overview 694
22.2 Features 695
22.3 Configuration 696
22.4 Registers 697
22.4.1 Calibration Unit Control Register 1: CUCR1 (Calibration Unit Control Register 1) 698
22.4.2 CR Clock Timer Data Register: CUTD1 (Calibration Unit Timer Data register 1) 699
22.4.3 Main Oscillation Timer Result Register 1: CUTR1 (Calibration Unit Timer Result register 1) 700
22.5 Operation 701
22.5.1 Measurement of Errors in CR Clock 702
22.5.2 Note 703

23. Power Consumption Control 705

23.1 Overview 706
23.2 Features 707
23.3 Configuration 708
23.4 Registers 709
23.4.1 Standby Control Register: STBCR (STandBy mode Control Register) 710

23.5 Operation 712
23.5.1 Clock Control 713
23.5.2 List of Clocks Supplied in Low-power Consumption Mode 715
23.5.3 Sleep Mode 716
23.5.4 Standby Mode: Watch Mode 723
23.5.5 Standby Mode: Stop Mode 728
23.5.6 Stop State of Microcontroller 733
23.5.7 Transition to Illegal Standby Mode 734
23.6 Usage Example 735

24. Low-Voltage Detection (Internal Low-Voltage Detection) 747

24.1 Overview 748
24.2 Features 749
24.3 Configuration 750
24.4 Register 751
24.4.1 Internal Low-Voltage Detection Register: LVD (Low-Voltage Detect internal power fall register) 752
24.5 Operation 754
24.5.1 Internal Low-voltage Detection 755
24.6 Notes 756

25. Low-Voltage Detection (External Low-Voltage Detection) 747

25.1 Overview 748
25.2 Features 749
25.3 Configuration 750
25.4 Registers 751
25.4.1 External Low-Voltage Detection Rise Detection Register: LVD5R (Low Voltage Detect external 5v Rise register) 752
25.4.2 External Low-Voltage Detection Fall Detection Register: LVD5F (Low-Voltage Detect external 5v Fall register) 753
25.5 Operation 755
25.6 Notes 756

26. Wild Register 757

26.1 Overview 758
26.2 Features 759
26.3 Configuration 760
26.4 Registers 761
26.4.1 Wild Register Data Enable Register: WREN (Wild Register ENable register) 763
26.4.2 Wild Register Address Register 00 to 15: WRAR00 to 15 (Wild Register Address Register 00 to 15) . 764
26.4.3 Wild Register Data Register 00 to 15: WRDR00 to 15 (Wild Register Data Register 00 to 15) 765
26.5 Operation 766
26.6 Usage Example 767

27. Clock Supervisor 769

27.1 Overview 770
27.2 Configuration 771
27.3 Register 772
27.3.1 Clock Supervisor Control Register: CSVCR (Clock SuperVisor Control Register) 773
27.4 Operation 775

27.4.1 Initial State 776
27.4.2 Stopping the CR Oscillator and the Clock Supervisor Function 777
27.4.3 Re-enabling the Clock Supervisor 778
27.4.4 Stop Mode 779
27.4.5 Watch Mode 780
27.4.6 Checking the Reset Factor Using the Clock Supervisor 781
27.4.7 Returning from the CR Clock 782

28. Regulator Control 783

28.1 Overview 784
28.2 Features 785
28.3 Configuration 786
28.4 Register 787
28.4.1 Regulator Output Voltage Select Register: REGSEL (REGulator output voltage SELect register) 788

29. Bus Performance Counters 789

29.1 Overview 790
29.2 Features 791
29.3 Configuration 792
29.4 Registers 793
29.4.1 BPC-A Control Register: BPCCRA (Bus Performance Counter Control Register A) 794
29.4.2 BPC-B Control Register: BPCCRB (Bus Performance Counter Control Register B) 795
29.4.3 BPC-C Control Register: BPCCRC (Bus Performance Counter Control Register C) 796
29.4.4 BPC-A Count Register: BPCTRA (Bus Performance CounTer Register A) 797
29.4.5 BPC-B Count Register: BPCTRB (Bus Performance CounTer Register B) 798
29.4.6 BPC-C Count Register: BPCTRC (Bus Performance CounTer Register C) 799
29.5 Operation 800
29.5.1 Setting 801
29.5.2 Starting and Stopping 803
29.5.3 Operation 804
29.5.4 Measurement and Result Processing 805

30. CRC 807

30.1 Overview 808
30.2 Features 809
30.3 Configuration 810
30.4 Registers 811
30.4.1 CRC Control Register: CRCCR (CRC Control Register) 812
30.4.2 CRC Initial Value Register: CRCINIT (CRC Initial value register) 813
30.4.3 Input Data Register: CRCIN (CRC INput data register) 814
30.4.4 CRC Register: CRCR (CRC Register) 815
30.5 Operation 816
30.5.1 CRC Definition 817
30.5.2 Reset Operation 818
30.5.3 Initialization 819
30.5.4 Byte and Bit Orders 820
30.5.5 CRC Calculation Sequence 821
30.5.6 Examples 822

31. RAMECC 829
31.1 Overview 830
31.2 Features 831
31.3 Configuration 832
31.4 Registers 833
31.4.1 Single-bit ECC Error Address Register XBS RAM: SEEARX 834
31.4.2 Double-bit ECC Error Address Register XBS RAM: DEEARX 835
31.4.3 ECC Error Control Register XBS RAM: EECSRX 836
31.4.4 ECC False Error Generation Address Register XBS RAM: EFEARX 837
31.4.5 ECC False Error Generation Control Register XBS RAM: EFECRX 838
31.5 Operation 840
31.5.1 RAMECC Function 841
31.5.2 Interrupt-related Register 842
31.5.3 Test Mode 843
31.5.4 Notes 844

32. Multi-Function Serial Interface 845
32.1 Overview 846
32.2 Features 847
32.3 Configuration 853
32.4 Registers 854
32.4.1 Common Registers 857
32.4.2 Registers for UART 869
32.4.3 Registers for CSIO 883
32.4.4 Registers for LIN 914
32.5 Operation of UART 947
32.5.1 Interrupt of UART 948
32.5.2 Operation of UART 957
32.5.3 Setup Procedure and Program Flow 976
32.6 Operation of CSIO 984
32.6.1 Interrupts of CSIO 985
32.6.2 Operation of CSIO 996
32.6.3 Setup Procedure and Program Flow 1049
32.7 Operation of LIN Interface (v2.1) 1051
32.7.1 Interrupt of LIN Interface (v2.1) manual mode 1052
32.7.2 Interrupts in LIN Interface (v2.1) Assist Mode 1063
32.7.3 Operation of Serial Timer 1077
32.7.4 Test Mode 1080
32.7.5 Operation of LIN Interface (v2.1) 1087
32.7.6 Setup Procedure and Program Flow 1123

33. CAN Controller 1133
33.1 Overview 1134
33.2 Features 1135
33.3 Configuration 1136
33.4 Registers 1137
33.4.1 Overview 1138

33.4.2 Overall Control Registers 1147
33.4.3 Message Interface Register 1159
33.4.4 Message Object 1169
33.4.5 Message Handler Registers 1175
33.5 Operation 1185
33.5.1 Message Object 1186
33.5.2 Message Transmission Operation 1189
33.5.3 Message Reception Operation 1194
33.5.4 FIFO Buffer Function 1201
33.5.5 Interrupt Function 1206
33.5.6 Bit Timing and CAN System Clock (fsys) Generation 1207
33.5.7 Test Mode 1210
33.5.8 Software Initialization 1217
33.5.9 CAN Wake Up Function 1218
33.6 Limitations 1219
33.6.1 Init Bit 1220

34. CAN Clock Prescaler 1223

34.1 Overview 1224
34.2 Features 1225
34.3 Configuration 1226
34.4 Registers 1227
34.4.1 CAN Prescaler Register: CANPRE 1228

35. 12-bit A/D Converter 1231

35.1 Overview 1232
35.2 Features 1233
35.2.1 Function of A/D Activation Compare 1234
35.2.2 Function of A/D Activation Arbitration 1236
35.2.3 Functions of 12-bit A/D Converter Control 1237
35.3 Configuration 1238
35.4 Registers 1240
35.4.1 Register of Analog Input Control 1242
35.4.2 Registers of A/D Activation Compare 1244
35.4.3 Registers of 12-BIT A/D Converter Control 1257
35.5 Operation 1263
35.5.1 Interrupt of A/D Activation Compare 1264
35.5.2 A/D Activation Compare Operation 1266
35.5.3 A/D Activation Arbitration Operation 1278
35.5.4 12-bit A/D Converter Operation 1282
35.6 Notes 1292

36. Flash Memory 1293

36.1 Overview 1294
36.2 Features 1295
36.3 Configuration 1296
36.3.1 Block Diagram 1297
36.3.2 Sector Configuration Diagram 1298

36.3.3 Sector Number and Flash Macro Number Correspondence Chart 1299
36.4 Registers 1300
36.4.1 Flash Control Register: FCTLR (Flash ConTroL Register) 1301
36.4.2 Flash Status Register: FSTR (Flash STatus Register) 1303
36.4.3 Flash Interface Control Register: FLIFCTLR (Flash I/F Control Register) 1304
36.4.4 Flash Interface Feature Extension Register 1: FLIFFER1 (FLash I/F Feature ExtensiONRegister 1) 1305
36.4.5 Flash Interface Feature Extension Register 2: FLIFFER2 (FLash I/F Feature ExtensiONRegister 2) 1306
36.5 Operation 1307
36.5.1 Access Mode Setting 1308
36.5.2 Writing Flash Memory by CPU 1311
36.5.3 Automatic Algorithm 1312
36.5.4 Reset Command 1319
36.5.5 Write Command 1320
36.5.6 Chip Erase Command 1323
36.5.7 Sector Erase Command 1324
36.5.8 Sector Erase Suspend Command 1327
36.5.9 Security Function 1328
36.5.10 Notes on Using Flash Memory 1333

37. WorkFlash Memory 1335

37.1 Overview 1336
37.2 Features 1337
37.3 Configuration 1338
37.3.1 Block Diagram 1339
37.3.2 Sector Configuration Diagram 1340
37.4 Registers 1341
37.4.1 WorkFlash Control Register: DFCTLR (WorkFlash ConTroL Register) 1342
37.4.2 WorkFlash Status Register: DFSTR (WorkFlash STatus Register) 1343
37.4.3 Flash Interface Control Register: FLIFCTLR (Flash I/F Control Register) 1344
37.5 Operation 1346
37.5.1 Access Mode Setting 1347
37.5.2 Writing Flash Memory by CPU 1350
37.5.3 Automatic Algorithm 1351
37.5.4 Reset Command 1359
37.5.5 Write Command 1360
37.5.6 Chip Erase Command 1363
37.5.7 Sector Erase Command 1364
37.5.8 Sector Erase Suspend Command 1367
37.5.9 Security Function 1368
37.5.10 Notes on Using WorkFlash Memory 1373

38. On Chip Debugger (OCD) 1375

38.1 Overview 1376
38.2 Features 1377
38.3 Configuration 1379
38.3.1 DEBUG I/F Clock 1381
38.4 Registers 1384

38.4.1 DBG Register 1385
38.4.2 User IO Register 1387
38.5 Operation 1391
38.5.1 OCDU Operating Mode 1392
38.5.2 Overview of DEBUG I/F 1395
38.5.3 Specification Restrictions at Connection to OCD Tool of This Series 1399
38.5.4 OCD-DSU ID Code and Mount Type Information on This Series 1406

39. Bus Diagnosis Function 1407

39.1 Overview 1408
39.2 Features 1409
39.3 Configuration 1410
39.4 Registers 1413
39.4.1 BUS DIaGnosis Status Register: BUSDIGSR 1414
39.4.2 BUS diagnosis TeST Register: BUSTSTR0/1 1416
39.4.3 BUS diagnosis ADdRess Register: BUSADR 1421
39.5 Operation 1422
39.5.1 Error Detection 1423
39.5.2 Test Function 1425
39.5.3 Notes 1426
39.5.4 Example of Operating Bus Diagnosis 1427

40. RAM Diagnosis Function 1433

40.1 Overview 1434
40.2 Features 1435
40.3 Configuration 1436
40.4 Registers 1437
40.4.1 TEST Error Address Register 0 XBS RAM: TEAR0X 1438
40.4.2 TEST Error Address Register 1 XBS RAM: TEAR1X 1440
40.4.3 TEST Error Address Register 2 XBS RAM: TEAR2X 1442
40.4.4 TEST Start Address Register XBS RAM: TASARX 1444
40.4.5 TEST End Address Register XBS RAM: TAEARX 1445
40.4.6 TEST Diagnosis Function Register XBS RAM: TTCRX 1446
40.4.7 TEST Initialization Function Register XBS RAM: TICRX 1449
40.4.8 TEST Software Reset Generation Control Register XBS RAM: TSRCRX 1451
40.4.9 TEST Fake Error Generation Control Register XBS RAM: TFECRX 1452
40.4.10 TEST Key Code Control Register XBS RAM: TKCCRX 1453
40.5 Operation 1454
40.5.1 RAM Diagnosis 1455
40.5.2 RAM Initialization 1456
40.5.3 Interrupt-Related Register 1457
40.5.4 RAM Diagnosis Fake Error Generation Procedure 1458
40.5.5 Number of Required Cycles 1459
40.5.6 Note 1461

41. Timing Protection Unit 1463

41.1 Overview 1464
41.2 Features 1465

41.3 Configuration 1466
41.4 Registers 1467
41.4.1 TPU Unlock Register: TPUUNLOCK (TPU Unlock Register) 1468
41.4.2 TPU Lock Status Register: TPULST (TPU Lock Status Register) 1469
41.4.3 TPU Access Violation Detection Register: TPUVST (TPU Access Violation Status Register) 1470
41.4.4 TPU Control Register: TPUCFG (TPU Configuration Register) 1471
41.4.5 TPU Timer Interrupt Request Register: TPUTIR (TPU Timer Interrupt Request Register) 1473
41.4.6 TPU Timer Status Register: TPUTST (TPU Timer Status Register) 1474
41.4.7 TPU Timer Interrupt Enable Register: TPUTIE (TPU Timer Interrupt Register) 1475
41.4.8 TPU Module ID Register: TPUTMID (TPU Module ID Register) 1476
41.4.9 TPU Timer Control Register 00 to 07: TPUTCN00-07 (TPU Timer Control Register 00-07) 1477
41.4.10 TPU Timer Control Register 10 to 17: TPUTCN10-17 (TPU Timer Control Register 10-17) 1479
41.4.11 TPU Counter Value Register 0 to 7: TPUTCC0-7 (TPU Timer Current Count Register 0-7) 1481
41.5 Operation 1482
41.5.1 TPU Control Register Access Protection 1483
41.5.2 Global Prescaler 1484
41.5.3 Interrupt Control 1485
41.5.4 Timer Operation 1486
41.5.5 Free-run Function 1487
41.5.6 Individual Prescaler Function 1488
41.5.7 Debug Support Function 1489
41.5.8 Operation Flow 1490

42. Clock Monitor 1491

42.1 Overview 1492
42.2 Features 1493
42.3 Configuration 1494
42.4 Registers 1495
42.4.1 Clock Monitor Configuration Registers: CMCFG 1496
42.5 Operation 1498
42.6 Setting 1499
42.7 Q&A 1500
42.7.1 How to Configure the Output Pin (MONCLK) 1501
42.7.2 How to Select the Output Frequency 1502
42.7.3 How to Enable or Disable Clock Monitor Output 1503
42.7.4 How to Set the Clock Output Mark Level 1504
42.8 Notes 1505

43. PWM 1507

43.1 Overview 1508
43.2 Features 1509
43.3 Configuration 1511
43.4 Registers 1516
43.4.1 Master Clock-related Registers 1521
43.4.2 PWM Common Status Registers 1545
43.4.3 PWM Generation-related Registers 1550
43.4.4 Fault Function-related Registers 1561

43.4.5 Soft Overwrite Register 1583
43.4.6 Dead Time Registers 1587
43.4.7 Blanking-related Registers 1594
43.4.8 A/D Converter Trigger Generation Registers 1605
43.5 Explanation of Operation 1615
43.5.1 PWM Interrupts 1616
43.5.2 Timer Operation for Master Clock Generation 1620
43.5.3 PWM Generation Operation 1630
43.5.4 Fault Control Operation 1643
43.5.5 Soft Overwrite Control Operation 1646
43.5.6 Dead Time Operation 1647
43.5.7 Blanking Generation Operation 1648
43.5.8 A/D Converter Trigger Generation Operation 1649
43.6 Notes 1650

44. PWM Dedicated Clock 1653

44.1 Overview 1654
44.2 Features 1655
44.3 Configuration 1656
44.4 Registers 1657
44.4.1 PWM PLL Division (M Division) Selection Register: PLL2DIVM 1658
44.4.2 PWM PLL Multiplication Rate (N Division) Selection Register: PLL2DIVN 1659
44.4.3 PWM PLL Multiplication Rate (K Division) Selection Register: PLL2DIVK 1660
44.4.4 PWM PLL Clock Output Control Register: CLKR2 1661
44.4.5 Clock Gear Configuration Setting Register 0: PWMCGRCR0 1663
44.4.6 Clock Gear Configuration Setting Register 1: PWMCGRCR1 1665
44.4.7 Clock Gear Configuration Setting Register 2: PWMCGRCR2 1666
44.5 Clock Gear Circuit 1667
44.6 Operation 1668
44.7 Frequency and Gear Calculation Examples 1670
44.8 Notes 1671

45. PWC with Buffers 1673

45.1 Overview 1674
45.2 Features 1675
45.3 Configuration 1677
45.4 Registers 1678
45.4.1 Registers of the PWC with Buffers 1680
45.5 Explanation of Operation 1708
45.5.1 Interrupts of the PWC with Buffers 1709
45.5.2 Operation of the PWC with Buffers 1711
45.6 Notes 1724

46. 12-bit A/D Converter (4-Channel Simultaneous Sampling) 1727

46.1 Overview 1728
46.2 Features 1729
46.2.1 A/D Activation Control Functions 1730
46.2.2 A/D Activation Arbitration Function 1731

46.2.3 12-bit A/D Converter (4-channel Simultaneous Sampling) Control Function 1732
46.3 Configuration 1733
46.4 Registers 1737
46.4.1 A/D Activation Control Registers 1739
46.4.2 12-bit A/D Converter (4-channel Simultaneous Sampling) Control Registers 1751
46.5 Explanation of Operation 1756
46.5.1 A/D Activation Control Interrupt 1757
46.5.2 A/D Activation Control Operation 1761
46.5.3 Operation of A/D Activation Arbitration 1772
46.5.4 Operation of 12-bit A/D Converter (4-channel Simultaneous Sampling) 1775
46.6 Notes 1786

47. Comparator 1787

47.1 Overview 1788
47.2 Features 1789
47.3 Configuration 1790
47.4 Registers 1791
47.4.1 Comparator Control Register: CMPCTL 1792
47.4.2 Comparator 0 D/A Converter Control Register: CMPDACR0 1797
47.4.3 Comparator 0 D/A Converter Output Setting Register: CMPDADR0 1798
47.4.4 Comparator 1 D/A Converter Control Register: CMPDACR1 1799
47.4.5 Comparator 1 D/A Converter Output Setting Register: CMPDADR1 1800
47.4.6 Comparator 2 D/A Converter Control Register: CMPDACR2 1801
47.4.7 Comparator 2 D/A Converter Output Setting Register: CMPDADR2 1802
47.4.8 Comparator Output Interrupt Flag Register: CMPINT 1803
47.4.9 Comparator Output Status Register: CMPST 1805
47.5 Explanation of Operation 1806
47.5.1 Comparator 1807
47.5.2 D/A Converter for Comparison Reference Voltage 1808
47.5.3 Comparator Output Polarity Selection 1809
47.5.4 Glitch Filter 1810
47.5.5 Interrupt Request 1811
47.6 Notes 1812

48. Slope Compensation 1813

48.1 Overview 1814
48.2 Features 1815
48.3 Configuration 1816
48.4 Registers 1817
48.4.1 Operation Control Register: SLPCNT 1818
48.4.2 Status Transition Trigger Edge Selection Register 1: SLPEDGESEL1 1819
48.4.3 Status Transition Trigger Edge Selection Register 2: SLPEDGESEL2 1821
48.4.4 Operation Control Switch Setting Register: SLPSWA, SLPSWB, SLPSWC 1823
48.4.5 Status Transition Trigger 1A Time Setting Register: SLP1A 1826
48.4.6 Status Transition Trigger 1B Time Setting Register: SLP1B 1827
48.4.7 Status Transition Trigger 1C Time Setting Register: SLP1C 1828
48.4.8 Status Transition Trigger 3 Time Setting Register: SLP3 1829

48.4.9 Status Transition Trigger 2A Time Setting Register: SLP2A 1830
48.4.10 Status Transition Trigger 2B Time Setting Register: SLP2B 1831
48.4.11 Status Transition Trigger 2C Time Setting Register: SLP2C 1832
48.4.12 Status Transition Trigger 4 Time Setting Register: SLP4 1833
48.4.13 Slope Amount Setting Register: SLPDADR 1834
48.4.14 Status Transition Trigger Transition Enable Setting Register 1: SLPSTMSEL1 1836
48.4.15 Status Transition Trigger Transition Enable Setting Register 2: SLPSTMSEL2 1839
48.4.16 Slope Transmission Error Correction Amount Display Register: SLPCVE 1841
48.5 Explanation of Operation 1842
48.5.1 Slope Compensation 1843
48.5.2 Setting the Slope Amount 1850
48.6 Notes 1851

Appendix 1853

A.1 I/O Map 1854
A.2 List of Interrupt Vector 1886
A.3 Pin States by CPU States 1889

Revision History 1891

Document Revision History 1891

1. Overview

This chapter explains the overview.

1.1 Overview
1.2 Features
1.3 Product Line-up
1.4 Function Overview
1.5 Block Diagram
1.6 Memory Map
1.7 Pin Assignment
1.8 Device Package
1.9 Explanation of Pin Functions
1.10 I/O Circuit Types

1.1 Overview

This section explains the overview of the MB91F552.

The MB91550 series is a Cypress 32-bit microcontroller designed for automotive devices. This series contains the FR81S CPU which is compatible with the FR family.

1.2 Features

This section explains the features of the MB91F552.

1.2.1 FR81S CPU Core

FR81S CPU core is shown.

- 32-bit RISC, load/store architecture, 5-stage pipeline
- Maximum operating frequency: 80 MHz (Source oscillation = 4.0 MHz and 20 multiplied (PLL clock multiplication system))
- General-purpose register : 32 bits × 16 sets
- 16-bit fixed length instructions (basic instruction), 1 instruction per cycle
- Instructions appropriate to embedded applications
  - ☐ Memory-to-memory transfer instruction
  - ☐ Bit processing instruction
  - ☐ Barrel shift order etc.
- High-level language support instructions
  - ☐ Function entry/exit instructions
  - ☐ Register content multi-load and store instructions
- Bit search instructions

  Logical 1 detection, 0 detection, and change-point detection
- Branch instructions with delay slot

  Decrease overhead during branch process
- Register interlock function

  Easy assembler writing
- Built-in multiplier and instruction level support

  Signed 32-bit multiplication: 5 cycles

  Signed 16-bit multiplication: 3 cycles
- Interrupt (PC/PS saving)

  6 cycles (16 priority levels)
- The Harvard architecture allows simultaneous execution of program and data access.
- Instruction compatibility with the FR Family
- Built-in memory protection function (MPU)
  - ☐ Eight protection areas can be specified commonly for instructions and the data.
  - ☐ Control access privilege in both privilege mode and user mode.
- Built-in FPU (floating point arithmetic)
  - ☐ IEEE754 compliant
  - ☐ Floating-point register 32-bit × 16 sets

1.2.2 Peripheral Functions

This section lists the peripheral functions.

- CPU function for clock generation (with SSCG function)
  - ☐ Main oscillation (4MHz to 16MHz)
  - ☐ PLL multiplication rate (1 to 20 times)
- PWM function for clock generation
  - ☐ Main oscillation (4 MHz to 16 MHz)
  - ☐ PLL multiplication rate (1 to 50 times)
- Built-in program flash memory capacity

  MB91F552:128+64KB
- Built-in data flash (WorkFlash) 64KB
- Built-in RAM capacity
  - ☐ Main RAM

    MB91F552:24KB
- General-purpose ports:

  MB91F552: 30sets

  DMA Controller
  - ☐ Up to 8 channels can be started simultaneously.
  - ☐ 2 transfer factors (Internal peripheral request and software)
- A/D converter 1 (successive approximation type)
  - ☐ 12-bit resolution : 8 channels × 1 unit
  - ☐ Conversion time : 1µs
- A/D converter 2 (successive approximation type simultaneous sampling of 4-channel input)
  - ☐ 12-bit resolution: Max. 4 channels × 1 unit
  - ☐ Conversion time :

    For 1-channel conversion: Min. 0.7 µs

    For 4-channel conversion: Min. 1.75 µs
- External interrupt input: 4 channels
  - ☐ Level ("H"/"L"), or edge detection (rising or falling) supported
- Multi-function serial communication (built-in transmission/reception FIFO memory) :3 channels

  < UART (Asynchronous serial interface) >
  - ☐ Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory
  - ☐ Parity or no parity is selectable.
  - ☐ Built-in dedicated baud rate generator
  - ☐ The external clock can be used as the transfer clock.
  - ☐ Parity, frame, and overrun error detection functions are provided
  - ☐ DMA transfer support

<CSIO (Synchronous serial interface) >

- ☐ Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory
- ☐ SPI supported; master and slave systems supported; 5-bit to 16-bit, 20-bit, 24-bit, 32-bit data length can be set.
- ☐ Built-in dedicated baud rate generator (Master operation)
- ☐ The external clock can be entered (Slave operation).
- ☐ Overrun error detection function is provided
- ☐ DMA transfer support
- ☐ Serial chip select SPI function

<LIN (Asynchronous Serial Interface for LIN) >

- ☐ Full-duplex double buffering system, 64-byte transmission FIFO memory, 64-byte reception FIFO memory
- ☐ LIN protocol revision 2.1 supported
- ☐ Master and slave systems supported
- ☐ Framing error and overrun error detection
- ☐ LIN synch break generation and detection; LIN synch delimiter generation
- ☐ Built-in dedicated baud rate generator
- ☐ The external clock can be adjusted by the reload counter
- ☐ DMA transfer support
- ☐ Hard assist function

- CAN Controller (CAN) : 1 channel
  - ☐ Transfer speed : Up to 1Mbps
  - ☐ 64-transmission/reception message buffering: 1 channel
- Reload timer: 16-bit × 5 channels
- Free-run timer: 16-bit × 1 channel
- Input capture: 16-bit × 1 channel (linked with the free-run timer)
- PWC: 2 channels
  - ☐ Max. 80 MHz operation
- PWM: 6 channels (2 channels × 3 pairs)
  - ☐ Max. 200 MHz operation
- Clock supervisor
  - ☐ Monitoring abnormality (by damaged quartz, etc.) of external main oscillation (4 MHz)
  - ☐ When abnormality is detected, it switches to the CR clock.
- Base timer : Max.4 channels
  - ☐ 16-bit timer
  - ☐ The timer mode is selected from PWM/PPG/PWC/reload.
  - ☐ A 32-bit timer can be used in 2 channels of cascade mode for the reload timer/PWC function.
- CRC generation

- Watchdog timer
  - ☐ Hardware watchdog
  - ☐ Software watchdog (An effective range of a clear counter can be set.)
- Slope compensation (constant current unit): 1 channel
- Comparator: 3 channels
- NMI (non-maskable interrupt)
- Interrupt controller
- Interrupt request batch read
  - ☐ Multiple interrupts from peripherals can be read by a series of registers.
- Low-power consumption mode
  - ☐ Sleep / Stop / Watch mode
- Power-on reset
- Low-voltage detection reset (external power supply and internal power supply are independently observed.)
- Device Package : LQFP-64
- CMOS 90nm Technology
- Power supply
  - ☐ 5V Power supply
  - ☐ The internal 1.2V is generated from 5V with the voltage step-down circuit.

1.3 Product Line-up

This section shows product line-up of the MB91F552.

Table 1-1. Product Line-up

	MB91F552
System Clock	On-chip PLL Clock multiple method
Minimum instruction execution time	12.5ns(80MHz)
Flash Capacity (Program)	128+64KB
Flash Capacity (Data)	64KB
RAM Capacity	24KB
DMA Transfer	8 channels
16-bit Base Timer	4 channels
Free-run Timer	16bit×1 channel
Input capture	16bit×1 channel
16-bit Reload Timer	5 channels
Clock Supervisor	Yes
External Interrupt	4 channels
A/D converter	12 bit×8 channels (1unit) Simultaneous sampling of 12 bit×4 channels input (1 unit)
Multi-Function Serial Interface	3 channels
CAN	64 msg×1 channel
Hardware Watchdog Timer	Yes
CRC Formation	Yes
Low-voltage detection reset	Yes
Flash Security	Yes
ECC Flash/WorkFlash	Yes
ECC RAM	Yes
Memory Protection Function (MPU)	Yes
Floating point arithmetic (FPU)	Yes
General-purpose port (#GPIOs)	30 ports
SSCG	Yes
CR oscillator	Yes
OCD (On Chip Debug)	Yes
TPU (Timing Protection Unit)	Yes
Key code register	Yes
Comparator	3 channels
Slope compensation (constant current unit)	1 channel
PWC	2 channels
PWM	2 channels×3pairs
NMI request function	Yes
Operation guaranteed temperature (T_A)	-40°C to +125°C
Power supply	4.5V to 5.5V
Package	LQFP-64

1.4 Function Overview

This section shows function overview of the MB91F552.

Table 1-2. Function Overview

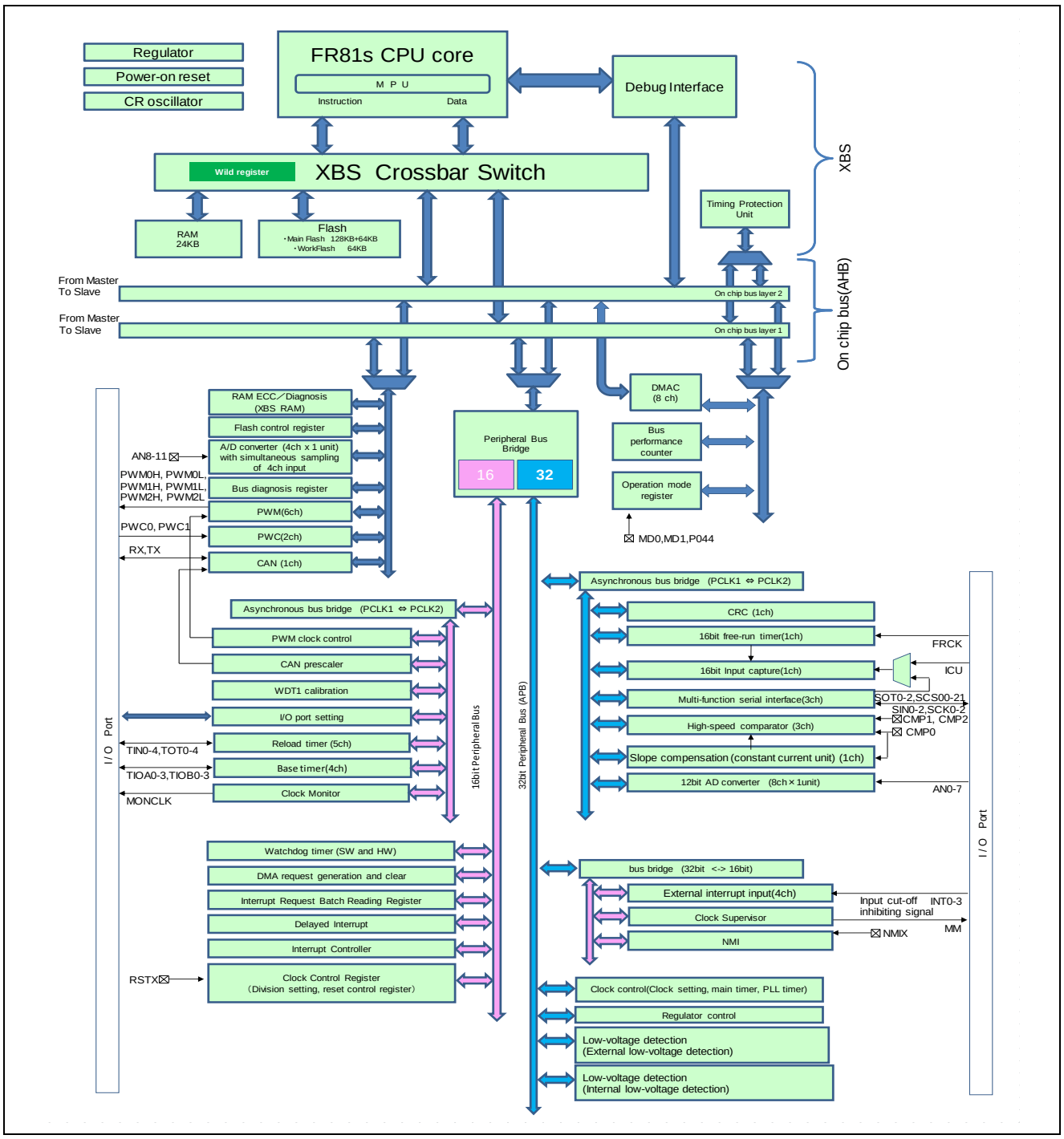
Function	Features
CPU	32-bit RISC microcontroller FR81S CPU core Built-in memory protection function (MPU) 8 channels Built-in floating-point operation (FPU)
Clock	Main oscillation: 4 MHz (up to 16 MHz can be input) PLL multiplication rate (for CPU): Up to 20 times of multiplication PLL multiplication rate (for PWM): Up to 50 times of multiplication Built-in 100 kHz CR oscillator
I/O ports	Each bit can be programmed for I/O or peripheral signals Pull-up (some pull-down) can be set.
Internal bus interface	On-chip bus: 32-bit, maximum operating frequency of 80 MHz
Peripheral bus interface	Maximum operating frequency: 40 MHz 32-bit peripheral bus, or 16-bit peripheral bus (R-bus) Note: Both of them operate in the same frequency.
Flash interface	Wild register function provided. For small sector (64 KB)
DMA controller	Up to 8 channels can be started simultaneously. The transfer factor (internal peripheral request or software) is selectable. Burst or block transfer mode is selectable. - When two or more interrupts are in one interrupt vector, it can select from which interrupt to generate the DMA request. - When two or more interrupts are in one interrupt vector, the interrupt cleared at the DMA transfer completion can be selected.
Base timer	16-bit timer Any of four PWM/PPG/PWC/reload timer functions can be selected and used. A 32-bit timer can be used in 2 channels of cascade mode for the reload timer/PWC function.
Free-run timer	16-bit up counter Free-run timer: 16-bit
Input capture	16-bit capture registers to detect a rising edge, a falling edge, or both edges. When an edge of pin input is detected, the counter value of free-run timer is latched and an interrupt request is generated. Cooperation with the free-run timer is as follows. Input capture: 16-bit -> Free-run timer Cooperation with LIN synch break/synch field is as follows. Input capture -> Selection from multi-function serial ch.0, ch.1, and ch.2
Reload timer	16-bit reload timer operation (The toggle output or one-shot output can be selected.) Event count function can be selected.
PWM	200 MHz operation - The software can change the cycle, duty, and phase shift.
PWC	80 MHz operation - Capture data has a buffer - Capture data and upper limit/lower limit value comparison functions
Delay interrupt	An interrupt for task switching is generated. The CPU interrupt request can be generated or canceled by the software.
External interrupt	4 channels, independent Interrupt factor: rising edge/falling edge/"L" level /"H" level can be selected. Support of edge input detection when returned to standby state.
A/D converter 1	The A/D converter is a single built-in unit with a 12-bit resolution. Able to sample the analog value from up to 8 channels input port. Conversion time: 12-bit A/D converter 1 µs External trigger activation Can be activated by an internal timer (16-bit reload timer/PWM are used). Has the function of selecting the sampling time for each channel.

Function	Features
A/D converter 2	The A/D converter is a single built-in unit with a 12-bit resolution. Able to sample the analog values for 4 channels simultaneously from an input port of up to 4 channels. Conversion time: 12-bit A/D converter For 1-channel conversion: 0.7 µs For 4-channel conversion: 1.75 µs
Multi-function serial	Any of UART/CSIO/LIN functions can be selected and used. Transmission FIFO memory 16-byte, and reception FIFO memory 16-byte provided Reception interrupt cause (3 types) - Reception error detection (parity, overrun, and frame error) - Detects FIFO reception of data up to an amount of its threshold. - Detects the idling period which is 8 x baud rate clock or more, when amount of the data received is less than FIFO threshold. Transmission interrupt factor (2 types) - No transmission operation. - Empty transmission FIFO memory (including the time of transmission) SPI (Serial Peripheral Interface) supported LIN protocol revision 2.1 supported
Interrupt controller	Detects an interrupt request. Sets an interrupt level.
Interrupt request batch read	A generation of multiple interrupts from peripherals can be read by a series of registers.
CAN interface	CAN Specifications Version 2.0, Part A and Part B satisfied Maximum of 64 message buffers x 1 channel supported Flexible composition of acceptance filter: Entire bit compare Entire bit mask 2 portion bit mask Up to 1 Mbps supported. CAN prescaler is mounted for the CAN operation clock CAN wakeup function CAN clock source can switch main clock/PLL clock.
Software watchdog	It counts while CPU is working. Stops counting when the CPU is stopped. The intervals can be selected from 16 types (PCLK x (2^{9} to 2^{24}) cycles). The lower limit of the term of validity to clear can be set up to 16 ways.
Hardware watchdog	CR-based CPU operation detection counter Used against program overrun Period: 218 ms to 655 ms (usually 328 ms, depending on the accuracy of the CR oscillation) Note that as shown above, a period of the CR oscillation clock varies widely due to the production process. Note that this does not necessarily mean any period can be set.
CRC generation	When data is sequentially written in the input registers, the CRC code is displayed in the result register.
Slope compensation (constant current unit)	The slope compensation voltage can be added to the DC-DC converter output voltage.
Comparator	Outputs the result of comparing a 10-bit D/A converter output value with an analog input value. Mounted channels: 3 channels
External low-voltage detection reset	Reset/interrupt generation at external low-voltage detection When an external power voltage falls below the detection voltage value, reset/interrupt is generated. Any of 7 types of detected voltage can be selected (3.7 V to 4.3 V).
Internal low-voltage detection reset	Reset generation at internal low-voltage detection Monitors 1.2 V power supply and generates the reset.
Low-power consumption mode	Sleep mode Stop mode Watch mode
NMI request	Non-maskable interrupt signal that is entered from NMIX pin.
Debug interface	Built-in OCD

1.5 Block Diagram

This section shows block diagram of the MB91F552.

MB91F552

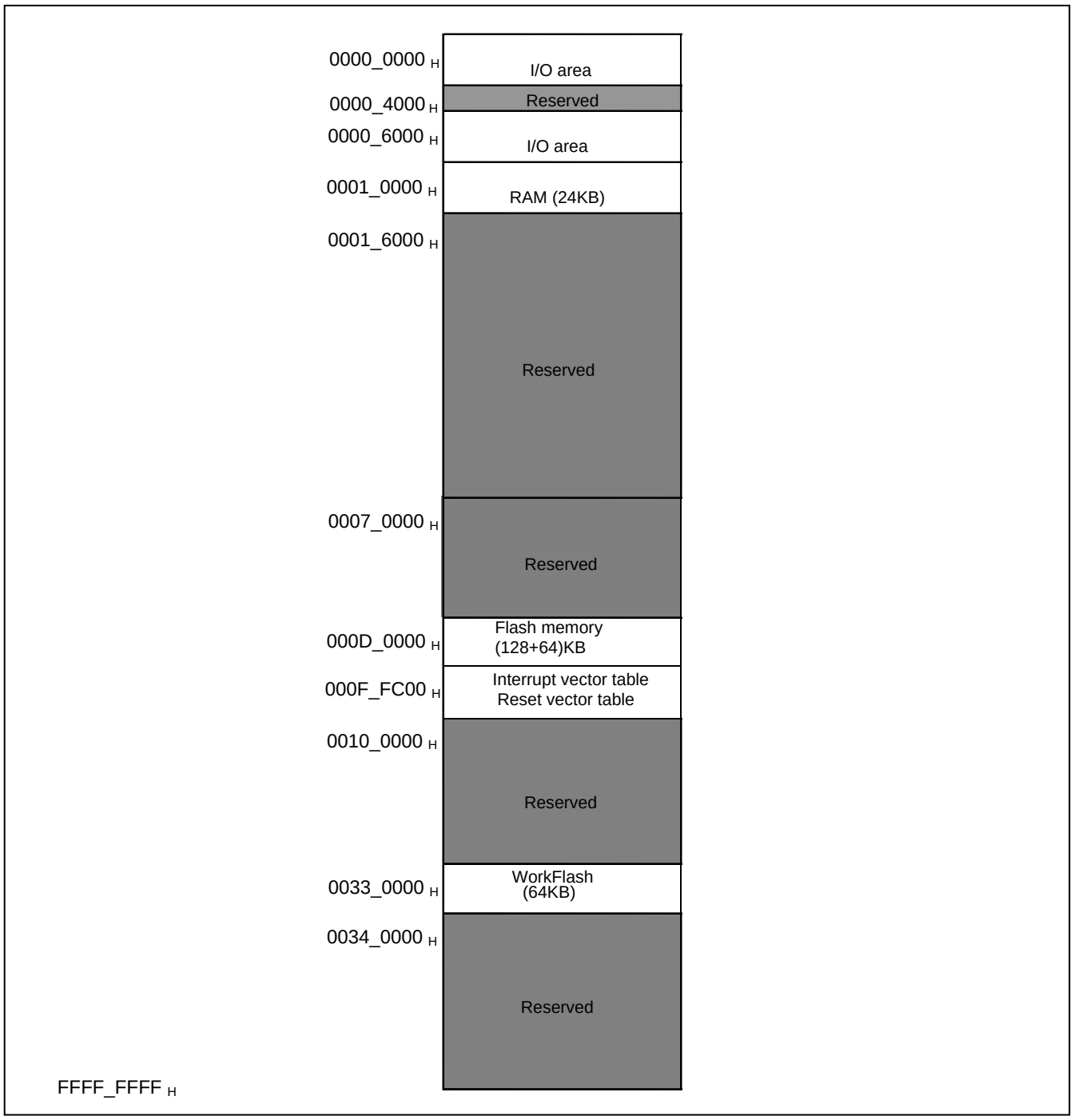


See "1.9.1 Pins of Each Function" for pins that can used by each function.

1.6 Memory Map

This section shows memory map of the MB91F552.

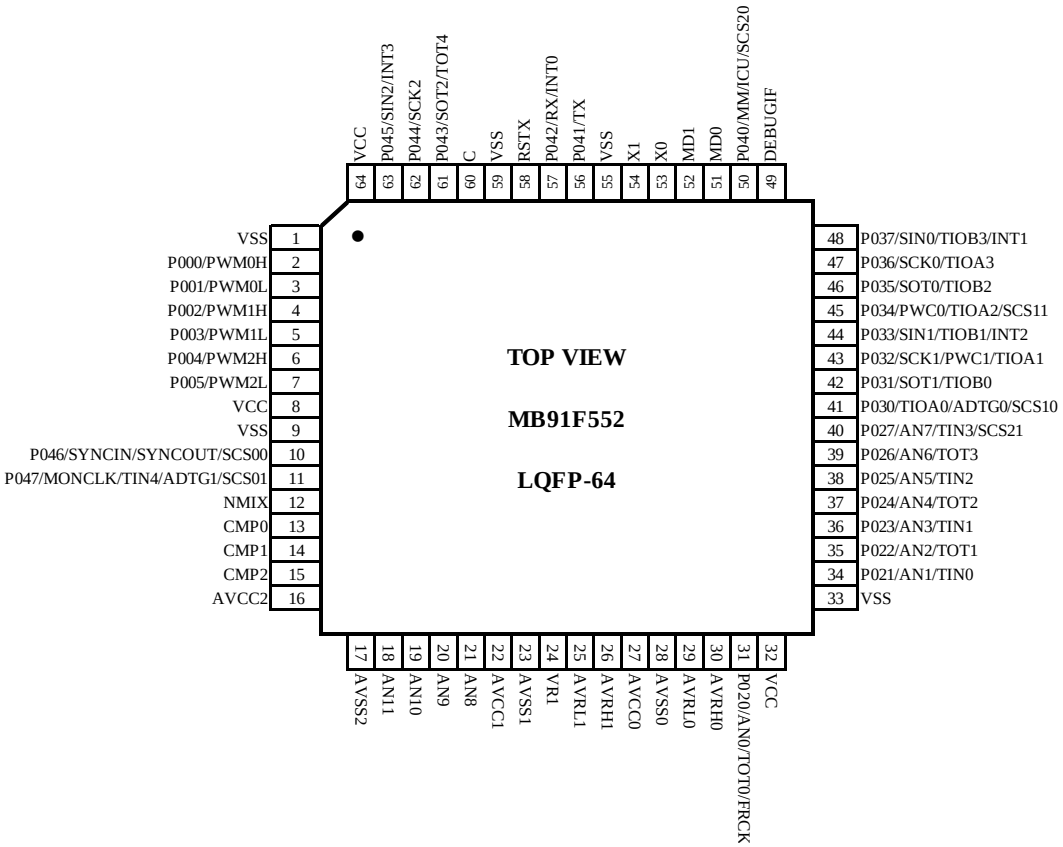
Figure 1-1. Memory Map MB91F552



1.7 Pin Assignment

This section shows pin assignment of the MB91F552.

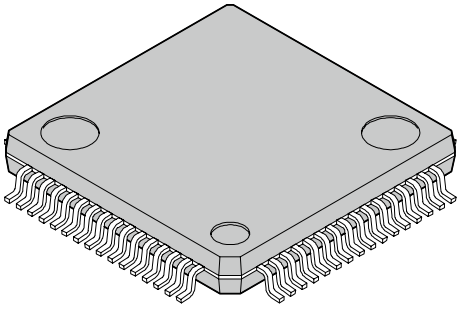
Pin Assignment MB91F552



1.8 Device Package

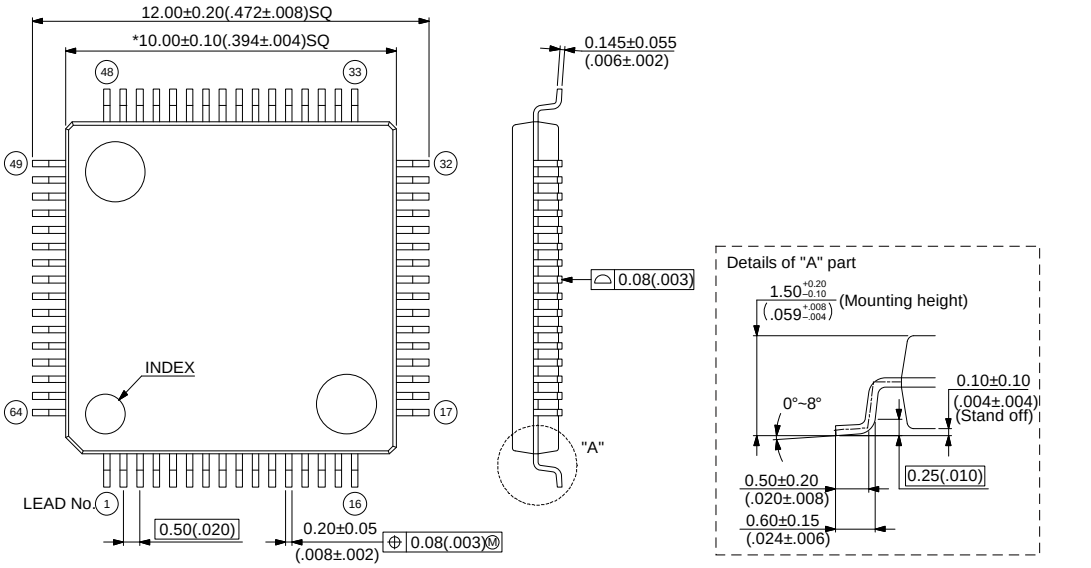
This section shows device package of the MB91F552.

Figure 1-2. LQFP-64 (FPT-64P-M24) External Dimensions

64-pin plastic LQFP		
	Lead pitch	0.50 mm
	Package width × package length	10.0 × 10.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.32 g
(FPT-64P-M24)	Code (Reference)	P-LFQFP64-10×10-0.50

64-pin plastic LQFP
(FPT-64P-M24)

Note 1) * : These dimensions do not include resin protrusion.
Note 2) Pins width and pins thickness include plating thickness.
Note 3) Pins width do not include tie bar cutting remainder.



ⓒ 2005-2010 FUJITSU SEMICONDUCTOR LIMITED F64036S-c-1-3

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

1.9 Explanation of Pin Functions

The pin function list of the MB91F552 is shown below.

Table 1-3. List of Pin Functions

Pin no.	Pin Name	Polarity	I/O circuit types	Function
2	P000 PWM0H	- -	A	General-purpose I/O port PWM ch.0-H
3	P001 PWM0L	- -	A	General-purpose I/O port PWM ch.0-L
4	P002 PWM1H	- -	A	General-purpose I/O port PWM ch.1-H
5	P003 PWM1L	- -	A	General-purpose I/O port PWM ch.1-L
6	P004 PWM2H	- -	A	General-purpose I/O port PWM ch.2-H
7	P005 PWM2L	- -	A	General-purpose I/O port PWM ch.2-L
10	P046 SYNCIN SYNCOUT SCS00	- - - -	B	General-purpose I/O port Master/Slave input pin for PWM parallel operation drive Master/Slave output pin for PWM parallel operation drive Multi-function serial ch.0 serial chip select 00 I/O pin
11	P047 MONCLK TIN4 ADTG1 SCS01	- - - - -	B	General-purpose I/O port Clock monitor output pin Reload timer ch.4 event input pin A/D converter ch.8-ch.11 external trigger input pin Multi-function serial ch.0 serial chip select 01 output pin
12	NMIX	N	L	Interrupt input pin without mask
13	CMP0	-	C	Comparator ch.0 input pin
14	CMP1	-	C	Comparator ch.1 input pin
15	CMP2	-	C	Comparator ch.2 input pin
18	AN11	-	C	A/D converter ch.11 analog input pin (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)
19	AN10	-	C	A/D converter ch.10 analog input pin (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)
20	AN9	-	C	A/D converter ch.9 analog input pin (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)
21	AN8	-	C	A/D converter ch.8 analog input pin (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)
31	P020 AN0 TOT0 FRCK	- - - -	D	General-purpose I/O port A/D converter ch.0 analog input pin Reload timer ch.0 output pin Free-run timer clock input pin
34	P021 AN1 TIN0	- - -	D	General-purpose I/O port A/D converter ch.1 analog input pin Reload timer ch.0 event input pin
35	P022 AN2 TOT1	- - -	D	General-purpose I/O port A/D converter ch.2 analog input pin Reload timer ch.1 output pin
36	P023 AN3 TIN1	- - -	D	General-purpose I/O port A/D converter ch.3 analog input pin Reload timer ch.1 event input pin

Pin no.	Pin Name	Polarity	I/O circuit types	Function
37	P024 AN4 TOT2	- - -	D	General-purpose I/O port A/D converter ch.4 analog input pin Reload timer ch.2 output pin
38	P025 AN5 TIN2	- - -	D	General-purpose I/O port A/D converter ch.5 analog input pin Reload timer ch.2 event input pin
39	P026 AN6 TOT3	- - -	D	General-purpose I/O port A/D converter ch.6 analog input pin Reload timer ch.3 output pin
40	P027 AN7 TIN3 SCS21	- - - -	D	General-purpose I/O port A/D converter ch.7 analog input pin Reload timer ch.3 event input pin Multi-function serial ch.2 serial chip select 21 output pin
41	P030 TIOA0 ADTG0 SCS10	- - - -	B	General-purpose I/O port Base timer ch.0 TIOA output pin A/D converter ch.0-ch.7 external trigger input pin Multi-function serial ch.1 serial chip select 10 I/O pin
42	P031 SOT1 TIOB0	- - -	B	General-purpose I/O port Multi-function serial ch.1 serial data output pin Base timer ch.0 TIOB input pin
43	P032 SCK1 PWC1 TIOA1	- - - -	E	General-purpose I/O port Multi-function serial ch.1 clock I/O pin PWC ch.1 input pin Base timer ch.1 TIOA I/O pin
44	P033 SIN1 TIOB1 INT2	- - - -	E	General-purpose I/O port Multi-function serial ch.1 serial data input pin Base timer ch.1 TIOB input pin INT2 external interrupt input pin
45	P034 PWC0 TIOA2 SCS11	- - - -	B	General-purpose I/O port PWC ch.0 input pin Base timer ch.2 TIOA output pin Multi-function serial ch.1 serial chip select 11 output pin
46	P035 SOT0 TIOB2	- - -	B	General-purpose I/O port Multi-function serial ch.0 serial data output pin Base timer ch.2 TIOB input pin
47	P036 SCK0 TIOA3	- - -	E	General-purpose I/O port Multi-function serial ch.0 clock I/O pin Base timer ch.3 TIOA I/O pin
48	P037 SIN0 TIOB3 INT1	- - - -	E	General-purpose I/O port Multi-function serial ch.0 serial data input pin Base timer ch.3 TIOB input pin INT1 external interrupt input pin
49	DEBUGIF	-	F	MDI I/O pin for debug (OCD)
50	P040 MM ICU SCS20	- - - -	B	General-purpose I/O port Clock supervisor main clock stop detection output pin Input capture input pin Multi-function serial ch.2 serial chip select 20 I/O pin
51	MD0	-	G	Mode pin 0
52	MD1	-	G	Mode pin 1
53	X0	-	H	Main clock oscillation input pin
54	X1	-	H	Main clock oscillation output pin
56	P041 TX	- -	B	General-purpose I/O port CAN transmission data output pin
57	P042 RX INT0	- - -	E	General-purpose I/O port CAN reception data input pin INT0 external interrupt input pin
58	RSTX	N	L	External reset input pin

Pin no.	Pin Name	Polarity	I/O circuit types	Function
61	P043 SOT2 TOT4	- - -	B	General-purpose I/O port Multi-function serial ch.2 serial data output pin Reload timer ch.4 output pin
62	P044 SCK2	- -	E	General-purpose I/O port Multi-function serial ch.2 clock I/O pin
63	P045 SIN2 INT3	- - -	E	General-purpose I/O port Multi-function serial ch.2 serial data input pin INT3 external interrupt input pin
16	AVCC2	-	-	Analog power supply pin for comparator and slope compensation circuit
22	AVCC1	-	-	Analog power supply pin for A/D converter, with 4ch. simultaneous sampling/hold function
27	AVCC0	-	-	A/D converter analog power supply pin
26	AVRH1	-	-	Upper limit reference voltage pin for A/D converter, with 4ch. simultaneous sampling/hold function
30	AVRH0	-	-	A/D converter upper limit reference voltage pin
17	AVSS2	-	-	GND pin for comparator and slope compensation circuit
23	AVSS1	-	-	GND pin for A/D converter, with 4ch. simultaneous sampling/hold function
25	AVRL1	-	-	Lower limit reference voltage pin for A/D converter, with 4ch. simultaneous sampling/hold function
24	VR1	-	-	Intermediate reference voltage pin for A/D converter, with 4ch. simultaneous sampling/hold function
28	AVSS0	-	-	A/D converter GND pin
29	AVRL0	-	-	A/D converter lower limit reference voltage pin
60	C	-	-	External capacity connection output pin
8 32 64	VCC	-	-	+5.0 V power supply pin
1 9 33 55 59	VSS	-	-	GND

1.9.1 Pins of Each Function

Pins of each function are shown below.

Table 1-4. Pins of A/D Converter (ch.0 to ch.11)

Function	Pin Name	Noise Filter	Pin Number
A/DC ch.8-ch.11 external trigger input pin	ADTG1	Yes	11
A/DC ch.0-ch.7 external trigger input pin	ADTG0	Yes	41
A/D converter analog input ch.0 pin	AN0	No	31
A/D converter analog input ch.1 pin	AN1	No	34
A/D converter analog input ch.2 pin	AN2	No	35
A/D converter analog input ch.3 pin	AN3	No	36
A/D converter analog input ch.4 pin	AN4	No	37
A/D converter analog input ch.5 pin	AN5	No	38
A/D converter analog input ch.6 pin	AN6	No	39
A/D converter analog input ch.7 pin	AN7	No	40
A/D converter analog input ch.8 pin	AN8	No	21
A/D converter analog input ch.9 pin	AN9	No	20
A/D converter analog input ch.10 pin	AN10	No	19
A/D converter analog input ch.11 pin	AN11	No	18
Analog power supply pin for A/D converter, with 4-channel simultaneous sampling/hold function	AVCC1	-	22
A/D converter analog power supply pin	AVCC0	-	27
Upper limit reference voltage pin for A/D converter, with 4-channel simultaneous sampling/hold function	AVRH1	-	26
A/D converter upper limit reference voltage pin	AVRH0	-	30
GND pin for A/D converter, with 4-channel simultaneous sampling/hold function	AVSS1	-	23
Lower limit reference voltage pin for A/D converter, with 4-channel simultaneous sampling/hold function	AVRL1	-	25
A/D converter GND pin	AVSS0	-	28
A/D converter lower limit reference voltage pin	AVRL0	-	29
Intermediate reference voltage pin for A/D converter, with 4-channel simultaneous sampling/hold function	VR1	-	24

Table 1-5. Pins of CAN

Function	Pin Name	Noise Filter	Pin Number
CAN reception data 0 input pin	RX	No	57
CAN transmission data 2 output pin	TX	-	56

Table 1-6. Pins of External Interrupt Input

Function	Pin Name	Noise Filter	Pin Number
INT0 external interrupt input pin	INT0	Yes	57
INT1 external interrupt input pin	INT1	Yes	48
INT2 external interrupt input pin	INT2	Yes	44
INT3 external interrupt input pin	INT3	Yes	63

Table 1-7. Pins of Multi-function Serial Interface (ch.0 to ch.2)

Function	Pin Name	Noise Filter	Pin Number
MFS ch.0 chip select 00 I/O pin	SCS00	No	10
MFS ch.0 chip select 01 output pin	SCS01	No	11
MFS ch.0 clock I/O pin	SCK0	No	47
MFS ch.0 serial data output pin	SOT0	-	46
MFS ch.0 serial data input pin	SIN0	No	48
MFS ch.1 chip select 10 I/O pin	SCS10	No	41
MFS ch.1 chip select 11 output pin	SCS11	No	45
MFS ch.1 clock I/O pin	SCK1	No	43
MFS ch.1 serial data output pin	SOT1	-	42
MFS ch.1 serial data input pin	SIN1	No	44
MFS ch.2 chip select 20 I/O pin	SCS20	No	50
MFS ch.2 chip select 21 output pin	SCS21	No	40
MFS ch.2 clock I/O pin	SCK2	No	62
MFS ch.2 serial data output pin	SOT2	-	61
MFS ch.2 serial data input pin	SIN2	No	63

Table 1-8. Pins of Input Capture

Function	Pin Name	Noise Filter	Pin Number
Input capture input pin	ICU	Yes	50

Table 1-9. Pin of Free-run Timer

Function	Pin Name	Noise Filter	Pin Number
Free-run timer clock input pin	FRCK	Yes	31

Table 1-10. Pins of Base Timer (ch.0 to ch.3)

Function	Pin Name	Noise Filter	Pin Number
Base timer ch.0 TIOA output pin	TIOA0	-	41
Base timer ch.0 TIOB input pin	TIOB0	Yes	42
Base timer ch.1 TIOA I/O pin	TIOA1	Yes	43
Base timer ch.1 TIOB input pin	TIOB1	Yes	44
Base timer ch.2 TIOA output pin	TIOA2	-	45
Base timer ch.2 TIOB input pin	TIOB2	Yes	46
Base timer ch.3 TIOA I/O pin	TIOA3	Yes	47
Base timer ch.3 TIOB input pin	TIOB3	Yes	48

Table 1-11. Pins of Reload Timer (ch.0 to ch.4)

Function	Pin Name	Noise Filter	Pin Number
Reload timer ch.0 output pin	TOT0	-	31
Reload timer ch.0 event input pin	TIN0	Yes	34
Reload timer ch.1 output pin	TOT1	-	35
Reload timer ch.1 event input pin	TIN1	Yes	36
Reload timer ch.2 event output pin	TOT2	-	37
Reload timer ch.2 event input pin	TIN2	Yes	38
Reload timer ch.3 output pin	TOT3	-	39
Reload timer ch.3 event input pin	TIN3	Yes	40
Reload timer ch.4 output pin	TOT4	-	61
Reload timer ch.4 event input pin	TIN4	Yes	11

Table 1-12. Pins of PWM

Function	Pin Name	Noise Filter	Pin Number
PWM ch.0-H	PWM0H	-	2
PWM ch.0-L	PWM0L	-	3
PWM ch.1-H	PWM1H	-	4
PWM ch.1-L	PWM1L	-	5
PWM ch.2-H	PWM2H	-	6
PWM ch.2-L	PWM2L	-	7
Master/Slave input pin for PWM parallel operation drive	SYNCIN	No	10
Master/Slave output pin for PWM parallel operation drive	SYNCOUT	-	10

Table 1-13. Pins of PWC

Function	Pin Name	Noise Filter	Pin Number
PWC ch.0 input pin	PWC0	No	45
PWC ch.1 input pin	PWC1	No	43

Table 1-14. Pins of Comparator

Function	Pin Name	Noise Filter	Pin Number
Comparator input ch.0 pin	CMP0	No	13
Comparator input ch.1 pin	CMP1	No	14
Comparator input ch.2 pin	CMP2	No	15
Comparator and slope compensation circuit pin Analog power supply pin	AVCC2	-	16
Comparator and slope compensation circuit GND pin	AVSS2	-	17

Table 1-15. Pin of Clock Monitor

Function	Pin Name	Noise Filter	Pin Number
Clock monitor output pin	MONCLK	-	11

Table 1-16. Pins of Port Function (General-Purpose I/O)

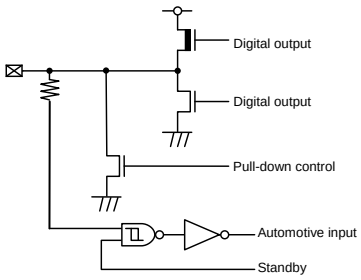
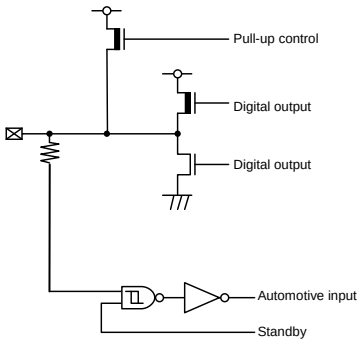
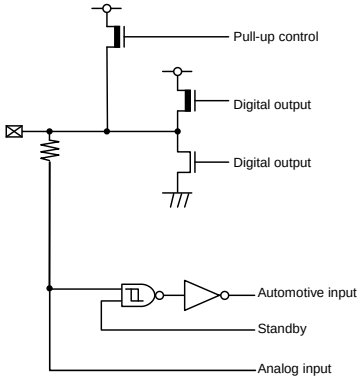
Function	Pin Name	Noise Filter	Pin Number
General-purpose I/O port	P000	Yes	2
General-purpose I/O port	P001	Yes	3
General-purpose I/O port	P002	Yes	4
General-purpose I/O port	P003	Yes	5
General-purpose I/O port	P004	Yes	6
General-purpose I/O port	P005	Yes	7
General-purpose I/O port	P020	Yes	31
General-purpose I/O port	P021	Yes	34
General-purpose I/O port	P022	Yes	35
General-purpose I/O port	P023	Yes	36
General-purpose I/O port	P024	Yes	37
General-purpose I/O port	P025	Yes	38
General-purpose I/O port	P026	Yes	39
General-purpose I/O port	P027	Yes	40
General-purpose I/O port	P030	Yes	41
General-purpose I/O port	P031	Yes	42
General-purpose I/O port	P032	Yes	43
General-purpose I/O port	P033	Yes	44
General-purpose I/O port	P034	Yes	45
General-purpose I/O port	P035	Yes	46
General-purpose I/O port	P036	Yes	47
General-purpose I/O port	P037	Yes	48
General-purpose I/O port	P040	Yes	50
General-purpose I/O port	P041	Yes	56
General-purpose I/O port	P042	Yes	57
General-purpose I/O port	P043	Yes	61
General-purpose I/O port	P044	Yes	62
General-purpose I/O port	P045	Yes	63
General-purpose I/O port	P046	Yes	10
General-purpose I/O port	P047	Yes	11

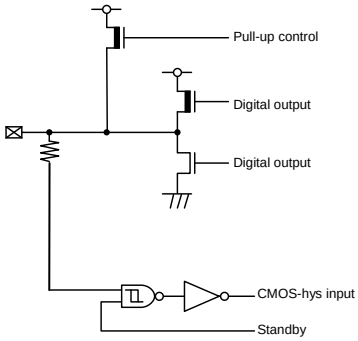
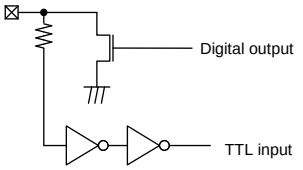
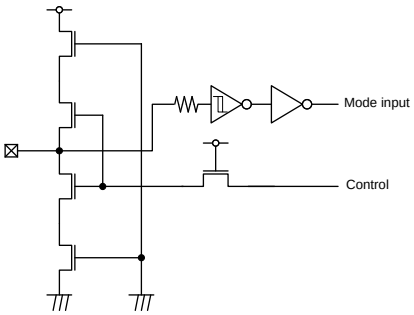
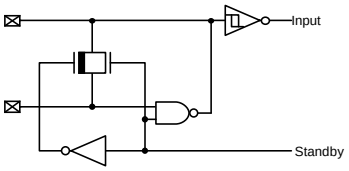
Table 1-17. Other Pins

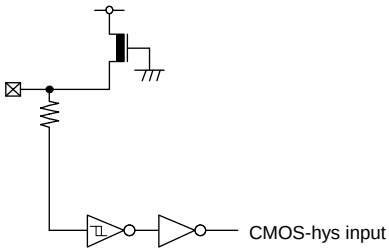
Function	Pin Name	Noise Filter	Pin Number
+5.0 V power supply pin	VCC	-	8
		-	32
		-	64
GND	VSS	-	1
		-	9
		-	33
		-	55
		-	59
External capacity connection output pin	C	-	60
Main clock oscillation input pin	X0	Yes	53
Main clock oscillation output pin	X1	-	54
Mode pin 0	MD0	-	51
Mode pin 1	MD1	-	52
Clock supervisor pin Main clock stop detection output pin	MM	-	50
Interrupt input pin without mask	NMIX	Yes	12
DEBUGIF I/O pin for debug (OCD)	DEBUGIF	Yes	49
External reset input pin	RSTX	Yes	58

1.10 I/O Circuit Types

This section shows I/O circuit types.

Type	Circuit	Remarks
A		■ General-purpose I/O port ■ Output 8mA ■ Pull-down resistor control 50kΩ ■ Automotive input
B		■ General-purpose I/O port ■ Output 4mA ■ Pull-up resistor control 50kΩ ■ Automotive input
C		■ Analog input
D		■ Analog input, General-purpose I/O port ■ Output 4mA ■ Pull-up resistor control 50kΩ ■ Automotive input

Type	Circuit	Remarks
E		■ General-purpose I/O port ■ Output 4mA ■ Pull-up resistor control 50kΩ ■ CMOS hysteresis input
F		■ Open-drain I/O ■ Output 25mA (Nch open drain) ■ TTL input
G		■ Mode I/O ■ CMOS hysteresis input
H		■ Main oscillation I/O

Type	Circuit	Remarks
L		■ CMOS hysteresis input ■ Pull-up resistor 50kΩ

2. Handling the Device

This chapter explains the notes on using this series.

2.1 Handling Precautions

2.2 Handling Device

2.3 Application Notes

2.1 Handling Precautions

Any semiconductor devices have inherently a certain rate of failure. The possibility of failure is greatly affected by the conditions in which they are used (circuit conditions, environmental conditions, etc.). This page describes precautions that must be observed to minimize the chance of failure and to obtain higher reliability from your Cypress semiconductor devices.

1. Precautions for Product Design

This section describes precautions when designing electronic equipment using semiconductor devices.

Absolute Maximum Ratings

Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of certain established limits, called absolute maximum ratings. Do not exceed these ratings.

Recommended Operating Conditions

Recommended operating conditions are normal operating ranges for the semiconductor device. All the device's electrical characteristics are warranted when operated within these ranges.

Always use semiconductor devices within the recommended operating conditions. Operation outside these ranges may adversely affect reliability and could result in device failure.

No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their sales representative beforehand.

Processing and Protection of Pins

These precautions must be followed when handling the pins which connect semiconductor devices to power supply and input/output functions.

1. Preventing Over-Voltage and Over-Current Conditions

   Exposure to voltage or current levels in excess of maximum ratings at any pin is likely to cause deterioration within the device, and in extreme cases leads to permanent damage of the device. Try to prevent such overvoltage or over-current conditions at the design stage.

2. Protection of Output Pins

   Shorting of output pins to supply pins or other output pins, or connection to large capacitance can cause large current flows. Such conditions if present for extended periods of time can damage the device.
   Therefore, avoid this type of connection.

3. Handling of Unused Input Pins

   Unconnected input pins with very high impedance levels can adversely affect stability of operation. Such pins should be connected through an appropriate resistance to a power supply pin or ground pin.

Latch-up

Semiconductor devices are constructed by the formation of P-type and N-type areas on a substrate. When subjected to abnormally high voltages, internal parasitic PNPN junctions (called thyristor structures) may be formed, causing large current levels in excess of several hundred mA to flow continuously at the power supply pin. This condition is called latch-up.

CAUTION: The occurrence of latch-up not only causes loss of reliability in the semiconductor device, but can cause injury or damage from high heat, smoke or flame. To prevent this from happening, do the following:

1. Be sure that voltages applied to pins do not exceed the absolute maximum ratings. This should include attention to abnormal noise, surge levels, etc.
2. Be sure that abnormal current flows do not occur during the power-on sequence.

Observance of Safety Regulations and Standards

Most countries in the world have established standards and regulations regarding safety, protection from electromagnetic interference, etc. Customers are requested to observe applicable regulations and standards in the design of products.

Fail-Safe Design

Any semiconductor devices have inherently a certain rate of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

Precautions Related to Usage of Devices

Cypress semiconductor devices are intended for use in standard applications (computers, office automation and other office equipment, industrial, communications, and measurement equipment, personal or household devices, etc.).

CAUTION: Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.

2. Precautions for Package Mounting

Package mounting may be either lead insertion type or surface mount type. In either case, for heat resistance during soldering, you should only mount under Cypress recommended conditions. For detailed information about mount conditions, contact your sales representative.

Lead Insertion Type

Mounting of lead insertion type packages onto printed circuit boards may be done by two methods: direct soldering on the board, or mounting by using a socket.

Direct mounting onto boards normally involves processes for inserting leads into through-holes on the board and using the flow soldering (wave soldering) method of applying liquid solder. In this case, the soldering process usually causes leads to be subjected to thermal stress in excess of the absolute ratings for storage temperature. Mounting processes should conform to Cypress recommended mounting conditions.

If socket mounting is used, differences in surface treatment of the socket contacts and IC lead surfaces can lead to contact deterioration after long periods. For this reason it is recommended that the surface treatment of socket contacts and IC leads be verified before mounting.

Surface Mount Type

Surface mount packaging has longer and thinner leads than lead-insertion packaging, and therefore leads are more easily deformed or bent. The use of packages with higher pin counts and narrower pin pitch results in increased susceptibility to open connections caused by deformed pins, or shorting due to solder bridges.

You must use appropriate mounting techniques. Cypress recommends the solder reflow method, and has established a ranking of mounting conditions for each product. Users are advised to mount packages in accordance with Cypress ranking of recommended conditions.

Lead-Free Packaging

CAUTION: When ball grid array (BGA) packages with Sn-Ag-Cu balls are mounted using Sn-Pb eutectic soldering, junction strength may be reduced under some conditions of use.

Storage of Semiconductor Devices

Because plastic chip packages are formed from plastic resins, exposure to natural environmental conditions will cause absorption of moisture. During mounting, the application of heat to a package that has absorbed moisture can cause surfaces to peel, reducing moisture resistance and causing packages to crack. To prevent, do the following:

1. Avoid exposure to rapid temperature changes, which cause moisture to condense inside the product. Store products in locations where temperature changes are slight.
2. Use dry boxes for product storage. Products should be stored below 70% relative humidity, and at temperatures between 5°C and 30°C.
   When you open Dry Package that recommends humidity 40% to 70% relative humidity.
3. When necessary, Cypress packages semiconductor devices in highly moisture-resistant aluminum laminate bags, with a silica gel desiccant. Devices should be sealed in their aluminum laminate bags for storage.
4. Avoid storing packages where they are exposed to corrosive gases or high levels of dust.

Baking

Packages that have absorbed moisture may be de-moisturized by baking (heat drying). Follow the Cypress recommended conditions for baking.

Condition: 125°C/24 h

Static Electricity

Because semiconductor devices are particularly susceptible to damage by static electricity, you must take the following precautions:

1. Maintain relative humidity in the working environment between 40% and 70%. Use of an apparatus for ion generation may be needed to remove electricity.
2. Electrically ground all conveyors, solder vessels, soldering irons and peripheral equipment.
3. Eliminate static body electricity by the use of rings or bracelets connected to ground through high resistance (on the level of 1 MΩ).
   Wearing of conductive clothing and shoes, use of conductive floor mats and other measures to minimize shock loads is recommended.
4. Ground all fixtures and instruments, or protect with anti-static measures.
5. Avoid the use of styrofoam or other highly static-prone materials for storage of completed board assemblies.

3. Precautions for Use Environment

Reliability of semiconductor devices depends on ambient temperature and other conditions as described above.

For reliable performance, do the following:

1. Humidity

   Prolonged use in high humidity can lead to leakage in devices as well as printed circuit boards. If high humidity levels are anticipated, consider anti-humidity processing.

2. Discharge of Static Electricity

   When high-voltage charges exist close to semiconductor devices, discharges can cause abnormal operation. In such cases, use anti-static measures or processing to prevent discharges.

3. Corrosive Gases, Dust, or Oil

   Exposure to corrosive gases or contact with dust or oil may lead to chemical reactions that will adversely affect the device. If you use devices in such conditions, consider ways to prevent such exposure or to protect the devices.

4. Radiation, Including Cosmic Radiation

   Most devices are not designed for environments involving exposure to radiation or cosmic radiation. Users should provide shielding as appropriate.

5. Smoke, Flame

   CAUTION: Plastic molded devices are flammable, and therefore should not be used near combustible substances. If devices begin to smoke or burn, there is danger of the release of toxic gases.

Customers considering the use of Cypress products in other special environmental conditions should consult with sales representatives.

2.2 Handling Device

This section explains the handling device.

Notes on handling device

This section explains the latch-up prevention and pin processing.

For latch-up prevention

If a voltage higher than VCC or a voltage lower than VSS is applied to an I/O pin, or if a voltage exceeding the ratings is applied between VCC and VSS pins, a latch-up may occur in CMOS IC. If the latch-up occurs, the power supply current increases excessively and device elements may be damaged by heat. Take care to prevent any voltage from exceeding the maximum ratings in device application.

Also, the analog power supply (AVCC, AVRH) and analog input must not be exceed the digital power supply (VCC) when the power supply to the analog system is turned on or off.

In the correct power-on sequence of the microcontroller, turn on the digital power supply (VCC) and analog power supplies (AVCC, AVRH) simultaneously. Or, turn on the digital power supply (VCC), and then turn on analog power supplies (AVCC, AVRH).

Treatment of unused pins

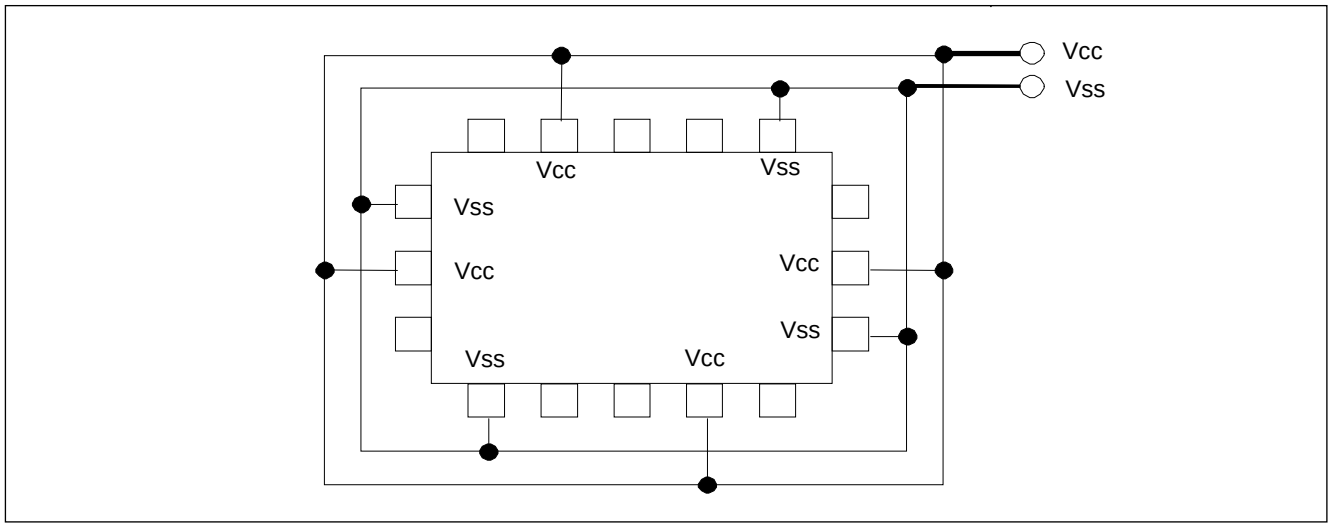
If unused input pins are left open, they may cause a permanent damage to the device due to malfunction or latch-up. Connect at least a 2kΩ resistor to each of the unused pins for pull-up or pull-down processing.

Also, if I/O pins are not used, they must be set to the output state for releasing or they must be set to the input state and treated in the same way as for the input pins.

Power supply pins

The device is designed to ensure that if the device contains multiple VCC or VSS pins, the pins that should be at the same potential are interconnected to prevent latch-up or other malfunctions. Further, connect these pins to an external power supply or ground to reduce unwanted radiation, prevent strobe signals from malfunctioning due to a raised ground level, and fulfill the total output current standard, etc. As shown in Figure 2-1, all Vss power supply pins must be treated in the similar way. If multiple Vcc or Vss systems are connected, the device cannot operate correctly even within the guaranteed operating range.

Figure 2-1. Power Supply Input Pins



The power supply pins should be connected to VCC and VSS pins of this device at the low impedance from the power supply source.

In the area close to this device, a ceramic capacitor having the capacitance larger than the capacitor of C pin is recommended to use as a bypass capacitor between VCC and VSS pins.

Crystal oscillation circuit

An external noise to the X0 or X1 pin may cause a device malfunction. The printed circuit board must be designed to lay out X0 and X1 pins, crystal oscillator (or ceramic resonator), and the bypass capacitor to be grounded to the close position to the device.

The printed circuit board artwork is recommended to surround the X0 and X1 pins by ground circuits.

Mode pins (MD1, MD0)

Connect the MD1 and MD0 mode pins to the VCC or VSS pin directly. To prevent an erroneous selection of test mode caused by the noise, reduce the pattern length between each mode pin and VCC or VSS pin on the printed circuit board. Also, use the low-impedance pin connection.

During power-on

To prevent a malfunction of the voltage step-down circuit built in the device, set the voltage rising time to have 50µs or longer (between 0.2V and 2.7V) during power-on.

Notes during PLL clock operation

When the PLL clock is selected and if the oscillator is disconnected or if the input is stopped, this clock may continue to operate at the free running frequency of the self-oscillator circuit built in the PLL clock. This operation is not guaranteed.

Treatment of A/D converter power supply pins

Connect the pins to have AVCC=AVRH=VCC and AVSS/AVRL=VSS even if the A/D converter is not used.

External clock is not supported

None of the external direct clock input can be used.

Power-on sequence of A/D converter power supplies and analog inputs

Be sure to turn on the digital power supply (Vcc) first, and then turn on the A/D converter power supplies (AVcc, AVRH, AVRL) and analog inputs (AN0 to AN7, AN8 to AN11). Also, turn off the A/D converter power supplies and analog inputs first, and then turn off the digital power supply (Vcc). When the AVRH pin voltage is turned on or off, it must not exceed AVCC. Even if a common analog input pin is used as an input port, its input voltage must not exceed AVcc. (However, the analog power supply and digital power supply can be turned on or off simultaneously.)

Power-on sequence of comparator and slope compensation power supply and analog inputs

Be sure to turn on the digital power supply (Vcc) first, and then turn on the comparator and slope compensation power supply (AVcc) and analog inputs (CMP0 to CMP2). Also, turn off the comparator and slope compensation power supply and analog inputs first, and then turn off the digital power supply (Vcc). (However, the analog power supply and digital power supply can be turned on or off simultaneously.)

Treatment of C pin

This device contains a voltage step-down circuit. A capacitor must always be connected to the C pin to assure the internal stabilization of the device. For the standard values, see the "Recommended Operating Conditions" of the latest data sheet.

Note: Please see the latest data sheet for a detailed specification of the operation voltage**.**

2.3 Application Notes

This section explains application notes.

2.3.1 Function Switching of a Multiplexed Port

2.3.2 Low-power Consumption Mode

2.3.3 Notes When Writing Data in a Register that Includes the Status Flag

2.3.1 Function Switching of a Multiplexed Port

Function switching of a multiplexed port is shown.

To switch between the PORT function and the multiplexed pin function, use the PFR (port function register). For details, see "Chapter: I/O Ports."

2.3.2 Low-power Consumption Mode

This section explains low-power consumption mode.

To transit to the sleep mode, watch mode, or stop mode, follow the procedure explained in the "Activating the sleep mode, watch mode, or stop mode" of "Chapter: Power Consumption Control."

Take the following notes when using a monitor debugger.

- Do not set a break point when the low-power consumption transition program operate.
- Do not execute an operation step when the low-power consumption transition program operate.

2.3.3 Notes When Writing Data in a Register that Includes the Status Flag

This section explains notes when writing data in a register that includes the status flag.

When writing data in a register that has a status flag (especially, an interrupt request flag) to control a function, it is important that care be taken to avoid erroneously clearing the status flag.

In other words, exercise caution when writing data so that the flag is not cleared for the status bit and the control bits have the desired value.

Especially, since the bit instruction cannot be used when the control bits are configured using multiple bits (the bit instruction can access a single bit only), data is written to the control bits and status flag simultaneously via Byte, Half-word, or Word access. However, during this time, take care not to erroneously clear any other non-targeted bits (in this case, the status flag bits).

Note: With the bit instruction, there is no need to exercise caution because it takes this point into account.

3. CPU

This chapter explains the CPU.

3.1 Overview

3.2 Features

3.3 CPU Operating Description

3.4 Pipeline Operation

3.5 Floating Point Operation Processing

3.6 Data Structure

3.7 Addressing

3.8 Programming Model

3.9 Reset and EIT Processing

3.10 Memory Protection Function (MPU)

3.1 Overview

This section explains the overview of the CPU.

The FR81 architecture is a microcontroller architecture that uses the FR family instruction set with improved floating point functionality, memory protection functionality and on-chip debugging functionality.

The integer family instruction set is compatible with the FR80 series.

For details, see "FR Family FR81 32-bit Microcontroller Programming Manual."

3.2 Features

This section explains features of the CPU.

The FR family is a CPU core for 32-bit RISC-based controllers equipped with a custom Cypress architecture. In particular, this architecture is optimal as the CPU core in microcontrollers designed for embedded control applications that require high-speed control.

General

- General-purpose register architecture (32-bit x 16)
- 32-bit address space (4 GB)
- 16-bit fixed instruction length (excluding immediate data transfer instructions)
- High-speed processing of basic instructions at one instruction per cycle using a 5-stage pipeline architecture
- 32-bit x 32-bit multiplication instruction that completes in 5 cycles
- 32-bit/32-bit division instruction by stepped division
- Direct addressing instructions for accessing peripherals
- High-speed interrupt processing that finishes in six cycles
- Single precision floating point arithmetic instructions
- Floating point register 32-bit 16
- Privilege mode and user mode
- FPU, instruction access, and data access exception functions
  - ☐ FPU exceptions
  - ☐ Instruction access protection violation exception
  - ☐ Data access protection violation exception
  - ☐ Illegal instruction exception (changed from undefined instruction exception)
  - ☐ Data access error exception
  - ☐ Non-existent FPU exception

Memory Protection Function (MPU)

- Eight protection areas can be specified common to instructions and data
- The protection areas are determined in a fixed order of precedence. (The areas can overlap)
- Areas are specified by a page address and a page size
  - ☐ Page size: Can be specified as 2^n bytes from 16 bytes
  - ☐ Page address: Misaligned address also supported
- The following access privileges are controlled using privilege mode and user mode
  - ☐ Instruction fetch (execution) permitted/forbidden
  - ☐ Read permitted/forbidden
  - ☐ Write permitted/forbidden
- The following attributes can be specified for each area
  - ☐ Bufferable/Non-bufferable

- Access privileges and attributes can be specified for unset areas
- On protection violation, an instruction access protection violation exception or data access protection violation exception occurs

Floating Point Operations

- IEEE754 compliant
- Support single precision
- Six exception sources are supported.
  - ☐ Underflow
  - ☐ Overflow
  - ☐ Division-by-zero
  - ☐ Invalid operation
  - ☐ Inexact
  - ☐ Inputs an denormalized number
  - ☐ The only rounding mode supported is nearest value
  - ☐ Denormalized numbers are truncated to 0 or generate an exception
  - ☐ Floating-point register: 32-bit x 16 sets
  - ☐ Multiply and Add, Multiply and Sub instructions supported
  - ☐ Division and square root operations supported

3.3 CPU Operating Description

This section explains the operation of the CPU.

3.3.1 CPU Operating Status

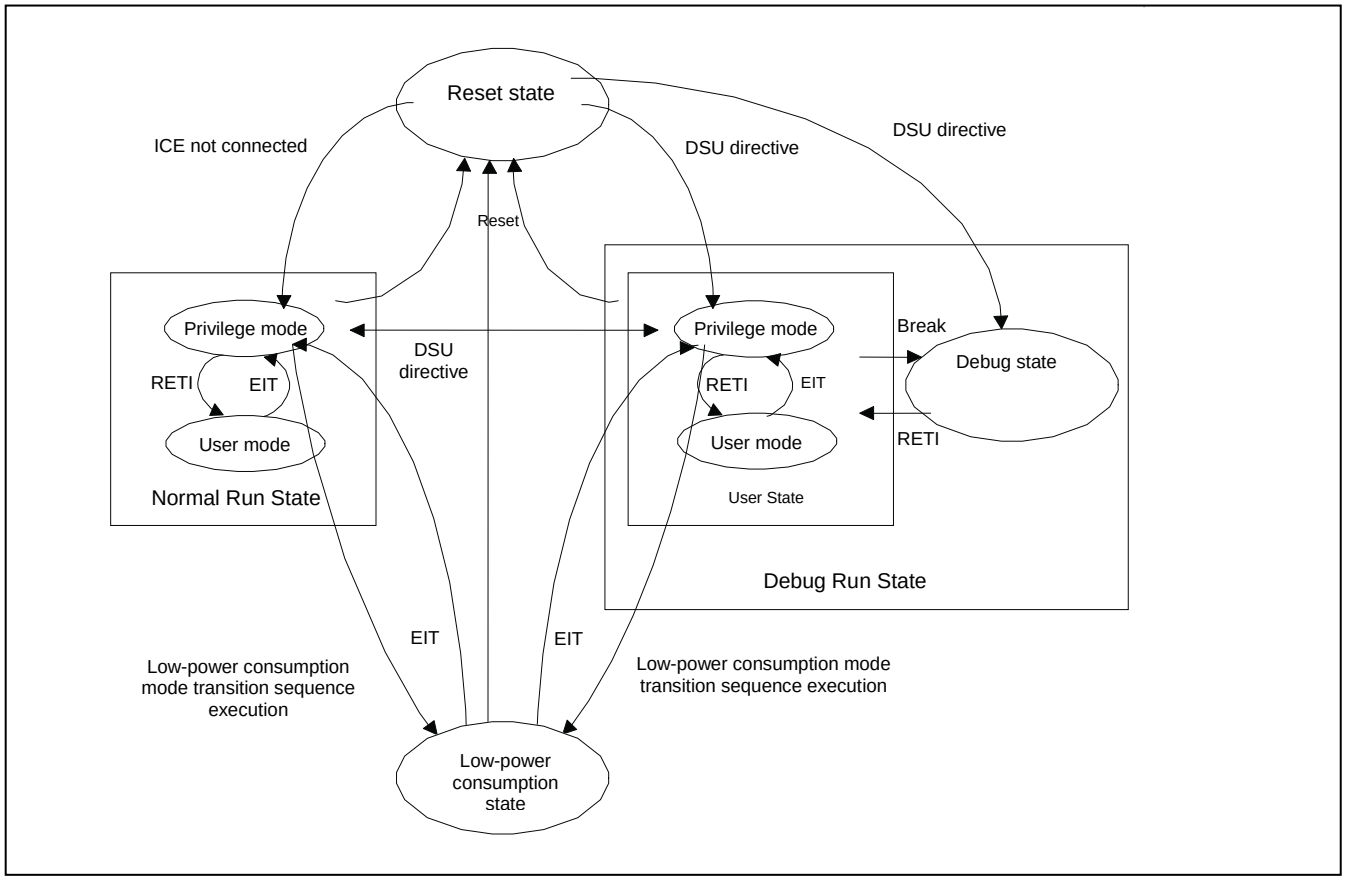
3.3.1 CPU Operating Status

The CPU operating status is shown below.

The CPU operation state includes the following states: reset state, normal run state, low-power consumption state, and debug run state.

The operating state transitions are shown below.

Figure 3-1. CPU Operating State Transition Diagram



3.3.1.1 Reset State

The reset state is shown below.

The reset state is the state when the CPU is being reset. Resets consist of two levels: initialize level and reset level. When an initialize level reset is issued, everything in the chip is initialized. For the reset level, others exclusive of the debug control functions, clocks, and reset control functions are initialized.

3.3.1.2 Normal Run State

The normal run state is shown below.

The normal run state is the state when sequential instruction and EIT processing are executed. The normal run state has privilege mode and user mode.

In user mode, there are restrictions on instructions and access destination, and there are instructions and access destinations that can only be executed in privilege mode. When the CPU enters the normal run state after reset is released, the CPU enters privilege mode, and changes to user mode when RETI is executed. The transition from user mode to privilege mode in the normal run state is triggered by reset or the EIT execution, and transition from privilege mode to user mode is triggered by the RETI execution.

3.3.1.3 Low-power Consumption State

The low-power consumption state is shown below.

The low-power consumption state is the state when the CPU is stopped to reduce the power consumption. The transition to the low-power consumption state is carried out by the standby control of the clock control unit. The low-power consumption state has three modes: sleep, stop and watch modes. Recovery from the low-power consumption state is carried out by interrupts.

3.3.1.4 Debug Run State

The debug run state is shown below.

The debug run state is the state when the CPU is connected to ICE and debug related functions are enabled. The debug run state has two states: a user state and a debug state. The transition between the debug run state and other states is basically carried via the reset state. However, the transition from the normal run state to the debug run state forcefully is also enabled.

The user state has a privilege mode and a user mode as the normal run state. However, when a break for debugging is carried out, the state changes to the debug state. In the debug state, instructions are executed in a privilege mode and all registers and memory can be accessed under the state when the memory protection function, etc. is disabled. The transition from a debug state to a user state is carried by the RETI instruction.

3.4 Pipeline Operation

This section explains the pipeline operation of the CPU.

In FR81, the common pipeline processing is carried out by the decode stage, and there are two types of pipelines such as an integer pipeline and a floating point pipeline from the execution stage. Although the completion between each pipeline processing differs from the sequence of instruction issuances, the processing results based on the program sequence are guaranteed.

For details, see "FR Family FR81 32-bit Microcontroller Programming Manual."

3.5 Floating Point Operation Processing

The floating point operation processing for the CPU is shown.

This series incorporates FPU.

For details, see "FR Family FR81 32-bit Microcontroller Programming Manual."

3.6 Data Structure

This section explains the data structure of the CPU.

The data types which can be handled with FR81 family CPU are the integer type, which can be handled with FR80 family or earlier, and the single precision floating point type.

For the integer type, little endian as the bit ordering and big endian as the byte ordering are used.

For details, see "FR Family FR81 32-bit Microcontroller Programming Manual."

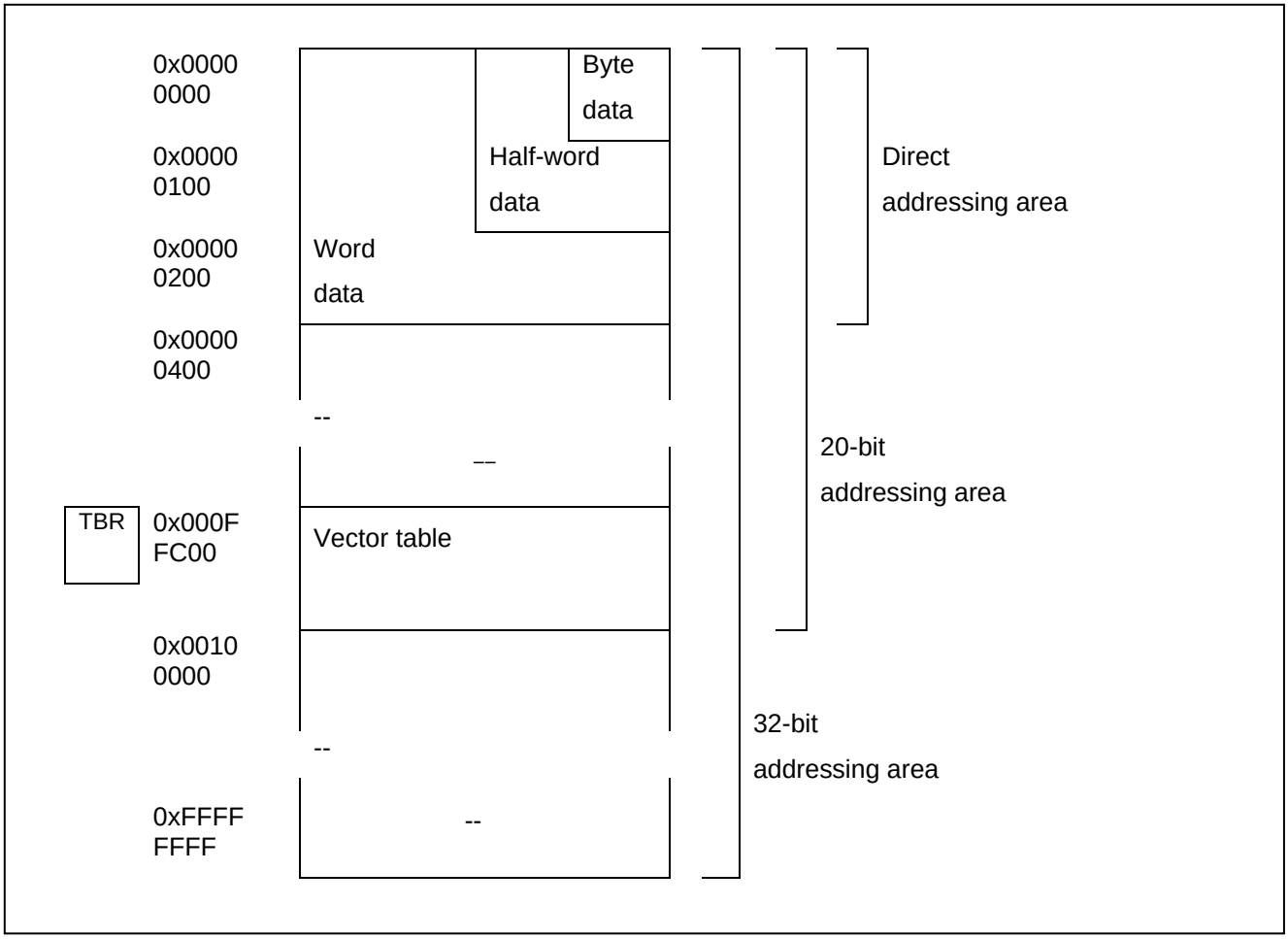
3.7 Addressing

This section explains addressing of the CPU.

A memory space is 32-bit linear.

The CPU manages the address space in byte. Specify a value of 32-bit for the address on the address space to access from the CPU. Figure 3-2 shows the address space.

Figure 3-2. Memory Map



The address space is also called memory space. The address space is the CPU-based logical address space. Address conversion is not performed. The CPU-based logical address is same as the physical address where memory and I/O are actually located.

For details, see "FR Family FR81 32-bit Microcontroller Programming Manual."

3.8 Programming Model

This section explains the programming model of the CPU.

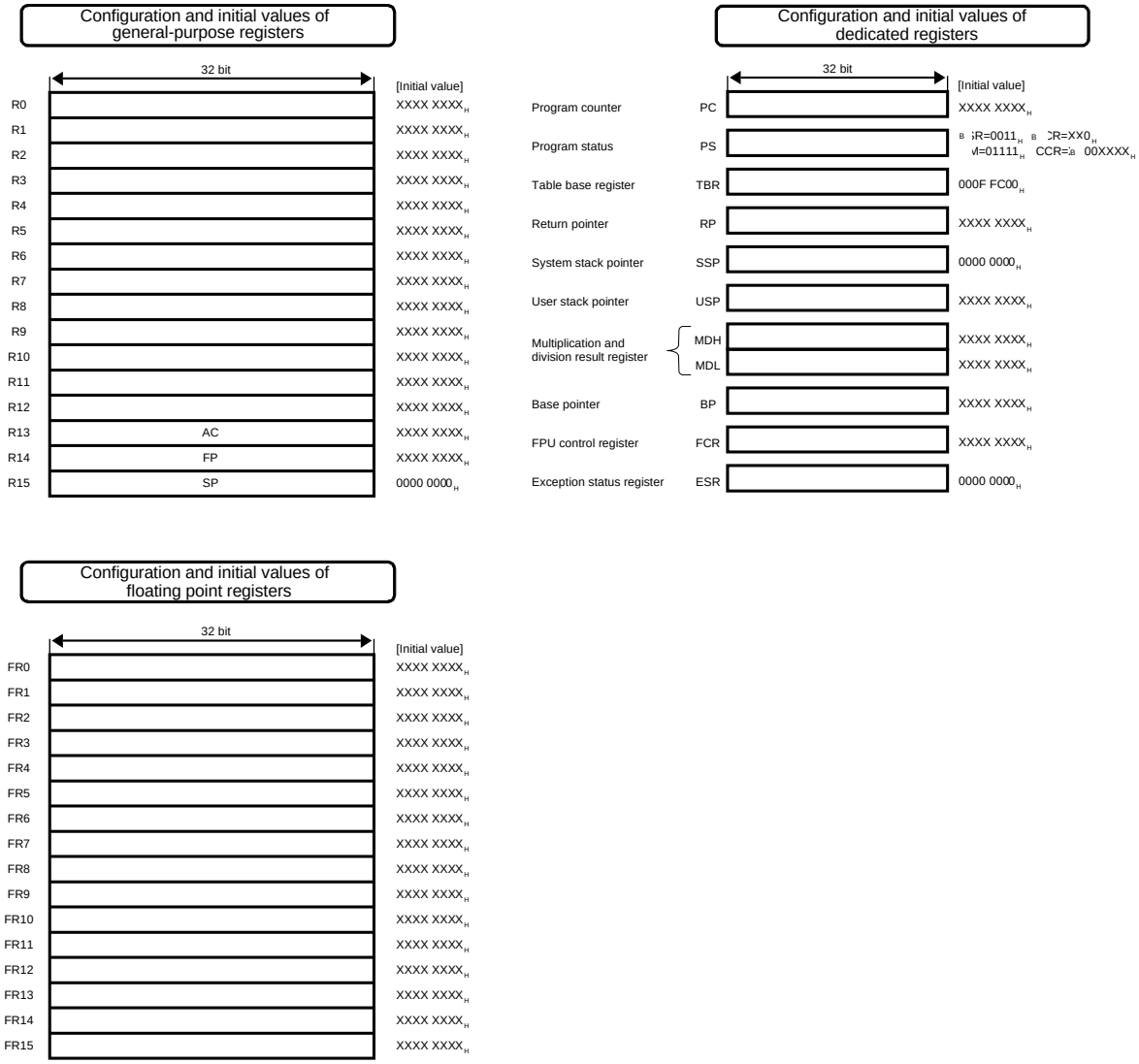
The CPU of FR81 has general-purpose registers, dedicated registers, and floating point registers. Besides these registers, the FR81 core has address-mapped system registers.

3.8.1 General-purpose Registers, Dedicated Registers, and Floating Point Registers

This section explains general-purpose registers, dedicated registers, and floating point registers.

Figure 3-3 shows the initial values for this series. For details of each register, see "FR Family FR81 32-bit Microcontroller Programming Manual."

Figure 3-3. Initial Values of General-purpose Registers, Dedicated Registers, and Floating Point Registers



3.8.2 System Register

The system register is shown below.

System register is an address mapping register for controlling system. These registers can be accessed only in the privilege mode. There are system registers as follows.

- Clock control-related register
- Reset control-related register
- Debug control-related register
- Memory protection-related register
- DMA-related register
- Watchdog timer register
- Wildregister control register
- FLASH control register
- TimingProtectionUnit register

When these registers are written and/or read in the user mode, the illegal instruction exception (data access error) occurs.

The access protection to system registers is judged on a priority bases than the memory protection function. Therefore, when user access to the system register area is enabled in the memory protection function and access is disabled in the privilege mode, those settings are disabled. Read and/or write is enabled only in the privilege mode and read and/or write is disabled in the user mode.

3.9 Reset and EIT Processing

This section explains reset and EIT processing.

Reset and EIT processing is the processing that is carried out by other than normal programs when Reset, Exception, Interrupt and Trap are detected.

For details, see "FR Family FR81 32-bit Microcontroller Programming Manual."

3.9.1 Reset

The reset is shown below.

Reset forcibly suspends operations currently running, initializes the device and restarts the program from the reset vector entry address.

Note:

In this series, the FixedVector function returns not the value written in the address of 0xF_FFFC on flash memory but the first address of + 0x0024 on flash memory to reset vector.
See "Chapter10: FixedVector Function" for details.

3.9.2 EIT Processing

The EIT processing is shown below.

The EIT processing suspends operations currently running, stores resumable information into memory, and transfers control to the predetermined processing program.

3.9.3 Vector Table

The vector table is shown.

Table 3-1. Vector Table

Interruption Factor	Interrupt Vector Number		Interrupt Level	Offset	Address at TBR Initial Value
	Decimal	Hexa-decimal			
Reset	0	00	-	0x3FC	0x000FFFFC
System reserved	1	01	-	0x3F8	0x000FFFF8
System reserved	2	02	-	0x3F4	0x000FFFF4
System reserved	3	03	-	0x3F0	0x000FFFF0
System reserved	4	04	-	0x3EC	0x000FFFEC
FPU exception	5	05	-	0x3E8	0x000FFFE8
Instruction access protection violation exception	6	06	-	0x3E4	0x000FFFE4
Data access protection violation exception	7	07	-	0x3E0	0x000FFFE0
Data access error interrupt	8	08	-	0x3DC	0x000FFFDC
INTE instruction	9	09	-	0x3D8	0x000FFFD8
Instruction break	10	0A	-	0x3D4	0x000FFFD4
System reserved	11	0B	-	0x3D0	0x000FFFD0
System reserved	12	0C	-	0x3CC	0x000FFFCC
System reserved	13	0D	-	0x3C8	0x000FFFC8
Illegal instruction exception	14	0E	-	0x3C4	0x000FFFC4
NMI request	15	0F	15(0xF) Fixed	0x3C0	0x000FFFC0
Peripheral interrupt #0	16	10	ICR00	0x3BC	0x000FFFBC
Peripheral interrupt #1	17	11	ICR01	0x3B8	0x000FFFB8
Peripheral interrupt #2	18	12	ICR02	0x3B4	0x000FFFB4
Peripheral interrupt #3	19	13	ICR03	0x3B0	0x000FFFB0
Peripheral interrupt #4	20	14	ICR04	0x3AC	0x000FFFAC
Peripheral interrupt #5	21	15	ICR05	0x3A8	0x000FFFA8
Peripheral interrupt #6	22	16	ICR06	0x3A4	0x000FFFA4
Peripheral interrupt #7	23	17	ICR07	0x3A0	0x000FFFA0
Peripheral interrupt #8	24	18	ICR08	0x39C	0x000FFF9C
Peripheral interrupt #9	25	19	ICR09	0x398	0x000FFF98
Peripheral interrupt #10	26	1A	ICR10	0x394	0x000FFF94
Peripheral interrupt #11	27	1B	ICR11	0x390	0x000FFF90
Peripheral interrupt #12	28	1C	ICR12	0x38C	0x000FFF8C
Peripheral interrupt #13	29	1D	ICR13	0x388	0x000FFF88

Interruption Factor	Interrupt Vector Number		Interrupt Level	Offset	Address at TBR Initial Value
	Decimal	Hexa-decimal			
Peripheral interrupt #14	30	1E	ICR14	0x384	0x000FFF84
Peripheral interrupt #15	31	1F	ICR15	0x380	0x000FFF80
Peripheral interrupt #16	32	20	ICR16	0x37C	0x000FFF7C
Peripheral interrupt #17	33	21	ICR17	0x378	0x000FFF78
Peripheral interrupt #18	34	22	ICR18	0x374	0x000FFF74
Peripheral interrupt #19	35	23	ICR19	0x370	0x000FFF70
Peripheral interrupt #20	36	24	ICR20	0x36C	0x000FFF6C
Peripheral interrupt #21	37	25	ICR21	0x368	0x000FFF68
Peripheral interrupt #22	38	26	ICR22	0x364	0x000FFF64
Peripheral interrupt #23	39	27	ICR23	0x360	0x000FFF60
Peripheral interrupt #24	40	28	ICR24	0x35C	0x000FFF5C
Peripheral interrupt #25	41	29	ICR25	0x358	0x000FFF58
Peripheral interrupt #26	42	2A	ICR26	0x354	0x000FFF54
Peripheral interrupt #27	43	2B	ICR27	0x350	0x000FFF50
Peripheral interrupt #28	44	2C	ICR28	0x34C	0x000FFF4C
Peripheral interrupt #29	45	2D	ICR29	0x348	0x000FFF48
Peripheral interrupt #30	46	2E	ICR30	0x344	0x000FFF44
Peripheral interrupt #31	47	2F	ICR31	0x340	0x000FFF40
Peripheral interrupt #32	48	30	ICR32	0x33C	0x000FFF3C
Peripheral interrupt #33	49	31	ICR33	0x338	0x000FFF38
Peripheral interrupt #34	50	32	ICR34	0x334	0x000FFF34
Peripheral interrupt #35	51	33	ICR35	0x330	0x000FFF30
Peripheral interrupt #36	52	34	ICR36	0x32C	0x000FFF2C
Peripheral interrupt #37	53	35	ICR37	0x328	0x000FFF28
Peripheral interrupt #38	54	36	ICR38	0x324	0x000FFF24
Peripheral interrupt #39	55	37	ICR39	0x320	0x000FFF20
Peripheral interrupt #40	56	38	ICR40	0x31C	0x000FFF1C
Peripheral interrupt #41	57	39	ICR41	0x318	0x000FFF18
Peripheral interrupt #42	58	3A	ICR42	0x314	0x000FFF14
Peripheral interrupt #43	59	3B	ICR43	0x310	0x000FFF10
Peripheral interrupt #44	60	3C	ICR44	0x30C	0x000FFF0C
Peripheral interrupt #45	61	3D	ICR45	0x308	0x000FFF08
Peripheral interrupt #46	62	3E	ICR46	0x304	0x000FFF04
Delay interrupt	63	3F	ICR47	0x300	0x000FFF00

Interruption Factor	Interrupt Vector Number		Interrupt Level	Offset	Address at TBR Initial Value
	Decimal	Hexa-decimal			
System reserved(For REALOS use)	64	40	-	0x2FC	0x000FFEFC
System reserved(For REALOS use)	65	41	-	0x2F8	0x000FFEF8
For INT instruction use	66	42	-	0x2F4	0x000FFEF4
	|	|		|	|
	255	FF		0x000	0x000FFC00

3.10 Memory Protection Function (MPU)

This section explains the memory protection function (MPU) of the CPU.

3.10.1 Overview

3.10.2 List of Registers

3.10.3 Description of Registers

3.10.4 Operations of Memory Protection Function

3.10.1 Overview

This section explains the overview of memory protection function (MPU) of the CPU.

This architecture supports a memory protection function. The memory protection function is a function that monitors access to a specified area and generates an exception on prohibited access. However, protection specified on system registers is ignored.

- Eight protection areas can be specified that are shared by instructions and data
- The protection area with the highest priority is area 0, with the priority decreasing for areas 1, 2, 3, etc. (The areas can overlap)
- Areas are specified by a page address and a page size
  - ☐ Page size: Can be specified in units of 2^n bytes from 16 bytes
  - ☐ Page address: Misaligned address also supported
- The following access privileges are controlled using privilege mode and user mode
  - ☐ Instruction fetch: Enabled/Disabled
  - ☐ Data Read: Enabled/Disabled
  - ☐ Data Write: Enabled/Disabled
- Access attributes are specified for each area
  - ☐ Buffer: Enabled/Disabled
- The access rights and attributes of undefined areas are controlled as a default area
- Protection violation exceptions occur when a protection violation occurs
- The register for the memory protection function can only be accessed in a privilege mode as system registers
- Data access error notification function
- I/O area (00000000_H to $0000FFFF_H$) is fixed buffer disabled

3.10.2 List of Registers

The list of registers is shown.

Table 3-2. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0310	Reserved		MPUCR		MPU control register
0x0314	Reserved				
0x0318	Reserved				
0x031C	Reserved				
0x0320	DPVAR				Data access protection violation address register
0x0324	Reserved		DPVSR		Data access protection violation status register
0x0328	DEAR				Data access error address register
0x032C	Reserved		DESR		Data access error status register
0x0330	PABR0				Protection area base address register 0
0x0334	Reserved		PACR0		Protection area control register 0
0x0338	PABR1				Protection area base address register 1
0x033C	Reserved		PACR1		Protection area control register 1
0x0340	PABR2				Protection area base address register 2
0x0344	Reserved		PACR2		Protection area control register 2
0x0348	PABR3				Protection area base address register 3
0x034C	Reserved		PACR3		Protection area control register 3
0x0350	PABR4				Protection area base address register 4
0x0354	Reserved		PACR4		Protection area control register 4
0x0358	PABR5				Protection area base address register 5
0x035C	Reserved		PACR5		Protection area control register 5
0x0360	PABR6				Protection area base address register 6
0x0364	Reserved		PACR6		Protection area control register 6
0x0368	PABR7				Protection area base address register 7
0x036C	Reserved		PACR7		Protection area control register 7

3.10.3 Description of Registers

Registers are shown.

3.10.3.1 MPU Control Register: MPUCR

3.10.3.2 Instruction Access Protection Violation Address Register: IPVAR

3.10.3.3 Instruction Access Protection Violation Status Register: IPVSR

3.10.3.4 Data Access Protection Violation Address Register: DPVAR

3.10.3.5 Data Access Protection Violation Status Register: DPVSR

3.10.3.6 Data Access Error Address Register: DEAR

3.10.3.7 Data Access Error Status Register: DESR

3.10.3.8 Protection Area Base Address Register 0 to 7: PABR0 to PABR7

3.10.3.9 Protection Area Control Register 0 to 7: PACR0 to PACR7

3.10.3.1 MPU Control Register: MPUCR

The bit configuration of the MPU control register (MPUCR) is shown.

The MPU control register controls whether the MPU is enabled or disabled, and configures the access permissions in privilege mode and user mode to default areas (areas not specified as protection areas).

MPUCR: Address 0312_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	PIE	PRE	PWE	UIE	URE	UWE	Reserved	BE
Initial value	0	0	0	0	0	0	-	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PAN[1:0]		DEE	MPE
Initial value	-	-	-	-	0	1	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,WX	R1,WX	R/W	R/W

[bit15] PIE (Privilege Mode Instruction Fetch Enable)

This bit is for permitting instruction fetch in privilege mode from the default areas (areas that have not been specified as protection areas).

PIE	Access to Default Area
0	Instruction fetch not permitted in privilege mode (initial value)
1	Instruction fetch permitted in privilege mode

[bit14] PRE (Privilege Mode Read Access Enable)

This bit is for permitting data read access in privilege mode from the default areas (areas that have not been specified as protection areas).

PRE	Access to Default Area
0	Read access not permitted in privilege mode (initial value)
1	Read access permitted in privilege mode

[bit13] PWE (Privilege Mode Write Access Enable)

This bit is for permitting data write access in privilege mode to the default areas (areas that have not been specified as protection areas).

PWE	Access to Default Area
0	Write access not permitted in privilege mode (initial value)
1	Write access permitted in privilege mode

[bit12] UIE (User Mode Instruction Fetch Enable)

This bit is for permitting instruction fetch in user mode from the default areas (areas that have not been specified as protection areas).

UIE	Access to Default Area
0	Instruction fetch not enable in user mode (initial value)
1	Instruction fetch enable in user mode

[bit11] URE (User Mode Read Access Enable)

This bit is for permitting data read access in user mode from the default areas (areas that have not been specified as protection areas).

URE	Access to Default Area
0	Read access not permitted in user mode (initial value)
1	Read access permitted in user mode

[bit10] UWE (User Mode Write Access Enable)

This bit is for permitting data write access in user mode to the default areas (areas that have not been specified as protection areas).

UWE	Access to Default Area
0	Write access not permitted in user mode (initial value)
1	Write access permitted in user mode

[bit9] Reserved

Always write "0" when writing. This bit reads out "0".

[bit8] BE (Buffer Enable)

The bit permits buffering to be used when performing data access to default areas (areas that are not specified as protection areas). When the use of buffering is forbidden, the CPU stops pipeline operation and waits for the data access to finish before starting the next operation. As a result, although the data access efficiency decreases, it is possible to perform data access synchronized to the instruction. Illegal instruction exceptions occur when there is an error during data access only if buffering is forbidden. When buffering is permitted, data access errors can be notified as interrupts.

BE	Bufferable Specification for the Default Area
0	Buffer disabled (initial value)
1	Buffer enabled

[bit7 to bit4] Reserved

These bits are reserved. Always write "0" when writing.

[bit3, bit2] PAN (Protection Area Number)

Indicates the number of implemented protection areas that can be specified. This bit is read-only and indicates the number of areas implemented in hardware.

PAN[1:0]	Number of Memory Protection Areas Implemented
00	Reserved
01	8 areas
10	12 areas
11	16 areas

[bit1] DEE (Data Access Error Interrupt Enable)

This bit permits interrupts to occur when a data access error occurs in areas where buffer operation is enabled. If a data access error occurs in an area where buffer operation is permitted while this bit is enabled, a data access error interrupt occurs. At this time, the address where the error occurred is stored in the data access error address register (DEAR), and the details of the access are stored in the data access error status register (DESR). If interrupts are disabled, the above registers are updated only.

DEE	Data Access Error Interrupt Enabled
0	Data access error interrupt disabled (initial value)
1	Data access error interrupt enable

[bit0] MPE (Memory Protection Unit Enable)

This bit is for enabling the memory protection function. If the memory protection function is disabled, buffering is configured as disabled for accesses to all areas.

MPE	Memory Protection Function
0	Memory protection function disabled (initial value)
1	Memory protection function enabled

3.10.3.2 Instruction Access Protection Violation Address Register: IPVAR

The bit configuration of the instruction access protection violation address register is shown.

This register stores the address where an instruction access protection violation occurred.

Also see "3.10.4.2 Instruction Access Protection Violation" and "3.10.4.7 Notes".

IPVAR: Address 0318_H (Access: Word)

	bit31	bit30	- - -	bit2	bit1	bit0
	IPVA[31:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R,WX	R,WX	- - -	R,WX	R,WX	R,WX

[bit31 to bit0] IPVA[31:0] (Instruction fetch Protection Violation Address)

This register stores the address where an instruction access protection violation occurred when a violation has not occurred in the instruction access protection violation status register (IPVSR.IPV=0). This is not aligned.

Note:

Using this register is prohibited.

3.10.3.3 Instruction Access Protection Violation Status Register: IPVSR

The bit configuration of the instruction access protection violation status register is shown.

This register indicates the status when an instruction access protection violation occurs.
The content of this register is updated by hardware only when IPV=0. Only writing "0" to the IPV bit has an effect. Writes to any other bits and writing "1" to IPV are ignored.
Also see "3.10.4.2 Instruction Access Protection Violation" and "3.10.4.7 Notes".

IPVSR: Address 031E$_H$ (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	-	-	-	-	-	-	-	-
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		SZ[1:0]		MD	Reserved		IPV
Initial value	-	-	0	0	0	-	-	0
Attribute	R0,W0	R0,W0	R,WX	R,WX	R,WX	R0,W0	R0,W0	R,W

[bit15 to bit6, bit2, bit1] Reserved

These bits are reserved. Always write "0" to these bits.

[bit5, bit4] SZ[1:0]

The access size when the violation occurred.

SZ[1:0]	Access Size
00	Byte
01	Half-word
10	Word
11	Reserved

[bit3] MD

Indicates the mode of the access.

MD	Operation Mode
0	Access in user mode
1	Access in privilege mode

[bit0] IPV (Instruction fetch Protection Violation)

This bit indicates that an instruction access protection violation occurred. In order to save the details of new protection violations, clear this bit.

IPV	Instruction Access Protection Violation
0	Instruction access protection violation not detected (initial value)
1	Instruction access protection violation detected

Note:

Using this register is prohibited.

3.10.3.4 Data Access Protection Violation Address Register: DPVAR

The bit configuration of the data access protection violation address register is shown.

The address where the data access protection violation occurs is saved.

DPVAR: Address 0320_H (Access: Word)

	bit31	bit30	- - -	bit2	bit1	bit0
	DPVA[31:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R,WX	R,WX	- - -	R,WX	R,WX	R,WX

[bit31 to bit0] DPVA[31:0] (Data Access Protection Violation Address)

This register stores the address where a data access protection violation occurred when a violation has not occurred in the data access protection violation status register (DPVSR.DPV=0). This register indicates the address requested by the CPU, and the address is not aligned.

3.10.3.5 Data Access Protection Violation Status Register: DPVSR

The bit configuration of the data access protection violation status register is shown.

This register indicates the status when a data access protection violation occurs.
The content of this register is updated by hardware only when DPV=0. Writing "0" to DPV only is valid. Writes to any other bits and writing "1" to DPV are ignored.

DPVSR: Address 0326_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RW[1:0]		SZ[1:0]		MD	Reserved		DPV
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R0,W0	R0,W0	R,W

[bit15 to bit8, bit2, bit1] Reserved

These bits are reserved. Always write "0" to these bits.

[bit7, bit6] RW[1:0] (Read/Write)

The access type when the violation occurred. When a read-modify-write is executed, because both read and write access rights are required and the determination is made in the initial read cycle, RW=01_B read (read-modify-write) even if the violation occurs in the write part of the read-modify-write.

RW[1:0]	Access Type
00	Read
01	Read (Read-modify-write)
10	Write
11	Reserved

[bit5, bit4] SZ[1:0]

The access size when the violation occurred.

SZ[1:0]	Access Size
00	Byte
01	Half-word
10	Word
11	Reserved

[bit3] MD

Indicates the mode of the access.

MD	Operation Mode
0	Access in user mode
1	Access in privilege mode

[bit0] DPV (Data Access Protection Violation)

This bit indicates that a data access protection violation occurred. In order to save the details of new protection violations, clear this bit.
Writing "0" to this bit only is valid. Writing "1" to the bit is ignored.

DPV	Data Access Protection Violation
0	Data access protection violation not detected (initial value)
1	Data access protection violation detected

3.10.3.6 Data Access Error Address Register: DEAR

The bit configuration of the data access error address register is shown.

This register stores the address where a data access error occurred.

DEAR: Address 0328_H (Access: Word)

	bit31	bit30	- - -	bit2	bit1	bit0
	DEA[31:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R,WX	R,WX	- - -	R,WX	R,WX	R,WX

[bit31 to bit0] DEA[31:0] (Data Access Error Address)

This register stores the address where a data access error occurred when a violation has not occurred in the data access error status register (DESR.DAE=0). If the protection violation occurred while accessing system registers, the access address from the CPU is stored as it is without being aligned. If the result of performing a bus access is an error, the address is aligned.

3.10.3.7 Data Access Error Status Register: DESR

The bit configuration of the data access error status register is shown.

This register indicates the status when a data access error occurs. The content of this register is updated by hardware only when DAE=0. Writing 0 to DAE only is valid. Writes to any other bits and writing 1 to DAE are ignored.

DESR: Address $032E_H$ (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RW[1:0]		SZ[1:0]		MD	Reserved		DAE
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R0,W0	R0,W0	R,W

[bit15 to bit8, bit2, bit1] Reserved

These bits are reserved. Always write 0 to these bits. These bits read out "0".

[bit7, bit6] RW[1:0] (Read/Write)

The access type when the error occurred.

RW[1:0]	Access Type
00	Read
01	Read (Read-modify-write)
10	Write
11	Reserved

[bit5, bit4] SZ[1:0]

The access size when the error occurred.

SZ[1:0]	Access Size
00	Byte
01	Half-word
10	Word
11	Reserved

[bit3] MD

Indicates the mode of the access.

MD	Operation Mode
0	Access in user mode
1	Access in privilege mode

[bit0] DAE (Data Access Error)

This bit indicates that a data access error occurred. In order to save the details of new data errors, clear this bit.
The interrupt request is withdrawn by clearing this bit when the data access error interrupt is enabled. Only "0" writing is effective to this bit. "1" writing is invalid.

DAE	Data Access Error
0	Data access error not detected (initial value)
1	Data Access Error detected

3.10.3.8 Protection Area Base Address Register 0 to 7: PABR0 to PABR7

The bit configuration of protection area base address register 0 to 7 is shown.

These registers set the base addresses of the protection areas for each MPU channel.

PABR0 to 7: Address 0330_H, 0338_H, 0340_H ... (Access: Word)

	bit31	bit30	- - -	bit10	bit9	bit8
	PABR[31:8]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PABR[7:0]							
Initial value	X	X	X	X	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R0,WX

[bit31 to bit0] PABR[31:0] (Protection Area Base Address Register)

These registers point to the base address of the protection area. The area from the address specified here to the size specified by the protection area control registers (PACR0 to 7) is the protection area. The address does not need to be aligned to the protection area size.
The lower 4 bits of the PABR register are fixed at0000_B.

3.10.3.9 Protection Area Control Register 0 to 7: PACR0 to PACR7

The bit configuration of protection area control register 0 to 7 is shown.

These registers set access permissions and restrictions for each MPU channel.

PACR0 to 7: Address 0336_H, $033E_H$, 0346_H ... (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	PIE	PRE	PWE	UIE	URE	UWE	Reserved	BE
Initial value	0	0	0	0	0	0	-	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ASZ[4:0]					Reserved		PAE
Initial value	0	0	0	0	0	-	-	0
Attribute	R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0	R/W

[bit15] PIE (Privilege Mode Instruction Fetch Enable)

This bit is for enabling instruction fetch in privilege mode for the specified protection area.

PIE	Access to the Specified Protection Area
0	Instruction fetch not permitted in privilege mode (initial value)
1	Instruction fetch permitted in privilege mode

[bit14] PRE (Privilege Mode Read Access Enable)

This bit is for enabling data read access in privilege mode for the specified protection area.

PRE	Access to the Specified Protection Area
0	Read access not permitted in privilege mode (initial value)
1	Read access permitted in privilege mode

[bit13] PWE (Privilege Mode Write Access Enable)

This bit is for enabling data write access in privilege mode for the specified protection area.

PWE	Access to the Specified Protection Area
0	Write access not permitted in privilege mode (initial value)
1	Write access permitted in privilege mode

[bit12] UIE (User Mode Instruction Fetch Enable)

This bit is for enabling instruction fetch in user mode for the specified protection area.

UIE	Access to the Specified Protection Area
0	Instruction fetch not enable in user mode (initial value)
1	Instruction fetch enable in user mode

[bit11] URE (User Mode Read Access Enable)

This bit is for enabling data read access in user mode for the specified protection area.

URE	Access to the Specified Protection Area
0	Read access not permitted in user mode (initial value)
1	Read access permitted in user mode

[bit10] UWE (User Mode Write Access Enable)

This bit is for enabling data write access in user mode for the specified protection area.

UWE	Access to the Specified Protection Area
0	Write access not permitted in user mode (initial value)
1	Write access permitted in user mode

[bit9] Reserved

Always write "0" when writing. This bit reads out "0".

[bit8] BE (Buffer Enable)

This bit permits buffering to be used during data access for the specified protection area. When the use of buffering is forbidden, the CPU stops pipeline operation and waits for the data access to finish before starting the next operation. As a result, although the data access efficiency decreases, it is possible to perform data access synchronized to the instruction. Illegal instruction exceptions occur when there is an error during data access only if buffering is forbidden. When buffering is permitted, data access errors can be notified as interrupts.

BE	Bufferable Specification for the Specified Protection Area
0	Buffer disabled (initial value)
1	Buffer enable

[bit7 to bit3] ASZ[4:0] (Area Size)

These bits specify the size of the specified protection area. The specified address does not need to be aligned to the sizes described below. Furthermore, if the lower limit of the area specified by the address and size exceeds $FFFFFFFF_H$, the lower limit of the area is treated as $FFFFFFFF_H$.

ASZ[4:0]	Size of the Specified Protection Area
00000_B	Reserved
00001_B	Reserved
00010_B	Reserved
00011_B	16 B
00100_B	32 B
00101_B	64 B
00110_B	128 B
00111_B	256 B
01000_B	512 B
01001_B	1 KB
01010_B	2 KB
01011_B	4 KB
01100_B	8 KB
01101_B	16 KB
01110_B	32 KB
01111_B	64 KB
10000_B	128 KB
10001_B	256 KB
10010_B	512 KB
10011_B	1 MB
10100_B	2 MB
10101_B	4 MB
10110_B	8 MB
10111_B	16 MB
11000_B	32 MB
11001_B	64 MB
11010_B	128 MB
11011_B	256 MB

ASZ[4:0]	Size of the Specified Protection Area
11100_B	512 MB
11101_B	1 GB
11110_B	2 GB
11111_B	4 GB

[bit2, bit1] Reserved

These bits are reserved. Always write "0" when writing.

[bit0] PAE (Protection Area Enable)

This bit is for enabling the memory protection function.

PAE	Memory Protection Area
0	Specified memory protection area disabled (initial value)
1	Specified memory protection area enabled

3.10.4 Operations of Memory Protection Function

The memory protection function is shown below.

3.10.4.1 Setting Up Memory Protection Areas

3.10.4.2 Instruction Access Protection Violation

3.10.4.3 Data Access Protection Violation

3.10.4.4 Data Access Error

3.10.4.5 Memory Protection Operation by Delay Slot

3.10.4.6 DEAR and DESR Update

3.10.4.7 Notes

3.10.4.1 Setting Up Memory Protection Areas

The setting up memory protection areas of the CPU is shown below.

The memory protection function is configured by settings whether instructions, data reads, and data writes are permitted or forbidden in privilege mode and user mode for a maximum of 8 protection areas specified by address and size, and default areas that are not contained in these protection areas. The buffer permitted or forbidden setting can also be configured for each area at the same time.

If there are overlaps between specified protection areas, the area with the smallest number takes precedence.

When the memory protection function is disabled (MPUCR.MPE=0), access is performed with access permitted to all areas and buffering disabled.

3.10.4.2 Instruction Access Protection Violation

The instruction access protection violation of the CPU is shown below.

The memory protection unit (MPU) monitors CPU instruction fetches and determines whether instruction fetches are permitted to the accessed areas. The instruction address when an instruction access protection violation exception occurs can be determined from the PC value saved on the system stack.

3.10.4.3 Data Access Protection Violation

The data access protection violation of the CPU is shown below.

The memory protection unit (MPU) monitors CPU data accesses and determines whether accesses (reads and writes) to the corresponding area are permitted. If an access was not permitted, the MPU stores that address and access information in the data access protection violation address register (DPVAR) and the data access protection violation status register (DPVSR). However, if data access protection violation information already exists in the above register (DPVSR.DPV=1), this is not overwritten. The data access that caused the violation at this time is not performed.

If a data access protection violation occurs during the execution of an instruction that performs multiple data accesses, the data accesses that had executed up until the violation occurred are not canceled. If a data access protection violation exception occurs during the LDM0, LDM1, STM0, STM1, FLDM, or FSTM instructions, the list of remaining registers is stored in the exception status register ESR.RL.

If a data access protection violation occurs during the EIT processing sequence or the RETI instruction, the CPU is halted and can only be recovered by break interrupt or reset.

3.10.4.4 Data Access Error

This section explains data access errors of the CPU.

If the following conditions are satisfied during a data access, this is treated as a data access error and the access information at that time are stored in the data access error address register (DEAR) and data access error status register (DESR). However, if data access error information already exists in the above register (DESR.DAE=1), this is not overwritten.

- System register access in user mode
- Bus error during data access

The operation after a bus error occurs during data access differs between accesses with buffering enabled and accesses with buffering disabled. System register accesses in user mode are always processed as illegal instruction exceptions (data access).

If a data access error occurs during access to an unbufferable area, the CPU processes this as an illegal instruction exception (data access error).

If a data access error occurs during access to a bufferable area, and if the data access error interrupt is enabled by MPU control register MPUCR.DEE=1, the data access error interrupt is triggered and the CPU performs data access error interrupt processing. If a data access error occurs during access to a bufferable area, because the CPU is executing a subsequence instruction, the PC value saved when the data access error interrupt occurs is not the PC value for the instruction that performed the data access.

If an illegal instruction exception (data access error) occurs during the execution of an instruction that performs multiple data accesses, the data accesses that had executed up until the error occurred are not canceled. If an illegal instruction exception (data access error) occurs during the LDM0, LDM1, STM0, STM1, FLDM, or FSTM instructions, the list of remaining registers is stored in the exception status register ESR.RL, and the bit indicating a data access error ESR.INV6 is set.

If an illegal instruction exception (data access error) occurs during the EIT processing sequence or the RETI instruction, the CPU is halted and can only be recovered by break interrupt or reset.

3.10.4.5 Memory Protection Operation by Delay Slot

The memory protection operation by a delay slot is shown.

The instruction arranged in the delay slot is processed as 16-bit. Therefore, the exception is generated as an illegal instruction exception (instruction that cannot be arranged in the delay slot) even if there are an instruction access protection violation factor and an instruction access error factor in the lower 16-bit by arranging 32-bit instruction in the delay slot.

3.10.4.6 DEAR and DESR Update

The DEAR and the DESR update are shown.

The data access error address register (DEAR) and the data access error status register (DESR) are renewed in the following cases.

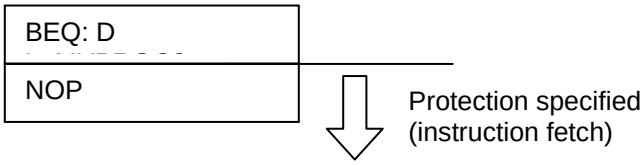
- System register access in user mode (illegal instruction exception)
- Bus error in buffer prohibition area access (illegal instruction exception)
- Bus error in buffer permission area access (data access error interrupt)

DEAR and DESR are renewed in the instruction that did the corresponding access and it is renewed to the asynchronization with the instruction operation in the case where the data access error interrupt is generated in the case where the illegal instruction exception is generated. It gives priority to the illegal instruction exception factor when the factor is generated at the same time.

3.10.4.7 Notes

This section explains notes of the Memory Protection Function (MPU).

- Access protection violation exception will occur when an instruction of access protection violation is executed. For details, see "FR Family FR81 32-bit Microcontroller Programming Manual." For details of the instruction access protection violation and the instruction access protection violation exception, also see "3.10.4.2 Instruction Access Protection Violation".
- If the boundary of delay slot overlaps with that of instruction access protection area as the following figure, the instruction access protection violation occurs regardless of whether the branch is established or not. PC with occurrence of exception is PC of delayed branch instruction.



4. Operation Mode

This chapter explains the operation mode.

4.1 Overview

4.2 Features

4.3 Configuration

4.4 Register

4.5 Operation

4.1 Overview

This section explains the overview of the operation mode.

This chapter explains the operation mode of this type of item decided after reset is released. See "Chapter: Power Consumption Control" for the mode of each power consumption control and the mode of each clock selection.

4.2 Features

This section explains features of the operation mode.

This device supports the following operation modes.

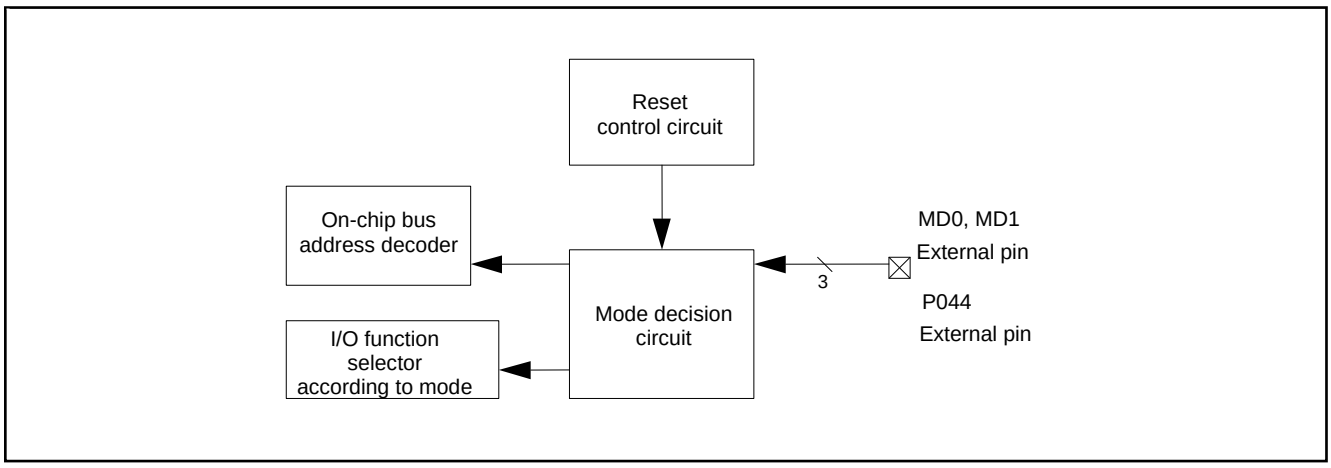
- User mode
  - ☐ The program starts from the built-in flash memory.

- Serial writer mode
  The built-in flash memory is programmed by using the serial writer.

4.3 Configuration

This section explains the configuration of the operation mode.

Figure 4-1. Block Diagram



4.4 Register

This section explains the register of the operation mode.

Address	Register				Register function
	+0	+1	+2	+3	
0x07FC	BMODR	Reserved	Reserved	Reserved	Bus Mode Register

4.4.1 Bus Mode Register: BMODR (Bus MODe Register)

The bit configuration of the bus mode register is shown.

This register indicates the mode that has been set during startup. The register data can be read only. Data writing does not affect on this register value.

BMODR: Address 07FC$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BMOD[7:0]							
Initial value	*	*	*	*	*	*	*	*
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

* It depends on operation mode.

[bit7 to bit0] BMOD[7:0]: Operation mode

These bits indicate the current operation mode. Data writing is ineffective.

BMOD[7:0]	Operation mode
0101xxxx	User mode
0111xx1x	Serial writer mode

4.5 Operation

This section explains operations of the operation mode.

4.5.1 MD0, MD1, P044 Pins Settings

4.5.2 Fetching the Operation Mode

4.5.3 Explanation of Each Operation Mode

4.5.1 MD0, MD1, P044 Pins Settings

MD0, MD1 and P044 pins settings are shown.

Table 4-1. Pin Settings

Operation mode	MD1	MD0	P044
User mode	0	1	-
Serial writer mode	1	0	1

Settings other than those shown in the table are prohibited.

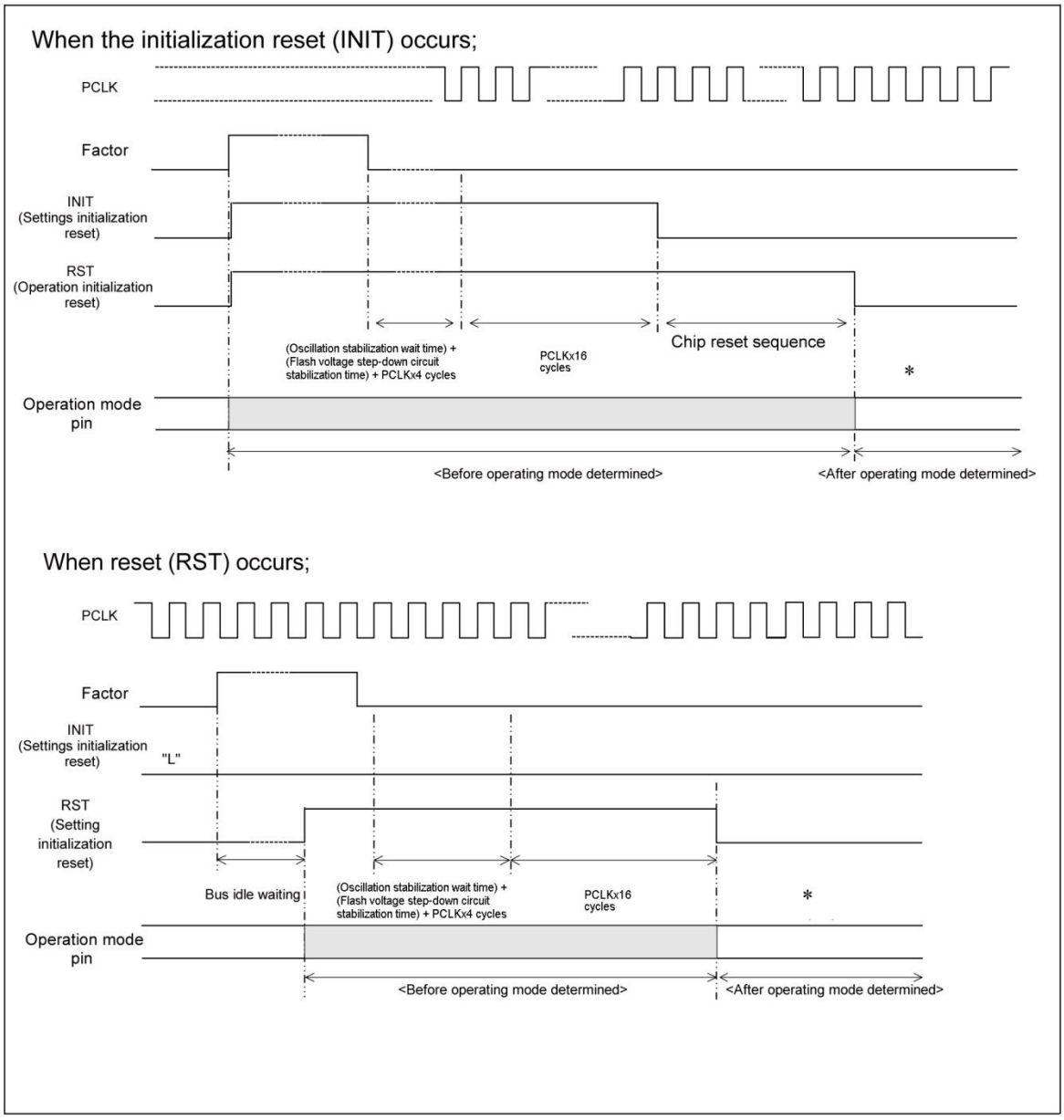
4.5.2 Fetching the Operation Mode

The fetching the operation mode is shown.

The operation mode is fetched by sampling the RST (Reset). During the time when an RST is issued and when it is released, the MD0, MD1 and P044 pin inputs must be determined. In user mode, the P044 pin does not need to be determined.

The following shows an operation sequence from an occurrence of reset cause to the determination of an operation mode.

Figure 4-2. Operation Mode Fetch Timing Chart



* Continue fixing MD0 and MD1 pins even after operating mode determined.

Note: When in serial writer mode, the P044 pin needs not be fixed after operating mode determined.

4.5.3 Explanation of Each Operation Mode

The each operation mode is shown.

The following details each operation mode.

4.5.3.1 User Mode

4.5.3.2 Serial Writer Mode

4.5.3.1 User Mode

The user mode is shown.

In this mode, internal I/O, built-in RAM, and built-in flash memory are enabled, and access to all other areas is disabled. The external pins function as peripheral functions or general-purpose ports.

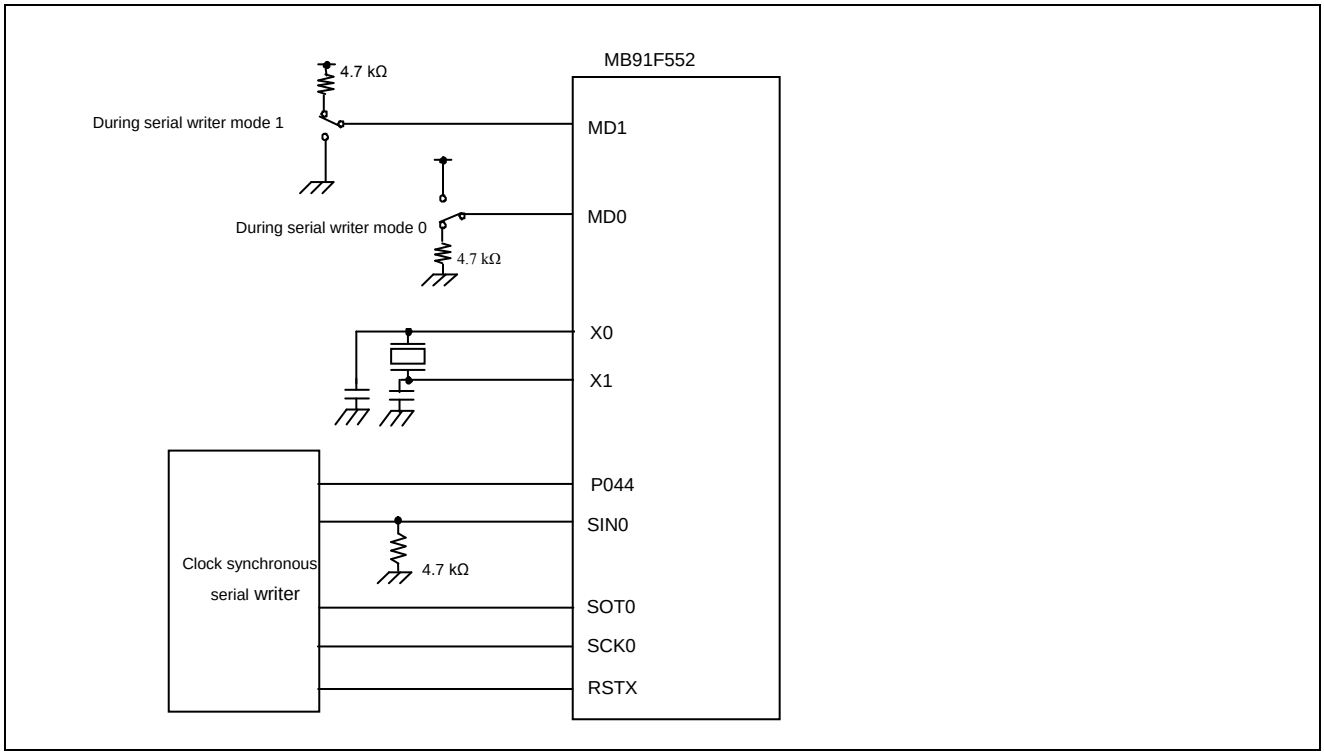
4.5.3.2 Serial Writer Mode

The serial writer mode is shown.

Table 4-2. Setting Pins in Serial Writer Mode

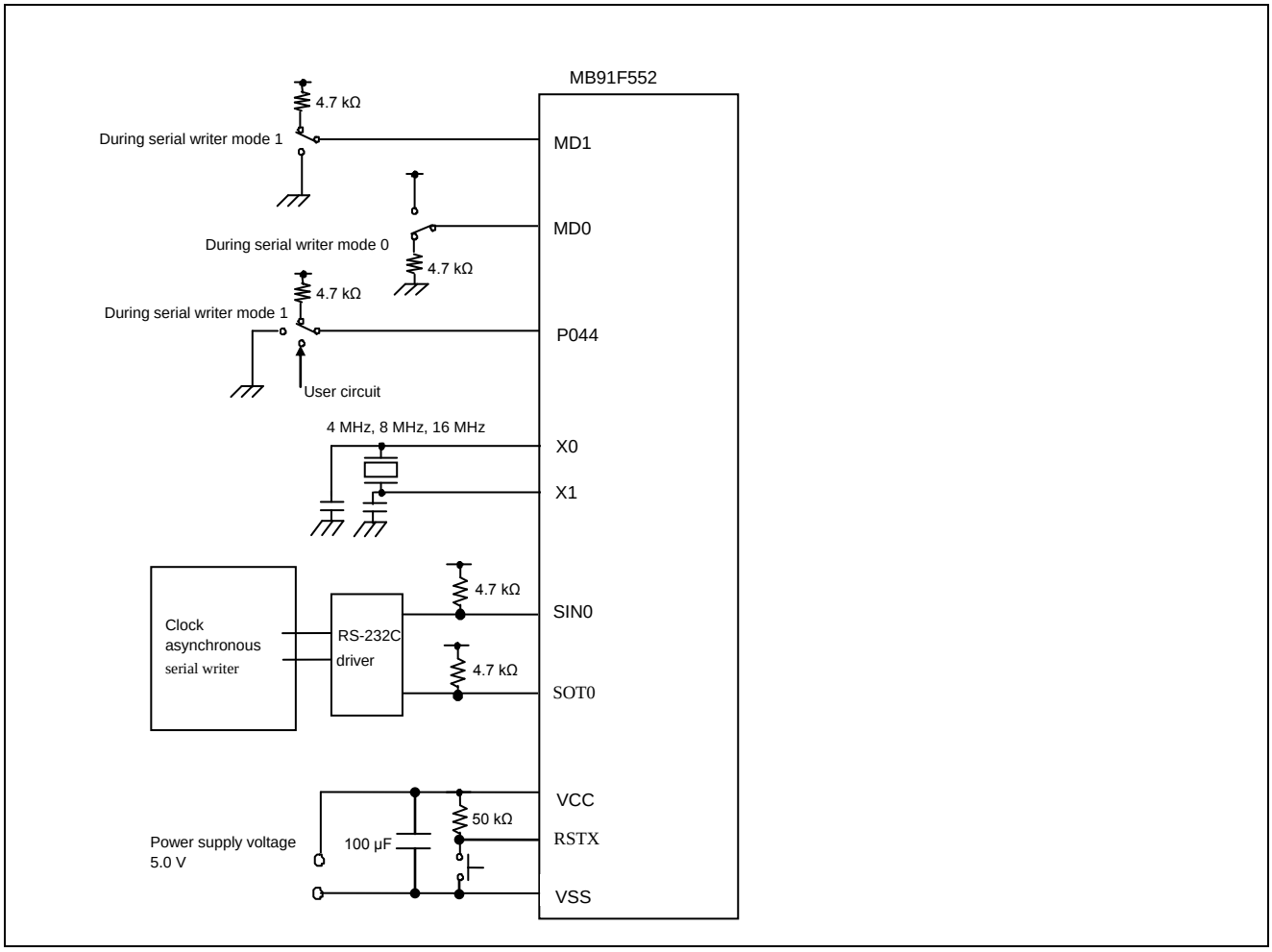
Pin number	Pin name	Serial writer mode		
		Pin device/connection destination (for on-board programming)	Input level	Output level
51	MD0	GND (Pull-down)	CMOS Hysteresis	-
52	MD1	VDD (Pull-up)	CMOS Hysteresis	-
58	RSTX	Reset input	CMOS Hysteresis	-
53	X0	Oscillation pin	-	-
54	X1	Oscillation pin	-	-
62	P044	Serial writer mode is started by adding pull-up resistor to external and, after releasing reset, setting level "H".	CMOS Hysteresis	CMOS
48	P037 SIN0	Setting the input of this pin to "H" until the start of communication enables clock asynchronous communication mode, while setting it to "L" enables clock synchronous communication mode. Serial writer mode starts, and at the point at which communication starts, this pin is used as a UART serial data input pin.	CMOS Hysteresis	CMOS
46	P035 SOT0	Serial writer mode starts and, at the point at which communication starts, this pin becomes a serial data output pin.	Automotive	CMOS
47	P036 SCK0	When the communication mode is clock synchronous communication mode, this pin becomes the serial clock input/output pin.	CMOS Hysteresis	CMOS
8, 32, 64	VCC	+5.0 V power supply	-	-
1, 9, 33, 55, 59	VSS	GND	-	-
16 22 27	AVCC2 AVCC1 AVCC0	+5.0 V power supply	-	-
17 23 28	AVSS2 AVSS1 AVSS0	GND	-	-
25 29	AVRL1 AVRL0	GND	-	-
26 30	AVRH1 AVRH0	+5.0 V power supply	-	-

Figure 4-3. Example of Connection for Serial Writer Mode Using Clock Synchronous Serial Writer



Pull-up and pull-down resistance is an example. Select the most appropriate resistance value for each system.

Figure 4-4. Example of Connection for Serial Writer Mode Using Clock Asynchronous Serial Writer



Pull-up and pull-down resistance and capacity values are examples. Select the most appropriate resistance value for each system.

5. Clock

This chapter explains the clock.

5.1 Overview

5.2 Features

5.3 Configuration

5.4 Registers

5.5 Operation

5.1 Overview

This section explains the overview of the clock.

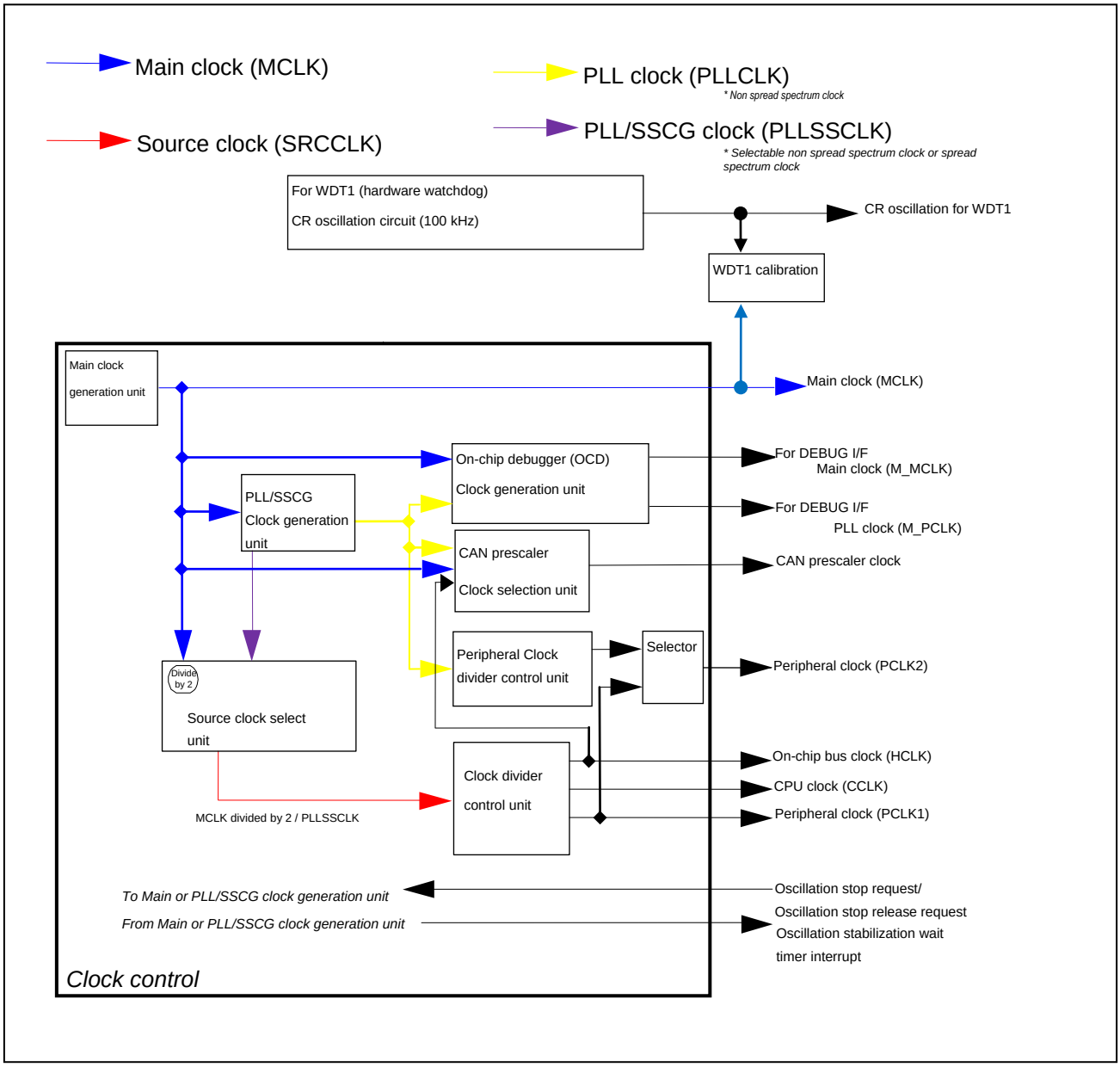
The built-in oscillation circuit generates a single clock, which generates individual clock systems on the chip.

- External pins for the built-in oscillation circuit:

  Main clock: Connects to the crystal resonator

- Generation of source clocks: Selects from the clocks which are multiplied by PLL/SSCG of main clock (MCLK) or divided by 2 of main clock.
- Division of source clock: Divides the source clock and generates operating clocks for supplying to each unit.

Figure 5-1. Diagram of the Clock Generation System



5.2 Features

This section explains features of the clock.

- A single clock on-chip oscillators are implemented.
- The main clock (MCLK) is multiplied by on-chip PLL/SSCG.
- Each clock has been forced not to supply by using the timer until it becomes stabilized (oscillation stabilization wait timer).
- Oscillation stabilization wait end interrupt can be generated.
- Main clock oscillation stabilization wait timer (main timer) can be used as a general-purpose interrupt interval timer after the main clock oscillation stabilizes.
- Implements a CR oscillation circuit for 100 kHz WDT1 clock.
- Generates the clock for CAN prescaler. The clock can be selected from PLL clock (PLLCLK) [non spread spectrum clock] and main clock (MCLK). When PLL stops when PLL clock is selected, on-chip bus clock (HCLK) is used.
- For the noise decrement, the SSCG clock [spread spectrum clock] can be selected as CPU and a clock of the peripheral functions.

5.3 Configuration

This section explains the configuration of the clock.

Figure 5-2. Connection Diagram of Clock (1)-1 Main Clock Generation Unit

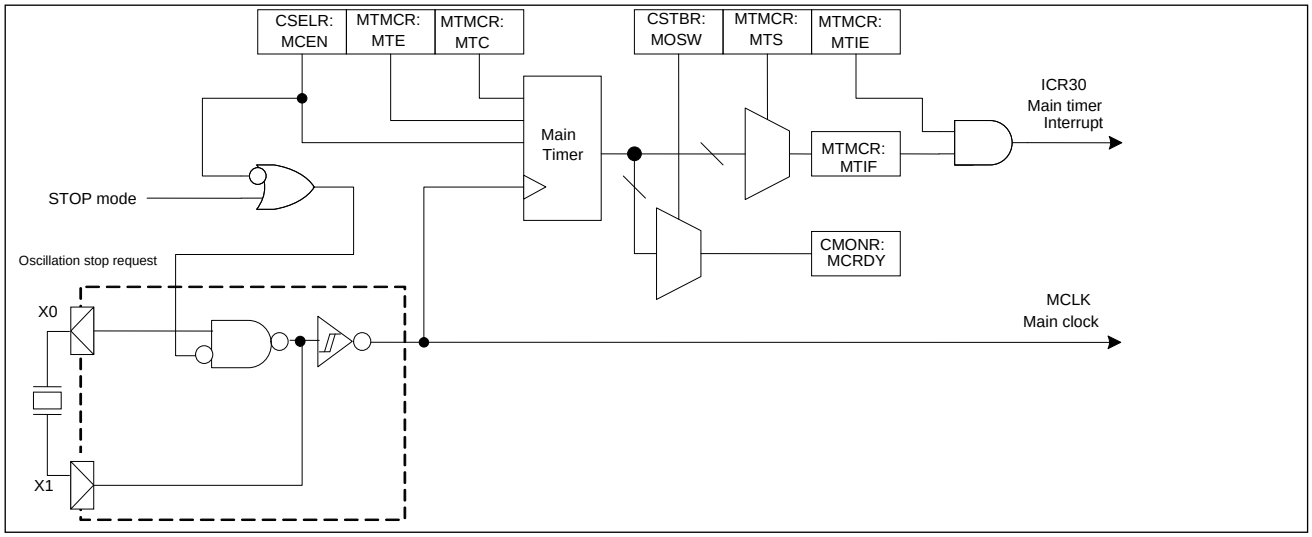


Figure 5-3. Connection Diagram of Clock (1)-2 PLL/SSCG Clock Generation Unit

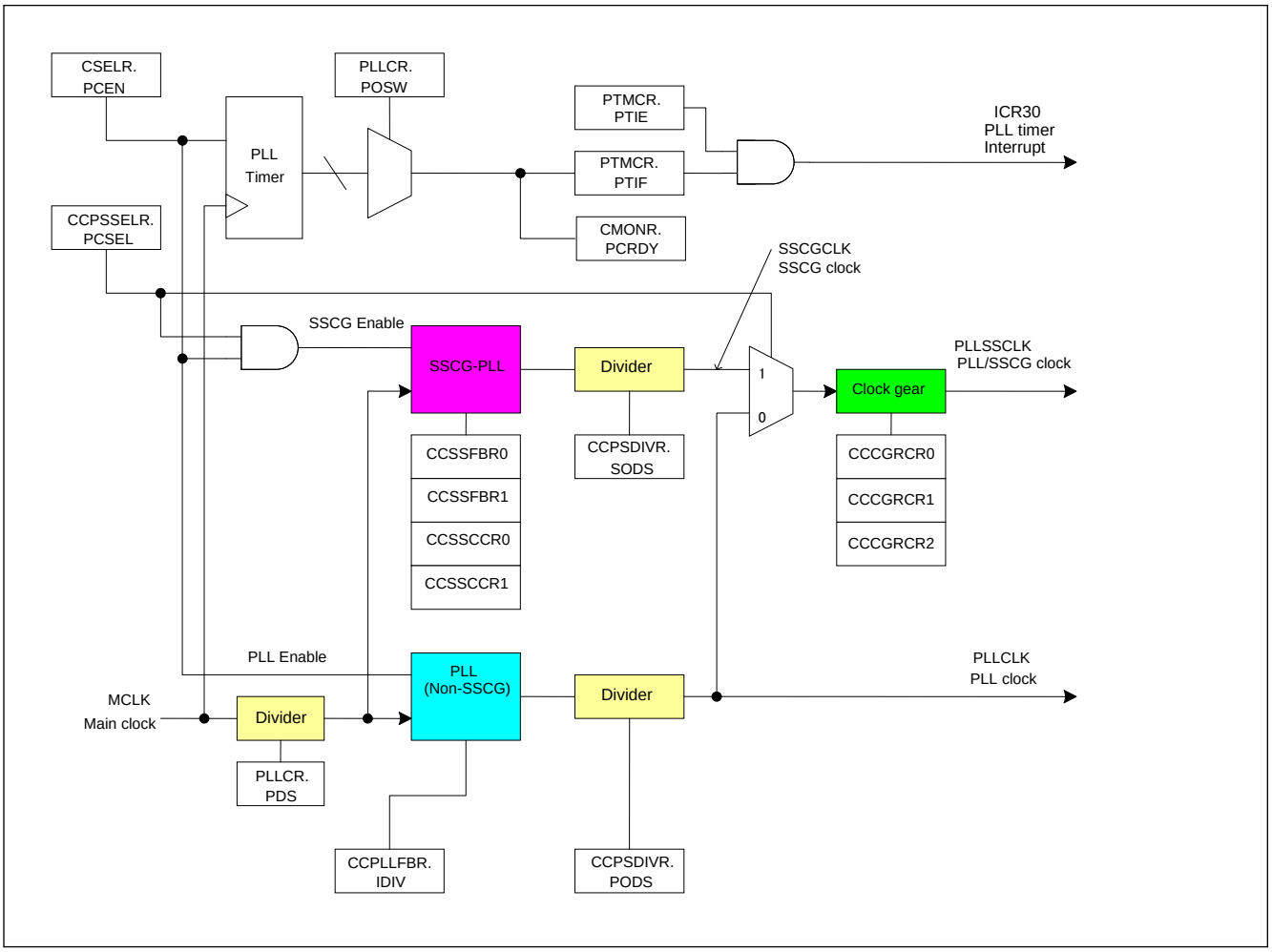


Figure 5-4. Connection Diagram of Clock (2) Source Clock Selection Unit

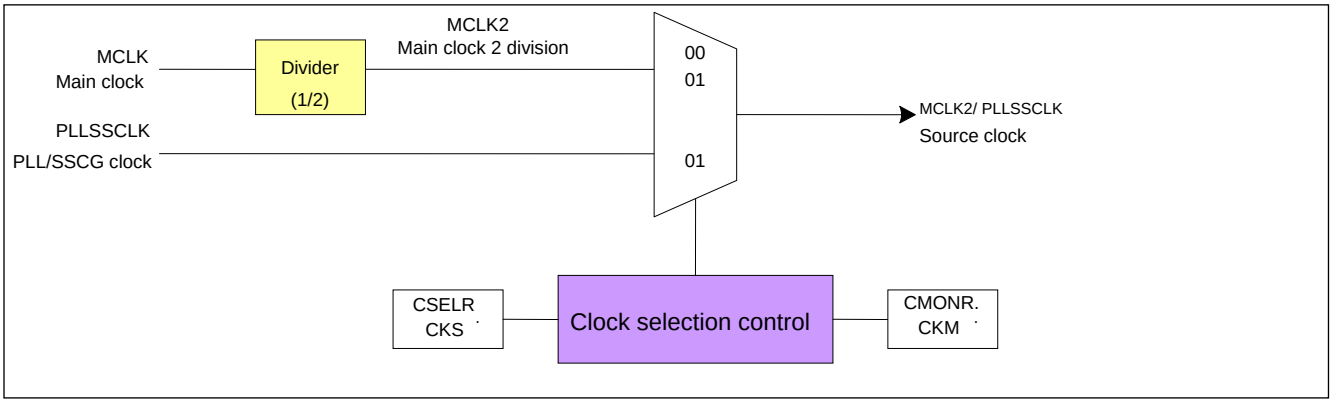


Figure 5-5. Connection Diagram of Clock (3) Divider Control

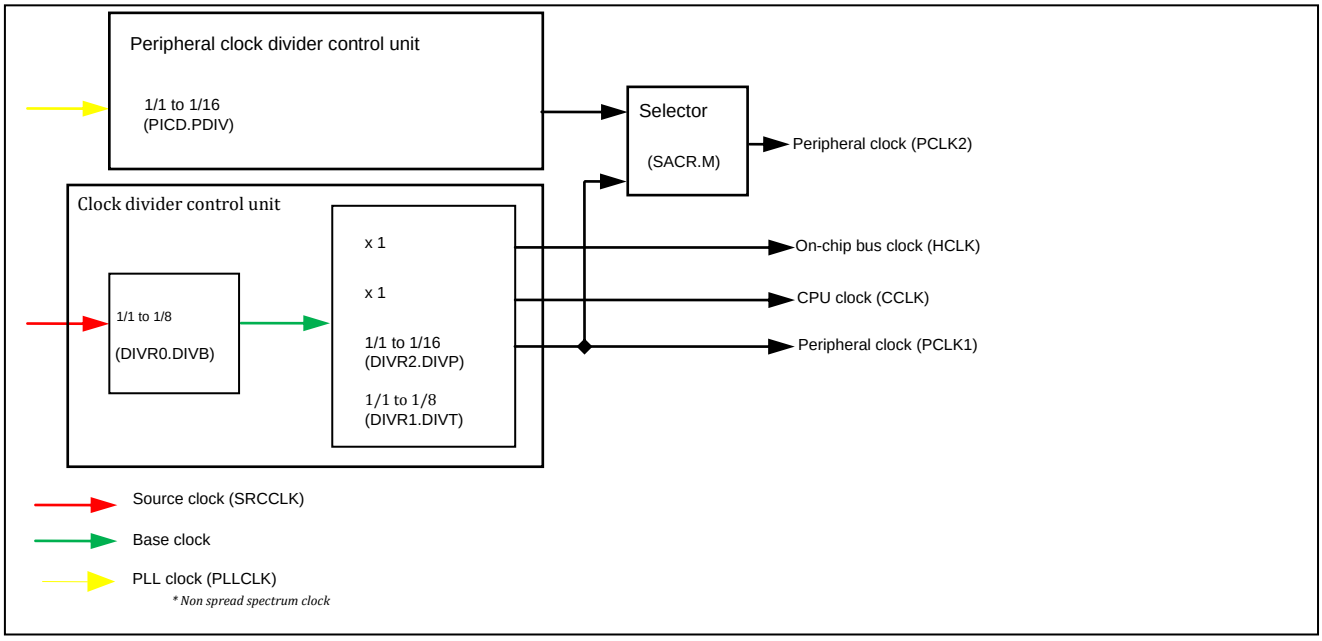


Figure 5-6. Connection Diagram of Clock (4) CAN Prescaler Clock Generation

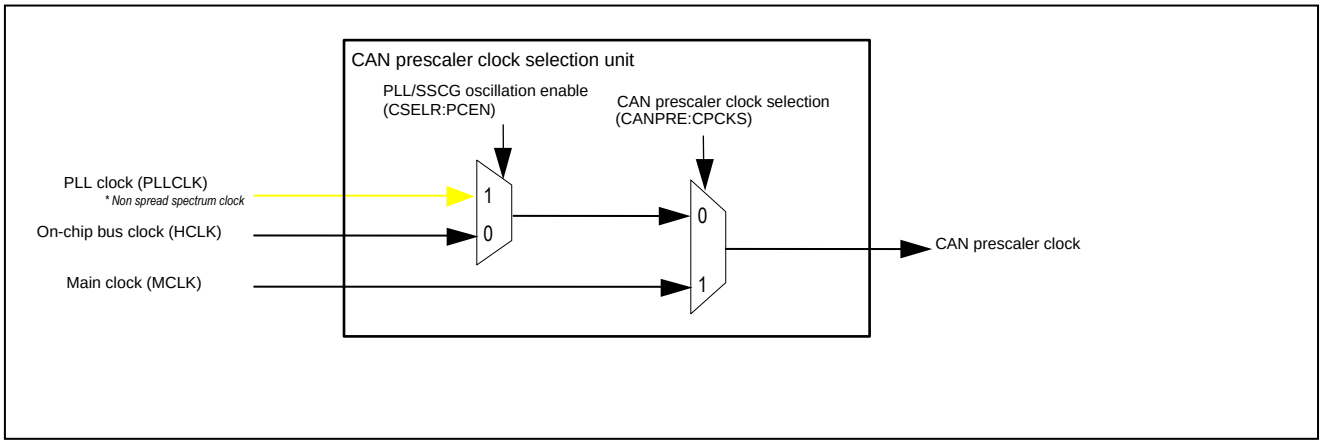


Figure 5-7.Diagram of the Clock System

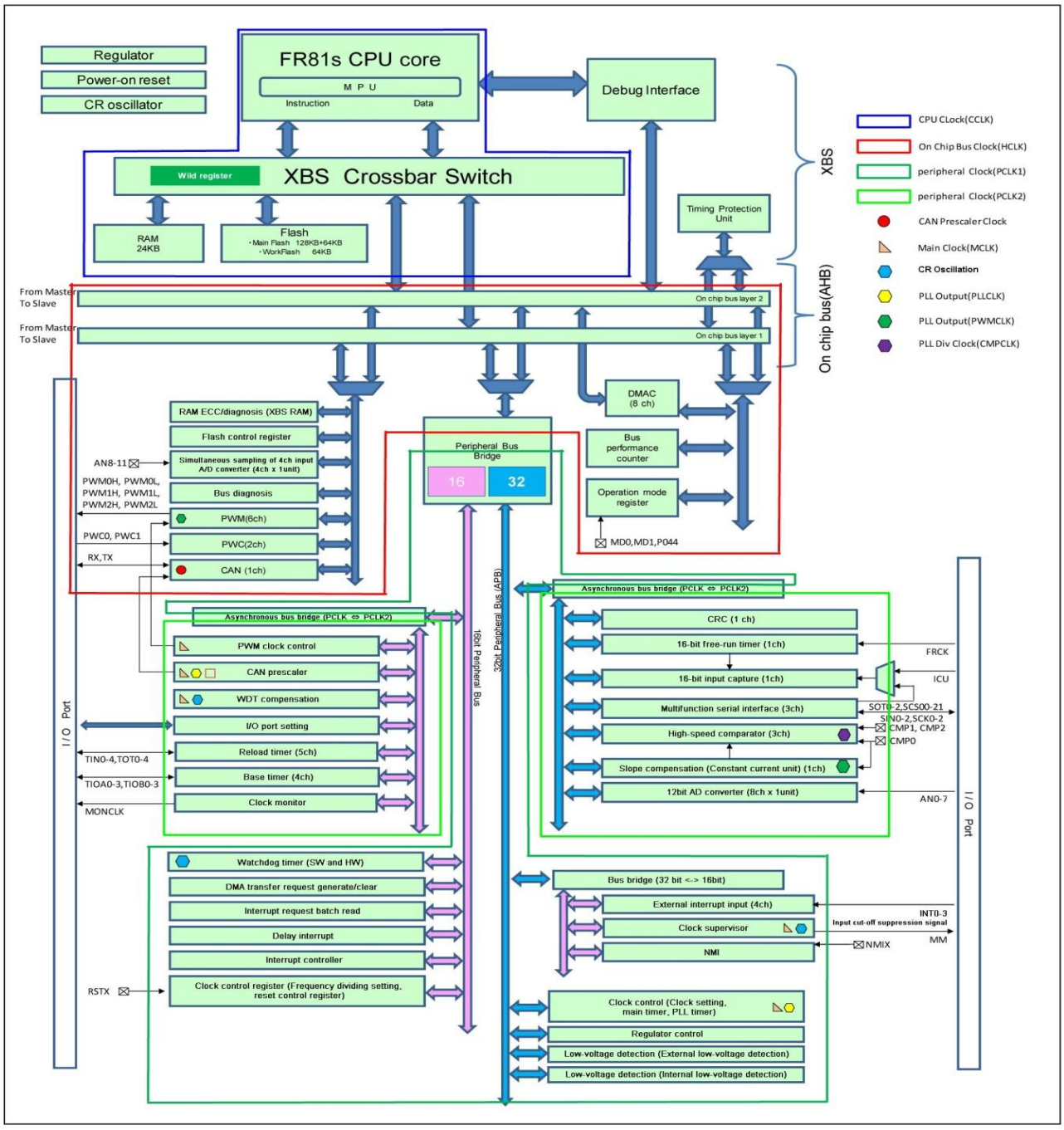


Table 5-1. Types of Functions That Use PCLK1/PCLK2

Functions That Use PCLK1	Functions That Use PCLK2
Watchdog timer (SW and HW)	PWM clock control
DMA transfer request generate/clear	CAN prescaler
Interrupt request batch read	WDT1 calibration
Clock control (frequency dividing setting)	I/O port setting
Clock control (Clock setting, Main timer, PLL timer)	Reload timer (5ch)
Reset control	Base timer (4ch)
Regulator control	Clock monitor
Delay interrupt	CRC
Interrupt controller	16-bit Free-run Timer (1ch)
External interrupt input	16-bit Input Capture (1ch)
Clock supervisor	Multi-function serial interface (3ch)
NMI	Comparator (3ch)
Low-voltage detection (External power supply low-voltage detection)	Slope compensation (1ch)
Low-voltage detection (Internal power supply low-voltage detection)	12-bit A/D Converter (8ch x 1unit)

5.4 Registers

This section explains registers of the clock.

Table 5-2. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0488	DIVR0	Reserved	DIVR2	Reserved	Frequency dividing setting register 0 Frequency dividing setting register 2
0x0510	CSELR	CMONR	MTMCR	Reserved	Clock source setting register Clock source monitor register Main timer control register
0x0514	PLLCR		CSTBR	PTMCR	PLL setting register Oscillation stabilization wait setting register PLL oscillation stabilization wait timer control register
0x0520	CCPSSELR	Reserved	Reserved	CCPSDIVR	PLL/SSCG clock selection register PLL/SSCG output clock frequency dividing setting register
0x0524	Reserved	CCPLLFBR	CCSSFBR0	CCSSFBR1	PLL feedback frequency dividing setting register SSCG feedback frequency dividing setting register 0 SSCG feedback frequency dividing setting register 1
0x0528	Reserved	CCSSCCR0	CCSSCCR1		SSCG configuration setting register 0 SSCG configuration setting register 1
0x052C	Reserved	CCCGRCR0	CCCGRCR1	CCCGRCR2	Clock gear configuration setting register 0 Clock gear configuration setting register 1 Clock gear configuration setting register 2
0x0530	Reserved	Reserved	Reserved	Reserved	Reserved
0x0534	Reserved	Reserved	Reserved	Reserved	Reserved
0x0538	Reserved	Reserved	Reserved	Reserved	Reserved
0x053C	Reserved	Reserved	Reserved	Reserved	Reserved
0x1000	SACR	PICD	Reserved	Reserved	Sync/Async Control Register Peripheral Interface Clock Divider

5.4.1 Frequency Dividing Setting Register 0: DIVR0 (DIVide clock configuration Register 0)

The bit configuration of the frequency dividing setting register 0 is shown.

This register controls division of clocks.

DIVR0: Address 0488_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DIVB[2:0]			Reserved				
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit5] DIVB[2:0] (DIVide ratio of Base clock): Base clock frequency dividing setting

These bits configure a division in the area where the base clock is generated from the source clock as follows.
The CPU operation clock and the on-chip bus clock (HCLK) have the same frequency as that of the base clock.

DIVB[2:0]	Division Ratio
000	No divide (Initial value)
001	2 division
010	3 division
011	4 division
100	5 division
101	6 division
110	7 division
111	8 division

[bit4 to bit0] (Reserved)

5.4.2 Frequency Dividing Setting Register 2: DIVR2 (DIVide clock configuration Register 2)

The bit configuration of the frequency dividing setting register 2 is shown.

This register controls division of clocks.

DIVR2: Address 048A$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DIVP[3:0]				Reserved			
Initial value	0	0	1	1	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit4] DIVP[3:0] (DIVide ratio of PCLK): Peripheral clock frequency dividing setting

These bits configure the division ratio when generating the peripheral clock (PCLK) from the base clock.

DIVP[3:0]	Base Clock -> PCLK Division Ratio
0000	No divide
0001	2 division
0010	3 division
0011	4 division (Initial value)
0100	5 division
0101	6 division
0110	7 division
0111	8 division
1000	9 division
1001	10 division
1010	11 division
1011	12 division
1100	13 division
1101	14 division
1110	15 division
1111	16 division

Note:

Set this register to peripheral clock (PCLK) to be sure to become 40 MHz or less.

[bit3 to bit0] (Reserved)

5.4.3 Clock Source Setting Register: CSELR (Clock source SELect Register)

The bit configuration of the clock source setting register is shown.

This register controls each clock source and selects a source clock (SRCCLK).

CSELR: Address 0510_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	PCEN	MCEN	Reserved			CKS[1:0]	
Initial value	0	0	1	0	0	0	0	0
Attribute	R,W0	R,W	R,W	R0,WX	R0,WX	R0,WX	R,W	R,W

Note:

The value set for this register and the value read out from this register are not actually controlled and selected. You can make sure that the value set for this register would really take effect by reading out CMONR. After making sure that the value of this register is the same as that of CMONR, rewrite the register. While switching clocks is in progress (CKS[1:0] ≠ CKM[1:0]), a write operation to this register will be ignored.

[bit7] Reserved

[bit6] PCEN (PLL Clock ENable): PLL oscillation enable

This bit controls the PLL/SSCG clock oscillation circuit as follows.

PCEN	Oscillation Control for PLL/SSCG Clock (PLLSSCLK)
0	Stop oscillation (Initial value).
1	Oscillate.

This bit cannot be rewritten when a PLL/SSCG clock (PLLSSCLK) is selected as the source clock. Also, this bit cannot be rewritten when the main oscillation is stopped or during the main oscillation stabilization wait time (CMONR.MCRDY = 0). Set this bit to "0" before switching to the stop mode.
Rewriting the MCEN bit with "0" causes this bit to set to "0".

Note:

PLL enters the status of the oscillation enable regardless of the value of this bit while communicating the MDI in high-speed.

[bit5] MCEN (Main Clock ENable): Main clock oscillation enable

This bit controls an oscillation circuit for main clock as follows.

MCEN	Oscillation Control for Main Clock
0	Stop oscillation.
1	Oscillate (Initial value).

This bit cannot be rewritten when a main clock (MCLK) or PLL/SSCG clock (PLLSSCLK) is selected as the source clock.
The oscillation circuit for main clock always stops regardless of the value of this bit when the stop mode is set.
The main timer is cleared when this bit is set to "0".

Note:

The main clock enters the status of the oscillation enable regardless of the value of this bit while communicating the MDI in low-speed.

[bit4 to bit2] Reserved

[bit1, bit0] CKS[1:0] (ClocK Select): Source clock selection

These bits select the source clock (SRCCLK) as follows.

CKS	Source Selection
00	Division of the main clock (MCLK) by 2 (Initial value)
01	Division of main clock (MCLK) by 2
10	PLL/SSCG clock (PLLSSCLK)
11	Setting prohibited (Writing has effect on operation)

However, when CKS[1:0] ≠ CKM[1:0], these bits cannot be rewritten. When the clock oscillation which you are trying to switch operations by these bits stops or is waiting for a stabilization (CMONR.xCRDY = 0), these bits cannot also be rewritten.
Possible combinations for changing these bits are shown below.

CKS Value before Change	Eligible Values	Rewritten Conditions	Ineligible Values
00	00, 01	MCRDY = 1	11
	10	PCRDY = 1	
01	00, 01	MCRDY = 1	10, 11
10	00	MCRDY = 1	01, 11
	10	PCRDY = 1	
11	Prohibited	Prohibited	Prohibited

Do not write the values which cannot be rewritten.

5.4.4 Clock Source Monitor Register: CMONR (Clock source MONitor Register)

The bit configuration of the clock source monitor register is shown.

This register displays the status of each clock source and the selected source clock (SRCCLK).
You can confirm that the value set at CSELR is really reflected in the actual status by reading this register.

CMONR: Address 0511_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	PCRDY	MCRDY	Reserved			CKM[1:0]	
Initial value	0	0	1	0	0	0	0	0
Attribute	R0,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX

Note:

If you have changed CSELR, do not write next value on CSELR until CMONR is equal to CSELR.

[bit7] Reserved

[bit6] PCRDY (PLL Clock ReaDY): PLL clock ready

This bit shows the PLL/SSCG clock (PLLSSCLK) status as follows.

PCRDY	PLL/SSCG Clock (PLLSSCLK) Status
0	Oscillation stops or in the oscillation stabilization wait status.
1	It is in the oscillation stabilization status and available for the source clock.

You cannot select a PLL/SSCG clock (PLLSSCLK) as the source clock when this bit is set to "0".

Note:

PCRDY = 1 may be read immediately after changing PCEN = 1 to 0.
PLL enters the status of the oscillation enable regardless of the value of this bit while communicating the MDI in high-speed.

[bit5] MCRDY (Main Clock ReaDY): Main clock ready

This bit shows the main clock (MCLK) status as follows.

MCRDY	Main Clock (MCLK) Status
0	Oscillation stops or in the oscillation stabilization wait status.
1	It is in the oscillation stabilization status and available for the source clock.

You cannot select a main clock (MCLK) or a PLL/SSCG clock (PLLSSCLK) as the source clock when this bit is set to "0".
The initial value of "1" for this bit means that it is oscillation stabilized at the first reset vector fetch after power-on reset, not that it is already oscillation stabilized immediately after power-on reset.

Note:

MCRDY = 1 may be read immediately after changing MCEN = 1 to 0.
The main clock enters the status of the oscillation enable regardless of the value of this bit while communicating the MDI in high-speed.

[bit4 to bit2] Reserved

[bit1, bit0] CKM[1:0] (ClocK Monitor): Source clock display

These bits show the source clock (SRCCLK) currently selected.

CKM[1:0]	Source Selection
00	Division of main clock (MCLK) by 2
01	Division of main clock (MCLK) by 2
10	PLL/SSCG clock (PLLSSCLK)
11	Reserved

5.4.5 Main Timer Control Register: MTMCR (Main clock TiMer Control Register)

The bit configuration of the main timer control register is shown.

This register controls the main timer which runs with the main clock (MCLK).

MTMCR: Address 0512_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MTIF	MTIE	MTC	MTE	MTS[3:0]			
Initial value	0	0	0	0	1	1	1	1
Attribute	R(RM1),W	R/W	R(RM0),W	R/W	R1,WX	R/W	R/W	R/W

Because the main timer is used for generating the oscillation stabilization wait time for main clock (MCLK), it can be used only after the main clock oscillation is stabilized.

The main timer is cleared when the main clock oscillation stops (MCEN = 0) or it is in the stop mode.

When the operation of the main timer is not allowed (MTE = 0), the main timer stops except that it is waiting for a main clock oscillation stabilization. The write operation to this register becomes enabled only when MCRDY = 1 except for MTIE. Thus a main timer clear executed by MTC = 1 in main clock oscillation stabilization wait status (MCEN = 1 and MCRDY = 0) is not effective.

When the main timer stops (MTE = 0) it will be cleared and while being cleared MTC = 1 will be read out.

At that time the main timer interrupt flag is not set. The main timer overflow period (MTS[3:0]) should be changed at the time when the main timer stops (MTE = 0).

When rewriting MTE = 1 with 0, the main timer will continue to operate until the MTC bit is set to "0". In this interval, the main timer interrupt flag may turn to "1". When writing MTC = 1, the main timer will continue to operate until the MTC bit is set to "0". In this interval, the main timer interrupt flag may turn to "1". If a MTE = 0 to 1 rewrite and a MTC = 1 write occur at the same time, the operation starts after a clear takes place, so the start will be delayed.

[bit7] MTIF (Main clock Timer Interrupt Flag): Main timer interrupt flag

The flag to indicate that an overflow happens in the interval for which the main timer has selected.

When the MTIE bit is "1" and this bit is set, a main timer interrupt request is generated.

Clear Factor	■ "0" write ■ A DMA transfer is generated by the main timer interrupt.
Set Factor	■ An overflow occurred in the interval set by MTS[3:0] ■ The end of oscillation stabilization wait time of the main clock after setting MCEN = 0 to 1. ■ The end of oscillation stabilization wait time of the main clock (MCLK) after exiting the stop mode. (A set will not take place at the end of oscillation stabilization wait time after reset by SINIT.)

Writing "1" to this bit is ineffective.

When the MTIE bit is set to "0", this bit will not be cleared by DMA transfer.

For read-modify-write instructions, "1" will be read out.

If a set factor and a clear factor occur at the same time, the set factor will take precedence.

[bit6] MTIE (Main clock Timer Interrupt Enable): Main timer interrupt enable

This bit controls interrupts by main timer overflow as follows.

MTIE	Main Timer Interrupt
0	Interrupt disabled (Initial value)
1	Interrupt enabled (The interrupt request is output when the MTIF bit is "1".)

[bit5] MTC (Main clock Timer Clear): Main timer clear

This bit clears the main timer.

MTC	Write	Read
0	Does nothing.	Operating normally
1	Clear the main timer.	Clearing the main timer

This bit automatically returns to "0" after writing "1".
For read-modify-write instructions, "0" will be read out.
When writing MTC = 1 at the time of MTC = 1, the second write will be ignored.

[bit4] MTE (Main clock Timer Enable): Main timer operation enable

This bit controls the operation of the main timer as follows.

MTE	Main Timer Operation
0	Operation disabled (Initial value)
1	Operation enabled

At the time of MTC = 1, MTE = 1 write is prohibited.

When you perform a PLL/SSCG clock oscillation stabilization wait, make sure to set this bit to "0" and stop the main timer.

[bit3 to bit0] MTS[3:0] (Main clock Timer interval Select): Main timer interval selection

These bits select the overflow interval of the main timer as follows.

MTS[3:0]	Main Timer Overflow Interval	At 4 MHz
1000	2^9 x main clock cycle	128.0 [µs]
1001	2^{10} x main clock cycle	256.0 [µs]
1010	2^{11} x main clock cycle	512.0 [µs]
1011	2^{12} x main clock cycle	1024.0 [µs]
1100	2^{13} x main clock cycle	2048.0 [µs]
1101	2^{14} x main clock cycle	4096.0 [µs]
1110	2^{15} x main clock cycle	8192.0 [µs]
1111	2^{16} x main clock cycle (Initial value)	16384.0 [µs]

The MTS[3] always reads "1".

Change MTS[3:0] at the time when the main timer stops (MTE = 0).

5.4.6 PLL Setting Register: PLLCR (PLL Configuration Register)

The bit configuration of the PLL setting register is shown.

This register configures the multiplication rate or division ratio in the PLL/SSCG clock oscillation circuit and the oscillation stabilization wait time.

PLLCR: Address 0514_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	POSW[3:0]				PDS[3:0]			
Initial value	1	1	1	1	0	0	0	0
Attribute	R1,WX	R,W	R,W	R,W	R,W	R,W	R,W	R,W

This register configures the multiplication rate in the PLL/SSCG clock oscillation circuit generating the PLL/SSCG clock (PLLSSCLK) from the main clock (MCLK).
When PLL/SSCG clock oscillation is allowed (CSELR.PCEN = 1), writing to this register has no effect.

[bit15, bit14] Reserved

Always write "0".

[bit13] Reserved

[bit12 to bit8] Reserved

Always write "0".

[bit7 to bit4] POSW[3:0] (Pll clock OSc Wait): PLL oscillation stabilization wait selection

These bits select the oscillation stabilization wait time for the PLL/SSCG clock (PLLSSCLK) as follows.

POSW[3:0]	PLL/SSCG Clock Oscillation Stabilization Wait time	At 4 MHz	At 8 MHz
1000	2^9 x main clock cycle	128.0 [µs]	64.0 [µs]
1001	2^{10} x main clock cycle	256.0 [µs]	128.0 [µs]
1010	2^{11} x main clock cycle	512.0 [µs]	256.0 [µs]
1011	2^{12} x main clock cycle	1024.0 [µs]	512.0 [µs]
1100	2^{13} x main clock cycle	2048.0 [µs]	1024.0 [µs]
1101	2^{14} x main clock cycle	4096.0 [µs]	2048.0 [µs]
1110	2^{15} x main clock cycle	8192.0 [µs]	4096.0 [µs]
1111	2^{16} x main clock cycle (Initial value)	16384.0 [µs]	8192.0 [µs]

POSW3 always reads "1".

Note:

The PLL/SSCG clock lock up time wait time specification in this product is 200 [µs]. Reserve the 200 [µs] wait time or more by either of the following methods.

- Select 256 [µs] POSW[3:0] or more.
- Reserve the 200 [µs] wait time or more by software processing, regardless of POSW[3:0] settings.

[bit3 to bit0] PDS[3:0] (Pll input clock Divider Select): PLL input clock divider selection

These bits select the main clock (MCLK) division for the PLL/SSCG input clock as follows.

PDS[3:0]	PLL/SSCG Input Clock Divider Selection
0000	PLL/SSCG input clock = Main clock / 1
0001	PLL/SSCG input clock = Main clock / 2
0010	PLL/SSCG input clock = Main clock / 3
0011	PLL/SSCG input clock = Main clock / 4
0100	PLL/SSCG input clock = Main clock / 5
0101	PLL/SSCG input clock = Main clock / 6
0110	PLL/SSCG input clock = Main clock / 7
0111	PLL/SSCG input clock = Main clock / 8
1000	PLL/SSCG input clock = Main clock / 9
1001	PLL/SSCG input clock = Main clock / 10
1010	PLL/SSCG input clock = Main clock / 11
1011	PLL/SSCG input clock = Main clock / 12
1100	PLL/SSCG input clock = Main clock / 13
1101	PLL/SSCG input clock = Main clock / 14
1110	PLL/SSCG input clock = Main clock / 15
1111	PLL/SSCG input clock = Main clock / 16

Follow the configuration steps for your appropriate PLL/SSCG and system specifications.
* For setting examples, see "5.5.1.2 PLL/SSCG Clock (PLLSSCLK)".

A set value is limited. For details on the setting time, see "5.5.1.3 Limitations when PLL/SSCG Clock Is Used".

5.4.7 Oscillation Stabilization Wait Setting Register: CSTBR (Clock STaBilization select Register)

The bit configuration of the oscillation stabilization wait setting register is shown.

This register configures the oscillation stabilization wait for each clock source.
The oscillation stabilization wait time set by this register will be used at the time when returning from the stop/watch mode. It will also be used for a period from the time when the oscillation of a clock which have not been selected as the source clock is allowed until the ready status (CMONR:*CRDY) of that clock switches to "1". If an oscillation stabilization wait is necessary at reset, it will always be set to the stabilization wait time selected as an initial value by this register. Write operations to MOSW[3:0] will not be effective at the main clock oscillation stabilization wait time (MCEN = 1 and MCRDY = 0).

CSTBR: Address 0516_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved			MOSW[3:0]			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,W0	R,W0	R,W0	R,W	R,W	R,W	R,W

[bit7] Reserved

[bit6 to bit4] Reserved

[bit3 to bit0] MOSW[3:0] (Main clock OSc Wait): Main clock oscillation stabilization wait selection

The main timer interval is set by the set value for MOSW[3:0].
These bits select the oscillation stabilization wait time for the main clock (MCLK) as follows.

MOSW[3:0]	Main Clock Oscillation Stabilization Wait Time	At 4 MHz
0000	2^{15} x main clock cycle (Initial value)	8 [ms]
0001	2^{1} x main clock cycle	500 [ns]
0010	2^{5} x main clock cycle	8 [µs]
0011	2^{6} x main clock cycle	16 [µs]
0100	2^{7} x main clock cycle	32 [µs]
0101	2^{8} x main clock cycle	64 [µs]
0110	2^{9} x main clock cycle	128 [µs]
0111	2^{10} x main clock cycle	256 [µs]
1000	2^{11} x main clock cycle	512 [µs]
1001	2^{12} x main clock cycle	1 [ms]
1010	2^{13} x main clock cycle	2 [ms]
1011	2^{14} x main clock cycle	4 [ms]
1100	2^{17} x main clock cycle	33 [ms]
1101	2^{19} x main clock cycle	131 [ms]
1110	2^{21} x main clock cycle	524 [ms]
1111	2^{23} x main clock cycle	2 [s]

Note:

Note that the determination detection is done while waiting for the oscillation stability when the cycle of the determination detection is shorter than a set cycle of this register when the Clock supervisor function is effective.

5.4.8 PLL Oscillation Stabilization Wait Timer Setting Register: PTMCR (PLL clock osc TiMer Control Register)

The bit configuration of the PLL oscillation stabilization wait timer setting register is shown.

This register controls the timer that works with the main clock that enters PLL/SSCG clock oscillation stabilization wait.
The PLL/SSCG clock oscillation stabilization wait timer is used only at the oscillation stabilization wait time of the PLL/SSCG clock.
The PLL/SSCG clock oscillation stabilization wait time becomes time set by PLLCR:POSW[3:0].
When PLL/SSCG clock oscillation is enabled (CSELR.PCEN = "1"), PLL/SSCG clock stabilization timer starts counting up.
After the oscillation stabilization time elapses, PLL/SSCG clock stabilization timer stops. Moreover, when PLL/SSCG clock oscillation stop (CSELR.PCEN = "0") is done, it is cleared.

PTMCR: Address 0517_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PTIF	PTIE	Reserved					
Initial value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PTIF (Pll clock osc wait Timer Interrupt Flag): PLL oscillation stabilization wait timer interrupt flag

This flag shows that the overflow at the time set by PLL oscillation stabilization wait selection (PLLCR: POSW [3:0]) was generated. If this bit is set when the PTIE bit is "1", PLL/SSCG clock oscillation stabilization wait timer interrupt request is generated.

Clear Factor	■ "0" write ■ Generation of DMA transfer with PLL/SSCG oscillation stabilization wait timer
Set Factor	■ End of the oscillation stabilization wait time for PLL/SSCG clock oscillation stabilization wait clock after PCEN = 0 to 1

Writing "1" to this bit is ineffective.

When the PTIE bit is "0", the clearness of this bit by the DMA transfer is not done.
For read-modify-write instructions, "1" will be read out.
If a set factor and a clear factor occur at the same time, the set factor will take precedence.

[bit6] PTIE (Pll clock osc wait Timer Interrupt Enable): PLL oscillation stabilization wait timer interrupt enable

This bit controls the interrupt by the overflow of PLL/SSCG clock oscillation stabilization wait timer as follows.

PTIE	Operation
0	Interrupt disabled (Initial value)
1	Interrupt enabled (The interrupt request is output when the PTIF bit is "1".)

[bit5 to bit0] Reserved

5.4.9 PLL/SSCG Clock Selection Register: CCPSSELR (CCtl Pll/Sscg clock SELect Register)

The bit configuration of the PLL/SSCG clock selection register is shown.

This register selects whether to use PLL or SSCG.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCPSSELR: Address 0520_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							PCSEL
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit7 to bit1] Reserved

[bit0] PCSEL (Pll Clock source SELect): PLL/SSCG Clock source selection

It selects the PLL/SSCG clock source.

PCSEL	PLL or SSCG
0	Select PLL.
1	Select SSCG.

Note:

SSCG (Because it is unused) always becomes a reset status for PCSEL = 0.
The PLL clock is supplied to CAN and OCDU for PCSEL = 1.

5.4.10 PLL/SSCG Output Clock Frequency Dividing Setting Register: CCPSDIVR (CCtl Pll/Sscg clock DIVide Register)

The bit configuration of the PLL/SSCG output clock frequency dividing setting register is shown.

This register sets the division ratio of the PLL/SSCG clock.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCPSDIVR: Address 0523_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	PODS[2:0]			Reserved	SODS[2:0]		
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W

[bit7] Reserved

[bit6 to bit4] PODS (Pll Oscillator Divider Select): Selection of PLL macro oscillation clock division ratio

These bits set the division ratio of the PLL clock.

PODS[2:0]	Division Ratio Setting
000	PLL clock = PLL macro oscillation clock /2
001	PLL clock = PLL macro oscillation clock /4
010	PLL clock = PLL macro oscillation clock /6
011	PLL clock = PLL macro oscillation clock /8
100	PLL clock = PLL macro oscillation clock /10
101	PLL clock = PLL macro oscillation clock /12
110	PLL clock = PLL macro oscillation clock /14
111	PLL clock = PLL macro oscillation clock /16

Note:

These bits can set only the even number dividing frequency. They cannot set the odd number dividing frequency.
Duty of the output clock becomes 50%.
Please set for the PLL clock to become 80 MHz or less. (The operation is not guaranteed if a frequency exceeding 80 MHz is set.)

[bit3] Reserved

[bit2 to bit0] SODS[2:0] (Sscg Oscillator Divider Select): Selection of SSCG macro oscillation clock division ratio

These bits set the division ratio of the SSCG clock.

SODS[2:0]	Division Ratio Setting
000	SSCG clock = SSCG macro oscillation clock /2
001	SSCG clock = SSCG macro oscillation clock /4
010	SSCG clock = SSCG macro oscillation clock /6
011	SSCG clock = SSCG macro oscillation clock /8
100	SSCG clock = SSCG macro oscillation clock /10
101	SSCG clock = SSCG macro oscillation clock /12
110	SSCG clock = SSCG macro oscillation clock /14
111	SSCG clock = SSCG macro oscillation clock /16

Note:

These bits can set only the even number dividing frequency. They cannot set the odd number dividing frequency.
Duty of the output clock becomes 50%.
Please set for the SSCG clock to become 80 MHz or less. (The operation is not guaranteed if a frequency exceeding 80 MHz is set.)

A set value is limited. For details on the setting time, see "5.5.1.3 Limitations when PLL/SSCG Clock Is Used".

5.4.11 PLL Feedback Frequency Dividing Setting Register: CCPLLFBR (CCtl PLL FB clock divide Register)

The bit configuration of the PLL feedback frequency dividing setting register is shown.

This register sets the multiplication rate of PLL.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCPLLFBR: Address 0525_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	IDIV[6:0]						
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7] Reserved

[bit6 to bit0] IDIV[6:0] (pll feedback Input DIVider ratio settings): PLL macro FB input division ratio setting

These bits set the PLL multiplication rate.

IDIV[6:0]	Division Ratio Setting
0000000 to 0001011	Setting is prohibited.
0001100	13
0001101	14
0001110	15
...	.....
1100010	99
1100011	100
1100100 to 1111111	Setting is prohibited.

A set value is limited. See "5.5.1.3 Limitations when PLL/SSCG Clock Is Used" when you set it.

5.4.12 SSCG Feedback Frequency Dividing Setting Register 0: CCSSFBR0 (CCtl SScg FB clock divide Register 0)

The bit configuration of the SSCG feedback frequency dividing setting register 0 is shown.

This register sets the multiplication rate N of SSCG. The multiplication rate of SSCG becomes P x N together with the setting of CCSSFBR1.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCSSFBR0: Address 0526_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		NDIV[5:0]					
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit7, bit6] Reserved

[bit5 to bit0] NDIV[5:0] (sscg feedback input N-DIVider ratio settings): SSCG macro FB input N division ratio setting

These bits set the multiplication rate N of SSCG.

NDIV[5:0]	Division Ratio Setting
000000	Setting is prohibited.
000001	2
000010	3
...	.....
111101	62
111110	63
111111	Setting is prohibited.

A set value is limited. For details on the setting time, see "5.5.1.3 Limitations when PLL/SSCG Clock Is Used".

5.4.13 SSCG Feedback Frequency Dividing Setting Register 1: CCSSFBR1 (CCtl SScg FB clock divide Register 1)

The bit configuration of the SSCG feedback frequency dividing setting register 1 is shown.

This register sets the multiplication rate P of SSCG. The multiplication ratio of SSCG becomes P x N along with the setting of CCSSFBR0.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCSSFBR1: Address 0527_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			PDIV[4:0]				
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W

[bit7 to bit5] Reserved

[bit4 to bit0] PDIV[4:0] (sscg feedback input P-DIVider ratio settings): SSCG macro FB input P division ratio setting

These bits set the multiplication rate P of SSCG.

PDIV[4:0]	Division Ratio Setting
00000	1
00001	2
00010	3
...	.....
11101	30
11110	31
11111	Setting is prohibited.

A set value is limited. See "5.5.1.3 Limitations when PLL/SSCG Clock Is Used" when you set it.

5.4.14 SSCG Configuration Setting Register 0: CCSSCCR0 (CCtl SSCg Config. Register 0)

The bit configuration of the SSCG configuration setting register 0 is shown.

This register sets various settings of SSCG.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCSSCCR0: Address 0529_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SFREQ[1:0]		SMODE	SSEN
Initial value	0	0	0	1	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W

[bit7 to bit5] Reserved

[bit4] Reserved

Writing has no effect on operation.

[bit3, bit2] SFREQ[1:0] (Spread spectrum modulation FREQuency settings): Spread spectrum modulation frequency settings

These bits set the spread spectrum modulation frequency of SSCG.

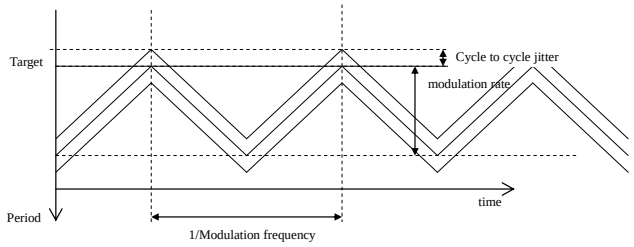
SFREQ[1:0]	Modulation Frequency
00	1/1024
01	1/2048
1x	1/4096

[bit1] SMODE (Spread spectrum modulation MODE settings): Spread spectrum modulation mode selection

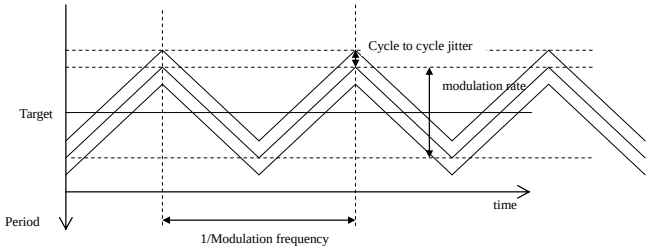
This bit sets the spread spectrum modulation mode of SSCG.

SMODE	Modulation Mode
0	Down Spread
1	Center Spread

Down Spread



Center Spread



[bit0] SSEN (Spread Spectrum ENable): Spread spectrum enable

This bit enables spread spectrum of SSCG.

SSEN	Spread Spectrum Enable
0	Spread spectrum disabled
1	Spread spectrum enabled

Note:

Spread spectrum modulation rate becomes 0% regardless of a setting of the CCSSCCR1:RATESEL when SSEN is set disabled.

5.4.15 SSCG Configuration Setting Register 1: CCSSCCR1 (CCtl SSCg Config. Register 1)

The bit configuration of the SSCG configuration setting register 1 is shown.

This register sets various settings of SSCG.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCSSCCR1: Address 052A$_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	RATESEL[2:0]			Reserved				
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R/W0	R/W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0

[bit15 to bit13] RATESEL[2:0] (spread spectrum modulation RATE SELection): Spread spectrum modulation rate selection

These bits set the spread spectrum modulation rate of SSCG.

RATESEL[2:0]	Modulation Rate
00x	0.5%
010	1%
011	2%
100	3%
101	4%
110	5%
111	Setting is prohibited.

[bit12 to bit10] Reserved

Writing to these bits has no effect.

[bit9 to bit0] Reserved

Always write "0".

5.4.16 Clock Gear Configuration Setting Register 0: CCCGRCR0 (CCtl Clock Gear Config. Register 0)

The bit configuration of the clock gear configuration setting register 0 is shown.

This register sets various settings of clock gear.

CCCGRCR0: Address 052D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	GRSTS[1:0]		Reserved				GRSTR	GREN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM0), W1	R/W

[bit7, bit6] GRSTS[1:0] (clock GeaR STatuS flags): Clock gear status flags

These bits display the status of Clock gear.

GRSTS[1:0]	Status
00	Stop in the state of clock gear low-speed oscillation or No use of clock gear (CCCGRCR0.GREN = 0) or In the status of PLL/SSCG reset (CSELR.PCEN = 0)
01	In operation of GEAR UP
10	Stop in the status of clock gear high-speed oscillation
11	In operation of GEAR DOWN

[bit5 to bit2] Reserved

[bit1] GRSTR (clock GeaR STaRt): Clock gear start

Writing "1" to this bit starts the operation of clock gear.
The operation of clock gear depends on the value of the GRSTS bits. (Gear up or gear down)

When GRSTS = 00

GRSTR	Operation
"0" write	Not affect the operation
"1" write	Start the operation of gear up.

When GRSTS = 01/11

GRSTR	Operation
"0" write	Not affect the operation
"1" write	Not affect the operation

When GRSTS = 10

GRSTR	Operation
"0" write	Not affect the operation
"1" write	Start the operation of gear down.

Note:

This bit can be written only when CSELR.CKS = 10 (PLL/SSCG clock (PLLSSCLK) selection) and CCCGRCR0.GREN = 1 (clock gear enable).
This bit is automatically cleared to "0" after the operation of clock gear up (down) complete. Also, this bit is cleared to "0" when CSELR.PCEN = 0 (PLL/SSCG clock oscillation stopped).
If a read-modify-write instruction is executed, "0" is always read from this bit. When writing is executed while this bit is "1", writing for the second and subsequent times is ignored.

[bit0] GREN (clock GeaR ENable): Clock gear enable

This bit enables the operation of clock gear.

GREN	Operation
0	Do not use clock gear.
1	Use clock gear.

Note:

This bit can be written only when PLL/SSCG clock oscillation is stopped (CSELR.PCEN = "0").

5.4.17 Clock Gear Configuration Setting Register 1: CCCGRCR1 (CCtl Clock Gear Config. Register 1)

The bit configuration of the clock gear configuration setting register 1 is shown.

This register sets various settings of clock gear.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCCGRCR1: Address 052E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	GRSTP[1:0]		GRSTN[5:0]					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7, bit6] GRSTP[1:0] (clock GeaR STeP select): Clock gear step selection

These bits select the step number at the time of clock gear up/down (the number of increment/decrement).

GRSTP[1:0]	Step Number
00	1
01	2
10	3
11	4

[bit5 to bit0] GRSTN[5:0] (clock GeaR STart step Number select): Clock gear start step number selection

These bits select the step number at the start of clock gear operation. These bits select the step number at the start of clock gear operation between 0 and 63.

GRSTN[5:0]	Step Number
000000	0
000001	1
000010	2
...	.....
111101	61
111110	62
111111	63

Note:

The gear does not operate at GRSTN = 111111 (number 63 of steps) setting.

5.4.18 Clock Gear Configuration Setting Register 2: CCCGRCR2 (CCtl Clock Gear Config. Register 2)

The bit configuration of the clock gear configuration setting register 2 is shown.

This register sets various settings of clock gear.
This register can be written only at PLL/SSCG clock oscillation stop (CSELR.PCEN = "0").

CCCGRCR2: Address 052F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	GRLP[7:0]							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit0] GRLP[7:0] (clock GeaR LooP number select): Clock gear loop number selection

These bits select the loop number of one step. The setting enabled number of iteration is between 1 to 256. Step is incremented/decremented when the number set to this bit is completed.

GRLP[7:0]	Loop Number
0000_0000	1
0000_0001	2
0000_0010	3
...	.....
1111_1101	254
1111_1110	255
1111_1111	256

5.4.19 Sync/Async Control Register

The bit configuration of the sync/async control register is shown.

This register selects the peripheral clock.

SACR: Address 1000_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							M
Initial Value	1	1	1	1	1	1	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W

[bit7 to bit1] Reserved

[bit0] M: Synchronous/asynchronous setting register of peripheral clock

This bit switches the peripheral clock when CPU selects the SSCG clock.

M	Synchronous/Asynchronous Setting
0	Synchronous (PLL/SSCG clock for CPU/peripheral)
1	Asynchronous (PLL/SSCG clock for CPU, PLL clock for peripheral)

5.4.20 Peripheral Interface Clock Divider

The bit configuration of peripheral interface clock divider is shown.

This register sets the dividing frequency of the peripheral clock.

PICD: Address 1001_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PDIV[3:0]			
Initial Value	1	1	1	1	0	0	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved

[bit3 to bit0] PDIV[3:0]: Peripheral clock division radio setting

These bits set the division ratio of the peripheral clock (PCLK2) from the PLL clock (PLLCLK) [non spread spectrum clock] at SACR.M = 1.

PDIV[3:0]	PLL Clock (PLLCLK)[Non Spread Spectrum Clock] -> PCLK2 Division Ratio
0000	No divide
0001	2 division
0010	3 division
0011	4 division (initial value)
0100	5 division
0101	6 division
0110	7 division
0111	8 division
1000	9 division
1001	10 division
1010	11 division
1011	12 division
1100	13 division
1101	14 division
1110	15 division
1111	16 division

Note:

Set this register so that the peripheral clock (PCLK2) definitely becomes 40 MHz or less.

5.5 Operation

This section explains operations of clock.

5.5.1 Oscillation Control
5.5.2 Oscillation Stabilization Wait
5.5.3 Selecting the Source Clock (SRCCLK)
5.5.4 Timer
5.5.5 Notes when Clocks Conflict
5.5.6 The Clock Gear Circuit
5.5.7 Operations during MDI Communications

5.5.1 Oscillation Control

This section explains oscillation control.

5.5.1.1 Main Clock (MCLK)

The main clock (MCLK) is shown.

The oscillation of the main clock stops on any of the following conditions.

- SINIT reset (See "Chapter: Reset")
- During the stop mode

After all the above conditions of the oscillation stop are cancelled and then the oscillation stabilization wait time which is set to CSTBR.MOSW[3:0] goes by, supplying the clock starts. The oscillation stabilization wait time specified by the initial value is required because CSTBR.MOSW[3:0] is initialized at the time of return from the reset input.

5.5.1.2 PLL/SSCG Clock (PLLSSCLK)

The PLL/SSCG clock (PLLSSCLK) is shown.

This LSI has PLL and SSCG (PLL which generates spread spectrum clock) and can select SSCG for reducing noise. The combinations of clocks which CPU and peripheral functions can select are as follows.

Table 5-3. Clock Mode

	Clock Mode		
	RUN1	**RUN2**	**RUN3**
CPU	PLL	SSCG	SSCG
CAN	PLL	PLL	PLL
Peripheral	PLL	SSCG	PLL
OCDU	PLL	PLL	PLL

The CPU/Peripheral (timer/communication) clock is selected by CCPSSELR.PCSEL. Also, when CPU is operated by the SSCG clock, peripheral (timer/communications) can be operated by the PLL clock. In this case, the peripheral clock is selected by SACR.M and divided by PICD.PDIV[3:0].

Note:

When the CPU is operated by the SSCG clock and the peripherals are operated by the PLL clock, because the asynchronization transfer enters between CPU and Peripheral, the penalty of 5 x PCLK2 to 8 x PCLK2 is added to the access cycle. In this case, the frequency of PCLK2 must be same as that of PCLK1.

Select synchronization with SACR:M when you want to make both CPU and Peripheral operation with the PLL clock.

The oscillation of the PLL/SSCG clock (PLLSSCLK) stops on any of the following conditions.

- After the occurrence of reset (the bus idle wait time before stop is required. See "Chapter: Reset").
- While the main clock oscillation stops (PCEN = 0)
- During the time of main clock oscillation stabilization wait (PCEN = 0)
- During the watch mode
- While a clock other than the PLL/SSCG clock (PLLSSCLK) are selected as the source clock and "0" is set to CSELR.PCEN

After all the above conditions of the oscillation stop are cancelled and then PLL/SSCG clock lock wait time which is set to PLLCR.POSW[3:0] goes by, supplying the clock starts. The PLL/SSCG clock oscillation stops until "1" is set to because CSELR.PCEN is initialized to "0" at the time of return from the reset input or the INIT status.

The formula for calculating the clock frequency and the multiplication rate related to PLL/SSCG is as follows:

(PLL/SSCG setting in Microcontroller unit)

- ☐ PLL/SSCG input clock frequency = (main oscillation frequency) / (PLLCR.PDS[3:0] division ratio)
- ☐ PLL multiplication rate = (CCPLLFBR.IDIV[6:0] FB input division ratio)
  SSCG multiplication rate = (CCSSFBR0.NDIV[5:0] FB input division ratio) x
- ☐ (CCSSFBR1.PDIV[4:0] FB input division ratio)
- ☐ PLL macro oscillation clock frequency = (PLL/SSCG input clock frequency) x PLL multiplication rate
  SSCG macro oscillation clock frequency = (PLL/SSCG input clock frequency) x SSCG multiplication rate
- ☐ PLL clock frequency = (PLL macro oscillation clock frequency) /
- ☐ (CCPSDIVR.PODS[2:0] division ratio)

- ☐ SSCG clock frequency = (SSCG macro oscillation clock frequency) /(CCPSDIVR.SODS[2:0] division ratio)

Figure 5-8. PLL Peripheral Block Diagram

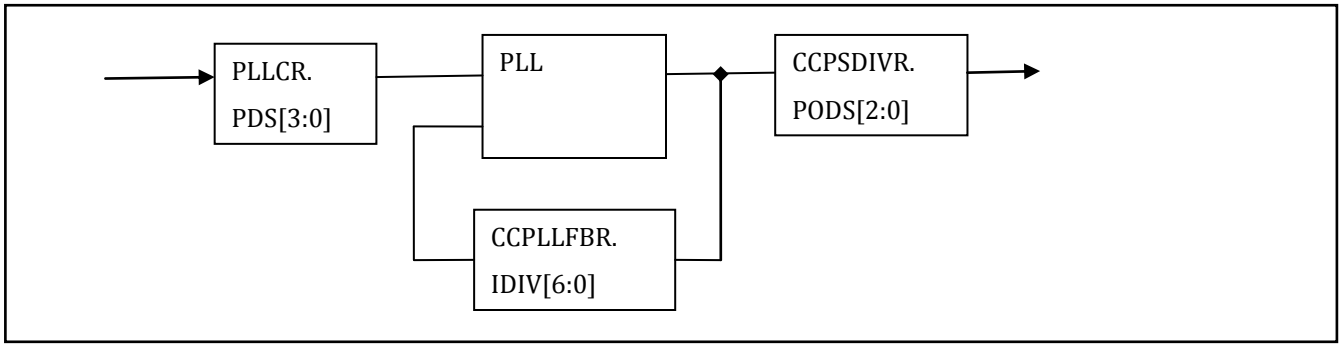
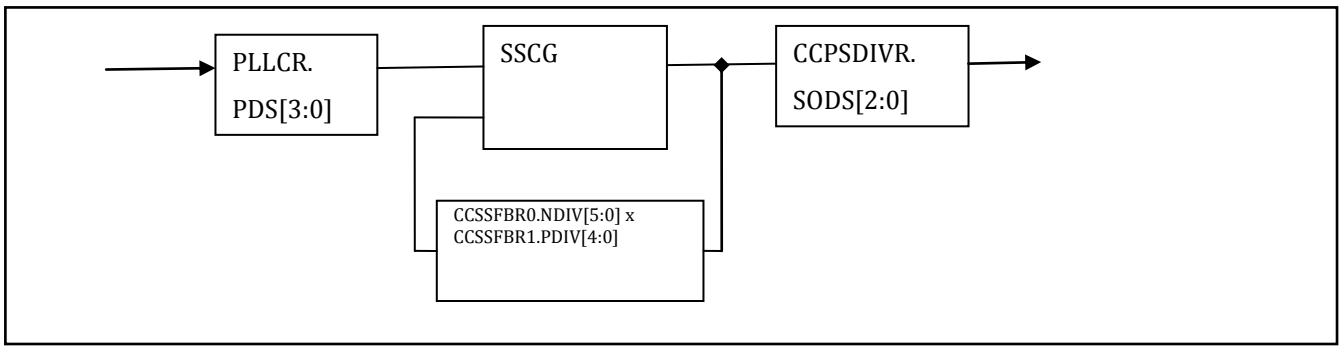


Figure 5-9. SSCG Peripheral Block Diagram



PLL/SSCG input clock, PLL/SSCG multiplication rate and PLL/SSCG macro oscillation clock must be set within the operating condition ranges for built-in PLL/SSCG in this series.

For the operating condition ranges of PLL/SSCG, see the following notes.

Note:

- In debug operation, PLL cannot stop because always supplying the PLL clock is required for MDI communication.
- Interrupts cannot be transferred normally in switching PLL-SSCG.
  Therefore, when switching PLL-SSCG synchronous/asynchronous, disable the interrupt from peripheral functions.
- The PLL/SSCG macro oscillation clock frequency has the upper bound and the lower bound. Set the multiplication rate of PLL/SSCG so as not to exceed the following range.
- PLL/SSCG in Microcontroller unit:
  - ☐ 200 MHz ≤ PLL macro oscillation clock frequency ≤ 320 MHz
  - ☐ 200 MHz ≤ SSCG macro oscillation clock frequency ≤ 288 MHz (Down Spread)

5.5.1.3 Limitations when PLL/SSCG Clock Is Used

The limitations of the PLL/SSCG clock used are shown.

Use the PLL/SSCG clock according to the following limitations.

Clock Control PLL Clock Frequency

Frequency (max)	FCTLR:FAW	CCPSSELR: PCSEL	Remarks
80 MHz	00	0	

Note:

Set PLLCR, CCPSDIVR and CCPLLFBR so as not to exceed frequency (max).
Set DIVR2, PICD to prevent the peripheral clock from exceeding 40 MHz.

Microcontroller Unit Clock Control SSCG Clock Frequency

Frequency (max)	FCTLR: FAW	CCPSSELR: PCSEL	CCSSCCR0: SSEN	CCSSCCR0: SMODE	CCSSCCR1: RATESEL	Remarks
72 MHz	00	1	1	0	000 to 110	Down Spread
72 MHz	00	1	1	1	000	Center Spread (0.5%)
72 MHz	00	1	1	1	010	Center Spread (1%)
72 MHz	00	1	1	1	011	Center Spread (2%)
71 MHz	00	1	1	1	100	Center Spread (3%)
71 MHz	00	1	1	1	101	Center Spread (4%)
70 MHz	00	1	1	1	110	Center Spread (5%)
72 MHz	01	1	0	0/1	000 to 110	Spread 0%
72 MHz	00	1	0	0/1	000 to 110	Spread 0%

Note:

Set CCPSDIVR, CCSSFBR0 and CCSSFBR1 so as not to exceed frequency (max).
Set DIVR2, PICD to prevent the peripheral clock from exceeding 40 MHz.

Relation between Modulation Rate and Division Ratio when SSCG is Used

CCSSCCR1:RATESEL[2:0]		CCSSFBR0:NDIV[5:0]		
Modulation Rate	Set Value	Range of Division Ratio	Set Value Lower Limit	Set Value Upper Limit
0.50%	00x	8 - 60	7_H	$3B_H$
1.00%	010	8 - 60	7_H	$3B_H$
2.00%	011	8 - 48	7_H	$2F_H$
3.00%	100	8 - 31	7_H	$1E_H$
4.00%	101	8 - 23	7_H	16_H
5.00%	110	8 - 18	7_H	11_H

5.5.2 Oscillation Stabilization Wait

Oscillation stabilization wait is shown.

This section describes oscillation stabilization wait for each clock input.

5.5.2.1 Conditions for Generating Stabilization Wait Time

Conditions for generating stabilization wait time are shown.

The cancellation of the oscillation stop control for each clock enters the oscillation stabilization wait status. After the oscillation stabilization wait time specified by each clock, the oscillation stabilization wait status is cancelled and supplying clock restarts.

The main clock (MCLK) enters the oscillation stabilization wait status when the oscillation stops before cancellation of reset because the setting register is initialized by reset. The main clock does not enter the oscillation stabilization wait status when the main clock oscillates by reset of INIT and RST level because the main clock oscillation does not stop by reset of INIT and RST level.

5.5.2.2 Selecting Stabilization Wait Time

Selecting the stabilization wait time is shown.

The stabilization wait time for each clock can be changed by setting of CSTBR and PLLCR.

- Initial values after reset for clock oscillation stabilization wait time
- Main clock: CSTBR.MOSW[3:0] bit 2^{15} x main clock period
- PLL/SSCG clock: PLLCR.POSW[3:0] bit 2^{16} x main clock period

The main oscillation stabilization wait time is always specified by the initial value because CSTBR.MOSW[3:0] is initialized by reset (INIT or RST). Except that case, the main oscillation stabilization wait time can be changed by setting to CSTBR.MOSW[3:0].

The PLL/SSCG clock lock wait time is always specified by the initial value because PLLCR.POSW[3:0] is initialized by reset (INIT or RST). Except that case, the PLL/SSCG clock lock wait time can be changed by setting to PLLCR.POSW[3:0]. Set "1" to CSELR.PCEN after setting to PLLCR.POSW[3:0]. For details, see the explanation of POSW in "5.4.6 PLL Setting Register: PLLCR (PLL Configuration Register)".

5.5.2.3 End of the Stabilization Wait Time

The end of the stabilization wait time is shown.

The operations are stopped while the clock which is selected as a source clock is the status of the oscillation stabilization wait time. The operations restart after the end of the oscillation stabilization wait time. You can verify that the clock which is not selected as the source clock has entered the oscillation stabilization wait time by checking the value of the ready bit corresponding to each clock for CMONR register when each clock is enabled.

Display of the clock oscillation stabilization wait status and the oscillation stabilization status

- Main clock: CMONR: MCRDY = "0", CMONR:MCRDY = "1"
- PLL/SSCG clock (PLLSSCLK): CMONR:PCRDY = "0", CMONR:PCRDY = "1"

5.5.3 Selecting the Source Clock (SRCCLK)

Selecting the source clock (SRCCLK) is shown.

This section explains the selection control of the source clock (SRCCLK) which functions as the operation clock.

5.5.3.1 Selecting the Source Clock at the Time of Initialization

Selecting the source clock at the time of initialization is shown.

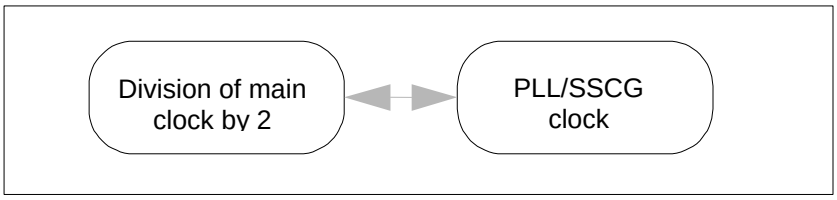
After reset (RST) the main clock (MCLK) divided by 2 is selected as the source clock. After program operation the source clock can be changed by setting CSELR.CKS[1:0].

5.5.3.2 Procedure of Switching the Source Clock

The procedure of switching the source clock is shown.

Set the oscillation stop as necessary because the value of the oscillation enable bit (CSELR.xCEN) is held, even though the source clock is switched.

Figure 5-10. Procedure of Switching the Source Clock



1. The main clock divided by 2 -> PLL/SSCG clock

   While selecting the main clock divided by 2 as the source clock (CMONR.CKM[1:0]=00)

   ↓

   PLL/SSCG multiplication rate, SSCG modulation, PLL/SSCG selection, setting PLL/SSCG clock lock wait time (setting PLLCR/CCPSSELR/CCPSDIVR/CCPLLFBR/CCSSFBR0/CCSSFBR1/CCSSCCR0/CCSSCCR1) --when PLL oscillation is not enabled--

   ↓

   Sets clock gear (CCCGRCR0.GREN/CCCGRCR1/CCCGRCR2)

   ↓

   Clears PLL/SSCG clock oscillation stabilization wait timer interrupt factor (PTIF = 0)

   ↓

   (as necessary) Sets PLL/SSCG clock oscillation stabilization wait timer interrupt enable (PTIE = 1)

   ↓

   PLL/SSCG clock oscillation begins (PCEN = 0 to 1)

   ↓

   PLL/SSCG clock lock wait loop (loop until when PCRDY = 1), or interrupt wait

   ↓

   Clears PLL/SSCG clock oscillation stabilization wait timer interrupt (PTIF = 0, PTIE = 0)

   ↓

   Switches from the source clock to PLL/SSCG clock (CSELR.CKS[1:0] = 00 to 10)

   ↓

   The clock gear begins (CCCGRCR0.GRSTR = 1)

   ↓

   Verifies that the clock gear high-speed oscillation is stopped (CCCGRCR0.GRSTS[1:0] = 10)

   ↓

   While selecting PLL/SSCG clock as the source clock (CMONR.CKM[1:0] = 10)

2. PLL/SSCG clock -> the main clock divided by 2

   ↓

   While selecting PLL/SSCG clock as the source clock (CMONR.CKM[1:0] = 10)

   ↓

   The clock gear begins (CCCGRCR0.GRSTR = 1)

   ↓

   Verifies that the clock gear low-speed oscillation is stopped (CCCGRCR0.GRSTS[1:0] = 00)

   ↓

   Switches the source clock to the main clock divided by 2 (CSELR.CKS[1:0] = 10 to 00)

   ↓

   While selecting the main clock as the source clock (CMONR.CKM[1:0] = 00)

Figure 5-11. Example of PLL/SSCG Mode Setting Main -> PLL/SSCG

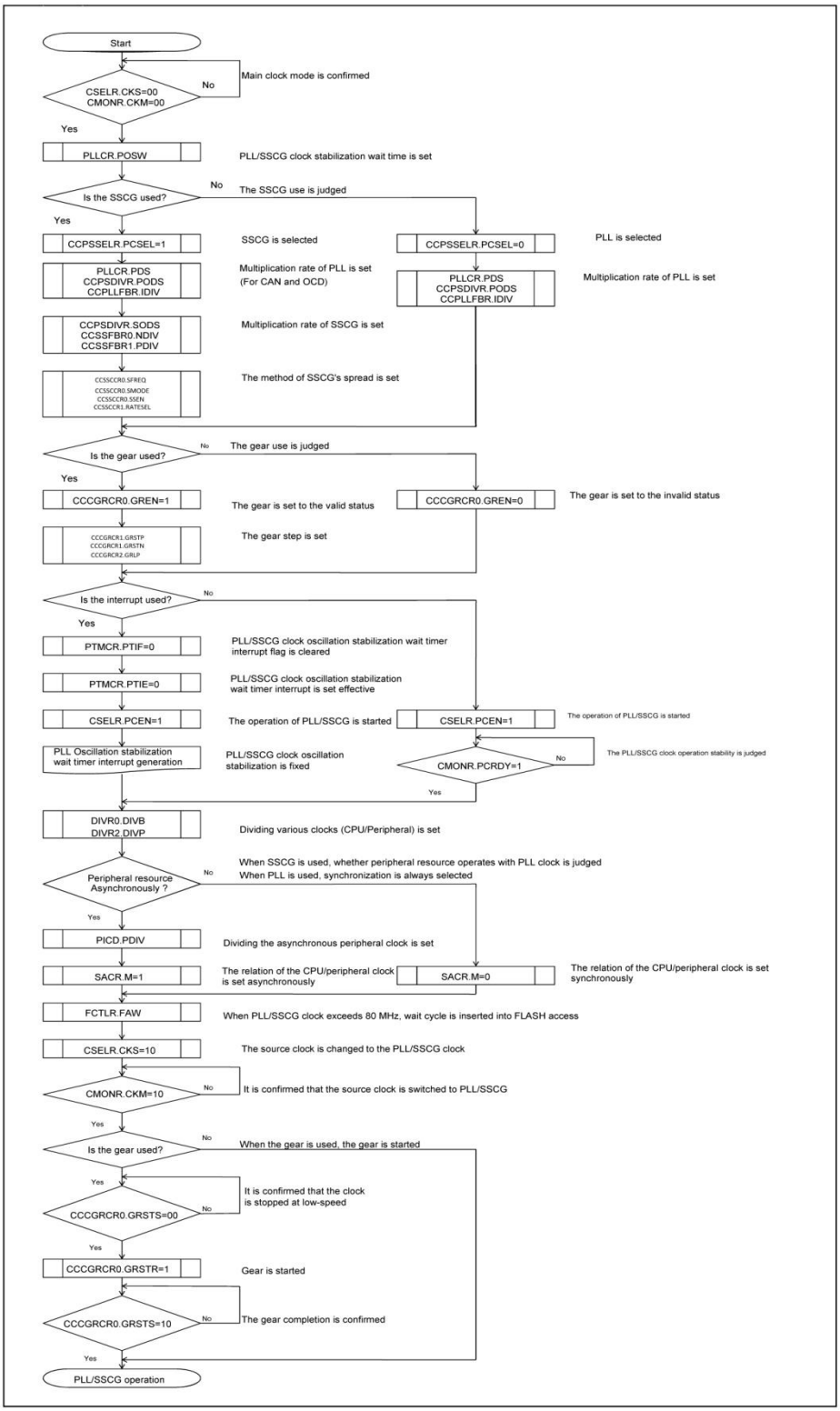
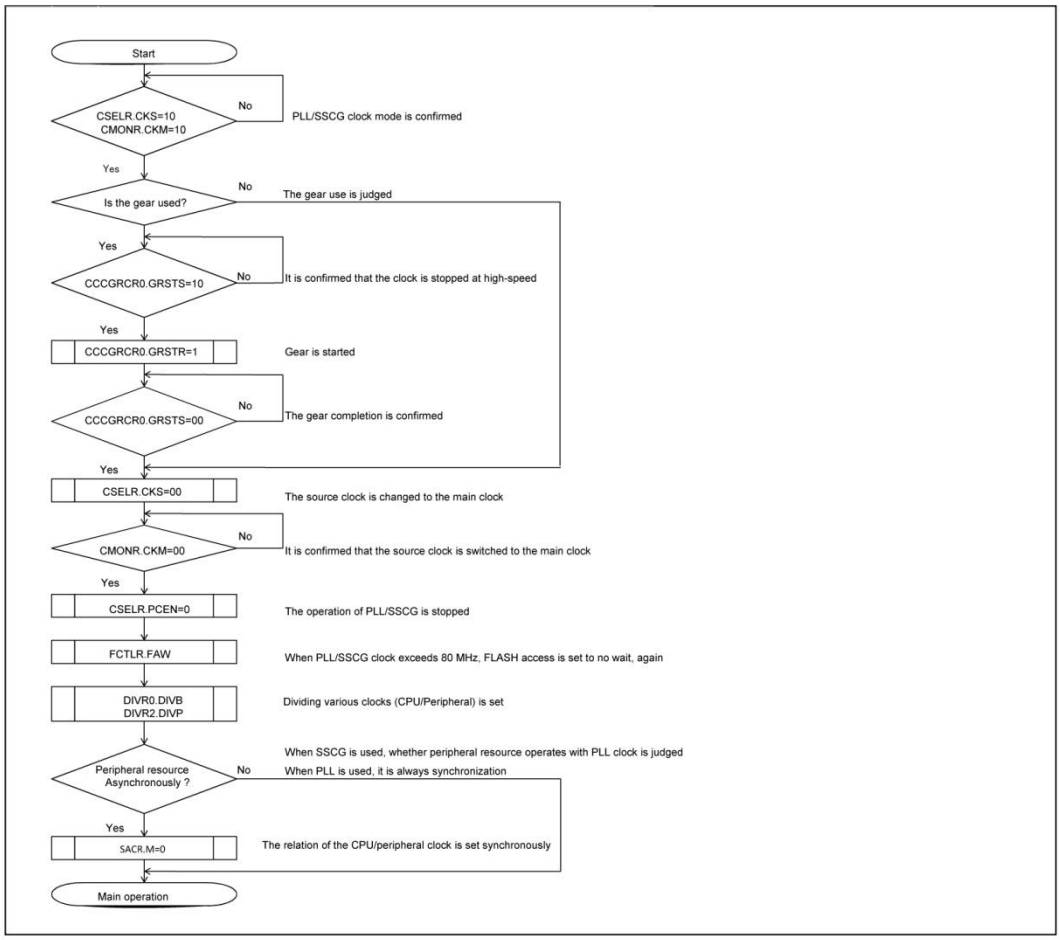


Figure 5-12. Example of PLL/SSCG Mode Setting PLL/SSCG -> Main



5.5.4 Timer

The timer is shown.

5.5.4.1 Main Clock Oscillation Stabilization Wait Timer (Main Timer)

The main clock oscillation stabilization wait timer (Main Timer) is shown.

The main timer is operated by the main clock (MCLK). This timer is used for the generation of the main clock oscillation stabilization wait time, and in main clock stabilization statuses other than those for oscillation stabilization wait, it can be used as the timer that generates an interrupt after the specified period.

5.5.4.2 PLL/SSCG Clock Oscillation Stabilization Wait timer (PLL Timer)

The PLL/SSCG clock oscillation stabilization wait timer (PLL Timer) is shown.

The PLL timer is operated by the main clock and only for generation of the PLL/SSCG oscillation stabilization wait time. This timer cannot be used for a general-purposed timer.

5.5.4.3 Setting

Setting is shown.

If the main timer operation is enabled (MTMCR.MTE = 1), the count operation of the main timer starts. If the main timer operation is disabled (MTMCR.MTE = 0), the count operation of the main timer stops and the main timer is cleared.

If the main timer is set to clear (MTMCR.MTC = 1), it is cleared. MTMCR.MTC = 1 is read until clear. The period of interrupt can be set by MTMCR.MTS[3:0]. When MTMCR.MTIE = 1, if MTMCR.MTIF = 1, the main timer interrupt occurs. MTMCR.MTIF is cleared by writing "0".

Note:

For setting the period of the timer interrupt (MTS), set the period to equal to or greater than PCLK x 5 clock. When the period of the timer interrupt is set to the extremely short time, the interrupt factor may not be set.

5.5.4.4 Procedure for Setting the Timer Interrupt

The procedure for setting the timer interrupt is shown.

This section describes the procedure for setting interrupt. The examples of the procedure for setting interrupt are shown as follows.

Sets the timer interrupt disable (MTMCR.MTIE = 0) and the interrupt flag clear (MTMCR.MTIF = 0)

↓

Sets the timer operation disable (MTMCR.MTE = 0)

↓

Verifies MTC = 0

↓

Sets the period of the timer (MTMCR.MTS = 1000 to 1111)

↓

Sets the timer interrupt enable (MTMCR.MTIE = 1)

↓

Sets the timer operation enable (MTMCR.MTE = 1)

↓

The interrupt occurs after setting time

↓

To the interrupt routine

↓

Sets the interrupt flag clear (MTMCR.MTIF = 0)*

↓

Verifies the interrupt flag (MTMCR.MTIF = 0)

↓

Program operations

↓

RETI

* Repeat reading until "0" is read because actual setting of the interrupt flag clear is delayed.

5.5.4.5 Timer Operations

Timer operations are shown.

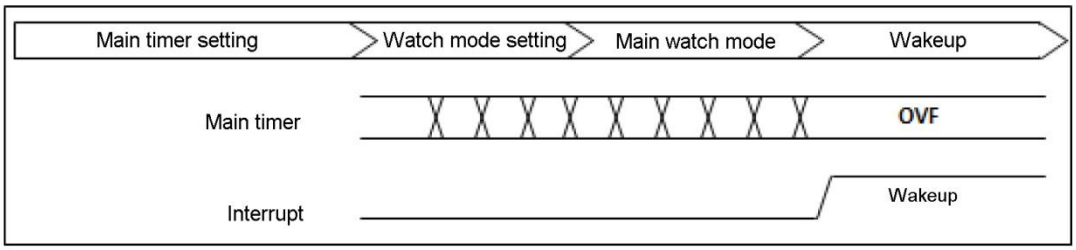
While MTMCR.MTE = 1, the main timer counts up by the main clock (MCLK). If the timer overflows by the period which is selected by MTMCR.MTS[3:0], MTMCR.MTIF is "1".

5.5.4.6 Watch Mode and Timer Interrupt

Watch mode and timer interrupt are shown.

Watch mode stops the specific functions and all operations other than timer. (See "Chapter: Power Consumption Control") The wake-up from the watch mode is enabled by using main timer interrupt. The example for switching of the watch mode in the setting of wake-up from the main timer is shown as follows.

Figure 5-13. Wake-up from the Watch Mode



5.5.5 Notes when Clocks Conflict

Notes when clocks conflict is shown.

Note that if peripheral interrupt activated by the frequency much lower than the CPU clock (CCLK) in the interrupt handler is cleared and the interrupt handler is immediately stopped, the peripheral cannot complete the internal process within the period of interrupt handler and the interrupt handler may be called in duplicate.

5.5.6 The Clock Gear Circuit

The clock gear circuit is shown.

When the main clock is switched to the PLL/SSCG clock or the PLL/SSCG clock is switched to the main clock, the power supply current fluctuates widely because the frequency fluctuates rapidly. Using the clock gear circuit in the part of the clock switching can gradually fluctuate the operating frequency from a low frequency to a high frequency or from a high frequency to a low frequency and therefore can reduce the fluctuation of the power supply current.

5.5.6.1 Procedure of Gear Up

The procedure of gear up is shown.

1. The clock of the start step set to the clock gear start step selection is output after the oscillation stabilization wait timer completes.
2. When the clock gear start (CCCGRCR0.GRSTR) is set to "1" and the rising is detected, the clock gear status flag (CCCGRCR0.GRSTS[1:0]) transits from "00" to "01". (gear up start)
3. The gear up is executed according to the clock gear step selection and the repeat number selection.
   The step number is the smaller and the repeat number is the larger that the operation changes the more gradually.
4. When the clock reaches the maximum step, the clock gear status flag (CCCGRCR0.GRSTS[1:0]) transits from "01" to "10". (the end of gear up, the gear stops)
   After this, a clock is output at the maximum step (64 steps).
5. After the gear stops, the clock gear start (CCCGRCR0.GRSTR) is cleared to "0" by hardware.

5.5.6.2 Procedure of Gear Down

The procedure of gear down is shown.

1. When the clock gear start (CCCGRCR0.GRSTR) is set to "1" and the rising is detected, the clock gear status flag (CCCGRCR0.GRSTS[1:0]) transits from "10" to "11". (gear down start)
2. The gear down is executed according to the clock gear step selection and the repeat number selection.
   The step number is the smaller and the repeat number is the larger that the operation changes the more gradually.
3. When the clock reaches the minimum step, the clock gear status flag (CCCGRCR0.GRSTS[1:0]) transits from "11" to "00". (the end of gear down, the gear stops)
   After this, the clock of the start step set for the clock gear start step selection is output.
4. After the gear stops, the clock gear start (CCCGRCR0.GRSTR) is cleared to "0" by hardware.

5.5.7 Operations during MDI Communications

Operations during MDI communications are shown.

The main oscillation is controlled so as not to be stopped during MDI communications even if the stop mode is transited to.

Moreover, during MDI high speed communication, the main oscillation is controlled so that the PLL reference clock is supplied even if CSELR.PCEN is cleared. The value of the register related to PLL is maintained and not updated. However, when software sets PLLCR.PCEN = 0, the value of the register related to PLL can be freely updated (written).

When a value set to the register related to PLL last time and a different value are written and the PLL/SSCG clock oscillation permission is assumed to be effective (CSELR.PCEN = 1), the frequency of the PLL clock is not updated. (PLL: because it maintains the locked status.)

Normally, always write the same value in the register related to PLL.

* The registers related to PLL are as follows.

- CCPSDIVR.PODS
- CCPLLFBR.IDIV
- PLLCR.PDS

6. Clock Reset State Transitions

This chapter explains clock reset state transitions.

6.1 Overview

6.2 Device States and Transitions

6.3 Device State and Regulator Mode Corresponding to those States

6.1 Overview

This section explains the overview of clock reset state transitions.

This chapter explains state transition of clock and reset. For features and settings of power consumption control state, see "Chapter: Power Consumption Control". For the operations of reset, see "Chapter: Reset". For the regulator mode, see "Chapter: Regulator Control".

6.2 Device States and Transitions

This section explains device states and transitions of clock reset state transitions.

6.2.1 Diagram of State Transitions

6.2.2 Explanation of Each States

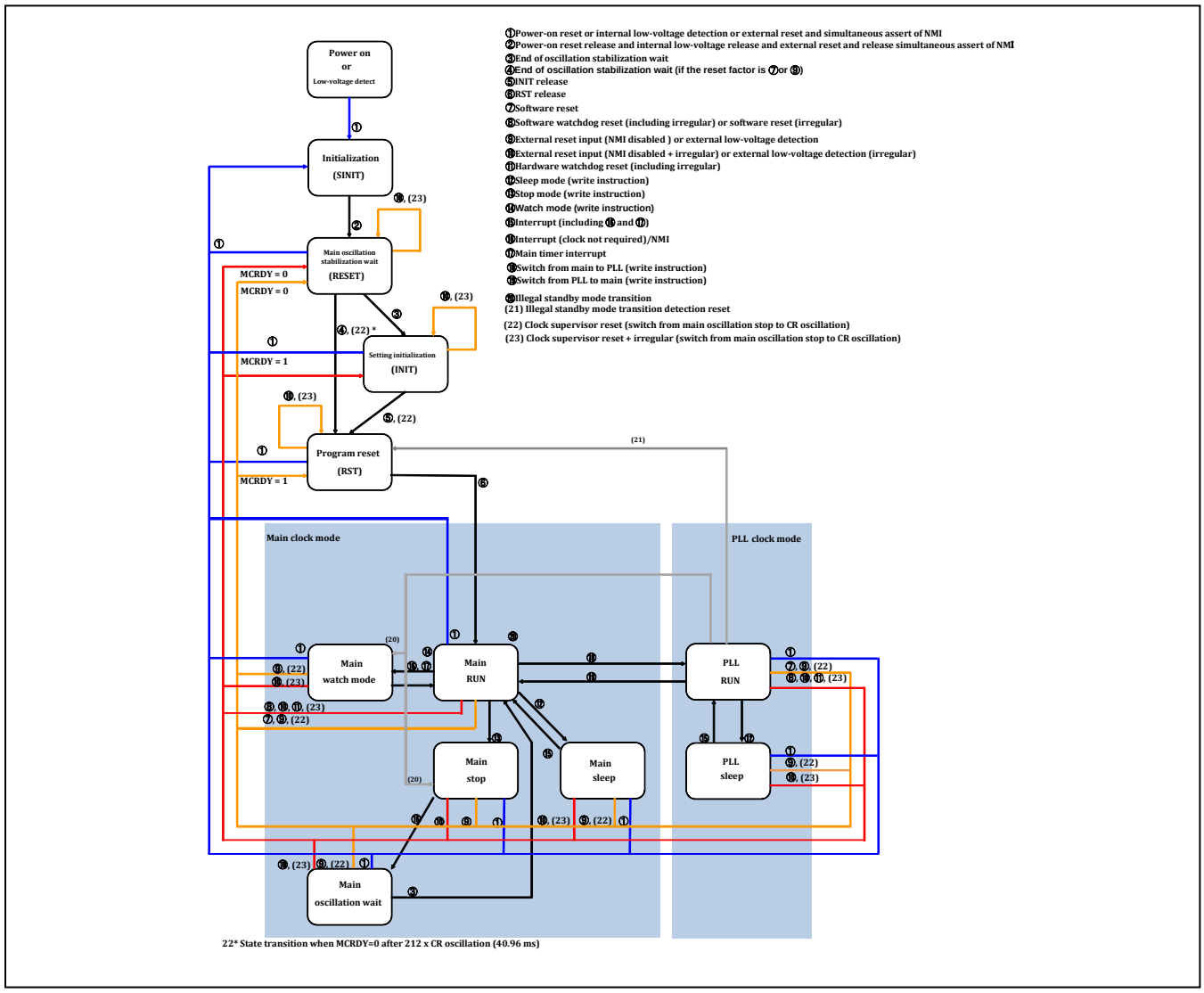
6.2.3 Priority of State Transition Requests

6.2.1 Diagram of State Transitions

This section shows diagram of state transitions.

The device state transitions for this series are shown below.

Figure 6-1. Diagram of Device State Transitions



Note:

The transition may be different from above diagram when connecting to OCD tool. See "Chapter: On Chip Debugger (OCD)" for details.

6.2.2 Explanation of Each States

This section explains each state.

Device operation states for this series are shown below.

RUN state (normal operation)

The program is running. All internal clocks supply and all circuits are ready to operate. High-impedance controls for the external pins in the stop state and watch mode state will be released.

Sleep mode

The program is not running. The state transits by program operations. There are some settings; one to stop program execution of the CPU only (CPU sleep mode) and the other to stop the CPU, on-chip bus and on-chip bus clock (HCLK) driven peripheral (bus sleep mode). For details, see "Chapter: Power Consumption Control".

Watch mode state

The devices are not running. The state transits by program operations. Internal circuits other than oscillation circuits (main clock generation unit) stop. Stop PLL oscillation before going into the watch mode state. It is also possible to use the external pins altogether (except for some pins) for high impedance by the settings. A specific enabled interrupt or main timer interrupt (clock not required) causes a transition to the RUN state. For details, see "Chapter: Power Consumption Control".

Stop state

The devices are not running. The state transits by program operations. All internal circuits will stop. Stop PLL oscillation before going into the stop mode state. It is also possible to use the external pins altogether (except for some pins) for high-impedance by the settings. Transits to the oscillation stabilization wait RUN state by NMI interrupt. For details, see "Chapter: Power Consumption Control".

Main oscillation stabilization wait (RUN) state

The devices are not running. Transits after returning from the stop state. All the internal circuits except for the timer operations for oscillation stabilization wait will stop. All internal clocks stop but the enabled oscillation circuits will still be running. After the elapse of the oscillation stabilization wait time interval set, transits to the RUN state (normal operation).

Main oscillation stabilization wait (reset) state

The devices are not running. Transits after returning from the initialization (SINIT) state. All the internal circuits except for the timer operations for oscillation stabilization wait will stop. All internal clocks stop but the main oscillation circuit will still be running. Outputs the program reset (RST) to the internal circuits. When the accepted reset level is an initialization reset, outputs also the setting initialization reset (INIT). After the elapse of the main clock oscillation stabilization wait time (2^{15} x main clock cycle), transits to the setting initialization (INIT) state.

Program reset (RST) state

The program is initialized. Transits after accepting the operation initialization reset (RST) request or at the end of the setting initialization (INIT) state. Outputs the program reset (RST) to the internal circuits. When transiting from the INIT state, OCD chip reset sequence (1026+3 PCLK cycles) will be performed.

Transits to the RUN state (normal operation) when removing the operation initialization reset (RST) request. For details, see "Chapter: Reset".

Setting initialization (INIT) state

All settings are initialized. Transits after accepting a setting initialization (INIT) request. The main oscillation circuit continues to run but the PLL will stop operations. Outputs a setting initialization (INIT) and a program reset (RST) to the internal circuits. Transits to the program reset (RST) state when removing the setting initialization (INIT) request and this state being released. For details, see "Chapter: Reset".

6.2.3 Priority of State Transition Requests

Priority of state transition requests is shown.

The state transition requests are prioritized in the following order in any states. However, since some requests are generated only in the specific states, they are enabled only in those states.

[Highest priority]	Initialization (SINIT) request
↓	Setting initialization (INIT) request
↓	The end of the oscillation stabilization wait time (generates an oscillation stabilization wait reset state and an oscillation stabilization wait RUN state only.)
↓	Program reset (RST) request
↓	Effective interrupt request (generates RUN, sleep, stop, watch mode states only)
↓	Stop mode request (register write) (generates RUN state only)
↓	Watch mode request (register write) (generates RUN state only)
[Lowest priority]	Sleep mode request (register write) (generates RUN state only)

6.3 Device State and Regulator Mode Corresponding to those States

Device state and regulator mode corresponding to those states are shown.

The regulator mode corresponding to each device state is shown in the following table. For regulator mode, see "Chapter: Regulator Control".

Table 6-1. Relationship between Device State and Regulator Mode

Device State	Main Clock	Regulator Mode
Main RUN	Oscillation	Main mode
Main sleep	Oscillation	Main mode
Main watch mode	Oscillation	Main mode
Main stop	Stop	Main mode
Main oscillation wait	Oscillation	Main mode
PLL RUN	Oscillation	Main mode
PLL sleep	Oscillation	Main mode

Note:

When OCD tool is connected, the regulator mode is a main mode in the above any tables.

7. Reset

This chapter explains the reset.

7.1 Overview

7.2 Features

7.3 Configuration

7.4 Registers

7.5 Operation

7.1 Overview

This section explains the overview of the reset.

When a reset factor is generated, the device terminates all programs and most of the hardware operations and initializes the state. This state is referred to as a reset.

7.2 Features

This section explains features of the reset.

This product, which has the following reset factors, issues a reset by accepting each factor to initialize the components in the device.

- Power-on reset
- RSTX pin input
- Watchdog reset 0 (software watchdog)
- Watchdog reset 1 (hardware watchdog)
- Software reset
- Illegal standby mode transition detection reset
- Flash security violation
- Internal low-voltage detection
- External low-voltage detection
- Clock supervisor reset

Other than the case of irregular reset (see "7.4.1 Reset Source Register: RSTRR (ReSeT Result Register)"), the contents of memory being accessed by the reset (RAM, Flash) will not be destroyed since all resets are issued once the completion of all bus accesses have been confirmed.

To issue a forced reset in case the bus does not return the response within a certain time frame, the device waits for the reset issue delay counter. If there is no response within the specified time frame, a reset will be issued regardless of whether the bus has responded. (Reset timeout)

See "Chapter: Clock Supervisor" for clock supervisor reset.

7.3 Configuration

This section explains the configuration of the reset.

Figure 7-1. Configuration Diagram of Reset

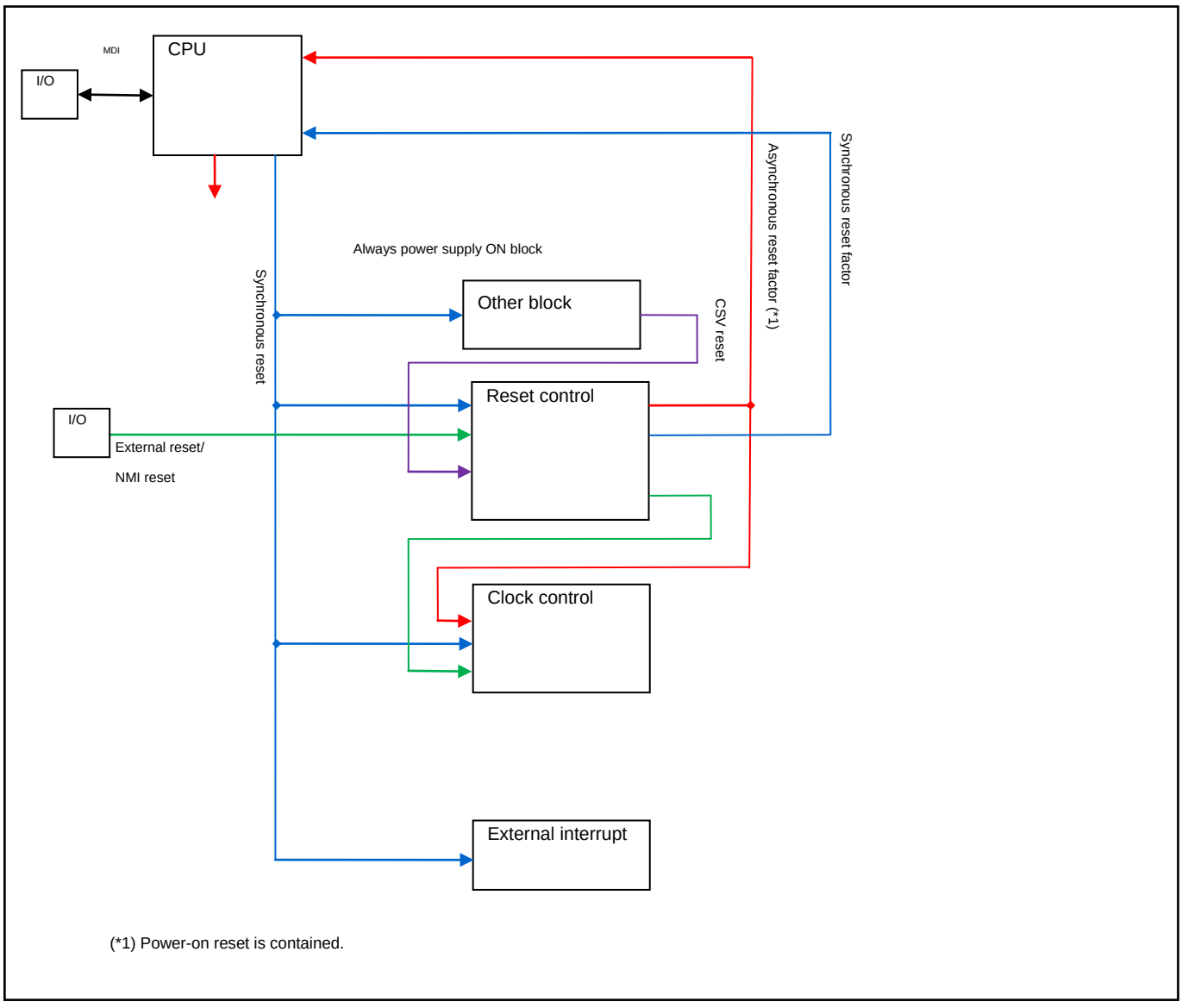


Figure 7-2. Configuration Diagram of Reset (Reset Control)

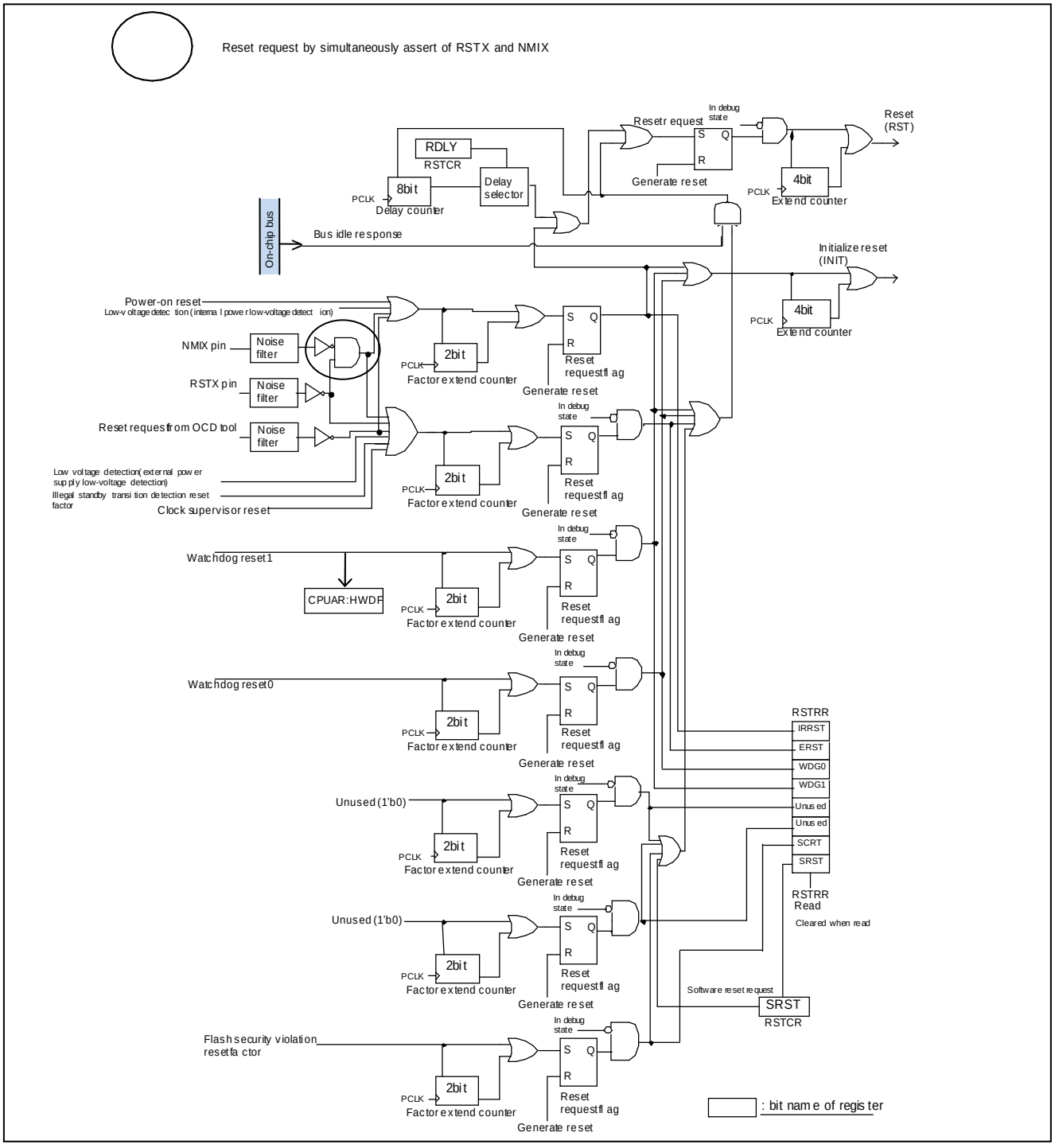
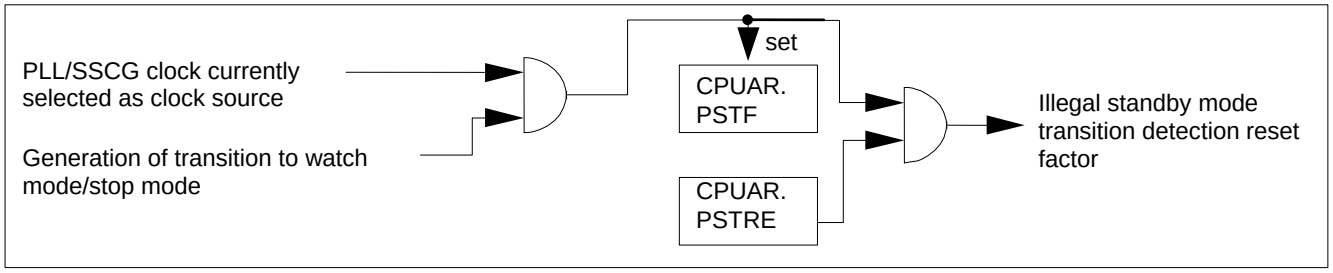


Figure 7-3. Generation Diagram of Illegal Standby Mode Transition Detection Reset Factor



7.4 Registers

This section explains the registers of the reset.

Table 7-1. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0480	RSTRR	RSTCR	Reserved	Reserved	Reset source register Reset control register
0x0518	Reserved	Reserved	CPUAR	Reserved	CPU abnormal operation register

Note:

Please note that the register of "Chapter: Power Consumption Control" is allocated in address 0x0482 and 0x0591.

7.4.1 Reset Source Register: RSTRR (ReSeT Result Register)

The bit configuration of the reset source register is shown.

This register displays various reset factors generated until just before.

RSTRR: Address 0480_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IRRST	ERST	WDG1	WDG0	Reserved		SCRT	SRST
Initial value	*	*	*	*	-	-	*	*
Attribute	R,WX	R,WX	R,WX	R,WX	RX,WX	RX,WX	R,WX	R,WX

* Due to a reset factor.

Note:

When this register is read out, all bits will be cleared.
This register is not cleared by reading in the debugging state.
Because each reset factor is masked in the debugging state, this register does not detect the reset factor either.

[bit7] IRRST (IRregular ReSeT): Irregular reset

This bit indicates that any of power-on reset, internal low-voltage detection, reset timeout, or simultaneous assert of RSTX and NMIX external pins has occurred, so that the bus access state when issuing a reset cannot be guaranteed. When this bit is "0" after the reset, no bus access was executed at the previous reset, which guarantees that memory contents have not been destroyed by the reset. When this bit is "1" after the reset, it is possible that a bus access was executed at the previous reset, which does not guarantee that memory contents have not been destroyed by the reset.

IRRST	Irregular Reset Detected
0	Irregular reset undetected
1	Irregular reset detected

This bit will be cleared when it is read out.

[bit6] ERST (External ReSeT): Reset pin input, illegal standby mode transition detection, external low-voltage detection, clock supervisor reset, simultaneous assert of RSTX and NMIX external pins

This bit indicates that there was a reset input from RSTX pin input, illegal standby mode transition detection reset, external low-voltage detection, clock supervisor reset or simultaneous assert of RSTX and NMIX external pins.
In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

ERST	RSTX Pin Reset Detection, Illegal Standby Mode Transition Detection, Clock Supervisor Reset, External Low-voltage Detection, or Simultaneous Assert of RSTX and NMIX External Pins
0	Undetected
1	Detected

This bit will be cleared when it is read out.

[bit5] WDG1 (WatchDoG reset 1): Watchdog reset 1

This bit indicates a reset from the watchdog timer 1.
In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

WDG1	Watchdog Timer 1 Reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.
The CPUAR register also has a flag that indicates a reset factor generation by the watchdog reset 1. The bit will not be cleared when the CPUAR register is read.

[bit4] WDG0 (WatchDoG reset 0): Watchdog reset 0

This bit indicates a reset from the watchdog timer 0.
In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

WDG0	Watchdog Timer 0 Reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

[bit1] SCRT (Flash SeCuRiTy violation): Flash security violation reset

This bit indicates that a flash memory security violation reset has occurred.
In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

SCRT	Flash Security Violation Reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

[bit0] SRST (Software ReSeT): Software reset

This bit indicates a reset by writing "1" to the RSTCR:SRST bit.
In case of a reset time out due to this reset factor, IRRST along with this bit will be "1".

SRST	Software Reset
0	Undetected
1	Detected

This bit will be cleared when it is read out.

7.4.2 Reset Control Register: RSTCR (ReSeT Control Register)

The bit configuration of the reset control register is shown.

This register controls various types of reset issuance.

RSTCR: Address 0481_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RDLY[2:0]			Reserved				SRST
Initial value	1	1	1	0	0	0	0	0
Attribute	R,W	R,W	R,W	R/W	R/W	R/W	R/W	R,W

[bit7 to bit5] RDLY[2:0] (Reset DeLaY): Reset issue delay

These bits set the reset timeout value. A reset will be issued if all bus operations become idle or the timer has counted to the reset timeout by this bit after a reset factor has been detected. (The latter is a case of irregular reset). These bits can be written for only once after the reset.

RDLY[2:0]	Reset Timeout Value
000	PCLK x 2 cycles
001	PCLK x 4 cycles
010	PCLK x 8 cycles
011	PCLK x 16 cycles
100	PCLK x 32 cycles
101	PCLK x 64 cycles
110	PCLK x 128 cycles
111	PCLK x 256 cycles (initial value)

[bit4 to bit1] Reserved

This has no effect on both writing and reading.

[bit0] SRST (Software ReSeT): Software reset

You will be able to generate a software reset request by reading RSTCR after writing "1" to this bit.
After you have written "1" to this bit, any values written to RSTCR will be ignored until a reset is generated, which means that register values cannot be changed.
Reading RSTCR while in the debugging state does not generate the reset.

SRST	Software Reset
0	No output (initial value)
1	The reset request is output by RSTCR reading.

7.4.3 CPU Abnormal Operation Register: CPUAR (CPU Abnormal operation Register)

The bit configuration of the CPU abnormal operation register is shown.

This register indicates the status and settings associated with the abnormal operation of CPU.

CPUAR: Address 051A$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PSTRE	Reserved				PMDF	PSTF	HWDF
Initial value	0	0	0	0	*	*	*	*
Attribute	R/W	R0,WX	R0,WX	R0,WX	RX,WX	R(RM1),W	R(RM1),W	R(RM1),W

* It will be initialized to "0" by RSTX pin asserts (including simultaneous assert with NMIX). For other reset factors, the bit will not be initialized.

[bit7] PSTRE (illegal PLL-run to STandby Reset Enable): Illegal standby mode transition detection reset enable

This bit configures whether or not to issue a reset when a watch mode or a stop mode transition has been detected (illegal standby mode transition) with the PLL clock selected as a clock source.
When enabled, a reset due to the illegal standby mode transition detection factor will be generated at a transition from the PLL-run state to watch mode or stop mode.

PSTRE	Description
0	Reset will not be generated (initial value)
1	Reset generation enabled

Note:

When you set this bit, make sure to clear the PSTF bit by writing "0" to the PSTF bit before setting this bit. If you set this bit before clearing the PSTF bit, a reset may be generated since the value of the PSTF bit after the power-on reset is indefinite.

[bit2] PMDF (PLL mode Main clock Down detection Flag): PLL mode main oscillation determination detection flag

When the clock supervisor does the main oscillation determination detection when PLL output is selected as a clock source, this bit is set. Moreover, the source clock is written automatically in main mode (CKS=CKM=00), and reset (RST level) is generated at once.
If a read-modify-write instruction is executed, "1" will be read out.

PMDF	Read	Write
0	The main oscillation determination detection is not in PLL mode. (initial value)	Clear this bit
1	The main oscillation determination detection is in PLL mode.	No effect

[bit1] PSTF (illegal PLL-run to STandby Flag): Illegal standby mode transition detection flag

This bit will be set when a watch mode or a stop mode transition has been detected (illegal standby mode transition) with the PLL clock selected as a clock source. Moreover, the source clock is written automatically in main mode (CKS=CKM=00). When the PSTRE bit is "1", reset (RST level) is generated.
This bit is cleared by writing "0".
If a read-modify-write instruction is executed, "1" will be read out.

PSTF	Read	Write
0	No illegal standby mode transition has been detected.	Clear this bit
1	Illegal standby mode transition has been detected.	No effect

[bit0] HWDF (Hardware WatchDog Flag): Hardware watchdog detection flag

When a reset factor for the watchdog timer 1 (hardware watchdog) has been detected, this bit will be set.
This bit is cleared by writing "0".
If a read-modify-write instruction is executed, "1" will be read out.

HWDF	Read	Write
0	No watchdog timer 1 (hardware watchdog) reset factor has been generated.	Clear this bit
1	Watchdog timer 1 (hardware watchdog) reset factor has been generated.	No effect

The set factor is given to priority when a set factor and a clear factor are generated at the same time.

Note:

There is a detection flag also in RSTRR.WDG1, and the factor disappears when read once because it is read clear. Because CPUAR.HWDF is maintained, the factor is maintained until clearing.

7.5 Operation

This section explains the reset operation.

This section explains each of the reset operations for this product.

7.5.1 Reset Level

The reset level is shown.

The following two levels of resets are available with this product.

- Initialize reset (INIT)
- Reset (RST)

Note:

Except the registers for debug interface (OCDU), the registers initialized by the reset of both levels are the same for this product.

7.5.1.1 Initialize Reset (INIT)

The initialize reset (INIT) is shown.

It initializes all register settings and the entire hardware. It terminates the CPU programs running, and the program counter will be initialized. All peripheral circuits will be initialized. A main oscillation circuit continues to run. If it was inactive, it starts running again. In this case, PLL becomes inactive.

This reset level is applied at a reset by the following reset factors.

- Irregular reset
- Watchdog reset 0, 1
  Only the following register will be initialized by this reset level.
- Register of the debug interface (OCDU)

7.5.1.2 Reset (RST)

The reset (RST) is shown.

It initializes the entire hardware and all registers except the ones initialized only by the initialize reset (INIT). It terminates the CPU programs running, and the program counter will be initialized. All peripheral circuits will be initialized.
When an initialize reset (INIT) is issued, a reset (RST) is issued at the same time.
The reset in the entire document indicates this reset level unless otherwise specified.

7.5.2 Reset Factor

The reset factor is shown.

This section explains each of the reset factors for this product.

7.5.2.1 Power-on Reset

The power-on reset is shown.

It is a reset factor generated when detecting the power has turned on.
All resets due to this reset factor are detected as an irregular reset and issue an initialize reset (INIT).

7.5.2.2 RSTX Pin Input

The RSTX pin input is shown.

It is a hardware reset input from the outside of the device.
Reset by this reset factor is detected as irregular reset only at the reset timeout or simultaneous assert of the NMIX pin.
Other than the irregular reset detection, a reset (RST) will be issued.

7.5.2.3 Watchdog Reset 0

The watchdog reset 0 is shown.

It is a hardware reset input from the FR81S-core built-in watchdog timer 0 (software watchdog).
Resets due to this reset factor will be detected as an irregular reset only at the time of reset timeout. Regardless of whether an irregular reset has been detected, an initialize reset (INIT) will be issued.

7.5.2.4 Watchdog Reset 1

The watchdog reset 1 is shown.

It is a hardware reset input from the FR81S-core built-in watchdog timer 1 (hardware watchdog).
Resets due to this reset factor will be detected as an irregular reset only at the time of reset timeout. Regardless of whether an irregular reset has been detected, an initialize reset (INIT) will be issued.

7.5.2.5 External Low-Voltage Detection Reset

The external low-voltage detection reset is shown.

Low-voltage detection (external voltage) is a hardware reset input from the low-voltage detection circuit located inside of the device.
Resets due to this reset factor will be detected as an irregular reset only at the time of reset timeout. Other than the irregular reset detection, a reset (RST) will be issued.
See "Chapter: Low-Voltage Detection (External Low-Voltage Detection)" for details on voltage detection.

7.5.2.6 Illegal Standby Mode Transition Detection Reset

The illegal standby mode transition detection reset is shown.

It is a hardware reset generated when a watch mode or a stop mode transition has been detected (illegal standby mode transition) with the PLL clock selected as a clock source.
Resets due to this reset factor will be detected as an irregular reset only at the time of reset timeout. Other than the irregular reset detection, a reset (RST) will be issued.

7.5.2.7 Internal Low-Voltage Detection Reset

The internal low-voltage detection reset is shown.

Low-voltage detection (internal voltage) is a hardware reset input from the low-voltage detection circuit located inside of the device.
Resets due to this reset factor will be detected as an irregular reset and an initialize reset (INIT) will be issued.
See "Chapter: Low-Voltage Detection (Internal Low-Voltage Detection)" for details on voltage detection.

7.5.2.8 Flash Security Violation Reset

The flash security violation reset is shown.

It is a reset issued when a violation of flash memory security protection has occurred.
Resets due to this reset factor will be detected as an irregular reset only at the time of reset timeout.
Other than the irregular reset detection, a reset (RST) will be issued.

7.5.2.9 Software Reset (RSTCR:SRST)

The software reset (RSTCR:SRST) is shown.

It is a software reset generated inside of the device.

This reset will be issued when you read RSTCR after writing "1" to the bit0:SRST bit of the RSTCR.
Resets due to this reset factor will be detected as an irregular reset only at the time of reset timeout. Other than the irregular reset detection, a reset (RST) will be issued.

[Example] Sample program of a software reset issue

```
LDI     #value_of_reset, R0     ; SRST bit="1"
LDI     #_RSTCR, R12            ;
STB     R0, @R12                ; Write
LDUB    @R12, R0                ; Read (generation of software reset request)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy process for pipeline adjustment
```

7.5.3 Reset Acceptance

The reset acceptance is shown.

This section explains the acceptance processing of each reset factor.

7.5.3.1 Generation of Reset Request

The generation of a reset request is shown.

A reset request will be generated when at least one reset factor is retrieved. The reset request will be notified to the internal bus controller, and the following processing will be executed.

- Stop the CPU programs running (same processing as sleep mode)
- Acquire bus control right of the on-chip bus
- Confirm that idle request has been notified to all busses

7.5.3.2 Acceptance of Reset Request

The acceptance of a reset request is shown.

Once all processing for the reset request completes, the component where a reset is issued accepts the reset request and issues a reset of which level corresponds to the reset factor. If the reset issue delay counter overflows (= reset timeout occurs), the reset request is accepted without waiting for the completion of reset request processing, and an irregular reset will be issued.

7.5.3.3 Reset Issue Delay Counter

The reset issue delay counter is shown.

As soon as a reset request is generated, the 8-bit reset issue delay counter starts counting. If the delay cycle specified by the bit7 to bit5:RDLY[2:0] bits of the RSTCR register has elapsed without a reset being issued and the counter overflows (= reset timeout occurs), an irregular reset will be issued.
The RDLY[2:0] bit of the RSTCR will be initialized by a reset. This bit can be rewritten for once only after a reset is released. If the delay cycle is set for a short time, it is more likely to generate an irregular reset. If the delay cycle is set for a long time, it might take a long time for a reset to be issued since the generation of a reset factor.

7.5.3.4 Irregular Reset

The irregular reset is shown.

If a reset is issued without confirming the completion of reset request processing, an irregular request will be generated. Once an irregular reset is generated, the following processing will be executed.

- Issue initialize reset (INIT) regardless of the type of reset factor.
- Set the bit7:IRRST bit of RSTRR register to "1".

When an irregular reset occurs, there is no guarantee that memory contents were not destroyed by the reset since a bus access may have been executed at the time of inputting the reset. The irregular reset does not necessarily mean that the memory contents were destroyed, but how the bus access was executed cannot be identified.

7.5.4 Reset Issue

This reset issue is shown.

A reset will be issued after a reset request has been accepted. This section explains each type of reset issue.

7.5.4.1 Super Initialize Reset (SINIT)

The super initialize reset (SINIT) is shown.

The super initialize reset (SINIT) will be issued first for power-on reset, internal low-voltage detection, or simultaneous assert of RSTX and NMIX. This reset is exclusively used for initializing the indefinite state of division circuits and so on.

While this reset is being issued, all clocks become inactive.

When this reset is issued, an initialize reset (INIT) and a reset (RST) will be always issued at the same time.

This reset initializes the clock control register.

This reset involves the wait time of main clock oscillation to be stabilized. Along with the control register initialization, the oscillation stabilization wait time is 2^{15} x main clock cycle.

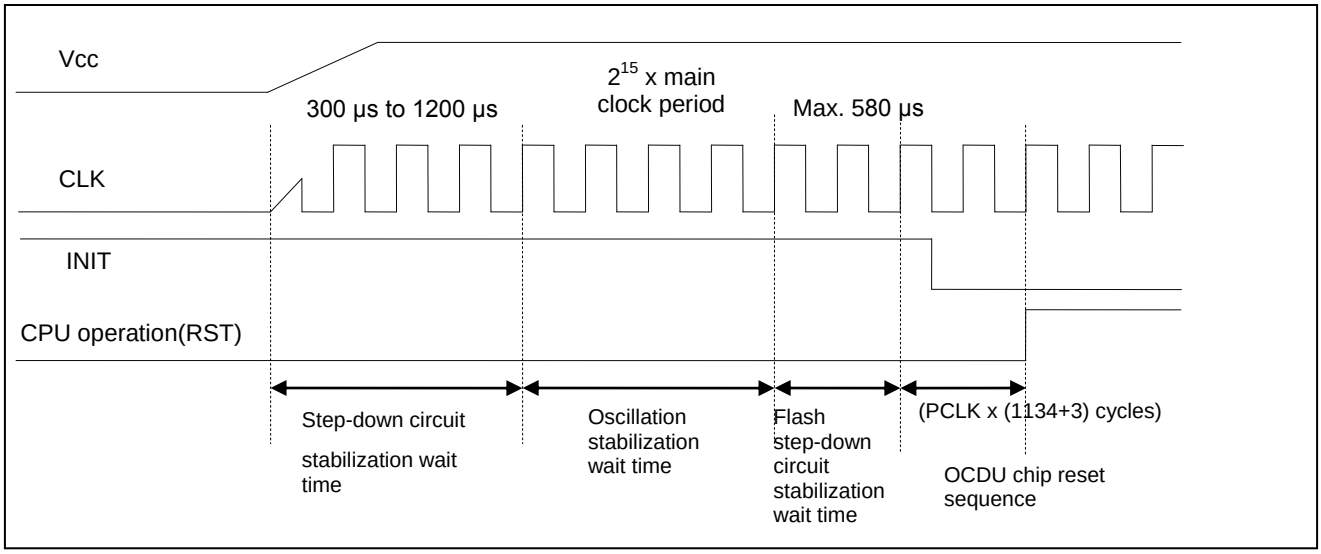
Table 7-2. Oscillation Stabilization Wait Time (SINIT)

Type	Main Clock Oscillation Stabilization Wait Time
Power-on reset	2^{15} x Main clock cycle
Internal low-voltage detection	2^{15} x Main clock cycle
Simultaneous assert of RSTX and NMIX	2^{15} x Main clock cycle

Note:

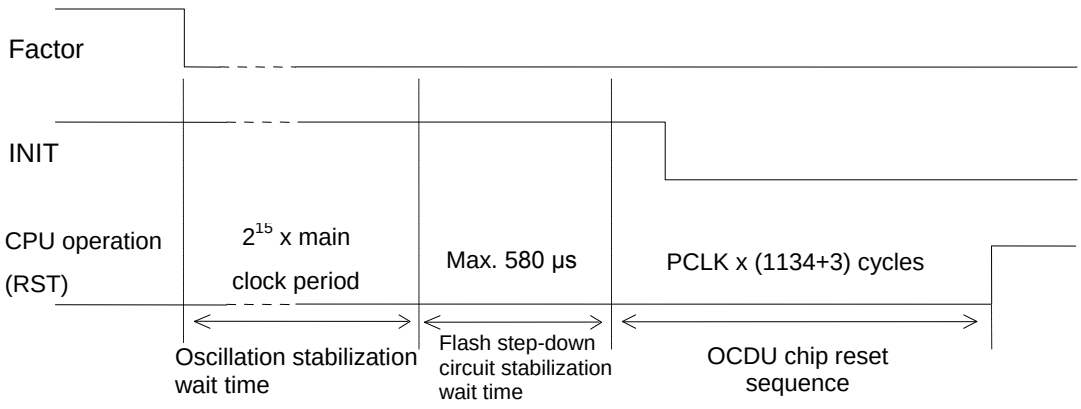
The oscillation stabilization wait times in the above table do not include the regulator stabilization wait times and FLASH stabilization wait times associated with power-on and voltage restore. These voltage step-down circuit stabilization times (300 µs to 1200 µs, and a maximum of 580 µs) are needed during a power-on reset.

Figure 7-4. Oscillation Stabilization Wait Time for Power-on Reset



The following describes each reset issue sequence after reset factors of this reset have been released.

Figure 7-5. Super Initialize Reset (SINIT) Sequence



Because the clock settings register is initialized by reset, the period of the peripheral clock (PCLK) is 8 times the period of the main clock (MCLK).

7.5.4.2 Initialize Reset (INIT)

The initialize reset (INIT) is shown.

If a reset factor of the initialize reset (INIT) level occurs, an initialize reset (INIT) and a reset (RST) will be issued at the same time. This reset is exclusively used for initializing the registers that cannot be initialized by a reset (RST).

While this reset is being issued, all clocks become active. When this reset is issued, a reset (RST) will be always issued at the same time. Although this reset initializes the clock control register, the oscillation of the clock does not change while the main clock (MCLK) is oscillating.

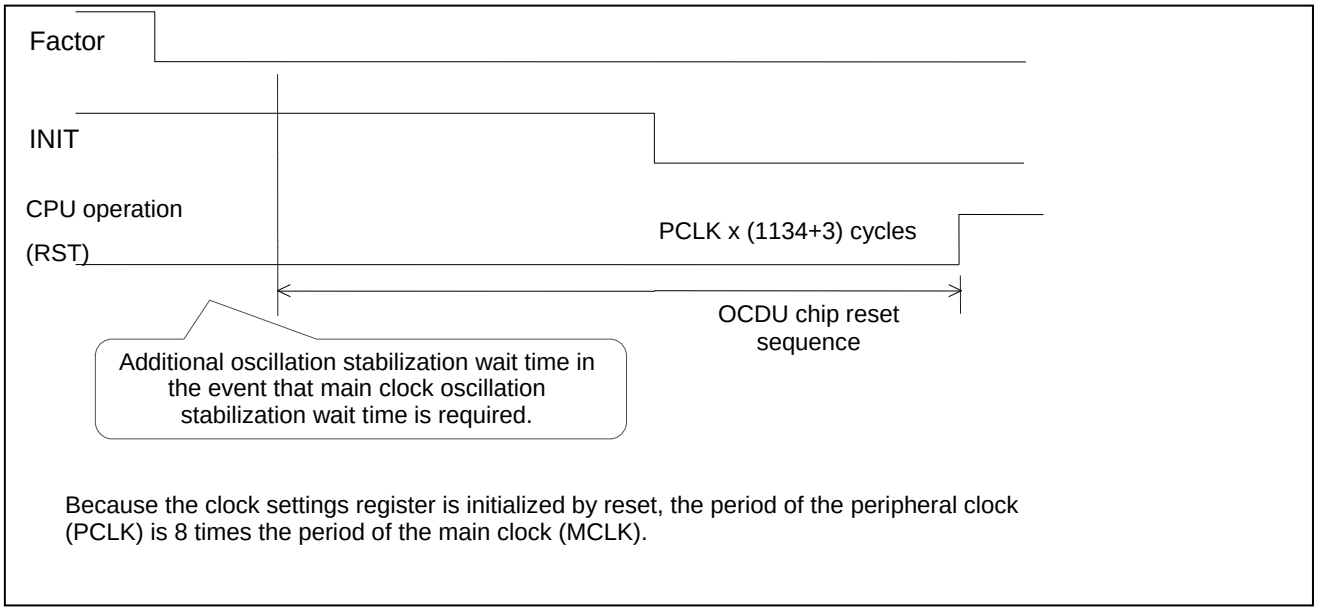
If the main clock is inactive such as in a stop mode, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15} x main clock cycle).

Table 7-3. Oscillation Stabilization Wait Time (INIT)

Is Main Clock Oscillation Inactive before Inputting a Reset?	Main Clock Oscillation Stabilization Wait Time
No	None
Yes	2^{15} x Main clock cycle

The following describes each reset issue sequence after reset factors of this reset have been released.

Figure 7-6. Initialize Reset (INIT) Sequence



7.5.4.3 Reset (RST)

The reset (RST) is shown.

If a reset factor that is not the initialize reset (INIT) level occurs, only a reset (RST) will be issued.

This reset is used for initializing the entire hardware except some registers (see "7.5.1.1Initialize Reset (INIT)").

While this reset is being issued, all clocks become active.

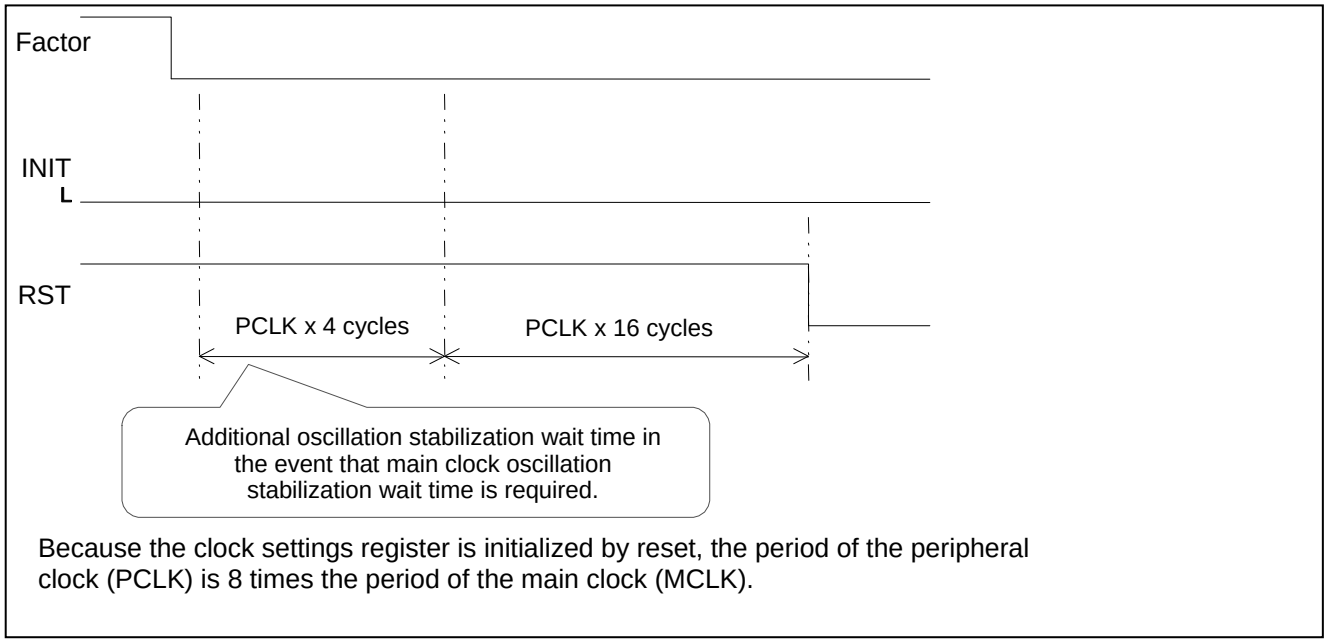
If the main clock is inactive such as in a stop mode before the reset, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15} x main clock cycle).

Table 7-4. Oscillation Stabilization Wait Time (RST)

Is Main Clock Oscillation Inactive before Inputting a Reset?	Main Clock Oscillation Stabilization Wait Time
No	None
Yes	2^{15} x Main clock cycle

The following describes each reset issue sequence after reset factors of this reset have been released.

Figure 7-7. Reset (RST) Sequence

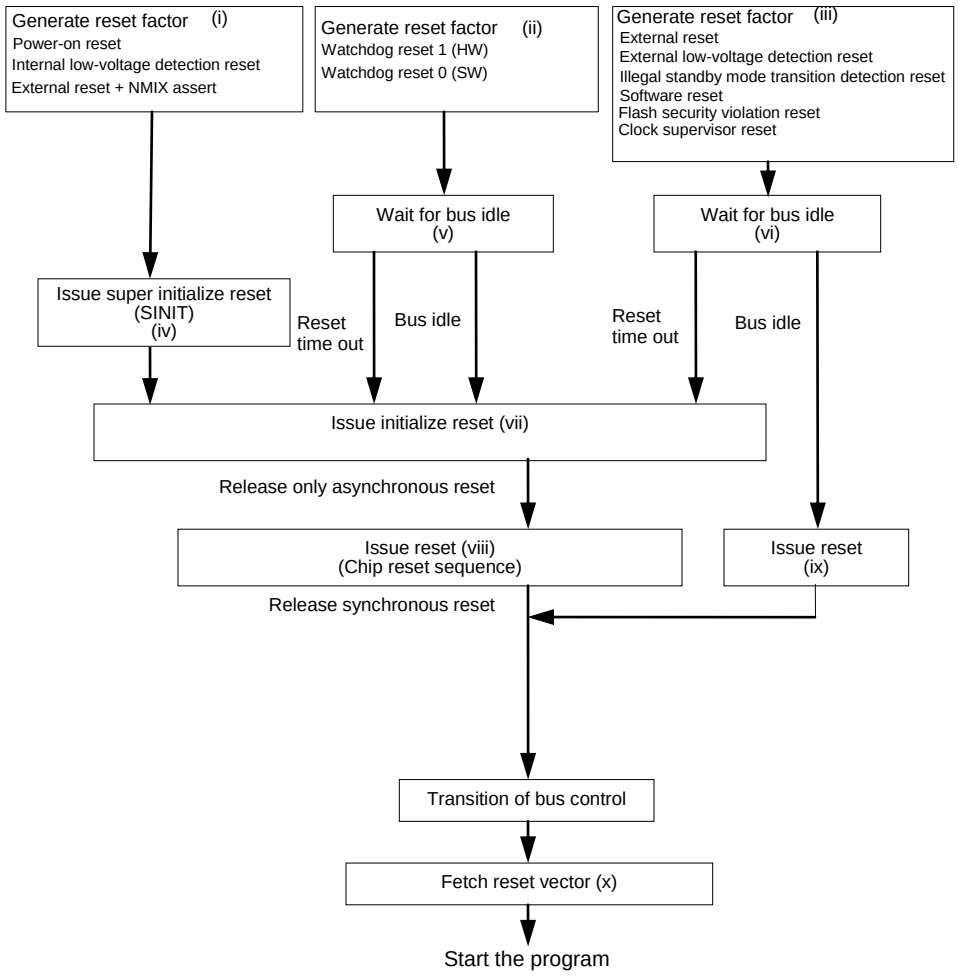


7.5.5 Reset Sequence

The reset sequence is shown.

This product transits from the initial state to start running the programs and hardware by disappearance of reset factors. A series of operations from this reset to the start of operation is called a reset sequence. This section explains the reset sequence.

Figure 7-8. Reset Sequence



Notes:

- If (i) occurs after (vii) or during (v) or (vi), the sequence restarts from (i).
  If (i) occurs after (B), the sequence restarts from (i).
  Refer to "Figure 6-1 Diagram of Device State Transitions" in Chapter of "Clock Reset State Transitions" for details.
- The main clock oscillation stabilization wait time is taken during (iv).
- The main clock oscillation stabilization wait time is taken during (vii), (viii), or (ix) if necessary (CMONR.MCRDY=0).
- Refer to Chapter of "FixedVector Function" for details on (x).
- At illegal standby mode transition detection reset, the status is already bus idle state after generating the reset factor, so the status moves to (ix)

7.5.5.1 Reset Cycle

The reset cycle is shown.

After the release of reset factors, the reset request is extended during the 4 x peripheral clock (PCLK) cycle. After that, a reset cycle will be maintained by the period of peripheral clock (PCLK) x 16 cycles for each reset level. Thus, the minimum number of issue cycles for each reset is 20 cycles. If it requires the main clock oscillation stabilization wait time, the cycle will be extended for the time required.

7.5.5.2 Reset Release

The reset release is shown.

Once a reset cycle has completed, each reset will be released and each hardware starts running. Right after the reset release, the mode control circuit functions as a bus master of on-chip bus.

7.5.5.3 Operating Mode Fix

The operating mode fix is shown.

The mode control circuit as a bus master will notify the operating mode, which was determined based on the mode setting value acquired, to each hardware component. Then, it will release the bus control of on-chip bus.

7.5.5.4 Transition of Bus Control

The transition of bus control is shown.

After the mode control circuit releases the bus control of on-chip bus, the CPU acquires the bus control and starts running bus operations by the CPU.

7.5.5.5 Reset Vector Fetch

The reset vector fetch is shown.

After the reset release, the CPU starts fetching the reset vector.

After CPU acquires the bus control, the CPU accesses the reset vector through on-chip bus and retrieves the acquired reset vector to the PC to start running programs.

7.5.5.6 Reset and Forced Break

The reset and forced break are shown.

If a forced break has occurred during the reset release, it accepts the forced break upon completion of the reset vector fetch. Thus, the PC value by the reset vector acquired will be saved at the emulator space side (stored at the E_BPCHR and E_BPCLR registers).

8. DMA Controller (DMAC)

This chapter explains the DMA controller (DMAC).

8.1 Overview

8.2 Features

8.3 Configuration

8.4 Registers

8.5 Operation

8.6 DMA Usage Examples

8.1 Overview

This section explains the overview of the DMA controller (DMAC).

DMAC is the module which performs the DMA (Direct Memory Access) transfer. DMA transfer controlled by this module enables the high speed transfer of variety of data without any interventions of a CPU, thus increases the system performance.

8.2 Features

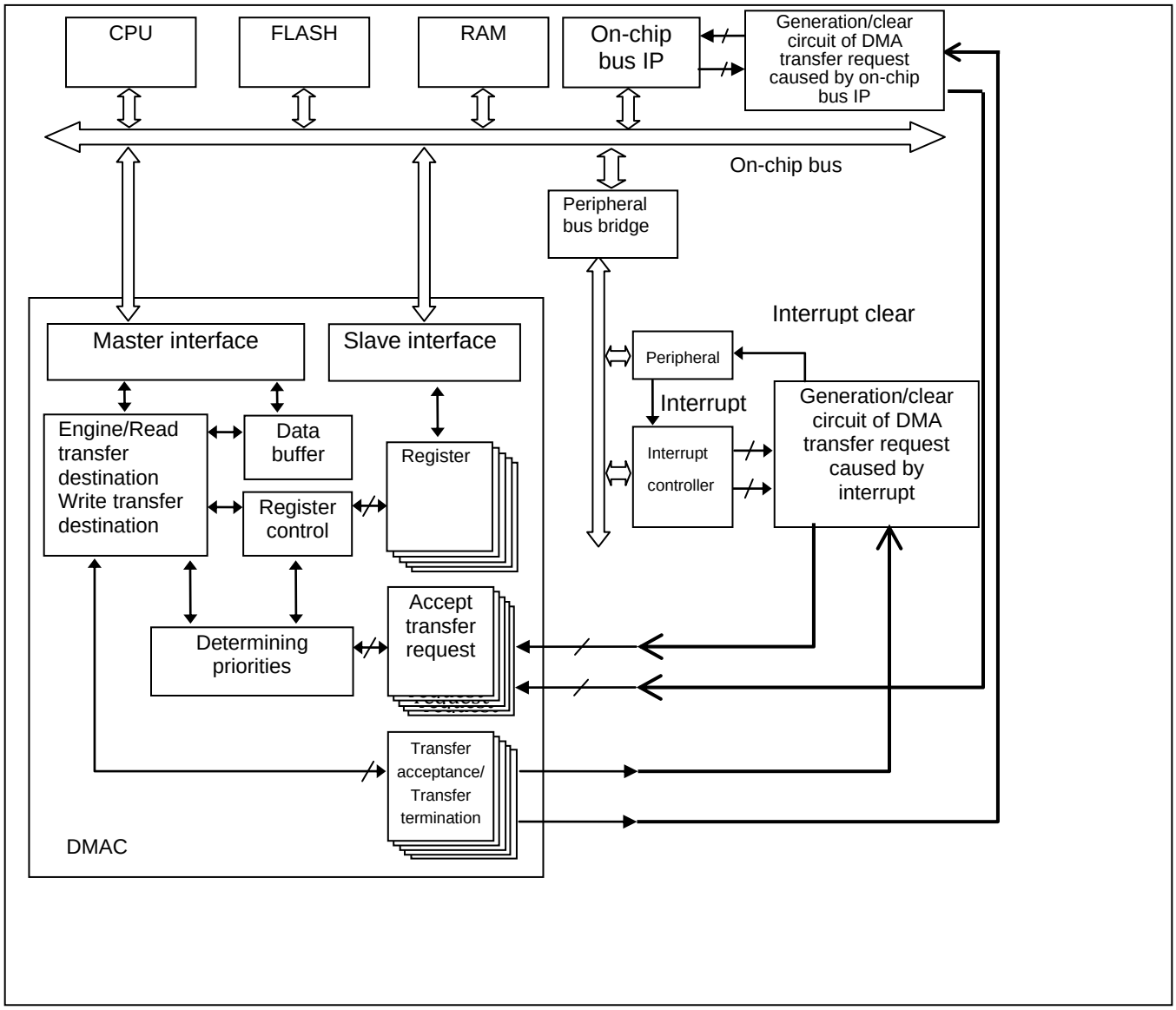
This section explains the features of the DMA controller (DMAC).

- Channels: 8 channels
- Address space: 32-bit address space (4 GB)
- Transfer mode: Block/burst transfer
- Address update: Increment/Decrement/Fixed (Address increment/decrement range: 1, 2, 4)
- Transfer size: 8-bits, 16-bits, 32-bits
- Block size: 1 to 16
- Transfer count: 1 to 65535
- Transfer request
  - ☐ Software transfer requests
  - ☐ Transfer requests by peripheral bus IP interrupt (for the transfer request by peripheral interrupt, you should select interrupt by channels. See "Chapter: Generation and Clearing of DMA Transfer Requests").
  - ☐ Transfer requests by on-chip bus IPs (A DMAC channel number corresponding to each on-chip bus IP cannot be selected. See "8.5.2 Table for On-chip Bus IPs and Corresponding DMAC Channels").
- Transfer stop request: Transfer stop request by interrupts
- Reload function: All channels can be specified for reload
  - ☐ Transfer source address reload
  - ☐ Transfer destination address reload
  - ☐ Transfer count reload
- Priority:
  - ☐ Fixed (ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7)
  - ☐ Or round robin
- Interrupt request: Normal completion interrupt requests, abnormal completion interrupt requests, and transfer suspend interrupt requests by transfer stop requests can be generated.

8.3 Configuration

This section explains the configuration of the DMA controller (DMAC).

Figure 8-1. Block Diagram



8.4 Registers

This section explains registers of the DMA controller (DMAC).

Table 8-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0C00	DCCR0				DMA channel control register 0
0x0C04	DCSR0		DTCR0		DMA channel status register 0 DMA transfer count register 0
0x0C08	DSAR0				DMA transfer source address register 0
0x0C0C	DDAR0				DMA transfer destination address register 0
0x0C10	DCCR1				DMA channel control register 1
0x0C14	DCSR1		DTCR1		DMA channel status register 1 DMA transfer count register 1
0x0C18	DSAR1				DMA transfer source address register 1
0x0C1C	DDAR1				DMA transfer destination address register 1
0x0C20	DCCR2				DMA channel control register 2
0x0C24	DCSR2		DTCR2		DMA channel status register 2 DMA transfer count register 2
0x0C28	DSAR2				DMA transfer source address register 2
0x0C2C	DDAR2				DMA transfer destination address register 2
0x0C30	DCCR3				DMA channel control register 3
0x0C34	DCSR3		DTCR3		DMA channel status register 3 DMA transfer count register 3
0x0C38	DSAR3				DMA transfer source address register 3
0x0C3C	DDAR3				DMA transfer destination address register 3
0x0C40	DCCR4				DMA channel control register 4
0x0C44	DCSR4		DTCR4		DMA channel status register 4 DMA transfer count register 4
0x0C48	DSAR4				DMA transfer source address register 4
0x0C4C	DDAR4				DMA transfer destination address register 4
0x0C50	DCCR5				DMA channel control register 5
0x0C54	DCSR5		DTCR5		DMA channel status register 5 DMA transfer count register 5
0x0C58	DSAR5				DMA transfer source address register 5
0x0C5C	DDAR5				DMA transfer destination address register 5
0x0C60	DCCR6				DMA channel control register 6

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0C64	DCSR6		DTCR6		DMA channel status register 6 DMA transfer count register 6
0x0C68	DSAR6				DMA transfer source address register 6
0x0C6C	DDAR6				DMA transfer destination address register 6
0x0C70	DCCR7				DMA channel control register 7
0x0C74	DCSR7		DTCR7		DMA channel status register 7 DMA transfer count register 7
0x0C78	DSAR7				DMA transfer source address register 7
0x0C7C	DDAR7				DMA transfer destination address register 7
0x0DF4	Reserved	Reserved	DNMIR	DILVR	DMA transfer suppression NMI flag register DMA transfer suppression interrupt level register
0x0DF8	DMACR				DMA control register
0x0DFC	Reserved				Reserved

8.4.1 DMA Control Register: DMACR (DMA Control Register)

This section explains the bit configuration of the DMA control register.

The DMA control register is a 32-bit register to control the entire DMAC (all channels). This register must be accessed as a 32-bit data.

DMACR: Address 0DF8$_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	DME	Reserved						
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	AT	Reserved						
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

[bit31] DME (DMA Enable): DMA operation enable

This bit controls the operation of the entire DMAC. When this bit is "0", a DMA transfer will not be performed even if operation of each channel is enabled. When this bit is "1", operations according to the settings for each channel are performed.

If "0" is written while a DMA transfer is in progress, the transfer is stopped in blocks specified in DCCRx.BLK.

DME	DMA Operation Enable
0	DMA operation disabled (initial value)
1	DMA operation enabled

[bit30 to bit16] Reserved

Always write "0" to these bits. The read value is "0".

[bit15] AT (Arbitration Type): Priority setting

This bit configures how to determine priority for each channel. If the priority is set to "fixed" (AT = 0), ascending order, ch.0 > ch.1 > ch.2 > ch.3, is taken. If the priority is set to "round robin" (AT = 1), DMAC makes the priority of the channel which started the transfer the lowest and raises the priority of following channels one by one. The decision on priority is made on each transfer of a block unit specified in DCCRx.BLK regardless of the priority setting.

AT	Priority Setting
0	Fixed (initial value)
1	Round robin

[bit14 to bit0] Reserved

Always write "0" to these bits. The read value is "0".

8.4.2 DMA Channel Control Register 0 to 7: DCCR0 to 7 (DMA Channel Control Register 0 to 7)

This section explains the bit configuration for DMA channel control register 0 to 7.

DMA channel control registers are 32-bit registers to control the operation of DMAC channels, which exists independently for each channel. This register must be accessed as a 32-bit data.

DCCR0 to 7: Address BASE + 0000_H (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	CE	Reserved				AIE	SIE	NIE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		RS[1:0]		Reserved		TM[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R/W	R/W	R0,W0	R0,W0	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ST	SAR	SAC[1:0]		DT	DAR	DAC[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TCR	Reserved	TS[1:0]		BLK[3:0]			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W

[bit31] CE (Channel Enable): Channel operation enable

This bit controls the operation of the channels. If the request source is set to "software", writing "1" to this bit starts a DMA transfer according to the configuration. In this case, the CE bit is automatically cleared when the transfer according to the transfer request completed.

If the request source is other than software, writing "1" to this bit makes channel operation enabled. After enabling operation, a DMA transfer starts when the corresponding transfer request is detected. In case of a request other than software, the CE bit will not be automatically cleared if transfer count reload (DCCRx.TCR) is specified. When transfer count reload is disabled, the CE bit will be cleared when all transfers are finished.

If "0" is written while the operation is going on regardless of the request source, stop transfer in blocks specified in DCCRx.BLK. When writing "1" again and detecting a new transfer request, the operation restarts.

CE	Channel Operation Enable
0	Disabled (initial value)
1	Enabled

[bit30 to bit27] Reserved

Always write "0" to these bits. The read value is "0".

[bit26] AIE (Abnormal completion Interrupt Enable): Abnormal completion interrupt enable

This bit controls the generation of interrupts when setting the prohibited values to the DMA channel control register (DCCR). The items not allowed to set to registers are listed below.

- Transfer mode: DCCRx.TM = 2'b10
- Transfer source address count: DCCRx.SAC = 2'b10
- Transfer destination address count: DCCRx.DAC = 2'b10
- Transfer size: DCCRx.TS = 2'b11
- Demand transfer mode by software request: DCCRx.RS = 2'b00 and DCCRx.TM = 2'b11

As for the interrupt factor, see the status register (DCSRx).

AIE	Abnormal Completion Interrupt Enable
0	Disabled (initial value)
1	Enabled

[bit25] SIE (Stop Interrupt Enable): Transfer suspend interrupt enabled by transfer stop requests

This bit controls the generation of interrupts when a DMA transfer is suspended by a transfer stop request from the transfer request source. As for the interrupt factor, see the status register (DCSRx).

SIE	Transfer Suspend Interrupt Enable
0	Disabled (initial value)
1	Enabled

[bit24] NIE (Normal Completion Interrupt Enable): Normal completion interrupt enable

This bit controls the generation of interrupts when completing DMA transfers successfully. After completing transfers as many times as set by transfer count (DTCRx.DTC) or when writing "1" to the corresponding channel's DCCRx.CE bit at the time the transfer count is "0", the operation will complete normally.
As for the interrupt factor, see the status register (DCSRx).

NIE	Normal Completion Interrupt Enable
0	Disabled (initial value)
1	Enabled

[bit23, bit22] Reserved

Always write "0" to these bits. The read value is "0".

[bit21, bit20] RS (Request Source): DMA transfer request source

These bits select the transfer request source for the channel.

Setting RS[1:0] = 2'b11 is prohibited because there will be no transfers requested by an on-chip bus IP on ch.1 to ch.3.

RS[1:0]	DMA Transfer Request Source
00	Software (initial value)
01	Interrupts
10	Reserved (setting is prohibited)
11	On-chip bus IP

[bit19, bit18] Reserved

Always write "0" to these bits. The read value is "0".

[bit17, bit16] TM (Transfer Mode): Transfer mode

These bits specify the DMA transfer mode.

TM[1:0]	Transfer Mode
00	Block transfer (initial value)
01	Burst transfer
10	Reserved (setting is prohibited)
11	Reserved (setting is prohibited)

[bit15] ST (Source Type): Transfer source type

The setting values are different depending on the combinations of DMA transfer request source (DCCRn.RS[1:0]), transfer source address (DSAR), and transfer destination address (DDAR). As for the setting, see “Setting the ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type)”.

ST	Transfer Source Type
0	See “Setting the ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type)”.
1	

[bit14] SAR (Source Address Reload): Transfer source address reload

This bit specifies the transfer source address register reload. When specifying a reload, the transfer source address register value is returned to the initial value at the end of the transfer. When disabling a reload, the transfer source address register will point to the next access address to the last address at the end of the transfer.

SAR	Transfer Source Address Reload Specified
0	Reload disabled (initial value)
1	Reload

[bit13, bit12] SAC (Source Address Count): Transfer source address count

These bits specify the address update once for each transfer of the transfer source address. The update values when specifying "increment/decrement" will be one of the values, 1, 2, 4 depending on the transfer size (DCCRx.TS).

SAC[1:0]	Transfer Source Address Count
00	Address increment (initial value)
01	Address decrement
10	Reserved (setting is prohibited)
11	Address fixed

[bit11] DT (Destination Type): Transfer destination type

The setting values are different depending on the combinations of DMA transfer request source (DCCR.RS[1:0]), transfer source address (DSAR), and transfer destination address (DDAR). See "Setting the ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type)".

DT	Transfer Destination Type
0	See "Setting the ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type)".
1	

[bit10] DAR (Destination Address Reload): Transfer destination address reload

This bit specifies the transfer destination address register reload. When specifying a reload, the transfer destination address register value is returned to the initial value at the end of the transfer.

When disabling a reload, the transfer destination address register will point to the next access address to the last address at the end of the transfer.

DAR	Transfer Destination Address Reload Specified
0	Reload disabled (initial value)
1	Reload

[bit9, bit8] DAC (Destination Address Count): Transfer destination address count

These bits specify the address update once for each transfer of the transfer destination address. The update values when specifying "increment/decrement" will be one of the values, 1, 2, 4 depending on the transfer size (DCCRx.TS).

DAC[1:0]	Transfer Destination Address Count
00	Address increment (initial value)
01	Address decrement
10	Reserved (setting is prohibited)
11	Address fixed

[bit7] TCR (Transfer Count Reload): Transfer count reload

This bit specifies the transfer count register reload.

When specifying a reload, the transfer count register value is returned to the initial value at the end of the transfer. If the transfer request source is set other than "software", DCCRx.CE bit will not be cleared at the end of the transfer and the operation will go into the transfer request wait state.

When disabling a reload, the transfer count register value at the end of the transfer will point to "0". In this case, DCCRx.CE bit will be cleared at the end of the transfer regardless of the transfer request source.

TCR	Transfer Count Reload
0	Reload disabled (initial value)
1	Reload

[bit6] Reserved

Always write "0" to this bit. The read value is "0".

[bit5, bit4] TS (Transfer Size): Transfer size

These bits specify the transfer size. DMA transfers will be performed once with the bit width specified here.

TS[1:0]	Transfer Size
00	8-bit: byte (initial value)
01	16-bit: half-word
10	32-bit: word
11	Reserved (setting is prohibited)

Set values to DSARx and DDARx registers so as not to cause a misalignment for the transfer size specified in these bits.

[bit3 to bit0] BLK (BlocK size): Block size

These bits specify the block size. 1 block transfer will be repeated for the number of blocks of the transfer size specified with DCCRx.TS bit.

BLK[3:0]	Transfer Count
0000	Once (initial value)
0001	Twice
0010	3 times
0011	4 times
0100	5 times
0101	6 times
0110	7 times
0111	8 times
1000	9 times
1001	10 times
1010	11 times
1011	12 times
1100	13 times
1101	14 times
1110	15 times
1111	16 times

8.4.3 DMA Channel Status Register 0 to 7: DCSR0 to 7 (DMA Channel Status Register 0 to 7)

This section explains the bit configuration for DMA channel status register 0 to 7.

These registers are 16-bit registers to indicate the status for each DMAC channel, which exist independently for each channel. These registers must be accessed as a 16-bit data.

DCSR0 to 7: Address BASE + 0004_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CA	Reserved						
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					AC	SP	NC
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W	R,W	R,W

[bit15] CA (Channel Active): Channel active

This bit indicates the channel operating state. Writing "1" to the corresponding DCCRx.CE bit for the channel makes it in the operating state. Completing transfers for as many times as set transfer count or writing "0" to DCCRx.CE makes the operation stop.

Writing to this bit is invalid.

CA	Channel Operating State
0	Stop state (initial value)
1	Channel operating

[bit14 to bit3] Reserved

Always write "0" to these bits. The read value is "0".

[bit2] AC (Abnormal Completion): Abnormal completion state

This bit indicates that a prohibited value has been set to the DMA channel control register (DCCR). The items not allowed to set to registers are listed below.

- Transfer mode: DCCRx.TM = 2'b10
- Transfer source address count: CCRx.SAC = 2'b10
- Transfer destination address count: DCCRx.DAC = 2'b10
- Transfer size: DCCRx.TS = 2'11
- Demand transfer mode by software request: DCCRx.RS = 2'b00 and DCCRx.TM = 2'b11

When having allowed the abnormal completion interrupt (DCCRx.AIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is invalid.

Make sure to clear this bit before enabling DMA operation. This bit will not be cleared automatically.

AC	Abnormal Completion State
0	Abnormal completion undetected (initial value)
1	Abnormal completion

[bit1] SP (StoP): Transfer suspend state by the transfer stop request

This bit indicates that a DMA transfer has been suspended by a transfer stop request from the transfer request source. When having allowed the transfer suspension interrupt (DCCRx.SIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is invalid.

Make sure to clear this bit before enabling DMA operation. This bit will not be cleared automatically.

SP	Transfer Suspend State
0	Transfer suspend undetected (initial value)
1	Transfer suspend

[bit0] NC (Normal Completion): Normal completion state

This bit indicates that DMA transfer has been completed successfully. After completing transfers as many times as set by transfer count or when writing "1" to the corresponding channel's "DCCRx.CE" bit at the time the transfer count is "0", the operation will complete normally. When having allowed the normal completion interrupt (DCCRx.NIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is invalid.

Make sure to clear this bit before enabling DMA operation. This bit will not be cleared automatically.

NC	Normal Completion State
0	Normal completion undetected (initial value)
1	Normal completion

8.4.4 DMA Transfer Count Register 0 to 7: DTCR0 to 7 (DMA Transfer Count Register 0 to 7)

This section explains the bit configuration for DMA transfer count register 0 to 7.

These registers are 16-bit registers to indicate the transfer count for each DMAC channel, which exist independently for each channel. These registers must be accessed as a 16-bit data.

DTCR0 to 7: Address BASE + 0006_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DTC[15:8]							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DTC[7:0]							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

[bit15 to bit0] DTC (DMA Transfer Count): DMA transfer count

These registers indicate the number of transfers. DMAC decreases a transfer count at the end of each block transfer and stops the transfer when the transfer count becomes "0". If "0" is set for transfer count, transfer will not be performed. Also, the dedicated reload register is provided. If DCCRx.TCR is "1", the value is returned to the initial value after data transfer.

8.4.5 DMA Transfer Source Register 0 to 7: DSAR0 to 7 (DMA Source Address Register 0 to 7)

This section explains the bit configuration for DMA transfer source register 0 to 7.

These registers are 32-bit registers to indicate the transfer source address of each DMAC channel, which exist independently for each channel. These registers must be accessed as a 32-bit data.

DSAR0 to 7: Address BASE + 0008_H (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	DSA[31:24]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	DSA[23:16]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DSA[15:8]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DSA[7:0]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

[bit31 to bit0] DDA[31:0] (DMA Destination Address): DMA transfer destination address

These registers indicate the transfer destination address. If an increment or a decrement is set by DCCRx.SAC, the address is updated according to the transfer size (DCCRx.TS). Also, the dedicated reload register is provided. If DCCRx.DAR is "1", the value is returned to the initial value after data transfer.

Set a value in these registers not to cause a misalignment against the transfer size to be set by DCCRx.TS.

If the DMA transfer request source has a peripheral interrupt (DCCRx.RS[1:0]=01), at least either the transfer source address (DSAR) or the transfer destination address (DDAR) must be within the address range of peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus. For details, see "Setting the ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type)".

8.4.6 DMA Transfer Destination Register 0 to 7: DDAR0 to 7 (DMA Destination Address Register 0 to 7)

This section explains the bit configuration for DMA transfer destination register 0 to 7.

These registers are 32-bit registers to indicate the transfer destination address of each DMAC channel, which exist independently for each channel. These registers must be accessed as a 32-bit data.

DDAR0 to 7: Address BASE + $000C_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	DDA[31:24]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	DDA[23:16]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DDA[15:8]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DDA[7:0]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

[bit31 to bit0] DDA[31:0] (DMA Destination Address): DMA transfer destination address

These registers indicate the transfer destination address. If an increment or a decrement is set by DCCRx.DAC, the address is updated according to the transfer size (DCCRx.TS). Also, the dedicated reload register is provided. If DCCRx.DAR is "1", the value is returned to the initial value after data transfer.

Set a value in these registers not to cause a misalignment against the transfer size to be set by DCCRx.TS.

If the DMA transfer request source has a peripheral interrupt (DCCRx.RS[1:0]=01), at least either the transfer source address (DSAR) or the transfer destination address (DDAR) must be within the address range of peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus. For details, see "Setting the ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type)".

8.4.7 DMA Transfer Suppression NMI Flag Register: DNMIR (DMA-halt by NMI Register)

This section explains the bit configuration for DMA transfer suppression flag register.

This register is 8-bit register to suppress DMA transfer by the user NMI. This register must be accessed as a 8-bit data.

DNMIR: Address $0DF6_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NMIH	Reserved						NMIHD
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

[bit7] NMIH (NMI Halt): DMA suppression flag (by NMI factor)

If the NMIHD bit is "0", this flag shows an occurrence of the user NMI request. The "H" level of NMI is detected, and this bit is set to "1". To restart DMA transfer, set this bit to "0".
Writing "1" to this bit is invalid.

NMIH	DMA Suppression Flag
0	DMA transfer is not suppressed. (initial value)
1	The DMA transfer has been stopped by user NMI.

[bit6 to bit1] Reserved

Always write "0" to these bits. The read value is "0".

[bit0] NMIHD (NMI Halt Disable): DMA suppression control (by NMI factor)

The control bit that stops DMA transfer if a user NMI request is generated.
If an NMI occurs when this bit is "0", the DMAC does not restart a new DMA transfer. During DMA transfer, the controller stops the current DMA transfer when a block unit transfer has completed.

NMIHD	DMA Suppression Control
0	Stop the DMA transfer by the user NMI. (initial value)
1	Do not stop the DMA transfer by the user NMI.

8.4.8 DMA Transfer Suppression Level Register: DILVR (DMA-halt by Interrupt Level Register)

This section explains the bit configuration for DMA transfer suppression level register.

This register is 8-bit register to control the DMA transfer suppression by peripheral interrupts. This register must be accessed as a 8-bit data.

DILVR: Address 0DF7$_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			LVL4	LVL[3:0]			
Initial Value	0	0	0	1	1	1	1	1
Attribute	R0,W0	R0,W0	R0,W0	R1,WX	R/W	R/W	R/W	R/W

[bit7 to bit5] Reserved

Always write "0" to these bits. The read value is "0".

[bit4 to bit0] LVL (Level): DMA suppression interrupt level

These bits set an interrupt level for suppression of DMA transfer. If a peripheral interrupt having an interrupt level higher than the one specified by this register occurs, the DMA transfer is suppressed. LVL4 is fixed to "1", but LVL[3:0] can be set to any level.

LVL[4:0]	DMA Suppression Control
11111	Suppress the DMA transfer when any peripheral interrupt request is issued. (initial value)
11110	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 1E$_H$ is issued.
11101	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 1D$_H$ is issued.
11100	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 1C$_H$ is issued.
11011	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 1B$_H$ is issued.
11010	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 1A$_H$ is issued.
11001	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 19$_H$ is issued.
11000	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 18$_H$ is issued.
10111	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 17$_H$ is issued.
10110	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 16$_H$ is issued.
10101	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 15$_H$ is issued.
10100	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 14$_H$ is issued.
10011	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 13$_H$ is issued.
10010	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 12$_H$ is issued.
10001	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 11$_H$ is issued.
10000	Do not suppress the DMA transfer when a peripheral interrupt request is issued.

8.5 Operation

This section explains the operation of the DMA controller (DMAC).

8.5.1 Configuration

8.5.2 Table for On-chip Bus IPs and Corresponding DMAC Channels

8.5.1 Configuration

This section explains the configuration for operation.

The following explains the setting items common to all channels and the items to be set separately for each channel.

8.5.1.1 Common Items for All Channels

The common Items for all channels is shown below.

This section explains the register settings for control of the entire DMAC.

DMA Operation Enable

The entire DMAC operation control can be set using the DMACR.DME.

- DMA operation disabled (DMACR.DME = 0)
- DMA operation enabled (DMACR.DME = 1)

Channel Priority

The decision method of the priority between channels can be set by DMACR.AT.

- Fixed priority (DMACR.AT = 0)
- Round robin (DMACR.AT = 1)

DMA Transfer Suppression Setting for Interrupt Occurrence

The DMA transfer suppression control during user NMI occurrence can be set by the DNMIR.NMIHD.

- Stops DMA transfer by the user NMI. (DNMIR.NMIHD = 0)
- Does not stop DMA transfer by the user NMI. (DNMIR.NMIHD = 1)

Also, an interrupt level, which precedes the DMA transfer when an interrupt occurs, can be set by DILVR.LVL. Allowed interrupt levels are 0x1F to 0x10.

8.5.1.2 Separate Items for Each Channel

The items set separately for each channel are shown.

The following explains both the items to be set separately for each channel and the register setup procedure.

Register Setup Procedure

The channel registers must be set in the following procedure. When you set the DCCRx.CE bit to "1", be sure to set the DTCRx to 1 or a higher value.

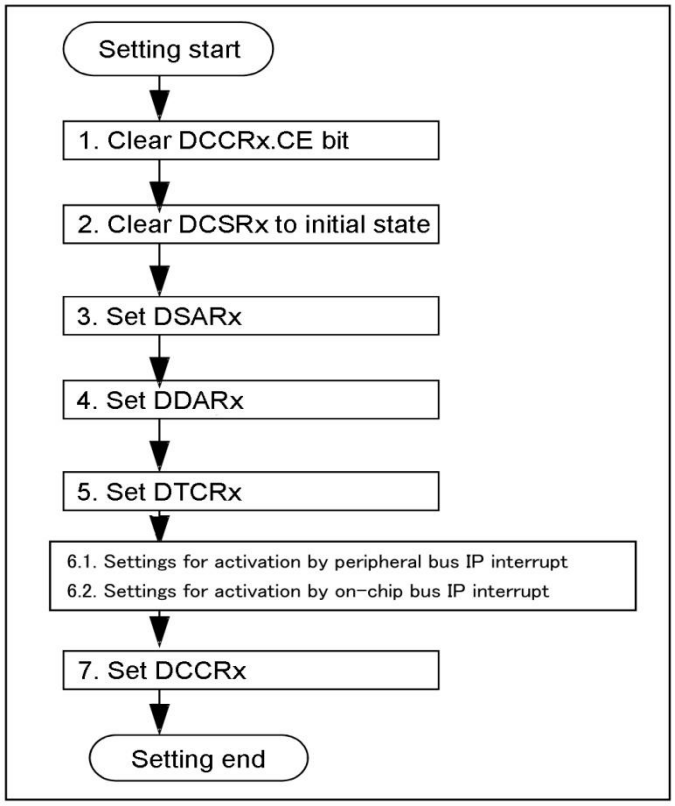
1. Clear the DCCRx.CE bit to disable the channel operation.
2. Clear each bit of DCSRx register to initialize the channel status flag.
3. Set the transfer source address (to be used when the transfer starts) in the DSARx register.
4. Set the transfer destination address (to be used when the transfer starts) in the DDARx register.
5. Set the transfer count in the DTCRx register. This count must be 1 or a larger value.

6.1 If transfer is started by a peripheral interrupt, the occurrence of each peripheral interrupt must be enabled and the ICSEL and IORR registers must be set. (See "Chapter: Generation and Clearing of DMA Transfer Requests" about the ICSEL and IORR registers.)

6.2 If transfer is started by an on-chip bus IP, the occurrence of each on-chip bus IP interrupt must be enabled and the ICSEL register must be set. (See "Chapter: Generation and Clearing of DMA Transfer Requests" about the ICSEL register.)

7. Set the DCCRx register. During this time, the channel operation is enabled when the DCCRx.CE bit is set.

Figure 8-2. Channel Register Setup Procedure



Transfer Source Address and the Transfer Destination Address Setting

Set the transfer source address (to be used when the transfer starts) using the DSARx.DSA.

Set the transfer destination address (to be used when the transfer starts) using the DDARx.DDA.

Align the transfer source and destination addresses based on the transfer size (DCCRx.TS), and ignore the lower 1 bit or lower 2 bits for 16-bit or 32-bit transfer size respectively.

Transfer Count Setting

Set the number of times of block transfer (repeated to the end of transfer) using the DTCRx.DTC. The transfer count can be 1 to 65535 times. The DMAC transfers data (1 block data), whose length in bytes is set by the transfer size and block size (see "Transfer Size and Block Size Setting") for the specified number of times.

Channel Operation Enable

Set the channel operation control using the DCCRx.CE.

- Disable the channel operation (DCCRx.CE = 0)
- Enable the channel operation (DCCRx.CE = 1)

When the software is selected at the transfer request source and when the DCCRx.CE bit is set, the channel operation is enabled and data transfer is started.

Interrupt Enable Setting

Enable an interrupt during abnormal completion, using the DCCRx.AIE.

- Disable an abnormal completion interrupt (DCCRx.AIE = 0)
- Enable an abnormal completion interrupt (DCCRx.AIE = 1)

Using the DCCRx.SIE, enable an interrupt to occur if data transfer is suspended by a transfer stop request.

- Disable a transfer suspend interrupt during detection of transfer stop request (DCCRx.SIE = 0)
- Enable a transfer suspend interrupt during detection of transfer stop request (DCCRx.SIE = 1)

Enable an interrupt during normal completion, using the DCCRx.NIE.

- Disable a normal completion interrupt (DCCRx.NIE = 0)
- Enable a normal completion interrupt (DCCRx.NIE = 1)

Transfer Request Source setting

Set the transfer request source to accept a transfer request using the DCCRx.RS.

- Request by software (DCCRx.RS = 00)
- Request by a peripheral bus IP interrupt (DCCRx.RS=01)
- Request by an on-chip bus interrupt (DCCRx.RS=11)

Transfer Mode Setting

Set the DMA transfer mode using the DCCRx.TM.

- Block transfer (DCCRx.TM = 00)
- Burst transfer (DCCRx.TM = 01)

Setting the ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type)

Set them by following the table definition below. The DMA transfer is not supported in combinations (5) and (9).

Table 8-2. ST Bit (Transfer Source Type) and DT Bit (Transfer Destination Type) Setting

	Combination of Transfer Request Source, Transfer Source, and Transfer Destination			DMA Transfer Support	ST and DT Bit Setting
	Transfer Request Source (DCCR.RS[1:0])	Transfer Source (DSAR)	Transfer Destination (DDAR)		
(1)	Request by Software (DCCR.RS=00)	Any Combination		Supported	ST=0, DT=0
(2)	Peripheral Bus Peripheral Interrupt (DCCR.RS=01)	●	□	Supported	ST=1, DT=0
(3)		□	●	Supported	ST=0, DT=1
(4)		●	●	Supported	ST=0, DT=1
(5)		□	□	Not supported	-
(6)	On-chip Bus Peripheral Interrupt(DCCR.RS=11)	○	■	Supported	ST=1, DT=0
(7)		■	○	Supported	ST=0, DT=1
(8)		○	○	Supported	ST=0, DT=1
(9)		■	■	Not supported	-

●: Address range of the peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus

□: Other address range

○: Address range of peripheral under control of on-chip bus

■: Other address range

If the ST and DT bits are set in a combination other than above, the interrupt may not be cleared automatically after occurrence of the DMA transfer request. Also, the interrupt of an on-chip bus peripheral not involved in the DMA transfer may be cleared.

	●Peripheral Bus Area	○On-chip Bus Area
Address Area	000000_H to $0002FF_H$ 000400_H to $0005FF_H$ $000E00_H$ to $001FFF_H$	000900_H to $000AFF_H$ 002000_H to $00EFFF_H$ 030000_H to $03FFFF_H$

Transfer Address Reload Setting

Using the DCCRx.SAR, set the reload control of transfer source address at the end of transfer.

- The transfer source address is not reloaded after the transfer. (The next access address after the last address is shown.) (DCCRx.SAR=0)
- The transfer source address is returned to the initial value at the end of transfer. (DCCRx.SAR=1)

Using the DCCRx.DAR, set the reload control of transfer destination address at the end of transfer.

- The transfer destination address is not reloaded after the transfer. (The next access address after the last address is shown.) (DCCRx.DAR=0)
- The transfer destination address is returned to the initial value at the end of transfer. (DCCRx.DAR=1)

Transfer Address Update Setting

Using the DCCRx.SAC, set the updating of transfer source address for DMA transfer.

- Address is increased. (DCCRx.SAC = 00)
- Address is decreased. (DCCRx.SAC = 01)
- Address is fixed. (DCCRx.SAC = 11)

Using the DCCRx.DAC, set the updating of transfer destination address for DMA transfer.

- Address is increased. (DCCRx.DAC = 00)
- Address is decreased. (DCCRx.DAC = 01)
- Address is fixed. (DCCRx.DAC = 11)

Transfer Count Reload Setting

Using the DCCRx.TCR, set the reload control of transfer count at the end of transfer.

- The transfer count is not reloaded after the transfer. (After the normal completion of transfer, the transfer count is set to 0.) (DCCRx.TCR=0)
- The transfer count is returned to the initial value at the end of transfer. (DCCRx.TCR=1)

Transfer Size and Block Size Setting

To set a transfer unit for DMA transfer (the byte count to be transferred as 1 block), set the transfer size and block size.

Using the DCCRx.TS, set the size of data to be sent by a single DMA transfer (8-bit/16-bit/32-bit).

- 8-bit (DCCRx.TS=00)
- 16-bit (DCCRx.TS=01)
- 32-bit (DCCRx.TS=10)

Using the DCCRx.BLK, set the DMA transfer count for 1-block data transfer. The block size can be 1 to 16 times. In the 1-block transfer, data having the bit width being set by the transfer size (DCCRx.TS), is transferred for the number of times being set by the block size.

8.5.1.3 Operations

This section explains DMAC operations.

This section explains the DMAC operations as follows.

1. Channel status check
2. Data transfer

Channel Status Check

Each DMAC channel status can be checked using the DCSRx register.

- When the channel operation is enabled (the channel is active), the DCSRx.CA bit is "1". When the channel is stopped, its status is shown as "0".
- If data transfer terminates abnormally, the DCSRx.AC bit is set to "1".
- If data transfer is suspended by the transfer stop request, the DCSRx.SP bit is set to "1".
- When data transfer terminates normally, the DCSRx.NC bit is set to "1".

Data writing to the DCSRx.CA bit is ignored.

The DCSRx.AC, DCSRx.SP, and DCSRx.NC bits must be cleared before the DMA transfer is allowed because these bits are not cleared automatically.

Data Transfer

The DMAC starts DMA transfer when the transfer source address and transfer destination address are set. By receiving a transfer source read instruction, this controller reads the data, having the bit width (8-bit/16-bit/32-bit) being set by DCCRx.TS, from the transfer source address, and temporarily stores it in the data buffer inside of the DMAC. By receiving a transfer destination write instruction, the controller writes the data temporarily stored in the DMAC into the transfer destination address.

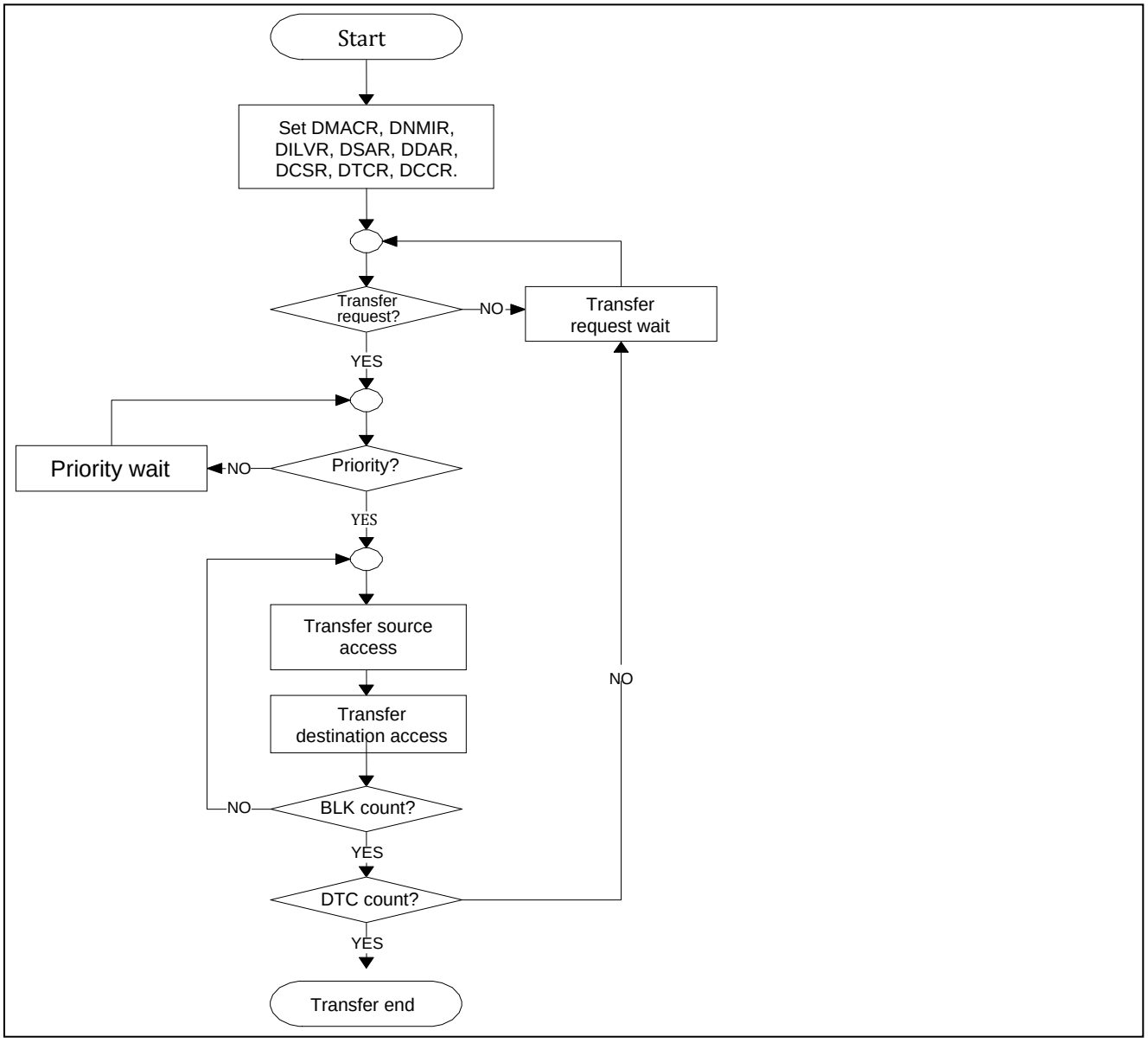
Transfer Mode

The transfer mode has block transfer mode or burst transfer mode.

- Block transfer mode

  1-time transfer request causes the 1 block transfer. When a transfer request is detected after the block transfer, the next 1-block transfer occurs. These operations are repeated until the end of data transfer. During 1-block data transfer, the data having the size specified by the DCCRx.TS bit is transferred for the number of times being set by the block size.

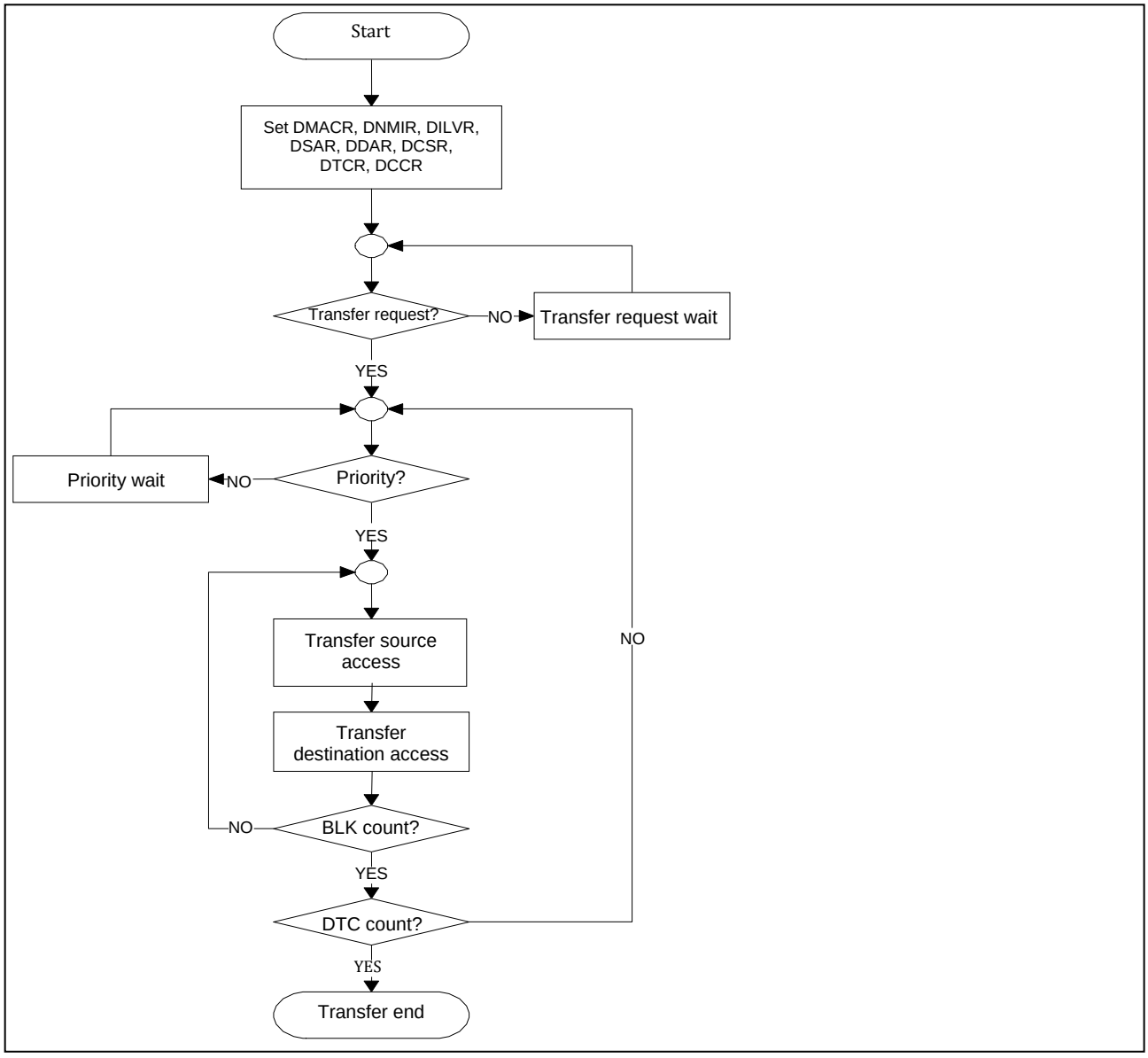
Figure 8-3. Each Transfer Mode (Block Transfer)



- Burst transfer mode

  1-time transfer request causes the continuous data transfer until the end of transfer. (Data having the size set by the DCCRx.TS bit is transferred continuously for the block size x number of transfers.)

Figure 8-4. Each Transfer Mode (Burst Transfer)



Transfer Request

The transfer request has a request by software or a request by interrupt. The following explains the relationship between the transfer request detection conditions and the transfer mode.

- Request by software

  If the DCCRx.CE bit is set to "1", a transfer request is detected. When the DMA operation is enabled (DMACR.DME=1), the priority is determined and the data transfer is started immediately. When the data transfer by the transfer request has terminated, the DCCRx.CE bit is cleared automatically.

- Request by interrupt

  If the channel operation is enabled (DCCRx.CE=1), a transfer request is awaited. If a peripheral interrupt, being set by the interrupt controller, has occurred, its transfer request is detected. When the DMA operation is enabled (DMACR.DME=1), the priority is determined and the data transfer is started immediately.

  When a transfer stop request is asserted from the peripheral, a transfer request is not detected.

  Also, an interrupt vector to be used for transfer request must be set for each channel. See "Chapter: Generation and Clearing of DMA Transfer Requests."

Note:

As the interrupt request from peripherals is detected by an edge, the transfer request cannot be detected even if the CE bit is reset from "0" to "1" while the interrupt request is enabled. The interrupt of the peripheral function should be enabled after the CE bit is set to "1".

Table 8-3. Relationship between Transfer Request Detection Conditions and Transfer Mode

	Block Transfer	Burst Transfer
Request by Software	Set the DCCRx.CE bit to "1".	Set the DCCRx.CE bit to "1".
Request by Interrupt	Edge detection	Edge detection

Also, the relationship between the detected transfer request and the DMACR.DME and DCCRx.CE bits is given on Table 8-4. If the DME bit or CE bit is cleared during transfer, the block transfer is stopped.

Table 8-4. Relationship between Transfer Requests and DME/CE Bits

		DME Bit	CE Bit
DME/CE Clear		The already detected transfer request is not cleared.	The already detected transfer request is cleared.
DME/CE Setting after the Transfer Interrupt	Block Transfer	When a new transfer request is detected, the data transfer is restarted based on the priority.	When a new transfer request is detected, the data transfer is restarted based on the priority.
	Burst Transfer	When the DME bit is set, the data transfer is restarted immediately based on the priority.	

Standby Recovery Request by DMA Transfer Request

If the MCU receives a transfer request in the standby mode, the DMAC requests the MCU to recover from the standby mode. If data transfer is enabled and if a transfer request is asserted by the transfer request source, a standby recovery is requested.

Channel Priority

If multiple transfer requests are issued, the DMAC starts data transfer on the channel having the highest priority. The channel priority can be fixed or can be set by round robin. The priority is determined for each block transfer or when data transfer ends.

- Fixed priority (DMACR.AT = 0)

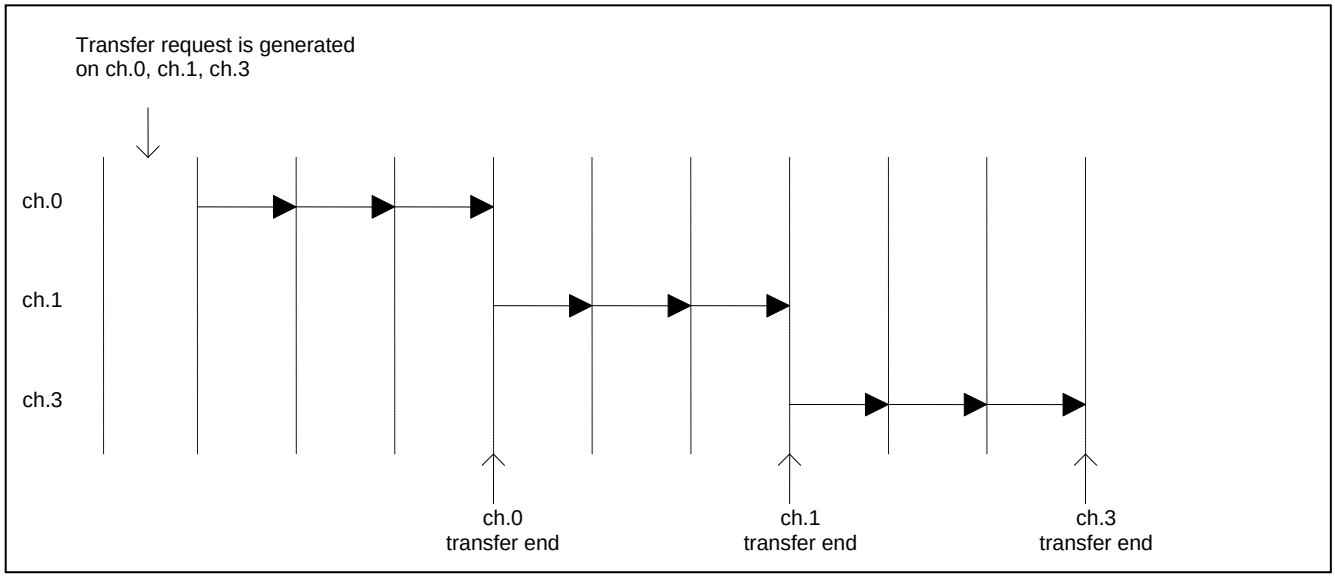
  The channel priority is fixed in the sequence of "ch.0 > ch.1 > ch.2 > ch.3". The following gives an example.

  Example 1: If transfer requests are issued on ch.0, ch.1 and ch.3 simultaneously, data transfer starts from ch. 0. When data transfer ends on ch.0, the next data transfer starts on ch.1. After data transfer on ch.1, the next data transfer starts on ch.3. The following gives transfer examples. Dotted lines in the figure show the block delimiters.

  Transfer request: Requests are issued for ch.0, ch.1 and ch.3 simultaneously.

  Setting: Ch.0, ch.1 and ch.3 are set to the burst transfer mode, and a data transfer count of 3.

Figure 8-5. Data Transfer Example 1 If Channel Priority Is Fixed

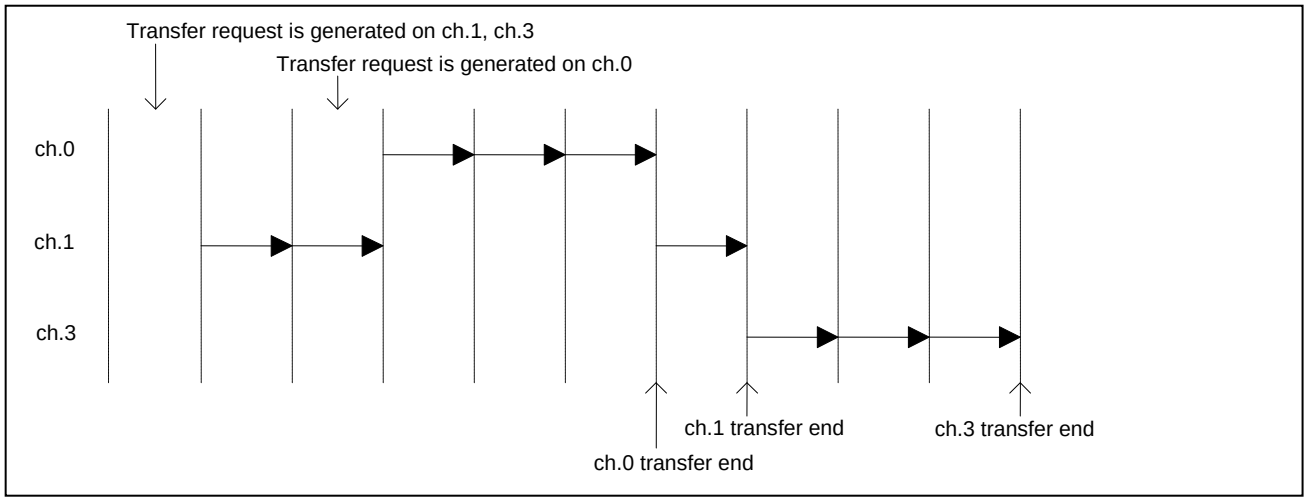


Example 2: If transfer requests are issued simultaneously for ch.1 and ch.3 and if a transfer request on ch.0 is issued during data transfer on ch.1, the data transfer on ch.1 is temporarily stopped and data transfer on ch.0 is started. During this time, the channel transition occurs in units of blocks. When the requested data transfer ends on ch.0, the data transfer is started on ch.1. Dotted lines in the figure show the block delimiters.

Transfer request: Requests are issued for ch.1 and ch.3 simultaneously. When data is transferred on ch.1, another request for transfer on ch.0 is issued.

Setting: Ch.0, ch.1 and ch.3 are set to the burst transfer mode, and data transfer count of 3.

Figure 8-6. Data Transfer Example 2 If Channel Priority Is Fixed



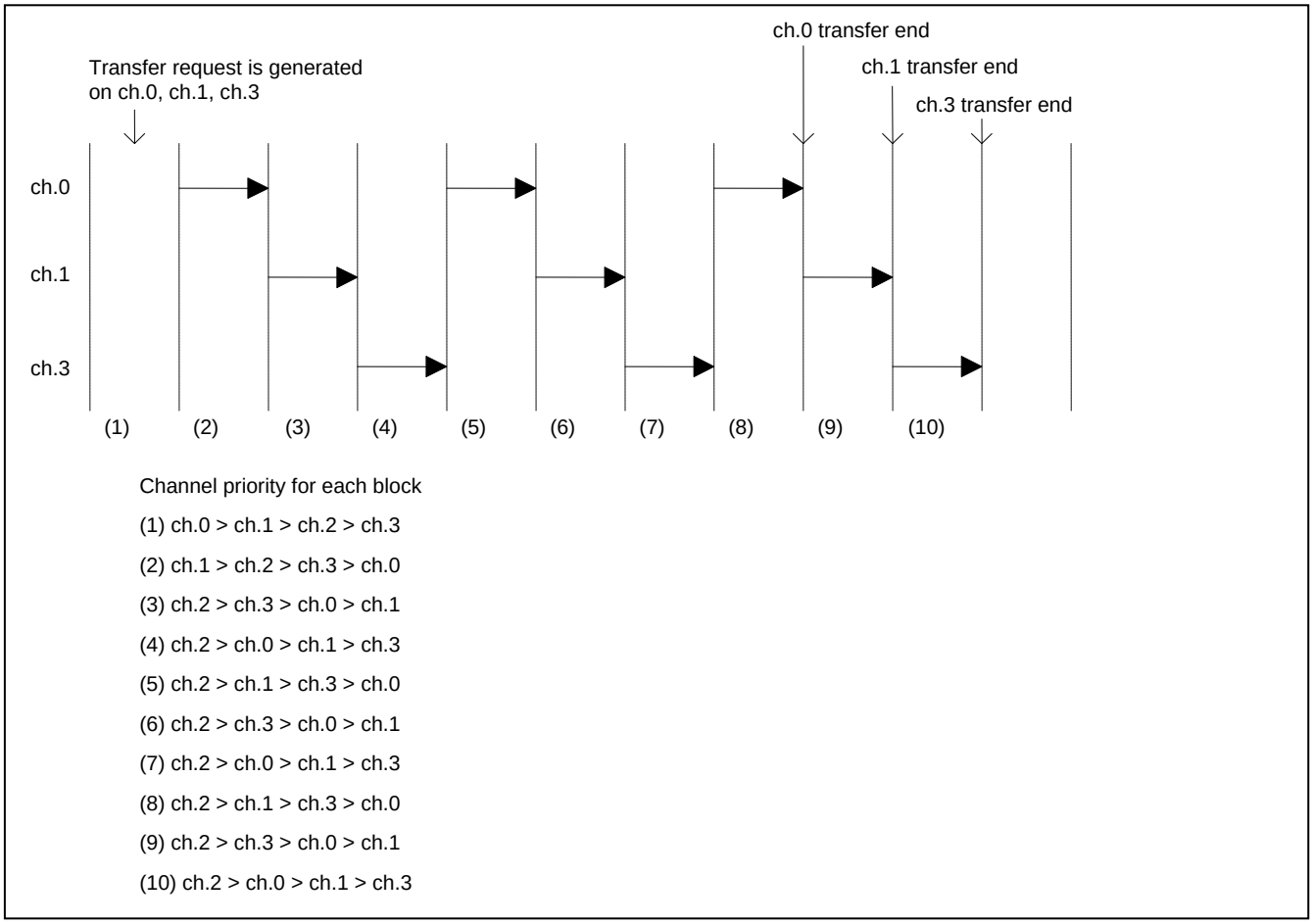
- Round robin (DMACR.AT = 1)

  When data transfer is started on a channel, its priority is set to the lowest level. A channel priority below this level is raised by one level. In the round robin, data transfer starts on a channel having the highest priority when a transfer request is issued. The priority of the channel where data transfer has started is dropped to the lowest level. The priority is determined for each of block data transfer, and data transfer is started on the channel having the highest priority. The following gives a transfer example. Dotted lines in the figure show the block delimiters.

  Example: Transfer request: Requests are issued for ch.0, ch.1 and ch.3 simultaneously.

  Setting: Ch.0, ch.1 and ch.3 are set to the burst transfer mode, and data transfer count of 3.

Figure 8-7. Data Transfer Example If Channel Priority Is Set by Round Robin



Updating of Transfer Address

The transfer source address and transfer destination address are updated each time data which size has been set by the DCCRx.TS is transferred. The address updating can be increasing, decreasing, or fixed. When increasing or decreasing, its address amount is determined by the transfer size (DCCRx.TS). If fixed, the address value does not change. Table 8-5 shows the address increasing or decreasing width during address updating. If an overflow occurs due to address updating, the relevant bit is discarded.

Table 8-5. Updating of Transfer Source Address and Transfer Destination Address

Address Setting		Transfer Size (TS)	Address Updating for Each Data Transfer	
Transfer Source (SAC)	Transfer Destination (DAC)		Transfer Source (DSA)	Transfer Destination (DDA)
Increments ("00")	Increments ("00")	8-bit ("00")	Increments by 1	Increments by 1
		16-bit ("01")	Increments by 2	Increments by 2
		32-bit ("10")	Increments by 4	Increments by 4
	Decrements ("01")	8-bit ("00")	Increments by 1	Decrements by 1
		16-bit ("01")	Increments by 2	Decrements by 2
		32-bit ("10")	Increments by 4	Decrements by 4
	Fixed ("11")	8-bit ("00")	Increments by 1	Not updated
		16-bit ("01")	Increments by 2	
		32-bit ("10")	Increments by 4	
Decrements ("01")	Increments ("00")	8-bit ("00")	Decrements by 1	Increments by 1
		16-bit ("01")	Decrements by 2	Increments by 2
		32-bit ("10")	Decrements by 4	Increments by 4
	Decrements ("01")	8-bit ("00")	Decrements by 1	Decrements by 1
		16-bit ("01")	Decrements by 2	Decrements by 2
		32-bit ("10")	Decrements by 4	Decrements by 4
	Fixed ("11")	8-bit ("00")	Decrements by 1	Not updated
		16-bit ("01")	Decrements by 2	
		32-bit ("10")	Decrements by 4	
Fixed ("11")	Increments ("00")	8-bit ("00")	Not updated	Increments by 1
		16-bit ("01")		Increments by 2
		32-bit ("10")		Increments by 4
	Decrements ("01")	8-bit ("00")		Decrements by 1
		16-bit ("01")		Decrements by 2
		32-bit ("10")		Decrements by 4
	Fixed ("11")	8-bit ("00")		Not updated
		16-bit ("01")		
		32-bit ("10")		

Reloading of Transfer Address

The DMAC can reload the transfer address after the specified number of data transfer has completed.

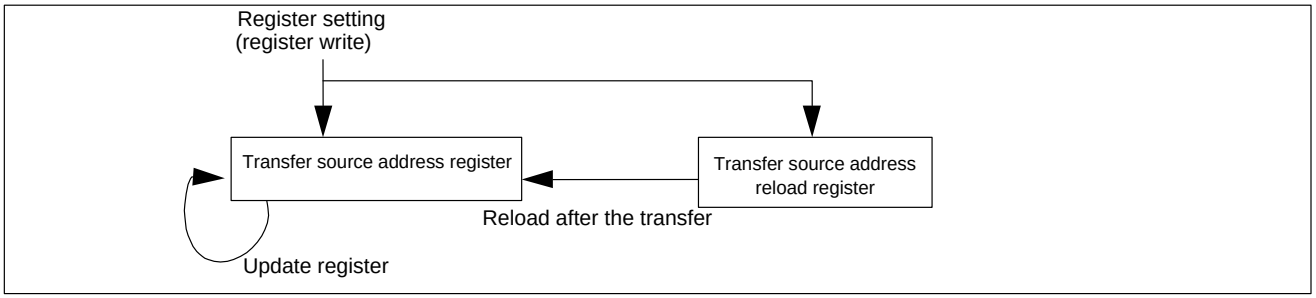
- Reloading of transfer source address

  If the reloading of transfer source address has been set, the DSARx.DSA bit is returned to the initial value after the data transfer.

  If the reloading of transfer source address is disabled, the DSARx.DSA bit indicates the next access address of the last address after the current data transfer.

  If the specified number of transfers is suspended or abnormally terminated, the DSARx.DSA bit indicates the next access address (after the terminated address) regardless of the reload setting of the transfer source address.

Figure 8-8. Reloading of Transfer Source Address Register



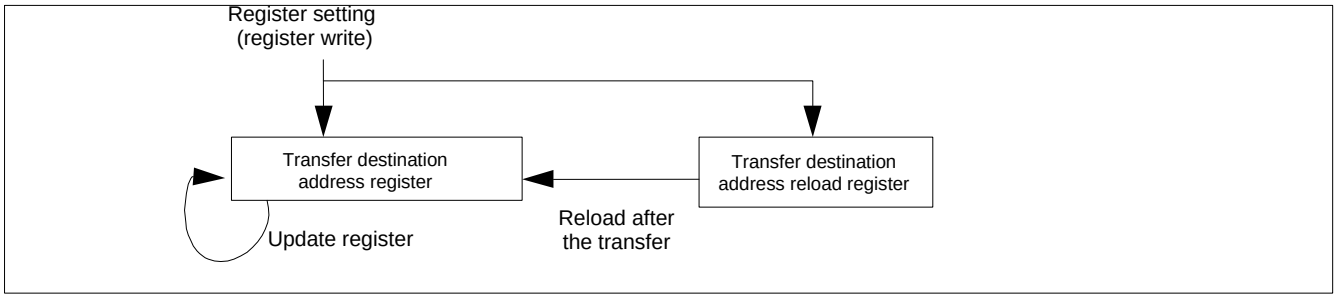
- Reloading of transfer destination address register

  If the reloading of the transfer destination address has been set, the DDARx.DDA bit is returned to the initial value after the data transfer.

  If the reloading of the transfer destination address is disabled, the DDARx.DDA bit indicates the next access address of the last address after the current data transfer.

  If the specified number of transfers is suspended or abnormally terminated, the DDARx.DDA bit indicates the next access address (after the terminated address) regardless of the reload setting of the transfer destination address.

Figure 8-9. Reloading of Transfer Destination Address Register



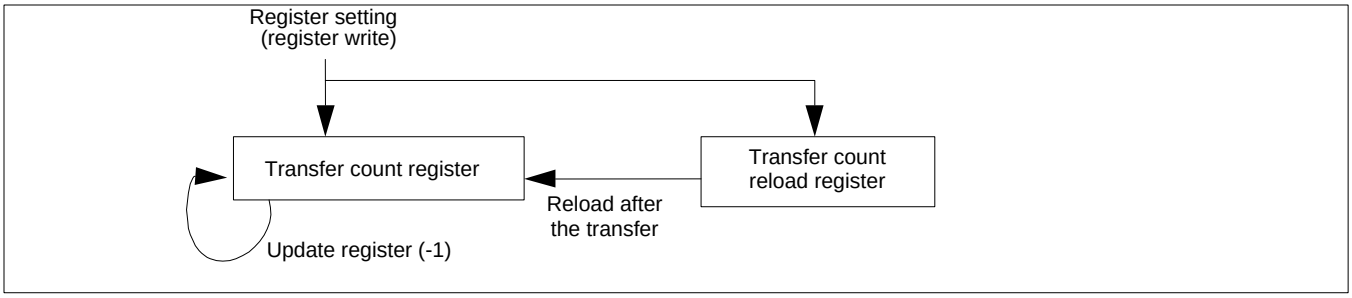
Reloading of Transfer Count

If the reloading of the transfer count has been set, the DTCRx.DTC bit is returned to the initial value after the data transfer.

If reloading of the transfer count is disabled, the DTCRx.DTC bit is set to "0" after the data transfer.

If the specified number of transfers is suspended or abnormally terminated, the DTCRx.DTC bit indicates the remaining transfer count regardless of the reload setting of the transfer count.

Figure 8-10. Reloading of Transfer Count Register



The DCCRx.CE bit status varies after the data transfer, depending on the reload setting of the transfer count. The following explains the relation between the transfer count reload setting and the transfer request source.

Table 8-6. DCCRx.CE Bit at the End of Transfer

	Software Request	Non-software Request
If the Reloading of Transfer Count Is Set	The DCCRx.CE bit is cleared	The DCCRx.CE bit is not cleared
If the Reloading of Transfer Count Is Disabled	The DCCRx.CE bit is cleared	The DCCRx.CE bit is cleared

Transfer Suspension

The DMAC suspends the DMA transfer due to the following causes.

- A suspension as the DMACR.DME bit is cleared
- A suspension as the DCCRx.CE bit is cleared
- A suspension caused by the transfer stop request by the transfer request source peripheral

Data transfer is suspended in units of blocks. If data transfer is suspended, the next transfer is not started. Data transfer is stopped. The settings to restart data transfer vary depending on the suspension cause.

- A suspension as the DCCRx.CE bit is cleared

  If the DMACR.DME bit is cleared, all channels are stopped from operating. After a block of data has been transferred on the current channel, the data transfer is suspended. To restart data transfer, set the DMACR.DME bit.

- A suspension as the DCCRx.CE bit is cleared

  If the DCCRx.CE bit is cleared, the channel is stopped from operating. After a block of data has been transferred, the data transfer is suspended. Also, as the DCCRx.CE bit is cleared, the already detected transfer request is cleared. To restart data transfer, set the DCCRx.CE bit for the stopped channel and issue a new transfer request.

- A transfer stop request from the transfer request source peripheral

  The following peripherals can issue a transfer stop request under certain conditions.

  (A) Multi-function serial interface
  If a PE, FRE, or ORE flag is set

If a transfer stop request is issued, the transfer is suspended after one block of the current data has been transferred. If the data transfer is suspended, the following occur.

- ☐ The SP bit of DMA channel status registers (DCSR0 to DCSR7) is set to "1".
- ☐ The CE bit of DMA channel control registers (DCCR0 to DCCR7) is set to "0".
- ☐ The already detected transfer request is cleared.

While a transfer stop request being issued, a new transfer request is rejected. Restart the DMA transfer in the following procedure.

1. Clear the flag described in paragraph (A) to make the transfer stop request invalid.
2. Set the SP bit of DMA channel status registers (DCSR0 to DCSR7) of the corresponding channel to "0".
3. Set the CE bit of DMA channel control registers (DCCR0 to DCCR7) to "1".
4. Issue a new transfer request.

Table 8-7. Settings to Restart the Suspended Data Transfer

	DME Clear	CE Clear	If a Transfer Stop Request from Transfer Request Source Peripheral Is Detected
Setting to Restart Transfer	(1) Set the DME bit	(1) Set the CE bit (2) Issue a transfer request	(1) The transfer request is negated (2) The SP bit is cleared (3) The CE bit is set (4) Issue a transfer request

Transfer Termination

Data transfer can terminate normally or abnormally.

- Normal termination

  The transfer terminates normally at the time when the transfers for the number of times set by the transfer count (DTCRx.DTC) end. When terminated normally, the DCSRx.NC bit of the corresponding channel is set. Also, the DCCRx.CE bit is cleared and data transfer is stopped. However, if the reloading of the transfer count has been set by non-software transfer request source, the DCCRx.CE bit of the channel is not cleared.
  If writing "1" to the corresponding channel's DCCRx.CE bit at the time the transfer count (DTCRx.DTC) is "0", the DCSRx.NC bit is set in the similar way as for the normal termination. Before setting the DCCRx.CE bit to "1", be sure to set the DTCRx.DTC bit to "1" or a larger value.

- Abnormal termination

  If an inhibited value is set in the register, data transfer terminates abnormally. When terminated abnormally, the DCSRx.AC bit of the corresponding channel is set. Also, the DCCRx.CE bit is cleared and data transfer is stopped.

The items not allowed to set to registers are listed below.

- ☐ Transfer mode: DCCRx.TM = 10
- ☐ Transfer source address count: DCCRx.SAC = 10
- ☐ Transfer destination address count: DCCRx.DAC = 10
- ☐ Transfer size: DCCRx.TS = 11
- ☐ Demand transfer mode by software request: DCCRx.RS = 00 and DCCRx.TM = 11

Interrupt Request

The DMAC can issue an interrupt request at normal termination of data transfer, at abnormal termination of data transfer, or at transfer suspension by a transfer stop request. When issuing an interrupt request, set the interrupt controller as well. Use the DMA channel status register (DCSRx) to check the interrupt request factor or to clear the interrupt request.

- Interrupt request at normal termination
  If the normal termination interrupt of a channel is enabled (DCCRx.NIE=1), the DMAC issues the interrupt request at the normal termination. However, the DCSRx.NC bit of the corresponding channel must be set regardless of the normal termination interrupt setting (DCCRx.NIE).
  Clear the interrupt request by clearing the DCSRx.NC bit of the corresponding channel.

- Interrupt request at abnormal termination
  If the abnormal termination interrupt of a channel is enabled (DCCRx.AIE=1), the DMAC issues the interrupt request at the abnormal termination. However, the DCSRx.AC bit of the corresponding channel is set regardless of the abnormal termination interrupt (DCCRx.AIE) setting.
  Clear the interrupt request by clearing the DCSRx.AC bit of the corresponding channel.

- A transfer suspension interrupt request by a transfer stop request
  If the transfer suspension interrupt of a channel is enabled (DCCRx.SIE=1), the DMAC issues the interrupt request if data transfer is suspended by a transfer stop request. However, the DCSRx.SP bit of the corresponding channel is set regardless of the transfer suspension interrupt (DCCRx.SIE) settings.
  Clear the interrupt request by clearing the DCSRx.SP bit of the corresponding channel.

- DMA transfer suppressing

The DMA transfer is suppressed due to the following causes.

- ☐ A DMA transfer suppress request from DSU/OCD (for debugging)
- ☐ NMI
- ☐ Peripheral interrupt

The DMA transfer is suppressed in units of blocks. If data transfer is suppressed, new data transfer does not start.Data transfer is stopped. The settings to restart data transfer vary depending on the DMA transfer suppress causes.

- DMA transfer suppressing request from DSU/OCD (for debugging)
  When the DMA transfer suppressing request by DSU/OCD is asserted, a new transfer does not start and a current transfer stops with the block unit. The acknowledge is not returned to the DMA transfer suppressing from DSU/OCD.
- DMA transfer suppressing by NMI
  If the NMIHD bit is set to "0", DMAC sets NMIH flag when user NMI occurs and suppresses DMA transfer after the current block has been transferred.
  Write "0" in the NMIH flag when you restart transfer.
- DMA transfer suppressing by peripheral interrupt
  If an interrupt having the level higher than the one specified in the DILVR register occurs, the DMA transfer is suppressed after the current block has been transferred.
  When the interrupt request is cleared and the interrupt level drops to LVL[4:0] or lower level, the DMA transfer restarts.

Table 8-8. LVL[4:0] Settings to Suppress DMA Transfer

LVL[4:0]	DMA Suppress Control
11111	Suppress the DMA transfer when any peripheral interrupt request is issued. (initial value)
11110	Suppress the DMA transfer when a peripheral interrupt request having a level higher than $1E_H$ is issued.
11101	Suppress the DMA transfer when a peripheral interrupt request having a level higher than $1D_H$ is issued.
11100	Suppress the DMA transfer when a peripheral interrupt request having a level higher than $1C_H$ is issued.
11011	Suppress the DMA transfer when a peripheral interrupt request having a level higher than $1B_H$ is issued.
11010	Suppress the DMA transfer when a peripheral interrupt request having a level higher than $1A_H$ is issued.
11001	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 19_H is issued.
11000	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 18_H is issued.
10111	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 17_H is issued.
10110	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 16_H is issued.
10101	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 15_H is issued.
10100	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 14_H is issued.
10011	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 13_H is issued.
10010	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 12_H is issued.
10001	Suppress the DMA transfer when a peripheral interrupt request having a level higher than 11_H is issued.
10000	Do not suppress the DMA transfer when a peripheral interrupt request is issued.

8.5.2 Table for On-chip Bus IPs and Corresponding DMAC Channels

The following on-chip bus IP is assigned to each DMAC channel.

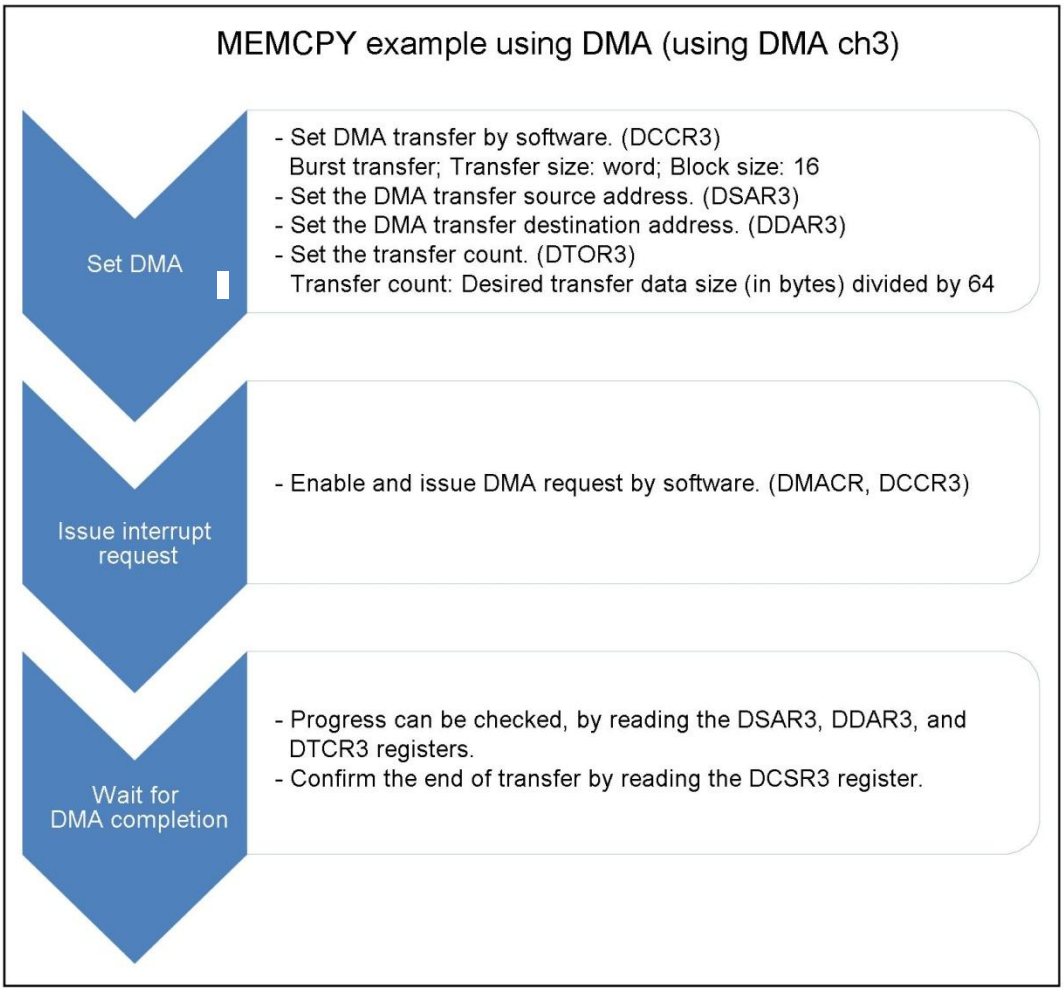
Channel	On-chip Bus IP
0	Transfer request by A/D converter 8/9/10/11 interrupt
1	No corresponding on-chip bus IP
2	No corresponding on-chip bus IP
3	No corresponding on-chip bus IP
4	PWC0 capture data upper limit interrupt 00/10/20/30 PWC0 capture data lower limit interrupt 00/10/20/30 PWC0 data buffer interrupt 00/10/20/30 PWC0 buffer overrun interrupt 00/10/20/30 Transfer requests by above interrupts
5	PWC1 capture data upper limit interrupt 01/11/21/31 PWC1 capture data lower limit interrupt 01/11/21/31 PWC1 data buffer interrupt 01/11/21/31 PWC1 buffer overrun interrupt 01/11/21/31 Transfer requests by above interrupts
6	PWC0 0 detection interrupt 00/10 Compare clear interrupt 00/10 Transfer requests by above interrupts
7	PWC1 0 detection interrupt 01/11 Compare clear interrupt 01/11 Transfer requests by above interrupts

8.6 DMA Usage Examples

This section explains DMA controller (DMAC) DMA usage examples.

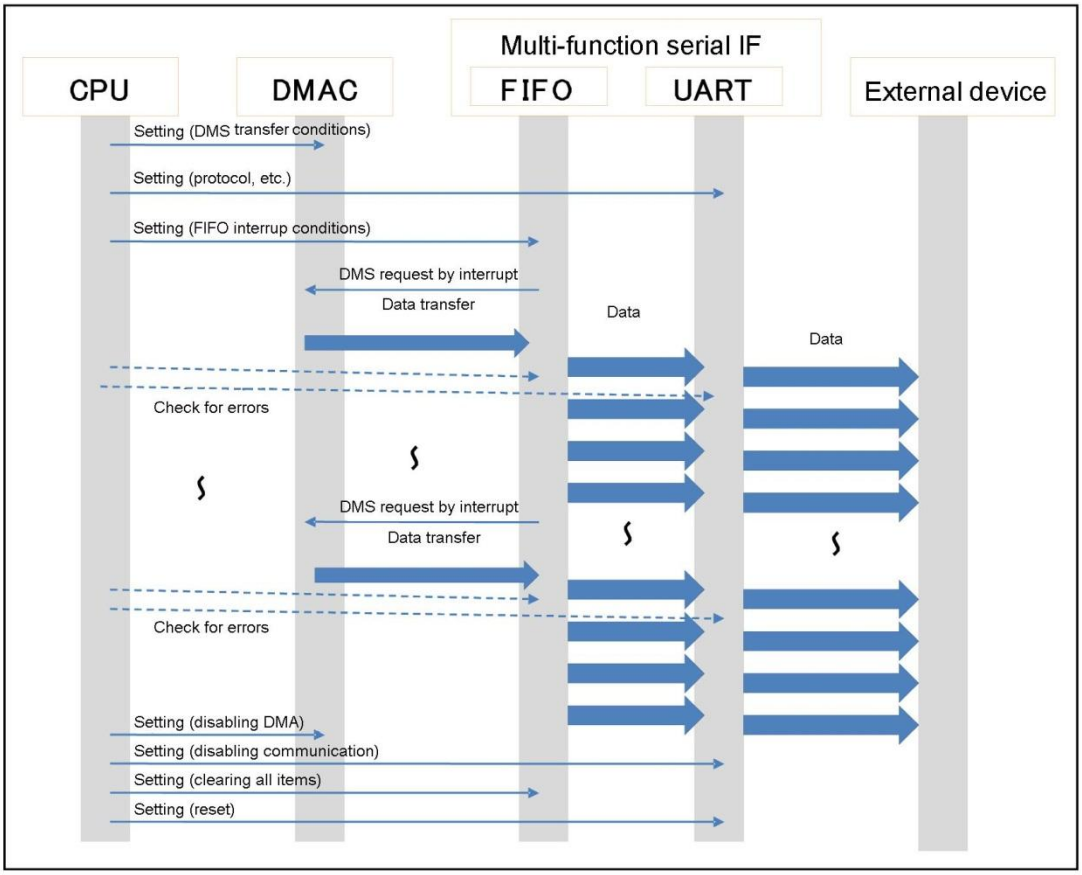
The following gives an example of memcpy instruction in every 64-byte data using the DMA. This is the simplest DMA transfer example.

Figure 8-11. Memcpy Example Using the DMA (ch.3 Is Used)



This is a communication example via the multi-function serial interface that uses the DMA. In this example, an interrupt of the multi-function serial interface is occupied by the DMA transfer request. Therefore, the CPU polls the status registers to check for an error occurrence.

Figure 8-12. Communication Example via the Multi-function Serial Interface That Uses DMA



9. Generation and Clearing of DMA Transfer Requests

This chapter explains the generation and clearing of DMA transfer requests.

9.1 Overview
9.2 Features
9.3 Configuration
9.4 Registers
9.5 Operation

9.1 Overview

This section explains the overview of the generation and clearing of DMA transfer requests.

This product can activate DMA transfer using interrupt requests from peripheral functions. Registers used to select interrupt requests that activate DMA transfer are provided for each DMA controller (DMAC) channel. If multiple interrupt requests are assigned to one interrupt vector number, it is also necessary to specify what interrupt request flag is to be cleared by the DMA controller (DMAC).

DMA controller (DMAC) registers allow DMA transfer request generation factors (transfer request sources) to be set on interrupt requests from peripheral functions. The interrupt requests to be used can be selected by specifying the value corresponding to the interrupt vector number.

9.2 Features

This section explains features of the generation and clearing of DMA transfer requests.

9.2.1 Transfer Request Generation Setting

9.2.2 Interrupt Clearing Setting

9.2.1 Transfer Request Generation Setting

The transfer request generation setting is shown.

For each 8-channel DMA transfer request, you need to specify what interrupt from interrupt vector numbers 0x10 (16 in decimal notation) to 0x3F (63 in decimal notation) is used to generate the DMA transfer request.

9.2.2 Interrupt Clearing Setting

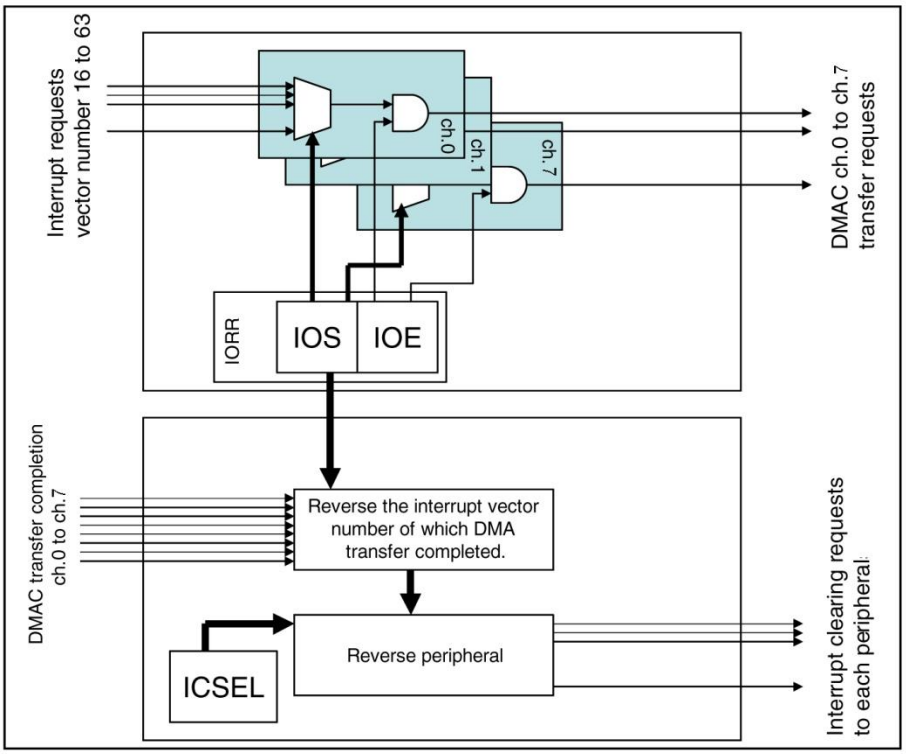
The interrupt clearing setting is shown.

After the DMA transfer ends, the interrupt source peripheral that has issued the interrupt request to be cleared is identified if the transfer request source is a vector number to which multiple interrupt source peripherals belong.

9.3 Configuration

This section explains the configuration of the generation and clearing of DMA transfer requests.

Figure 9-1. Block Diagram



9.4 Registers

This section explains registers of the generation and clearing of DMA transfer requests.

Table 9-1. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0400	ICSEL0	Reserved	ICSEL2	ICSEL3	DMA clear request register 0 (for vector number #16) DMA clear request register 2 (for vector number #18) DMA clear request register 3 (for vector number #19)
0x0404	Reserved	ICSEL5	Reserved	Reserved	DMA clear request register 5 (for vector number #39)
0x0408	Reserved	Reserved	Reserved	ICSEL11	DMA clear request register 11 (for vector number #46)
0x040C	Reserved	ICSEL13	ICSEL14	ICSEL15	DMA clear request register 13 (for vector number #52) DMA clear request register 14 (for vector number #53) DMA clear request register 15 (for vector number #54)
0x0410	ICSEL16	ICSEL17	ICSEL18	ICSEL19	DMA clear request register 16 (for vector number #55) DMA clear request register 17 (for vector number #56) DMA clear request register 18 (for vector number #57) DMA clear request register 19 (for vector number #58)
0x0414	ICSEL20	ICSEL21	ICSEL22	ICSEL23	DMA clear request register 20 (for vector number #59) DMA clear request register 21 (for vector number #60) DMA clear request register 22 (for vector number #61) DMA clear request register 23 (for vector number #45)
0x0438	Reserved	ICSEL25	Reserved	Reserved	DMA clear request register 25 (for vector number #48)
0x0490	IORR0	IORR1	IORR2	IORR3	IO transfer request register 0 IO transfer request register 1 IO transfer request register 2 IO transfer request register 3
0x0494	IORR4	IORR5	IORR6	IORR7	IO transfer request register 4 IO transfer request register 5 IO transfer request register 6 IO transfer request register 7

9.4.1 DMA Request Clear Register 0: ICSEL0 (Interrupt Clear SELect register 0)

The bit configuration of DMA request clear register 0 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #16).

ICSEL0: Address 0400_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					EISEL[2:0]		
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

[bit2 to bit0] EISEL[2:0] (External Interrupt request SELection): Interrupt clear selection bits for external interrupts 0 to 3

EISEL[2:0]	Clear Target
000	External interrupt 0
001	External interrupt 1
010	External interrupt 2
011	External interrupt 3
100 to 111	Reserved (Does not clear any interrupts)

9.4.2 DMA Request Clear Register 2: ICSEL2 (Interrupt Clear SELect register 2)

The bit configuration of DMA request clear register 2 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #18).

ICSEL2: Address 0402_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						RTSEL0[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1,0] RTSEL0 (Reload Timer SELection): Interrupt clear selection bit for reload timer 0/1/4

RTSEL0[1:0]	Clear Target
00	Reload timer 0
01	Reload timer 1
10	Reload timer 4
11	Reserved (Does not clear any interrupts)

Note:

Setting RTSEL0[1:0]= "11" is prohibited. During this setting, no interrupt clear will be selected.

9.4.3 DMA Request Clear Register 3: ICSEL3 (Interrupt Clear SELect register 3)

The bit configuration of DMA request clear register 3 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #19).

ICSEL3: Address 0403_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							RTSEL1
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit0] RTSEL1 (Reload Timer SELection): Interrupt clear selection bit for reload timer 2/3

RTSEL1	Clear Target
0	Reload timer 2
1	Reload timer 3

9.4.4 DMA Request Clear Register 5: ICSEL5 (Interrupt Clear SELect register 5)

The bit configuration of DMA request clear register 5 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #39).

ICSEL5: Address 0405_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					SG_RX_SEL1[2:0]		
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

[bit2 to bit0] SG_RX_SEL1[2:0] (SG_RX SELection1): Interrupt clear selection bits for 16-bit free-run timer 0 detection/compare clear

SG_RX_SEL1[2:0]	Clear Target
000	Reserved (Does not clear any interrupts)
001	Reserved (Does not clear any interrupts)
010	16-bit free-run timer 0 detection
011	16-bit free-run timer 0 compare clear
100	Reserved (Does not clear any interrupts)
101 to 111	Reserved (Does not clear any interrupts)

Note:

Setting SG_RX_SEL1[2:0]= "000", "001" and "100" to "111" are prohibited. During this setting, no interrupt clear will be selected.

9.4.5 DMA Request Clear Register 11: ICSEL11 (Interrupt Clear SELect register 11)

The bit configuration of DMA request clear register 11 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #46).

ICSEL11: Address $040B_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					PMSTSEL[2:0]		
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

[bit2 to bit0] PMSTSEL[2:0] (PLL, Main SELection): Interrupt clear selection for main timer/PLL timer

PMSTSEL[2:0]	Clear Target
000	Main timer
001	Reserved (Does not clear any interrupts)
010	PLL timer
011	Reserved (Does not clear any interrupts)
100	Reserved (Does not clear any interrupts)
101	Reserved (Does not clear any interrupts)
110	Reserved (Does not clear any interrupts)
111	Reserved (Does not clear any interrupts)

Note:

Setting PMSTSEL[2:0]= "001" and "011 to 111" are prohibited. During this setting, no interrupt clear will be selected.

9.4.6 DMA Request Clear Register 13: ICSEL13 (Interrupt Clear SELect register 13)

The bit configuration of DMA request clear register 13 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #52).

ICSEL13: Address 040D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						CMPSEL[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] CMPSEL[1:0] (CMP SELection): Interrupt clear selection bits for comparator output detection

CMPSEL[1:0]	Clear Target
00	Comparator output detection interrupt 0
01	Comparator output detection interrupt 1
10	Comparator output detection interrupt 2
11	Reserved (Does not clear any interrupts)

Note:

Setting CMPSEL[1:0]= "11" is prohibited. During this setting, no interrupt clear will be selected.

9.4.7 DMA Request Clear Register 14: ICSEL14 (Interrupt Clear SELect register 14)

The bit configuration of DMA request clear register 14 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #53).

ICSEL14: Address 040E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PWCSEL0[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] PWCSEL0[1:0] (PWC SELection0): Interrupt clear 00,10 selection bits for PWC0ch 0 detection 00,10/PWC0ch compare clear

PWCSEL0[1:0]	Clear Target
00	PWC0ch 0 detection interrupt 00
01	PWC0ch 0 detection interrupt 10
10	PWC0ch compare clear interrupt 00
11	PWC0ch compare clear interrupt 10

9.4.8 DMA Request Clear Register 15: ICSEL15 (Interrupt Clear SELect register 15)

The bit configuration of DMA request clear register 15 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #54).

ICSEL15: Address 040F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PWCSEL1[3:0]			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R,W	R,W	R,W	R/W

[bit3 to bit0] (PWC SELection1): Interrupt clear selection bits for PWC0ch capture data upper limit 00,10,20,30/PWC0ch capture data lower limit 00,10,20,30/PWC0ch data buffer 00,10,20,30/PWC0ch buffer overrun 00,10,20,30

PWCSEL1[3:0]	Clear Target
0000	PWC0ch capture data upper limit interrupt 00
0001	PWC0ch capture data upper limit interrupt 10
0010	PWC0ch capture data upper limit interrupt 20
0011	PWC0ch capture data upper limit interrupt 30
0100	PWC0ch capture data lower limit interrupt 00
0101	PWC0ch capture data lower limit interrupt 10
0110	PWC0ch capture data lower limit interrupt 20
0111	PWC0ch capture data lower limit interrupt 30
1000	PWC0ch data buffer interrupt 00
1001	PWC0ch data buffer interrupt 10
1010	PWC0ch data buffer interrupt 20
1011	PWC0ch data buffer interrupt 30
1100	PWC0ch buffer overrun interrupt 00
1101	PWC0ch buffer overrun interrupt 10
1110	PWC0ch buffer overrun interrupt 20
1111	PWC0ch buffer overrun interrupt 30

9.4.9 DMA Request Clear Register 16: ICSEL16 (Interrupt Clear SELect register 16)

The bit configuration of DMA request clear register 16 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #55).

ICSEL16: Address 0410_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PWCSEL2[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] PWCSEL2[1:0] (PWC SELection2): Interrupt clear 01,11 selection bits for PWC1ch 0 detection 01,11/PWC1ch compare clear

PWCSEL2[1:0]	Clear Target
00	PWC1ch 0 detection interrupt 01
01	PWC1ch 0 detection interrupt 11
10	PWC1ch compare clear interrupt 01
11	PWC1ch compare clear interrupt 11

9.4.10 DMA Request Clear Register 17: ICSEL17 (Interrupt Clear SELect register 17)

The bit configuration of DMA request clear register 17 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #56).

ICSEL17: Address 0411_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PWCSEL3[3:0]			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R,W	R,W	R,W	R/W

[bit3 to bit0] (PWC SELection3): Interrupt clear selection bits for PWC1ch capture data upper limit 01,11,21,31/PWC1ch capture data lower limit 01,11,21,31/PWC1ch data buffer 01,11,21,31/PWC1ch buffer overrun 01,11,21,31

PWCSEL3[3:0]	Clear Target
0000	PWC1ch capture data upper limit interrupt 01
0001	PWC1ch capture data upper limit interrupt 11
0010	PWC1ch capture data upper limit interrupt 21
0011	PWC1ch capture data upper limit interrupt 31
0100	PWC1ch capture data lower limit interrupt 01
0101	PWC1ch capture data lower limit interrupt 11
0110	PWC1ch capture data lower limit interrupt 21
0111	PWC1ch capture data lower limit interrupt 31
1000	PWC1ch data buffer interrupt 01
1001	PWC1ch data buffer interrupt 11
1010	PWC1ch data buffer interrupt 21
1011	PWC1ch data buffer interrupt 31
1100	PWC1ch buffer overrun interrupt 01
1101	PWC1ch buffer overrun interrupt 11
1110	PWC1ch buffer overrun interrupt 21
1111	PWC1ch buffer overrun interrupt 31

9.4.11 DMA Request Clear Register 18: ICSEL18 (Interrupt Clear SELect register 18)

The bit configuration of DMA request clear register 18 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #57).

ICSEL18: Address 0412_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						AD_SEL[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] AD_SEL1[1:0] (AD SELection1): Interrupt clear selection for A/DC ch.8 to ch.11

AD_SEL[1:0]	Clear Target
00	A/D converter ch.8
01	A/D converter ch.9
10	A/D converter ch.10
11	A/D converter ch.11

9.4.12 DMA Request Clear Register 19: ICSEL19 (Interrupt Clear SELect register 19)

The bit configuration of DMA request clear register 19 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #58).

ICSEL19: Address 0413_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							BTSEL2
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit0] BTSEL2 (BaseTimer Selection2): Interrupt clear selection bits for BaseTimer2 IRQ0, IRQ1

BTSEL2	Clear Target
0	BaseTimer2 IRQ0
1	BaseTimer2 IRQ1

9.4.13 DMA Request Clear Register 20: ICSEL20 (Interrupt Clear SELect register 20)

The bit configuration of DMA request clear register 20 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #59).

ICSEL20: Address 0414_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							BTSEL3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit0] BTSEL3 (BaseTimer Selection3): Interrupt clear selection bits for BaseTimer3 IRQ0, IRQ1

BTSEL3	Clear Target
0	BaseTimer3 IRQ0
1	BaseTimer3 IRQ1

9.4.14 DMA Request Clear Register 21: ICSEL21 (Interrupt Clear SELect register 21)

The bit configuration of DMA request clear register 21 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #60).

ICSEL21: Address 0415_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						BT_SG_SEL0[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] BT_SG_SEL0[1:0] (BT_SG Selection0): Interrupt clear selection bits for Base Timer0 IRQ0, IRQ1

BT_SG_SEL0[1:0]	Clear Target
00	Base Timer0 IRQ0
01	Base Timer0 IRQ1
10	Reserved (Does not clear any interrupts)
11	Reserved (Does not clear any interrupts)

Note:

Setting BT_SG_SEL0[1:0]= "10" and "11" are prohibited. During this setting, no interrupt clear will be selected.

9.4.15 DMA Request Clear Register 22: ICSEL22 (Interrupt Clear SELect register 22)

The bit configuration of DMA request clear register 22 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #61).

ICSEL22: Address 0416_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						BT_SG_SEL1[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] BT_SG_SEL1[1:0] (BT_SG_Selection1): Interrupt clear selection bits for Base Timer1 IRQ0, IRQ1

BT_SG_SEL1[1:0]	Clear Target
00	Base Timer1 IRQ0
01	Base Timer1 IRQ1
10	Reserved (Does not clear any interrupts)
11	Reserved (Does not clear any interrupts)

Note:

Setting BT_SG_SEL1[1:0]= "10" and "11" are prohibited. During this setting, no interrupt clear will be selected.

9.4.16 DMA Request Clear Register 23: ICSEL23 (Interrupt Clear SELect register 23)

The bit configuration of DMA request clear register 23 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #45).

ICSEL23: Address 0417_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						MFS_SEL0[1:0]	
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit1, bit0] MFS_SEL0[1:0] (MFS_Selection): Interrupt clear selection bits for ICU0

MFS_SEL0[1:0]	Clear Target
00	Reserved (Does not clear any interrupts)
01	16-bit ICU0
10	Reserved (Does not clear any interrupts)
11	Reserved (Does not clear any interrupts)

Note:

Setting MFS_SEL0[1:0]= "00" and "10" to "11" are prohibited. During this setting, no interrupt clear will be selected.

9.4.17 DMA Request Clear Register 25: ICSEL25 (Interrupt Clear SELect register 25)

The bit configuration of DMA request clear register 25 is shown below.

These bits select a peripheral to clear the interrupt (assigned to interrupt vector number #48).

ICSEL25: Address 0439_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			AD_SEL[4:0]				
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit4 to bit0] AD_SEL[4:0] (AD_Selection): Interrupt clear selection bits for A/DC ch.0 to ch.7

AD_SEL[4:0]	Clear Target
00000	A/D converter ch.0
00001	A/D converter ch.1
00010	A/D converter ch.2
00011	A/D converter ch.3
00100	A/D converter ch.4
00101	A/D converter ch.5
00110	A/D converter ch.6
00111	A/D converter ch.7
01000 to 11111	Reserved (Does not clear any interrupts)

Note:

Setting AD_SEL[4:0]= "01000" to "11111" are prohibited. During this setting, no interrupt clear will be selected.

9.4.18 IO Transfer Request Setting Register 0 to 7: IORR0 to 7 (IO triggered DMA Request Register for ch.0 to 7)

The bit configuration of IO transfer request setting register 0 to 7 is shown below.

If the DMA transfer request generation factor is specified as a peripheral interrupt request, these registers identify the vector number of the interrupt request that has generated the DMA transfer request.
An instance of these registers is provided for each DMA controller (DMAC) channel.

IORR0 to 7: Address 0490_H to 0497_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	IOE	IOS[5:0]					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit6] IOE (IO Enabled): Transfer request enable bit

When an interrupt request specified by the IOS5 to IOS0 bits has been generated, this bit is used to notify the DMA controller (DMAC) for the pertinent channel whether to output the DMA transfer request.

IOE	Function
0	Do not output No DMA transfer request. The interrupt request generated by the peripheral is not used as a DMA transfer request (Initial value).
1	Output DMA transfer request.

[bit5 to bit0] IOS (IO triggered DMA transfer request Select): Transfer request selection bits

These bits are used to identify the interrupt request of the vector number that is used as the transfer request source by the DMA controller (DMAC) for the channel corresponding to these registers.

IOS[5:0]	Interrupt Vector Number (Hexadecimal)
000000	0x10 (Initial value)
000001	0x11
000010	0x12
000011	0x13
000100	0x14
000101	0x15
:	:
101100	0x3C
101101	0x3D
101110	0x3E
101111	0x3F
11xxxx	Reserved

Note:

You cannot configure a setting that causes interrupt requests with the same interrupt vector number to be transfer requests from multiple DMA channels. (Example: Simultaneous setting of IORR0=0x42 and IORR1=0x42)

9.5 Operation

This section explains the operation of the generation and clearing of DMA transfer requests.

9.5.1 Configuration

9.5.2 Notes

9.5.1 Configuration

The configuration of the operation is shown.

The sequence for setting DMA transfer using a peripheral under peripheral bus control is as follows.

1. On the IORR, set the interrupt vector number of the transfer request source peripheral and the IOE bit.
2. If multiple peripherals are assigned to the vector number selected in step 1, set ICSEL.
3. Set the interrupt configuration-related registers for the peripheral.
4. Configure the DMAC.

The sequence for setting DMA transfer using a peripheral under on-chip bus control is as follows.

1. If multiple peripherals are assigned to the vector number of the interrupt of the peripheral that is the source of the transfer request, set ICSEL.
2. Set the interrupt configuration-related registers for the peripheral.
3. Configure the DMAC.

9.5.2 Notes

The notes are shown.

- Do not change the IORR and ICSEL registers when the DMAC has enabled DMA transfer requests issued by peripherals under peripheral bus control.
- Do not change the ICSEL register when the DMAC has enabled DMA transfer requests issued by peripherals under on-chip bus control.
- Peripherals to which resource numbers (RN) are not assigned (see "Appendix") cannot use the feature for clearing interrupts after the completion of DMA transfer. It should therefore be noted that once such a peripheral has requested DMA transfer, the interrupt will not be cleared after the completion of the requested DMA transfer.
- Interrupt requests used as transfer requests are considered as interrupt requests addressed to the CPU. Therefore, configure the interrupt controller to disable interrupts. (ICR register)

10. FixedVector Function

This chapter explains the FixedVector function.

10.1 Overview
10.2 Features
10.3 Configuration
10.4 Registers
10.5 Operation
10.6 Notes

10.1 Overview

This section explains the overview of the FixedVector function.

The FixedVector function is a function for returning the start address of flash memory + 0x0024 instead of the content of flash memory at the address (0xF_FFFC) corresponding to the interrupt vector on reset.

10.2 Features

This section explains the features of the FixedVector function.

- Interrupt vector on reset returned by the FixedVector function:
  - ☐ MB91F552 0x000D_0024

10.3 Configuration

This section explains the configuration of the FixedVector function.

See "Figure 36-2" in "Chapter: Flash Memory" for the configuration diagram.

10.4 Registers

This section explains the registers of the FixedVector function.

None

10.5 Operation

This section explains the operation of the FixedVector function.

10.5.1 Operation After Reset Released

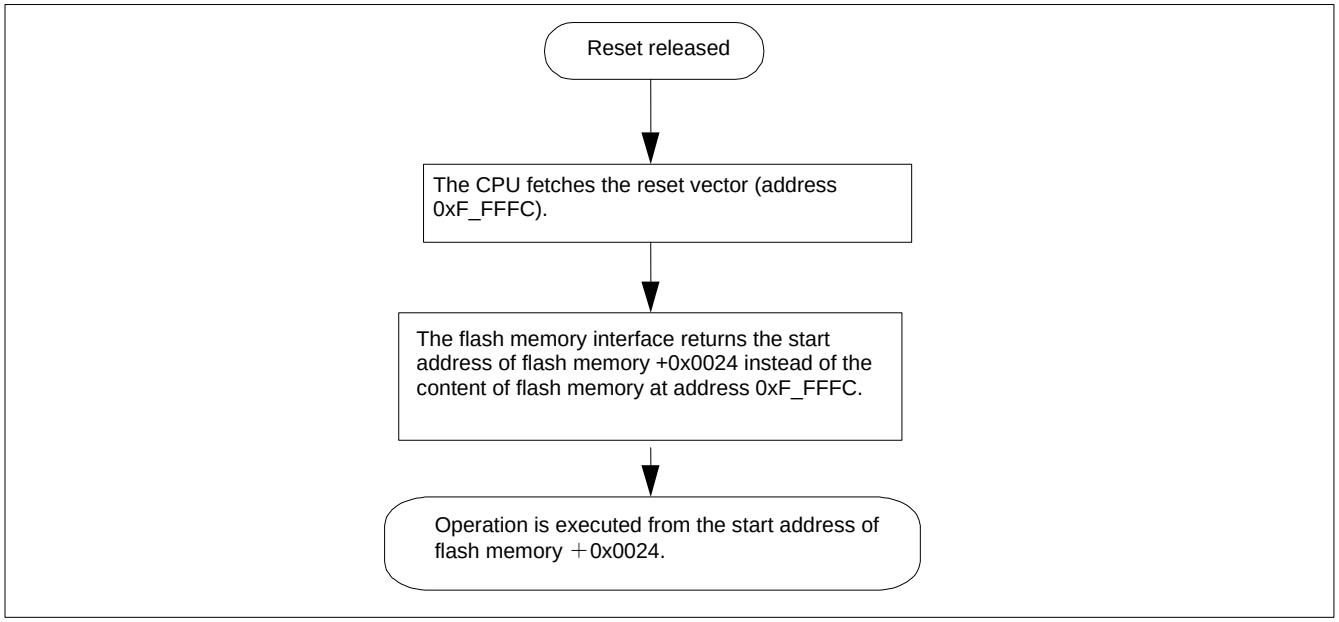
10.5.2 Usage

10.5.1 Operation After Reset Released

The operation after reset released is shown below.

In the following flow, the start address of flash memory + 0x0024 is returned instead of the content of 0xF_FFFC in flash memory when the reset is released.

Figure 10-1. Operation Flow after Reset



10.5.2 Usage

The usage is shown below.

After the reset is released, this series executes from the start address of flash memory + 0x0024 instead of the value written at address 0x000F_FFFC.

10.6 Notes

This section explains the notes of the FixedVector function.

During reads from addresses 0x000F_FFFC to 0x000F_FFFF other than reset vector fetch (Example: the call destination when INT #00H is executed while TBR is its initial value (=0x000F_FC00)), the content of flash memory at the addresses 0x000F_FFFC to 0x000F_FFFF is returned.

11. I/O Ports

This chapter explains the I/O ports.

11.1 Overview

11.2 Features

11.3 Configuration

11.4 Registers

11.5 Operation

11.1 Overview

This section explains the overview of the I/O ports.

This section explains the setting for assigning to the external pins (peripherals) and using external pins as the I/O port.

11.2 Features

This section explains features of the I/O ports.

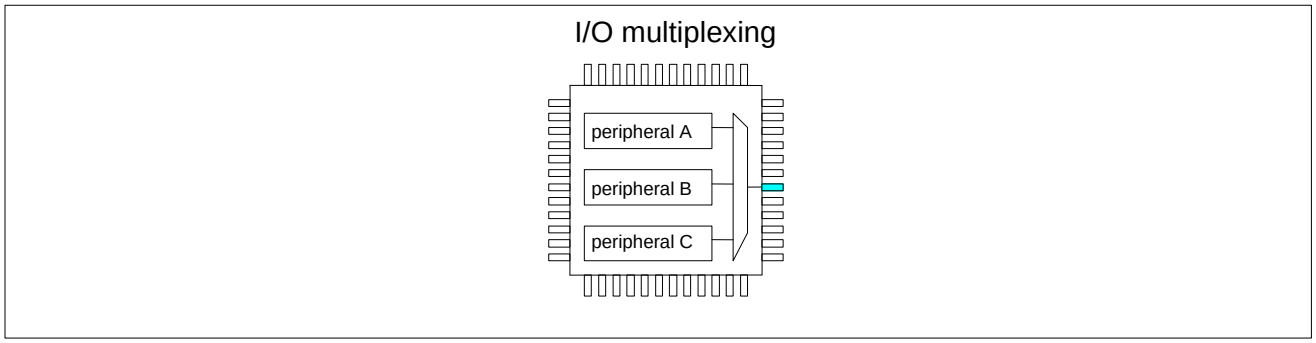
I/O multiplexing

If the I/O of multiple peripherals is assigned to one external pin, one of these peripherals is selected to be used.

PORT function

External pins can be used for general-purpose I/O: if they are used for output, their values can be set and if they are used for input, input values assigned to them can be read.

Figure 11-1. Diagram of I/O Multiplexing



Key code function

This function is for error writing protection. If writing is not executed to the key code register (KEYCDR) according to the specified method, writing to the target register will become invalid. Also, word access for the target register cannot be executed.

The following are the key code target registers.

- Data direction register
- Port function register
- Extended port function register
- Port pull-up/down enable register
- Port input enable register
- Analog input control register

11.3 Configuration

This section explains the configuration of the I/O ports.

No configuration diagram is provided.

11.4 Registers

This section explains registers of the I/O ports.

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0000	PDR00	Reserved	PDR02	PDR03	Port data register 00, 02 to 04
0x0004	PDR04	Reserved	Reserved	Reserved	
0x0E00	Reserved	Reserved	DDR02	DDR03	Data direction register 02 to 04 (Key code target registers)
0x0E04	DDR04	Reserved	Reserved	Reserved	
0x0E20	PFR00	Reserved	PFR02	PFR03	Port function register 00, 02 to 04 (Key code target registers)
0x0E24	PFR04	Reserved	Reserved	Reserved	
0x0E40	PDDR00	Reserved	PDDR02	PDDR03	Input data direct read register 00, 02 to 04
0x0E44	PDDR04	Reserved	Reserved	Reserved	
0x0E60	Reserved	Reserved	EPFR02	Reserved	Extended port function register 02, 26, 35 (Key code target registers)
0x0E64	Reserved	Reserved	Reserved	Reserved	
0x0E68	Reserved	Reserved	Reserved	Reserved	
0x0E6C	Reserved	Reserved	Reserved	Reserved	
0x0E70	EPFR16	Reserved	Reserved	Reserved	
0x0E74	Reserved	Reserved	Reserved	Reserved	
0x0E78	Reserved	Reserved	EPFR26	Reserved	
0x0E7C	Reserved	Reserved	Reserved	Reserved	
0x0E80	Reserved	Reserved	Reserved	EPFR35	
0x0EC0	PPER00	Reserved	PPER02	PPER03	Port pull-up/down enable register 00, 02 to 04 (Key code target registers)
0x0EC4	PPER04	Reserved	Reserved	Reserved	
0x0F40	PORTEN	Reserved	Reserved	Reserved	Port input enable register (Key code target registers)
0x0F44	KEYCDR		Reserved	Reserved	Key cord register
0x01B8	Reserved	EPFR65	Reserved	Reserved	Extended port function register 65, 86, 88 (Key code target registers)
0x01CC	Reserved	Reserved	EPFR86	Reserved	
0x01D0	EPFR88	Reserved	Reserved	Reserved	

11.4.1 Port Data Register 00, 02 to 04: PDR00, 02 to 04 (Port Data Register 00, 02 to 04)

The bit configuration of port data register 00, 02 to 04 is shown below.

These registers hold the output levels of the pins corresponding to individual ports that are in output mode.

PDR00, PDR02 to PDR04: Address 0000_H, 0002_H, (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
Initial value	X	X	X	X	X	X	X	X
Attribute	R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W	R,RM/W

[bit7 to bit0] P (Port): Port data setting bits

These bits set the output level of external pins P000, P001, and so on when the ports are in output mode.

PDR00.P[5:0] is for external pins P005 to P000
PDR02.P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	Output of "0"
1	Output of "1"

The value read by a read-modify instruction is determined based on the combination with the data direction register (DDR).

DDR	Reading by Read-modify Instruction	PDR Reading Value
1	No	The PDR value can be read.
1	Yes	The PDR value can be read.
0	No	The pin value can be read.
0	Yes	The PDR value can be read.

PDR00.P[7:6] are reserved bits. These bits always read "1". Writing has no effect on operation.

11.4.2 Data Direction Register 02 to 04: DDR02 to 04 (Data Direction Register 02 to 04)

The bit configuration of data direction register 02 to 04 is shown below.

These registers set the I/O directions of the pins when they function as ports. If a pin is to be used for input for a peripheral, the corresponding bit must be set for input. P000 to P005 cannot be used as output ports.
DDR02 to DDR04 are key code target registers.

DDR02 to DDR04: Address $0E02_H$, $0E03_H$, (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit0] P (Port): Data direction selection bits

These bits set the I/O direction of external pins P020, P021, and so on when the ports are in output mode.
DDR02.P[7:0] is for external pins P027 to P020
DDR03.P[7:0] is for external pins P037 to P030
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	Input (initial value)
1	Output

11.4.3 Port Function Register 00, 02 to 04: PFR00, 02 to 04 (Port Function Register 00, 02 to 04)

The bit configuration of port function register 00, 02 to 04 is shown below.

These registers specify whether or not the pins are used to function as ports. If a pin is to be used as a peripheral input pin, the corresponding bit must be set for the port function.
PFR00, PFR02 to PFR04 are key code target registers.

PFR00, PFR02 to PFR04: Address $0E20_H$, $0E22_H$, (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
Initial value	*	*	*	*	*	*	*	*
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

* For the initial value of each register, see "I/O Map" in "Appendix."

[bit7 to bit0] P (Port): Port function selection bits

These bits set the port function.
PFR00.P[5:0] is for external pins P005 to P000
PFR02.P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	Port function or peripheral input pin (initial value)
1	Peripheral I/O (bidirectional) pin, peripheral output pin

PFR00.P[7:6] are reserved bits. These bits always read "1". Writing has no effect on operation.

11.4.4 Input Data Direct Register 00, 02 to 04: PDDR00, 02 to 04 (Port Data Direct Register 00, 02 to 04)

The bit configuration of input data direct register 00, 02 to 04 is shown below.

These registers can always show the voltage levels of individual external pins. These registers can always be read unconditionally.

PDDR00, PDDR02 to PDDR04: Address $0E40_H$, $0E42_H$, (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
Initial value	X	X	X	X	X	X	X	X
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7 to bit0] P (Port): Read bits

The value of the external pins can be read.
PDDR00.P[5:0] is for external pins P005 to P000
PDDR02.P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	"L" level
1	"H" level

PDDR00.P[7:6] are reserved bits. These bits always read "1". Writing has no effect on operation.

11.4.5 Port Pull-up/down Enable Register 00, 02 to 04: PPER00, 02 to 04 (Port Pull-up/down Enable Register 00, 02 to 04)

The bit configuration of port pull-up/down enable register 00, 02 to 04 is shown below.

These registers enable pull-up or pull-down of each port.
PPER00, PPER02 to PPER04 are key code target registers.

PPER02 to 04: Address $0EC2_H$, $0EC3_H$, (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit0] P (Port): Pull-up/down enable selection bits

PPER02.P[7:0] is for external pins P027 to P020
PPER03.P[7:0] is for external pins P037 to P030
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	Pull-up disabled (initial value)
1	Pull-up enabled

See "List of Pin Functions" and "I/O Circuit Types" of "Chapter: Overview" for the existence of pull-up function.

PPER00: Address $0EC0_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		P[5:0]					
Initial value	1	1	0	0	0	0	0	0
Attribute	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit5 to bit0] P (Port): Pull-up/down enable selection bits

PPER00.P[5:0] is for external pins P005 to P000
The assignment is as shown above.

P[n]	Operation
0	Pull-down disabled (initial value)
1	Pull-down enabled

See "List of Pin Functions" and "I/O Circuit Types" of "Chapter: Overview" for the existence of pull-down function.
PPER00.P[7:6] are reserved bits. These bits always read "1". Writing has no effect on operation.

11.4.6 Extended Port Function Register 02, 16, 26, 35, 65, 86, 88: EPFR02, 16, 26, 35, 65, 86, 88 (Extended Port Function Register 02, 16, 26, 35, 65, 86, 88)

The bit configuration of extended port function register 02, 16, 26, 35, 65, 86, 88 is show below.

These registers control I/O multiplexing. Unlike other port registers, these registers have an enable bit for each peripheral, rather than for each pin.

EPFR02, 16, 26, 35, 65, 86, 88 are key code target registers.

11.4.6.1 Extended Port Function Register 02: EPFR02 (Extended Port Function Register 02)

The bit configuration of extended port function register 02 is shown.

This register selects reload timer output enable. (I/O multiplexing)

EPFR02: Address $0E62_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			TOT4E	TOT3E	TOT2E	TOT1E	TOT0E
Initial value	1	1	1	0	0	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W

[bit7 to bit5] Reserved

These bits always read "1". Writing has no effect on operation.

[bit4] TOT4E: Reload timer TOT4 output enable

[bit3] TOT3E: Reload timer TOT3 output enable

[bit2] TOT2E: Reload timer TOT2 output enable

[bit1] TOT1E: Reload timer TOT1 output enable

[bit0] TOT0E: Reload timer TOT0 output enable

TOTnE (n=0 to 4)	Operation
0	Reload timer TOTn output disabled (initial value)
1	Reload timer TOTn output enabled

11.4.6.2 Extended Port Function Register 16: EPFR16 (Extended Port Function Register 16)

The bit configuration of extended port function register 16 is shown.

This register enables master/slave output for PWM parallel operation drive. (I/O multiplexing)

EPFR16: Address $0E70_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							SYNOE
Initial value	1	1	1	1	1	1	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W

[bit7 to bit1] Reserved

These bits always read "1". Writing has no effect on operation.

[bit0] SYNOE: PWM parallel operation drive master/slave output enable

SYNOE	Operation
0	PWM parallel operation drive master/slave output disabled (initial value)
1	PWM parallel operation drive master/slave output enabled

11.4.6.3 Extended Port Function Register 35: EPFR35 (Extended Port Function Register 35)

The bit configuration of extended port function register 35 is shown.

This register enables multifunction serial interface output. (I/O multiplexing)

EPFR35: Address $0E83_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		SOT2E	SCK2E	SOT1E	SCK1E	SOT0E	SCK0E
Initial value	1	1	0	0	0	0	0	0
Attribute	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit6] Reserved

These bits always read "1". Writing has no effect on operation.

[bit5] SOT2E: Multifunction serial interface SOT2 output enable

[bit4] SCK2E: Multifunction serial interface SCK2 output enable

[bit3] SOT1E: Multifunction serial interface SOT1 output enable

[bit2] SCK1E: Multifunction serial interface SCK1 output enable

[bit1] SOT0E: Multifunction serial interface SOT0 output enable

[bit0] SCK0E: Multifunction serial interface SCK0 output enable

SOTnE (n=0 to 2)	Operation
0	Multifunction serial interface SOTn output disabled (initial value)
1	Multifunction serial interface SOTn output enabled

SCKnE (n=0 to 2)	Operation
0	Multifunction serial interface SCKn output disabled (initial value)
1	Multifunction serial interface SCKn output enabled

11.4.6.4 Extended Port Function Register 65: EPFR65 (Extended Port Function Register 65)

The bit configuration of extended port function register 65 is shown.

These registers enable serial chip select output.

EPFR65: Address $01B9_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		SCSO21E	SCSO20E	SCSO11E	SCSO10E	SCSO01E	SCSO00E
Initial value	1	1	0	0	0	0	0	0
Attribute	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit6] Reserved

These bits always read "1". Writing has no effect on operation.

[bit5] SCSO21E: Multifunction serial interface SCS21 output enable

[bit4] SCSO20E: Multifunction serial interface SCS20 input/output enable

[bit3] SCSO11E: Multifunction serial interface SCS11 output enable

[bit2] SCSO10E: Multifunction serial interface SCS10 input/output enable

[bit1] SCSO01E: Multifunction serial interface SCS01 output enable

[bit0] SCSO00E: Multifunction serial interface SCS00 input/output enable

SCSOnE (n=00, 01, 10, 11, 20, 21)	Operation
0	Input and output from the SCSn pin disabled (initial value) *
1	Input and output from the SCSn pin enabled

*: SCSOnE (n=01, 11, 21) is output only.

11.4.6.5 Extended Port Function Register 86: EPFR86 (Extended Port Function Register 86)

The bit configuration of extended port function register 86 is shown.

This register selects CAN output enable. (I/O multiplexing)

EPFR86: Address $01CE_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					TX1E	Reserved	
Initial value	1	1	1	1	1	0	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R1,WX	R1,WX

[bit7 to bit3, bit1 to bit0] Reserved

These bits always read "1". Writing has no effect on operation.

[bit2] TX1E CAN transmission data output enable

TX1E	Operation
0	CAN output disabled (initial value)
1	CAN output enabled

11.4.6.6 Extended Port Function Register 26: EPFR26 (Extended Port Function Register 26)

The bit configuration of extended port function register 26 is shown.

This register selects base timer output enable. (I/O multiplexing)

EPFR26: Address $0E7A_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TIA3E	Reserved	TIA2E	Reserved	TIA1E	Reserved	TIA0E
Initial value	1	0	1	0	1	0	1	0
Attribute	R1,WX	R/W	R1,WX	R/W	R1,WX	R/W	R1,WX	R/W

[bit7] Reserved

"1" is always read. Writing has no effect on operation.

[bit6] TIA3E: Base timer TIOA3 output enable

[bit5] Reserved

"1" is always read. Writing has no effect on operation.

[bit4] TIA2E: Base timer TIOA2 output enable

[bit3] Reserved

"1" is always read. Writing has no effect on operation.

[bit2] TIA1E: Base timer TIOA1 output enable

[bit1] Reserved

"1" is always read. Writing has no effect on operation.

[bit0] TIA0E: Base timer TIOA0 output enable

TIAnE (n=0 to 3)	Operation
0	Base timer TIOAn output disabled (initial value)
1	Base timer TIOAn output enabled

11.4.6.7 Extended Port Function Register 88: EPFR88 (Extended Port Function Register 88)

The bit configuration of extended port function register 88 is shown.

This register enables clock monitor output. (I/O multiplexing)

EPFR88: Address $01D0_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							MONKCLKE
Initial value	1	1	1	1	1	1	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W

[bit7 to bit1] Reserved

These bits always read "1". Writing has no effect on operation.

[bit0] MONKCLKE: Clock monitor output pin select

MONKCLKE	Operation
0	MONCLK output disabled (initial value)
1	MONCLK output enabled

11.4.7 Port Input Enable Register: PORTEN (PORT ENable register)

The bit configuration of the port input enable register is shown below.

This register releases the port input block. At a power-on reset, inputs to most pins are blocked in order to avoid pass-through current fluctuations before the ports are configured by software. See "A.3. Pin States by CPU States" in "Appendix" for the pin that becomes input blocked. After each port pin is configured according to its function, enable port input with the global port enable (PORTEN.GPORTEN).

The PORTEN is the key code target register.

PORTEN: Address $0F40_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							GPORTEN
Initial value	1	1	1	1	1	1	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W

[bit7 to bit1] Reserved

These bits always read "1". Writing has no effect on operation.

[bit0] GPORTEN (Global PORT ENable): Global input block release

GPORTEN	Operation
0	Most pins are set to block input. See "A.3. Pin States by CPU States" in "Appendix" for the pins that are input-blocked. (Initial value)
1	Input block by this bit is released.

11.4.8 KEY CoDe Register: KEYCDR (KEY CoDe register)

The bit configuration of key code register is shown.

This register sets register writing that includes the error writing protection function.
If writing to this register is not executed according to the specified method, writing to the target register will become invalid.
This register is only enabled for half-word access.

KEYCDR: Address $0F44_H$ (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	KEY1	KEY0	SIZE	RADR12	RADR11	RADR10	RADR9	RADR8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RADR7	RADR6	RADR5	RADR4	RADR3	RADR2	RADR1	RADR0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

[bit15, bit14] KEY1, KEY0: Key code

Key code setting bits. Write continuously to these bits in the following order: "00", "01", "10", and "11".

Note:

When the writing order becomes different, the key code setting will become invalid. Reset them from the beginning.

[bit13] SIZE: Access size

This bit sets the access size for writing to the key code target register. Write the same data to the bit when writing the key codes "00", "01", "10", and "11", in this order.

SIZE	Description
0	Set byte access
1	Set half-word access

Notes:

- When different data is written while the key codes "00", "01", "10", and "11" are being written, the key code setting becomes invalid. Set it again from the beginning.
- Word access for the key code target register is prohibited.

[bit12 to bit0] RADR[12:0]: Port address

These bits set the lower 13 bits of the address for the key code target register. Write the same data to the bits when writing the key codes "00", "01", "10", and "11", in this order.

Notes:

- When different data is written while the key codes "00", "01", "10", and "11" are being written, the key code setting becomes invalid. Set it again from the beginning.
- Key code setting might be canceled because of the DMA transfer. Read the value written in the object register, and confirm whether the value has been changed.

11.5 Operation

This section explains operations of I/O ports.

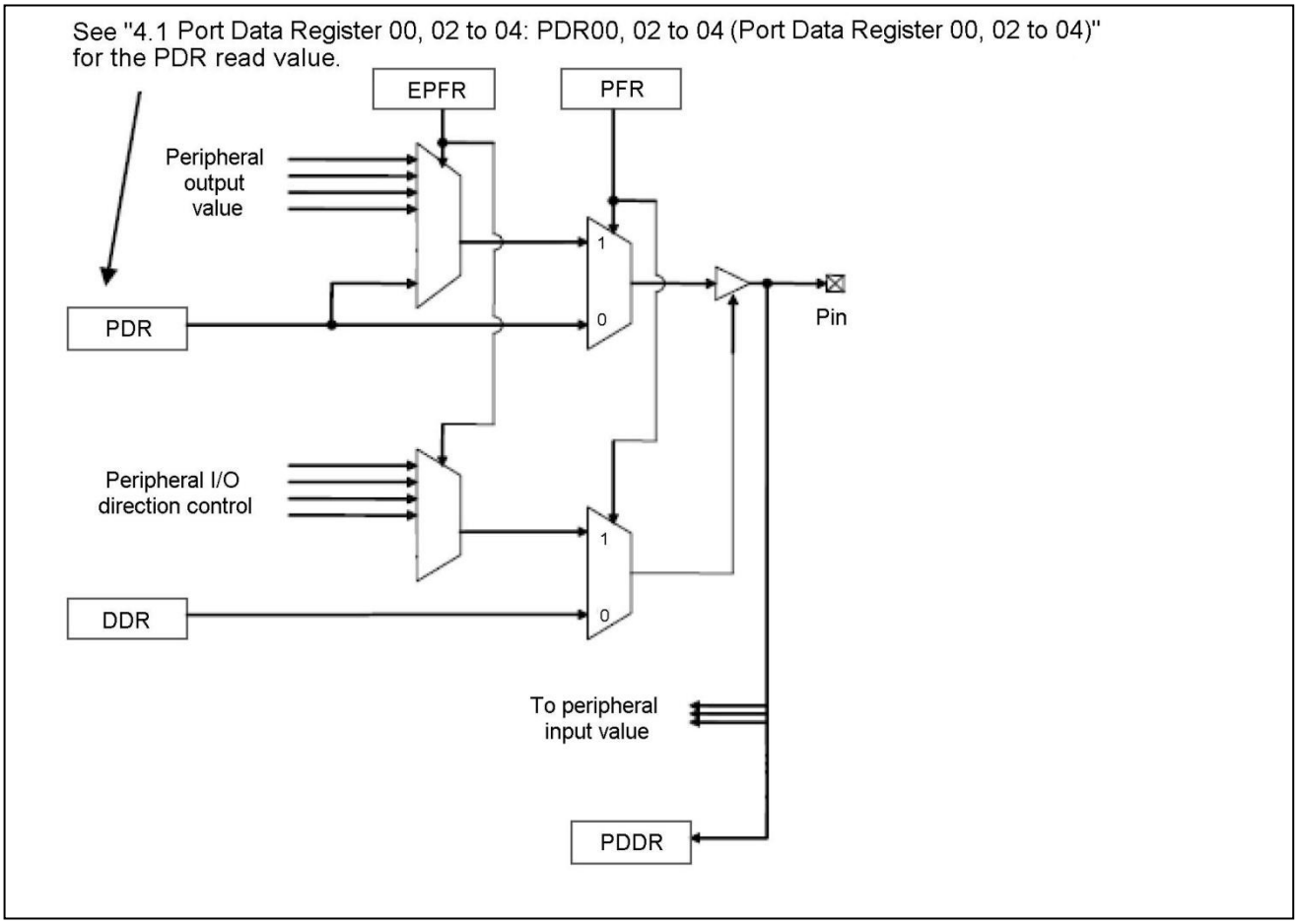
11.5.1 Pin I/O Assignment
11.5.2 EPFR Setting Priority
11.5.3 Noise Filter
11.5.4 Input Blocked by GPORTEN
11.5.5 Notes on Pins with the A/D Converter Function
11.5.6 Settings When Using the Base Timer TIOA1 and TIOA3 Pins
11.5.7 Key Code Register Function Settings
11.5.8 Notes on Switching the I/O Port Function
11.5.9 Input blocked when specific peripheral functions are used

11.5.1 Pin I/O Assignment

The pin I/O assignment is shown below.

Pin I/O assignment is explained here. The I/O direction of each pin is controlled based on the configuration shown below.

Figure 11-2. Configuration of Pin I/O Directions, Output Value Selection, and Input Value Retrieval



As explained in the pertinent section concerning pin assignment, first change the PFR setting to enable the port function. Since the pin functions as a port, also set the DDR and PDR values in advance if necessary. Note that the I/O direction of the pin is once set as specified by the DDR.

For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode." For details on how to set it, see "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)".

11.5.1.1 Peripheral I/O (Bidirectional) Pin Assignment

The peripheral I/O (bidirectional) pin assignment is shown below.

Preparation

- Since the pin once functions as a port as the result of step (1), set the DDR and PDR values in advance if necessary.
- For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode." For details on how to set it, see "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)".

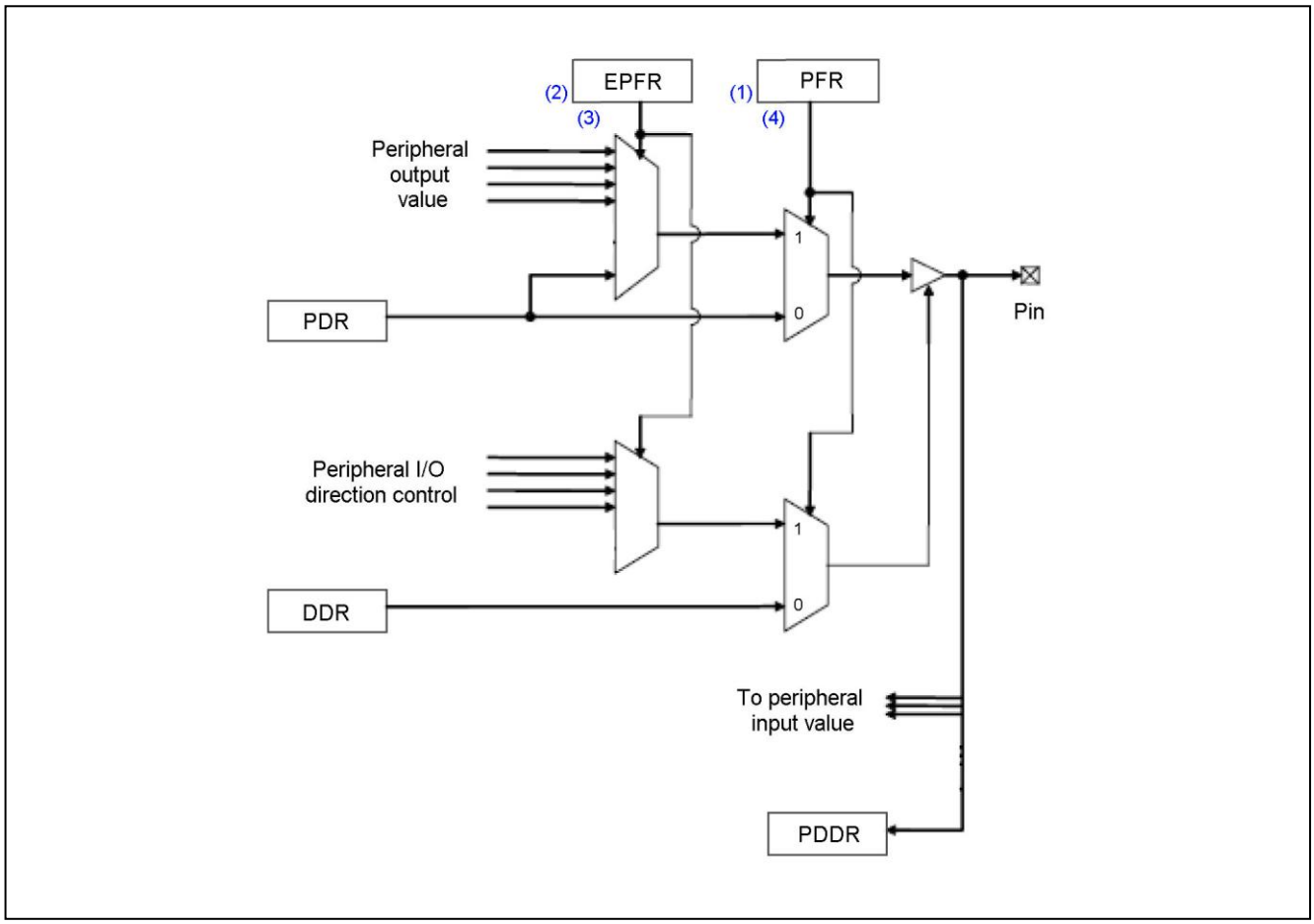
(1) Set the PFR for the applicable pin to enable the port function.

(2) Disable the EPFRs for all other peripherals that use the relevant pin.

(3) If the relevant peripheral is one of the targets of I/O multiplexing, set the EPFR of the relevant peripheral.

(4) Set the PFR for the peripheral.

Figure 11-3. Peripheral I/O Assignment Procedure



11.5.1.2 Peripheral Input Assignment

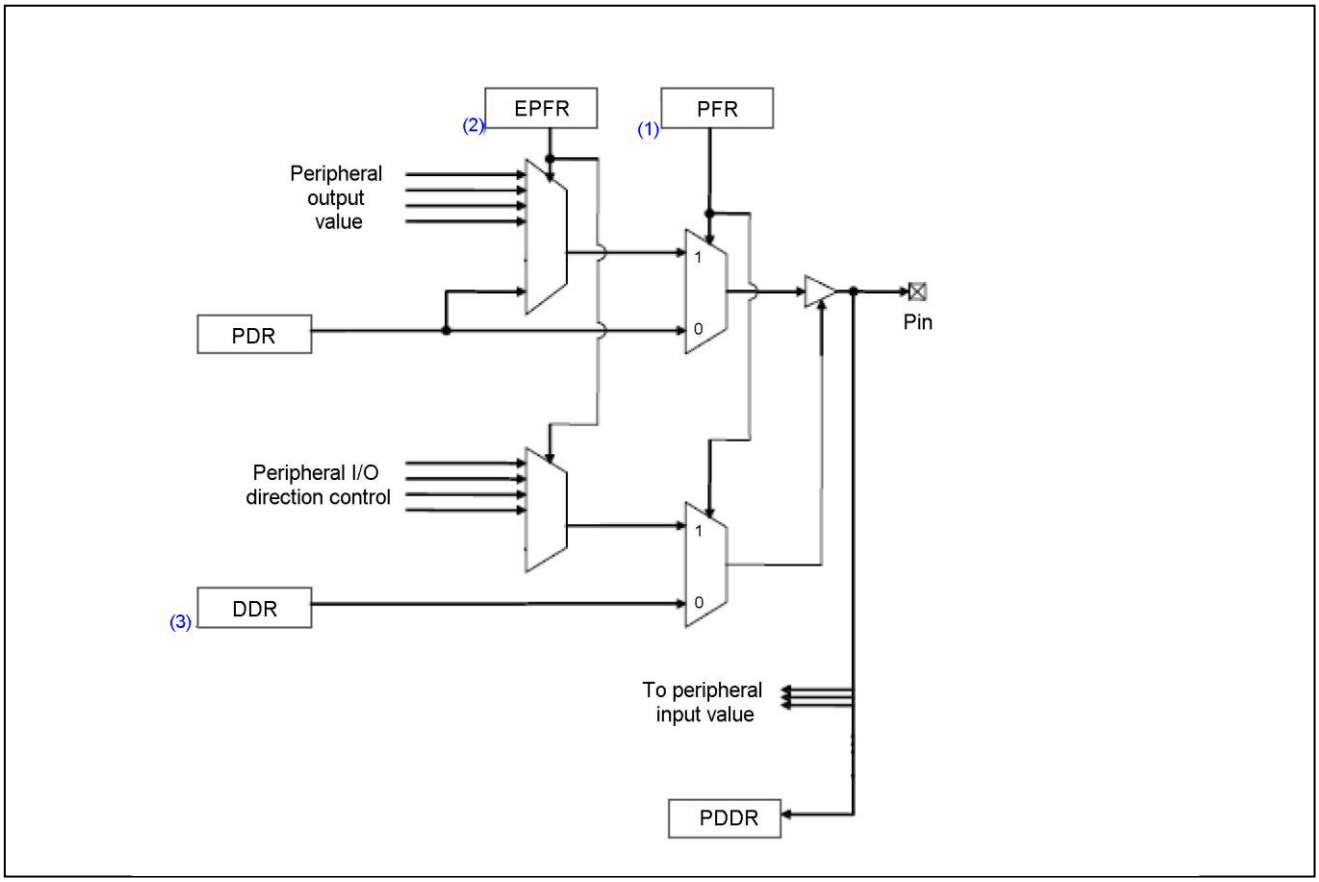
The peripheral input assignment is shown below.

Preparation

- Since the pin once functions as a port as the result of step (1), set the DDR and PDR values in advance if necessary.
- For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode." For details on how to set it, see "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)".

(1) Set the PFR for the applicable pin to enable the port function.
(2) Disable the EPFRs for all other peripherals that use the relevant pin.
(3) Set the DDR for input.

Figure 11-4. Peripheral Input Assignment Procedure



Note:

As shown in the figure above, if the pin is set for peripheral output etc., its output value is supplied to other peripheral inputs sharing the same pin.

11.5.1.3 Peripheral Output Assignment

The peripheral output assignment is shown below.

The setting method is the same as that described in "11.5.1.1 Peripheral I/O (Bidirectional) Pin Assignment."

Preparation

- Since the pin once functions as a port as the result of step (1), set the DDR and PDR values in advance if necessary.
- For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode." For details on how to set it, see "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)".

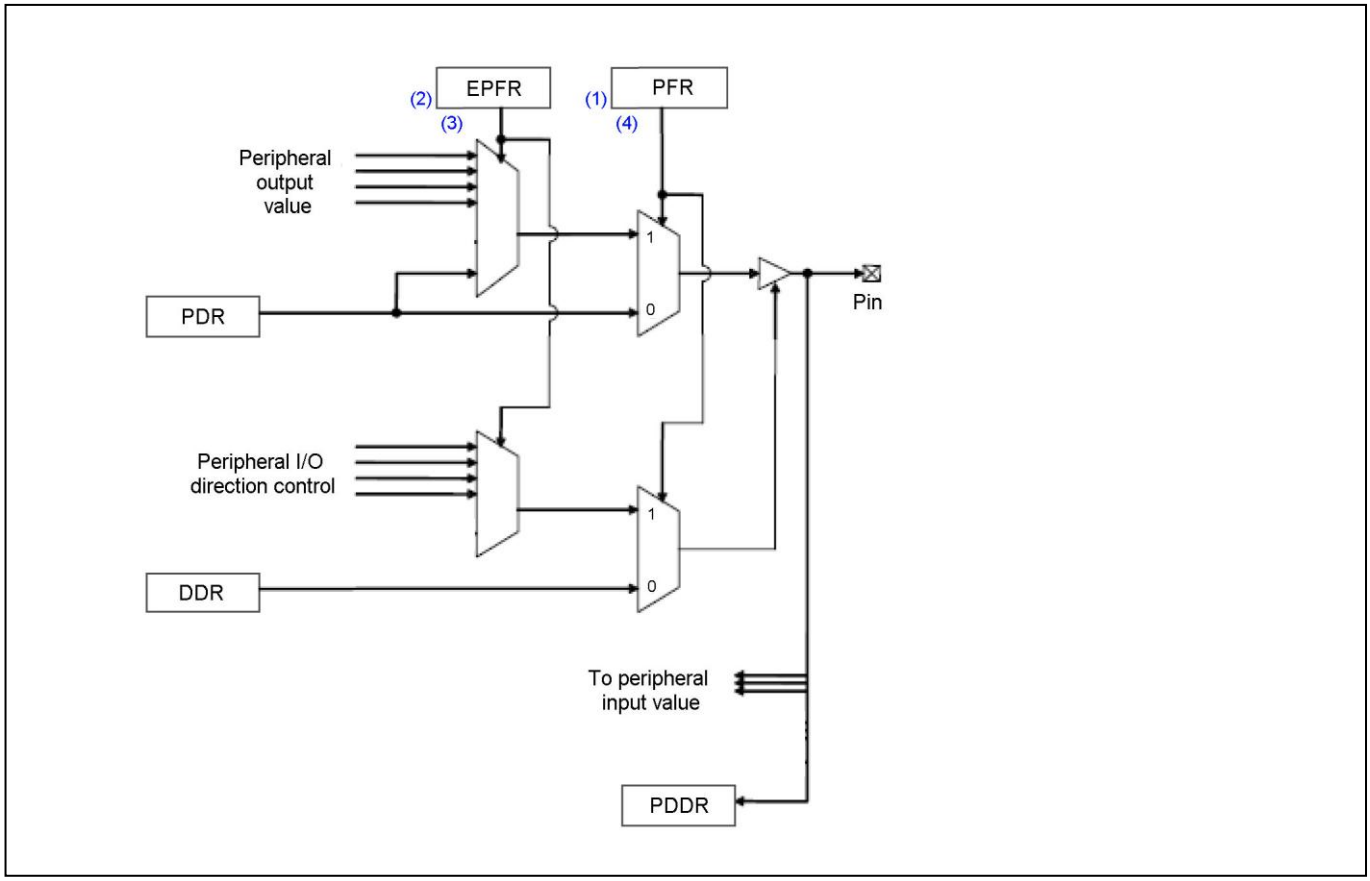
(1) Set the PFR for the applicable pin to enable the port function.

(2) Disable the EPFRs for all other peripherals that use the relevant pin.

(3) If the relevant peripheral is one of the targets of I/O multiplexing, set the EPFR of the relevant peripheral.

(4) Set the PFR for the peripheral.

Figure 11-5. Peripheral Output Assignment Procedure



11.5.1.4 Port Function (Input) Assignment

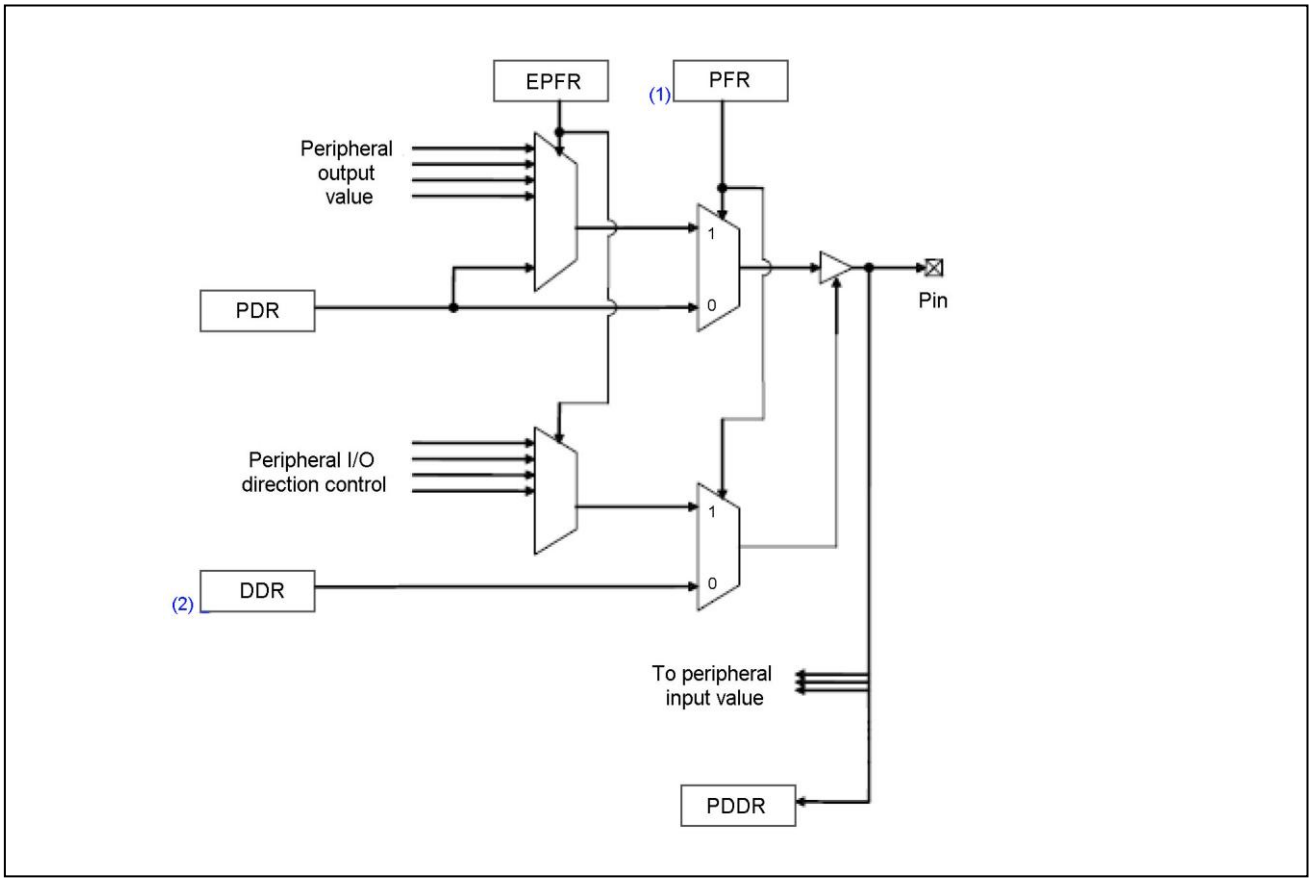
The port function (input) assignment is shown below.

Preparation

- For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode." For details on how to set it, see "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)".

(1) Set the PFR to enable the port function.
(2) Set the DDR for input.

Figure 11-6. Port Function (Input) Assignment Procedure



11.5.1.5 Port Function (Output) Assignment

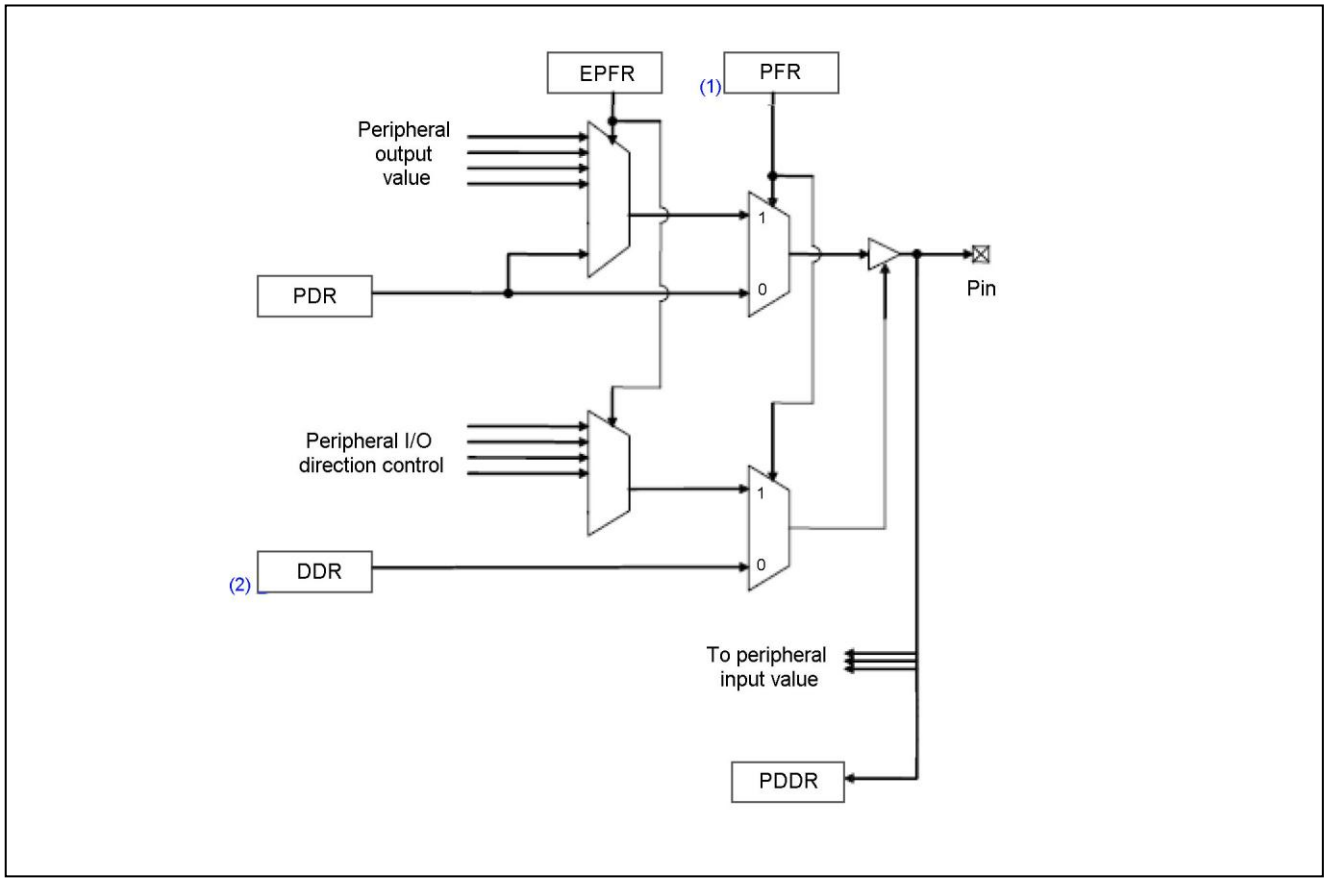
The port function (output) assignment is shown below.

Preparation

- For a pin with the A/D converter function, set the applicable bit in the analog input enable register (ADER) of the A/D converter to "Port I/O mode." For details on how to set it, see "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)".

(1) Set the PFR to enable the port function.
(2) Set the DDR for output.

Figure 11-7. Port Function (Output) Assignment Procedure



Note:

P000 to P005 cannot be used as output ports.

11.5.1.6 A/D Converter Input Assignment

The A/D converter input assignment is shown below.

1. Set the analog input enable register (ADER) of the A/D converter to analog input mode. See "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)".
   Since the A/D converter assignment is given the highest priority, no other configuration is required.

11.5.2 EPFR Setting Priority

The EPFR setting priority is explained below.

If the PFR is set for the peripheral and multiple EPFR settings are overlapping for a single pin, the valid peripheral is determined based on the following priorities.

1. Clock supervisor
2. WM
3. Multifunction serial interface
4. Base timer
5. Reload timer
6. Clock monitor

Note:

Clock supervisor output is not assigned the EPFR registers. Set it with CSVCR.OUTE.

11.5.3 Noise Filter

The noise filter is shown.

If an external pin is used to receive input for the following functions, the value that is entered through the noise filter is treated as the input level.

- Port function
- External interrupt request
- Free-run timer
- Reload timer
- Input capture
- A/D converter trigger input
- Base timer

Note:

For details, see "Pins of Each Function" in "Chapter: Overview".

11.5.4 Input Blocked by GPORTEN

The input blocked function by GPORTEN is explained below.

The majority of pins become the input blocked to avoid the change of the penetration current before the port is set with software at power-on reset. See "A.3. Pin States by CPU States" in "Appendix" for the pins that are input-blocked. For the input block release method, see "11.4.7 Port Input Enable Register: PORTEN (PORT ENable register)".

When the state of a pin to be the input blocked state is read during the input blocked by GPORTEN, "0" is always read out.

11.5.5 Notes on Pins with the A/D Converter Function

Notes on pins with the A/D converter function are shown below.

When using a pin with the A/D converter function to perform a different function, set the relevant bit of the A/D converter analog input enable register (ADER) to "Analog input disable" in advance. For details on how to set it, see "Chapter: 12-Bit A/D Converter (4-Channel Simultaneous Sampling)." If analog input is enabled, inputs from ports and from peripheral functions are fixed at "0" and outputs are fixed at Hi-Z regardless of the port function register (PFR00 to PFR04) and extended port function register (EPFR02, EPFR26, EPFR35, EPFR65, EPFR86, EPFR88) settings.

11.5.6 Settings When Using the Base Timer TIOA1 and TIOA3 Pins

The settings when using the base timer TIOA1 and TIOA3 pins are shown below.

To use the base timer TIOA1 pin, it must be set for input for base timer I/O mode 1 and set for output for all cases other than base timer I/O mode 1. To use the base timer TIOA1 pin, it must be set for peripheral input for base timer I/O mode 1 (see "11.5.1.2 Peripheral Input Assignment") and set for peripheral output for all cases other than base timer I/O mode 1 (see "11.5.1.3 Peripheral Output Assignment"). The same is true for the base timer TIOA3 pin.

11.5.7 Key Code Register Function Settings

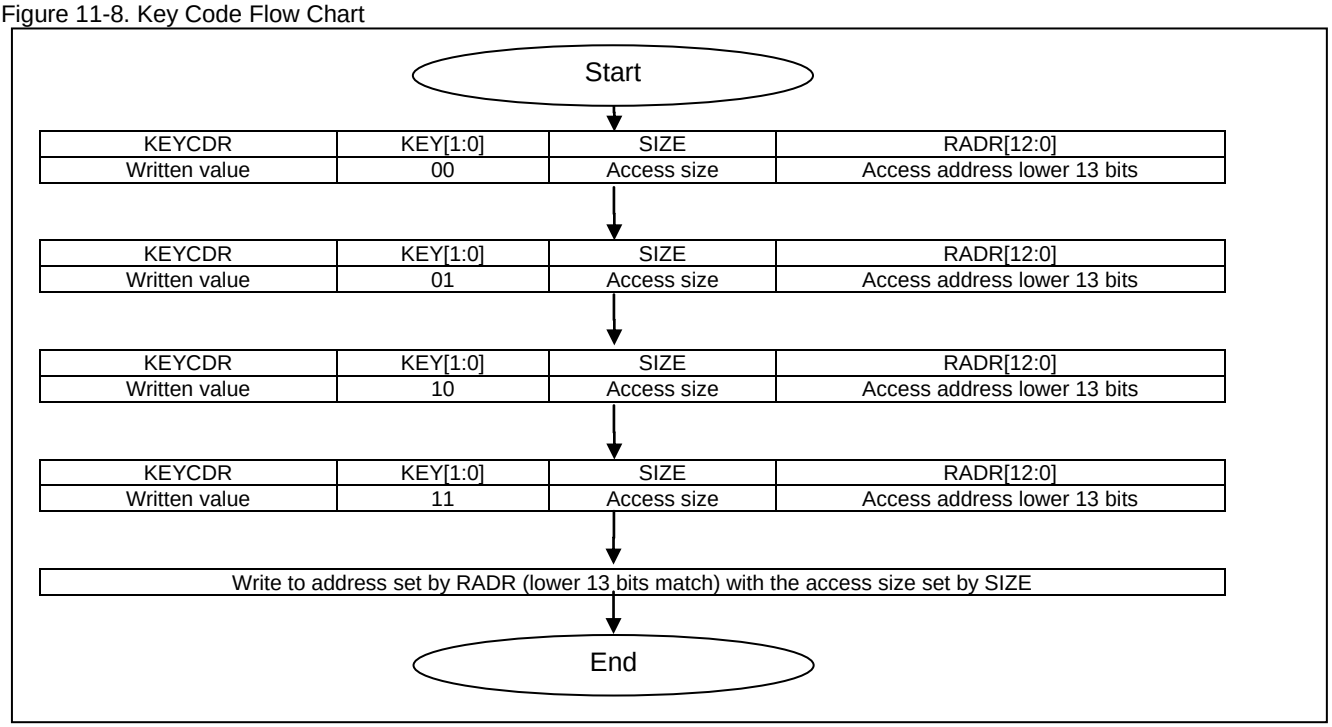
Setting when using the key code register is shown.

The following settings are necessary for the key code register (KEYCDR) in order to write to the key code target register.

- Set KEY1+KEY0+Access Size (SIZE)+Access address (RADR[12:0]) to the key code register using half-word.
- Write (KEY1, KEY0) continuously according to the order (0,0), (0,1), (1,0), and (1,1). Set the address and access size to the same value four times when (KEY1, KEY0) is written four times.

The following is a flow chart.

Figure 11-8. Key Code Flow Chart



If the following conditions apply, the key code will not be released and writing will not be executed to the target register. In this case, it is necessary to set the key code register again from the beginning.

- When writing order for (KEY1, KEY0) is different
- When the data written to the SIZE bit is changed in the middle
- When the data written to the RADR bit is changed in the middle
- When the access size written to the SIZE bit is different from the size when accessing the actual target register
- When the address (lower 13 bits) written to the RADR bit is different from the address (lower 13 bits) when accessing the actual target register
- When the key code register and register related to the port are read while writing to the key code register

Notes:

- Key code setting might be canceled because of the DMA transfer. Read the value written in the object register, and confirm whether the value has been changed.
- While debugging by the on-chip debugger (OCD), the key code setting is canceled when the break function is executed during the key code setting.
- The DDR, PFR, EPFR, PPER, ADER, and PORTEN are the key code target registers. It is necessary to set the key code in order to execute writing.

11.5.8 Notes on Switching the I/O Port Function

Notes on switching the I/O port function are shown below.

When the I/O port is switched from the port function to peripheral function or vice versa, the PDR value may be output momentarily.

It happens if port function is changed from "input to output" or "output to input" at the time of switching.

If this output may cause a problem for the system, please write a value to PDR in advance at a level that will not cause a problem.

11.5.9 Input blocked when specific peripheral functions are used

A note regarding blocked input when specific peripheral functions are used is shown below.

When a pin is used as the A/D function and the state of the pin is read, "0" is always read.

12. Interrupt Control (Interrupt Controller)

This chapter explains the interrupt control (interrupt controller).

12.1 Overview

12.2 Features

12.3 Configuration

12.4 Registers

12.5 Operation

12.1 Overview

This section explains overview of the interrupt control (interrupt controller).

The interrupt controller performs arbitration of interrupt requests.

12.2 Features

This section explains features of the interrupt control (interrupt controller).

This module is composed of the following parts.

- ICR register
- Interrupt priority determination circuit
- Interrupt level and interrupt vector generation circuit

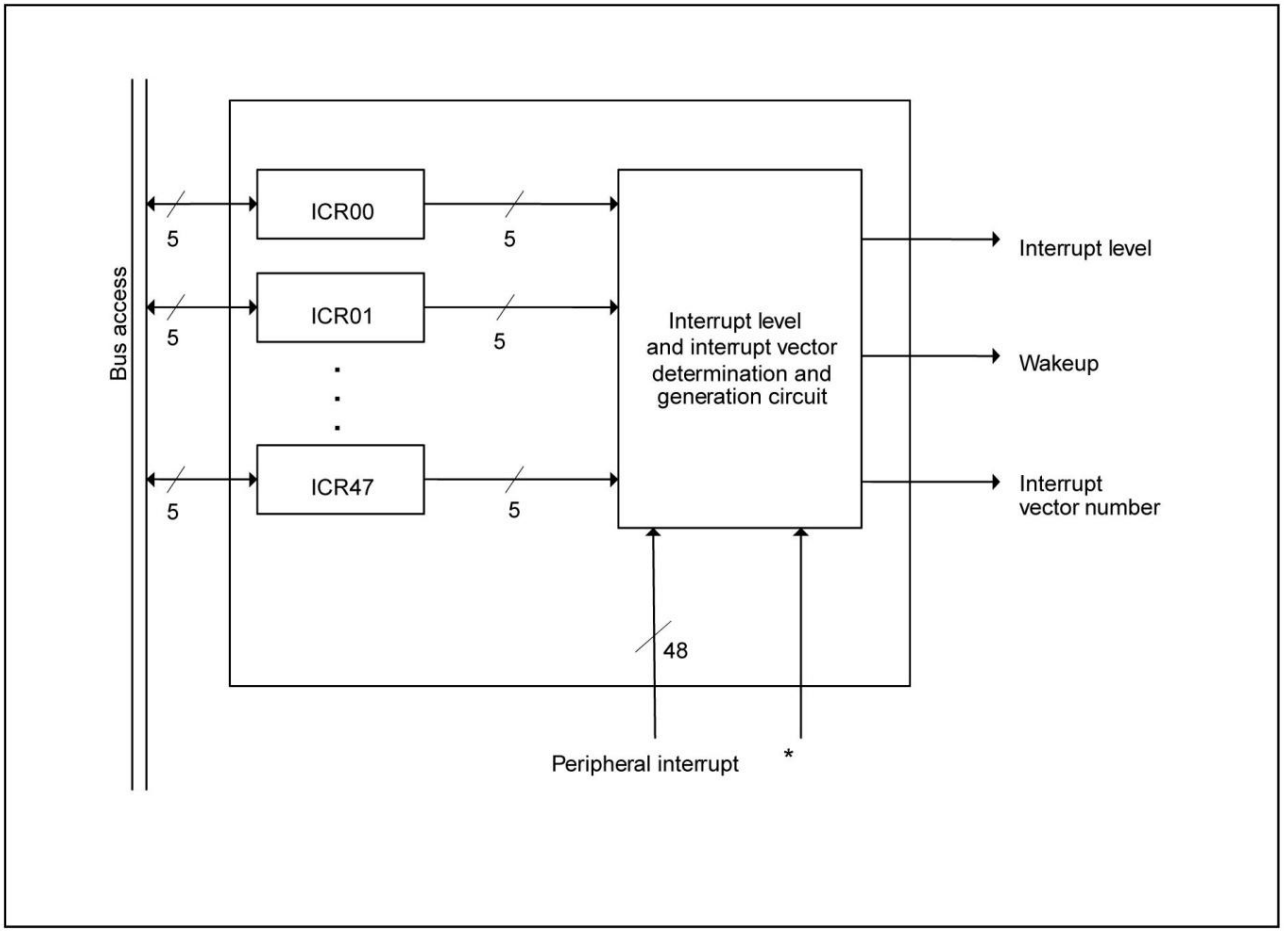
This module has the following functions.

- Detecting NMI requests and peripheral interrupt requests
- Priority determination (by level and interrupt vector)
- Transmitting the interrupt level of the factor with the highest priority to the CPU
- Transmitting the interrupt vector number of the factor with the highest priority to the CPU
- Generating wakeup requests by NMI / interrupts that occur with a level other than "11111"

12.3 Configuration

This section explains the configuration of the interrupt control (interrupt controller).

Figure 12-1. Block Diagram



*: NMI or (XBS RAM double bit error generation) or TPU violation or Error generation at internal bus diagnosis.

12.4 Registers

This section explains the registers of the interrupt control (interrupt controller).

Table 12-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0440	ICR00	ICR01	ICR02	ICR03	Interrupt control registers 00 to 47
0x0444	ICR04	ICR05	ICR06	ICR07	
0x0448	ICR08	ICR09	ICR10	ICR11	
0x044C	ICR12	ICR13	ICR14	ICR15	
0x0450	ICR16	ICR17	ICR18	ICR19	
0x0454	ICR20	ICR21	ICR22	ICR23	
0x0458	ICR24	ICR25	ICR26	ICR27	
0x045C	ICR28	ICR29	ICR30	ICR31	
0x0460	ICR32	ICR33	ICR34	ICR35	
0x0464	ICR36	ICR37	ICR38	ICR39	
0x0468	ICR40	ICR41	ICR42	ICR43	
0x046C	ICR44	ICR45	ICR46	ICR47	

12.4.1 Interrupt Control Registers 00 to 47: ICR00 to ICR47 (Interrupt Control Register 00 to 47)

The bit configuration of the interrupt control registers 00 to 47 is shown below.

1 register is provided for each interrupt input to set the level for the corresponding interrupt request.

ICR00-47: Address 0440-046F$_H$ (Access: Byte, Half-word, Word)

bit	7	6	5	4	3	2	1	0
	Reserved			IL[4:0]				
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W

[bit4 to bit0] IL[4:0] (Interrupt Level control): Interrupt level control

The interrupt level setting bits specify the interrupt level for the corresponding interrupt request. An interrupt request is masked in the CPU if the interrupt level set in these registers is greater than or equal to the level mask value in the ILM register of the CPU. These bits are initialized to "5'b11111" on reset.

The correspondence between the configurable interrupt level settings bits and the interrupt levels is shown below.

IL[4:0]	Interrupt Level		
10000	16	Configurable highest level	
10001	17	↑	(High)
10010	18	|	
10011	19	|	
10100	20	|	
10101	21	|	
10110	22	|	
10111	23	|	
11000	24	|	
11001	25	|	
11010	26	|	
11011	27	|	
11100	28	|	
11101	29	|	
11110	30	↓	(Low)
11111	31	Interrupts disabled	

IL4 is fixed at 1. Writing has no effect.

12.5 Operation

This section explains the operation of the interrupt control (interrupt controller).

12.5.1 Setting
12.5.2 Starting
12.5.3 Determining Priorities
12.5.4 Recovering From Stop Mode

12.5.1 Setting

This section explains the setting of the interrupt control (interrupt controller).

1. Configure the ICR register of the interrupt vector number corresponding to the peripheral for which you want to generate the interrupt.
2. Configure the peripheral where you want to generate the interrupt. (Configure interrupt output as enabled on the peripheral.)

12.5.2 Starting

This section explains the starting of the interrupt control (interrupt controller).

Start the configured peripheral.

12.5.3 Determining Priorities

The determining priorities are shown below.

This module selects the highest priority interrupt among interrupt factors that occur simultaneously and outputs the interrupt level and interrupt vector number for the interrupt factors to the CPU.

The criteria for determining the priority of interrupt factors are as follows.

1. NMI
2. Factors that meet the following conditions
   - ☐ If the value of the interrupt level is not 31 (5'b11111). (31 indicates interrupts disabled.)
   - ☐ The factors where the value of the interrupt level is the smallest
   - ☐ When the interrupt level is the same (except for 31), the factors that has the smallest interrupt vector number from amongst these

If no interrupt factors is selected by the above criteria, 31 (5'b11111) is output as the interrupt level. The interrupt vector number at this time is undefined.

12.5.4 Recovering From Stop Mode

The recovering from stop mode is shown below.

The function for using an interrupt request to recover from stop mode is performed by this module. If an interrupt request (the interrupt level is anything other than "5’b11111") is generated from a peripheral (including NMI), a request is generated to the clock control unit to recover from stop mode.
As the interrupt priority judgment unit restarts operation once the clock supply starts after recovery from stop mode, the CPU is able to execute instructions until the interrupt priority judgment unit produces a result.
For interrupts that are not used as sources for recovering from stop mode, set the interrupt level of the corresponding interrupt control registers (ICR00 to ICR47) to "5’b11111" (interrupts disabled).

13. External Interrupt Input

This chapter explains the external interrupt input.

13.1 Overview
13.2 Features
13.3 Configuration
13.4 Registers
13.5 Operation
13.6 Setting
13.7 Q&A
13.8 Notes

13.1 Overview

This section explains the overview of the external interrupt input.

Interrupt request input from external interrupt input pins (INT0 to INT3).

13.2 Features

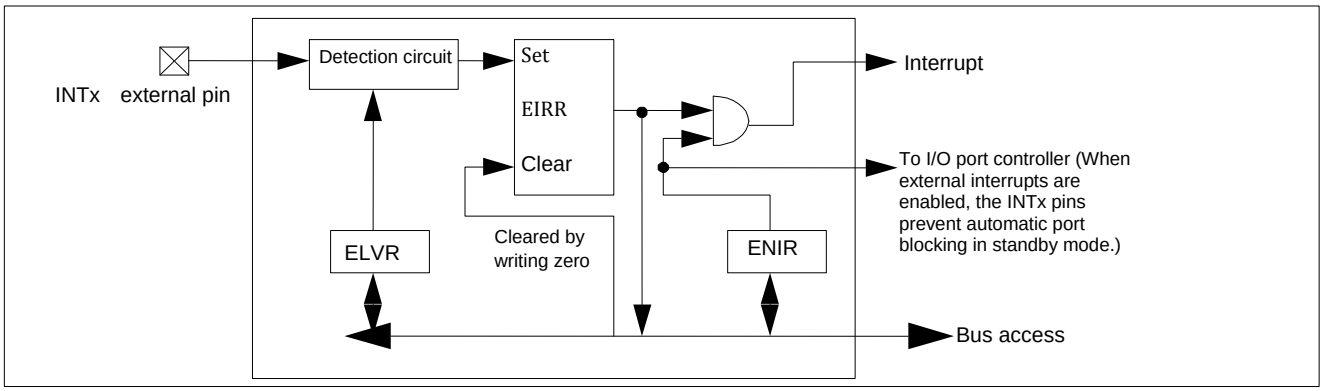
This section explains features of the external interrupt input.

- 4 types of external interrupt input pins (INT0 to INT3)
- Interrupt detection factors:4 types: ("L" level, "H" level, rising edge, and falling edge)

13.3 Configuration

This section explains the configuration of the external interrupt input.

Figure 13-1. Block Diagram



13.4 Registers

This section explains registers of the external interrupt input.

Channel	Base_addr	External Pins
		INT
0	0x0550	INT0
1	0x0550	INT1
2	0x0550	INT2
3	0x0550	INT3

Table 13-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0550	EIRR0	ENIR0	ELVR0		External interrupt factor register 0 External interrupt enable register 0 External interrupt request level register 0
0x0554	Reserved	Reserved	Reserved	Reserved	Reserved (Be sure to write "0".)

13.4.1 External Interrupt Factor Register 0: EIRR0 (External Interrupt Request Register 0)

The bit configuration of external interrupt factor register 0 (EIRR0) is shown below.

This register holds information that an external interrupt factor has been generated.

EIRR0: Address 0550_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	ER3	ER2	ER1	ER0
Initial Value	X	X	X	X	X	X	X	X
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W

[bit7 to bit4] Reserved bits

Be sure to write "0" to these bits.

[bit3 to bit0] ER3 to ER0 (External interrupt Request3-0): External interrupt request bits

Flags to indicate that there is an interrupt request by INT external pin input. Writing "0" will clear it.

ERn	Meaning	
	Read	Write
0	No external interrupt request	Clear
1	External interrupt request exists.	Do not influence operation.

Notes:

- EIRR0:ER0 corresponds to INT0 pin, EIRR0:ER1 to INT1 pin, EIRR0:ER2 to INT2 pin, EIRR0:ER3 to INT3 pin.
- Writing "0" to this bit has no meaning.
- The values read with read-modify-write (RMW) instructions will always be "1".
- When external interrupt detection condition is at "L" level or "H" level, the corresponding bit will be set again if the external interrupt pin input is at an active level after clearing each bit in the EIRR register.
- The factor bit in the interrupt factor register may be set by changing interrupt request level register. Initialize the interrupt factor register after changing the interrupt request level register.
- The value after resetting this register depends on the pin state after the reset.
- This register will be initialized by all reset factors.

13.4.2 External Interrupt Enable Register 0: ENIR0 (ENable Interrupt request Register 0)

The bit configuration of external interrupt enable register 0 (ENIR0) is shown below.

This register enables external interrupt inputs.

ENIR0: Address 0551_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	EN3	EN2	EN1	EN0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved bits

Be sure to write "0" to these bits.

[bit3 to bit0] EN3 to EN0 (interrupt ENable): External interrupt enable bits

These bits perform mask controls of interrupt requests from external pin INT inputs.

ENn	Operations at the Detection of an External Pin
0	Interrupt request mask. Hold interrupt requests but does not output them. (Initial value)
1	Interrupt request enabled. Enable interrupt requests.

Notes:

- ENIR0:EN0 corresponds to INT0 pin, ENIR0:EN1 to INT1 pin, ENIR0:EN2 to INT2 pin, ENIR0:EN3 to INT3 pin.
- This register will be initialized by all reset factors.

13.4.3 External Interrupt Request Level Register 0: ELVR0 (External interrupt LeVel Register 0)

The bit configuration of external interrupt request level register 0 (ELVR0) is shown below.

This register selects detection conditions for external interrupt requests.

ELVR0: Address 0552_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit8] Reserved bits

Be sure to write "0" to these bits.

[bit7 to bit1] LB3 to LB0 (Level select B): Level select B

[bit6 to bit0] LA3 to LA0 (Level select A): Level select A

These bits select detection conditions for external interrupt requests. Combination of 2 bits, LA bit and LB bit will be used.

LBn	LAn	Detection Conditions
0	0	"L" level detection (Initial value)
0	1	"H" level detection
1	0	Rising edge detection
1	1	Falling edge detection

When the request input is a level (LAn, LBn="00" or "01"), the corresponding bit (ERn) will turn back to "1" if INTn pin input is still in the effective levels after setting the external interrupt request bit (ERn) to "0".

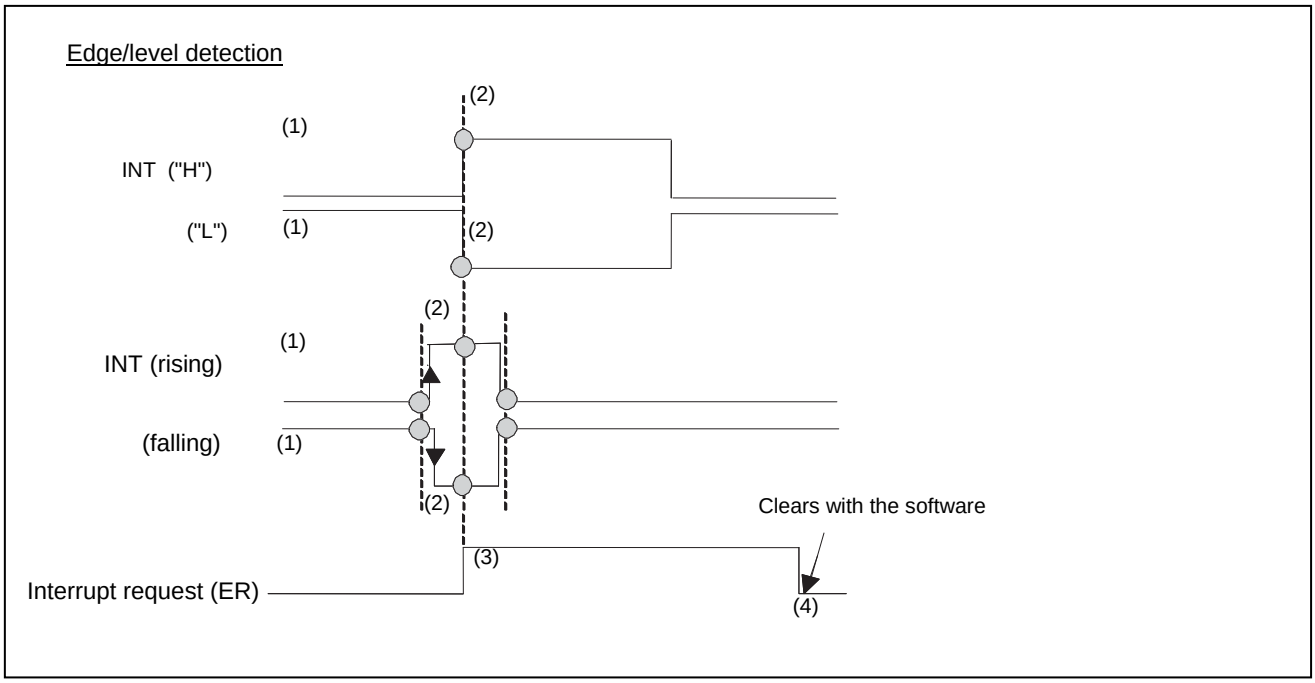
Notes:

- ELVR0:LA/LB0 corresponds to INT0 pin, ELVR0:LA/LB1 to INT1 pin, ELVR0:LA/LB2 to INT2 pin, ELVR0:LA/LB3 to INT3 pin.
- The factor bit in the interrupt factor register may be set by changing the interrupt request level register. Initialize the interrupt factor register after changing the interrupt request level register.
- This register will be initialized by all reset factors.

13.5 Operation

This section explains the operation of the external interrupt input.

Figure 13-2. Operation Diagram



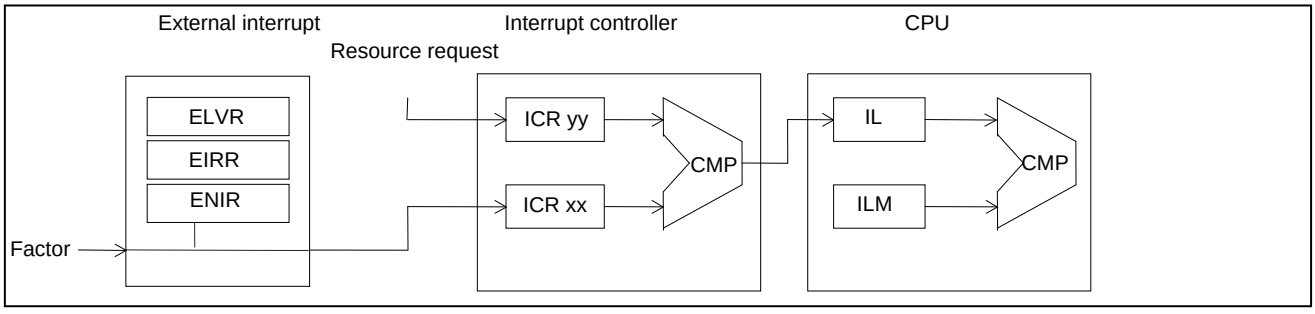
(1) External interrupt signal (INT) input.

(2) Detects interrupt signals (level/edge).

(3) Generates interrupt requests.

(4) Clears interrupt requests with the software.

Figure 13-3. Operation of External Interrupt



1. Operation of external interrupt
   This module generates the interrupt request signal to the interrupt controller when a request set in the ELVR register is input in the corresponding pin after setting a request level and the enable register. The corresponding interrupt will be generated when the interrupt from this peripheral function was found to have the highest priority in the result for examining the priority in interrupts concurrently occurred in the interrupt controller.

2. Transition to standby mode
   Channels not to be used should be moved to disable state before letting them go into the standby mode. The external pins are blocked for input at the standby mode. The external pins have input blocked during standby mode. However, for the channels with external interrupts enabled, the external pins are input-enabled.

3. Setting procedure of external interrupts
   When setting registers which reside in the external interrupt unit, follow the steps shown below.

   (1) Disable the corresponding bit for the enable register.
   (2) Set the corresponding bit for the request level setting register.
   (3) Read the request level register.
   (4) Clear the corresponding bit for the factor register.
   (5) Enable the corresponding bit for the enable register.
   (Note that concurrent writes of 16-bit data are allowed in step (4) and (5).)

   The enable register must be disabled before you can set the registers in this module. The factor register must be cleared before you can set the enable register to enable state.

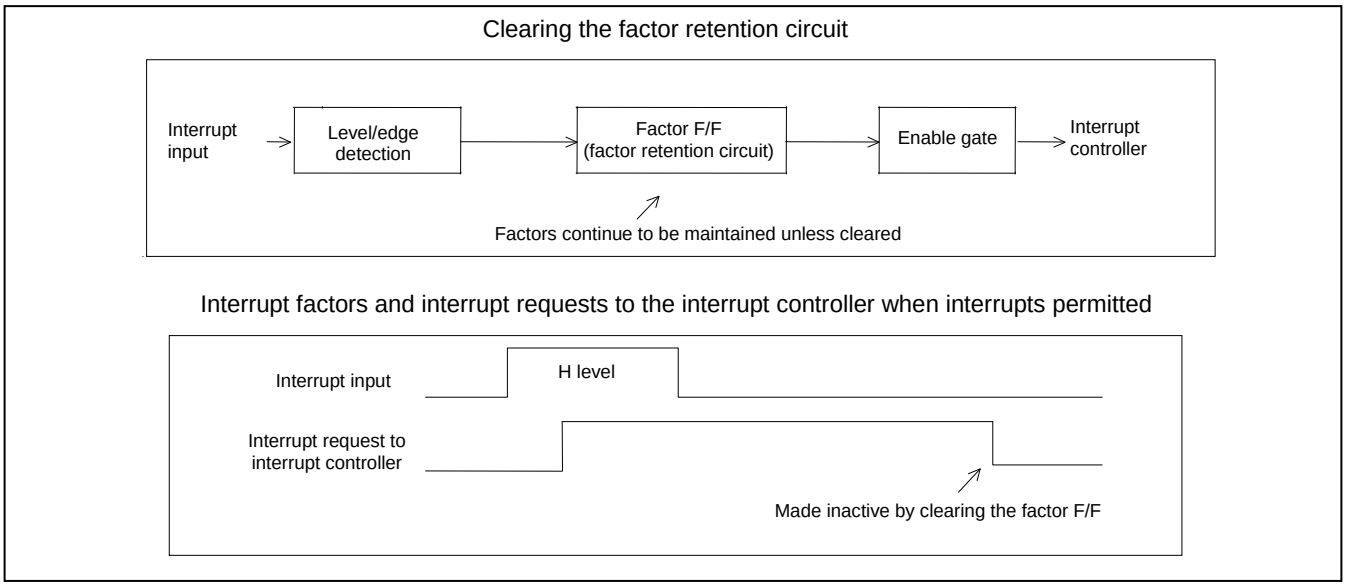
   This has to be done to avoid generating erroneous interrupt factors at the time of setting register or in interrupt enable state.

4. External interrupt factor

   Requests to the interrupt controller will continue to be active although a request input from outside is canceled, because there is an internal factor retention circuit.

   To cancel requests going toward the interrupt controller, the factor register should be cleared.

Figure 13-4. Clearing the Factor Retention Circuit and Interrupt Factor and Interrupt Request to Interrupt Controller in Interrupt Enable State



13.6 Setting

This section explains settings of the external interrupt input.

Table 13-2. Necessary Settings for Using External Interrupts

Settings	Setting Register	Setting Method
Detection level settings	External interrupt request level setting register (ELVR0)	See "About detection levels and their setting procedures" in "13.7 Q&A".
Specifying external pins to be used for input.	See "Chapter: I/O Ports".	See "Chapter: I/O Ports".
External interrupt	An input from the external pin -> Input signal to pins INT0 to INT3	-

13.7 Q&A

This section explains Q&A of the external interrupt input.

About detection levels and their setting procedures

4 levels: ("L" level, "H" level, rising edge, falling edge)

Set the detection level bits as follows: (ELVR0:LBn, LAn) (n=0 to 3).

Operation Modes	Detection Level Bits (LBn, LAn) n=0 to 3
To perform "L" level detection	Set "00".
To perform "H" level detection	Set "01".
To perform rising edge detection	Set "10".
To perform falling edge detection	Set "11".

How to make external pins to use for input

See "Chapter: I/O Ports".

About interrupt related registers

See "Chapter: Interrupt Control (Interrupt Controller)".

About interrupt types

Interrupt factors are only for external interrupts. There are no select bits.

How to enable/disable/clear interrupts

Interrupt request enable flag, interrupt request flag

Interrupt enable setting is done by the interrupt enable bit (ENIR0:EN0 to EN3).

Operation	Interrupt Enable Bit (ENn)
To disable interrupt requests	Set "0".
To enable interrupt requests	Set "1".

Interrupt request clear is done by the interrupt request bit (EIRR0:ER0 to ER3).

Operation	Interrupt Request Bit (ERn)
To clear interrupt requests	Write "0".

13.8 Notes

This section explains the notes of the external interrupt input.

14. NMI Input

This chapter explains the NMI input.

14.1 Overview
14.2 Features
14.3 Configuration
14.4 Register
14.5 Operation
14.6 Usage Example

14.1 Overview

This section explains the overview of the NMI input.

NMI (Non Maskable Interrupt) is the non-maskable interrupt signal that is entered from the NMIX pin. The NMI can be used as a source for recovering from stop mode.

14.2 Features

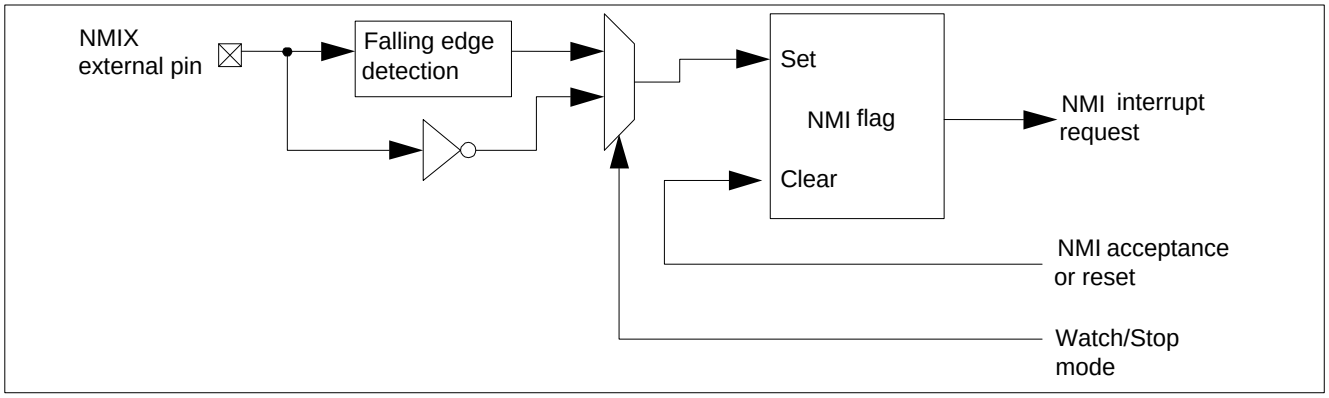
This section explains features of the NMI input.

Can be used in both stop mode and watch mode.

14.3 Configuration

This section explains the configuration of the NMI input.

Figure 14-1. Block Diagram



14.4 Register

This section explains the register of the NMI input.

This function has no register.

14.5 Operation

This section explains the operation of the NMI input.

NMI interrupt level

The NMI has the highest level among the user interrupts and cannot be masked. As an exception, the NMI is masked after reset until the ILM is set by the CPU.

NMI external pin

In stop mode, this pin detects the L level, and at other times it detects the falling edge.

Interrupt request output

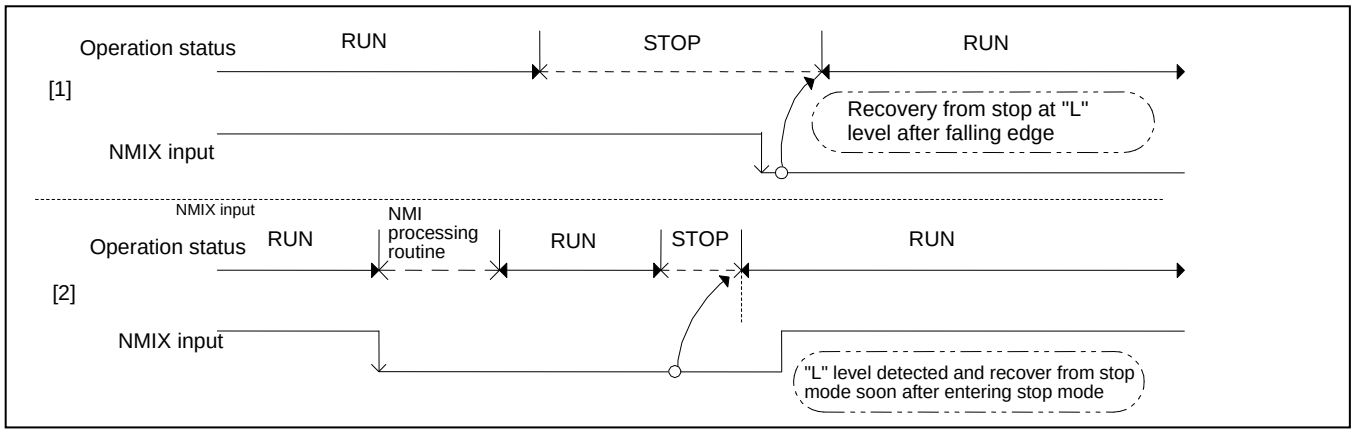
The NMI request detector has an NMI flag that is set for an NMI request and is cleared only if an interrupt for the NMI itself is accepted or reset occurs. The NMI flag cannot be read or written.

Read IRPR15H register to judge whether the NMI is caused by the NMIX external pin or the other factors. For details of this register, see "Interrupt Request Batch Read".

Recovering from stop mode

When switching to stop mode, if an "L" level is input to the NMIX, an NMI request is output to the interrupt controller and the CPU recovers from stop mode. If the CPU switches to stop mode without returning the input level of the NMIX pin to the "H" level after the NMI processing routine has finished in normal mode (not stop mode), the CPU recovers immediately after switching to stop mode (see [2] in Figure 14-2). Return the input level of the NMIX pin to the "H" level before entering stop mode so that the input level of the NMIX pin is set to the "L" level in stop mode.

Figure 14-2. Recovering from Stop Mode



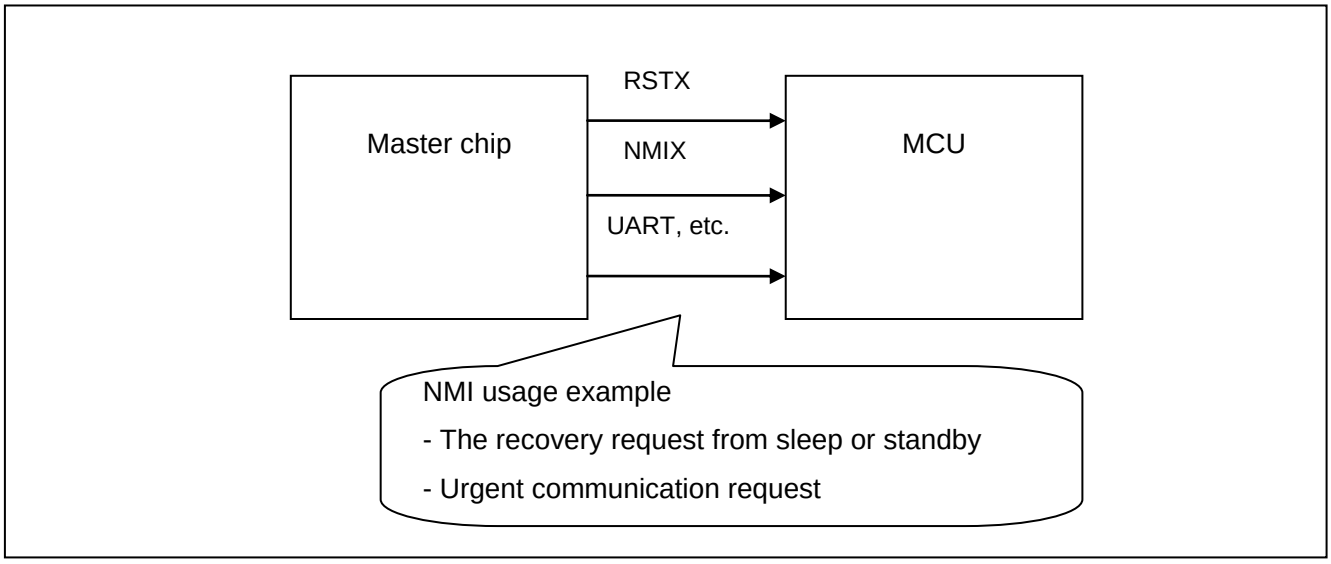
Note: The watch mode is similarly controlled.

14.6 Usage Example

This section explains a usage example of the NMI input.

This section gives an example of using the NMI function.

Figure 14-3. Usage Example



15. Delay Interrupt

This chapter explains the delay interrupt.

15.1 Overview
15.2 Features
15.3 Configuration
15.4 Registers
15.5 Operation
15.6 Restrictions

15.1 Overview

This section explains the overview of the delay interrupt.

The delay interrupt is a function for generating interrupts for the OS (operating system) to switch between tasks.

This function allows interrupt requests to the CPU to be generated and canceled by software.

15.2 Features

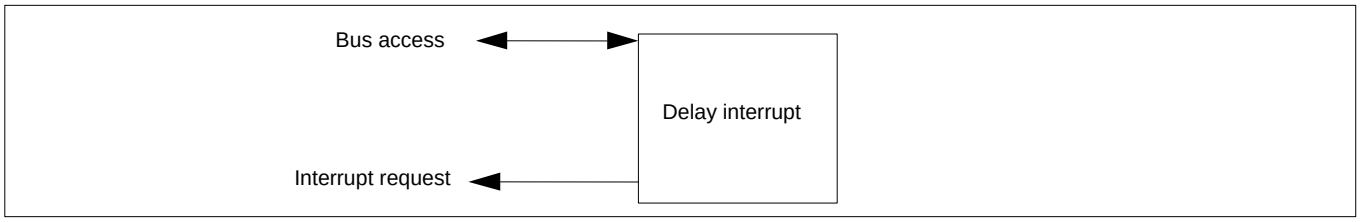
This section explains features of the delay interrupt.

The delay interrupt can be generated by writing to a register.

15.3 Configuration

This section explains the configuration of the delay interrupt.

Figure 15-1. Block Diagram



15.4 Registers

This section explains registers of the delay interrupt.

Address	Registers				Register function
	+0	+1	+2	+3	
0x0044	DICR	Reserved	Reserved	Reserved	Delay Interrupt Control Register

Delay Interrupt Control Register: DICR (Delay Interrupt Control Register)

This register controls the delay interrupts.

DICR: Address 0044_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							DLYI
Initial value	1	1	1	1	1	1	1	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W

[bit0] DLYI (DeLaY Interrupt enable): Delay Interrupt Enable Bit

This bit generates and clears the delay interrupt factor.

DLYI	Description
"0" write	Clears the delay interrupt factor.
"1" write	Generates the delay interrupt factor.

15.5 Operation

This section explains the operation description of the delay interrupt.

The delay interrupts are used to generate interrupts for task switching. Using this function allows interrupt requests to the CPU to be generated and canceled by software.

Interrupt Vector Number

The delay interrupts are allocated to the interrupt sources with the highest interrupt vector number.

In this core, delay interrupts are allocated to interrupt vector number 63 (0x3F).

DLYI Bit of the DICR Register

Writing "1" to this bit generates a delay interrupt factor. Writing "0" to this bit cancels the delay interrupt source.

This bit functions like a standard interrupt factor flag and should be cleared in the interrupt routine at the same time as when switching a task.

15.6 Restrictions

This section explains restrictions of the delay interrupt.

Do not use delay interrupts for DMA transfer requests.

16. Interrupt Request Batch Read

This chapter explains the overview, features, and configuration of the interrupt request batch read.

16.1 Overview

16.2 Features

16.3 Configuration

16.4 Registers

16.5 Operation

16.1 Overview

This section explains the overview of the interrupt request batch read.

This module can read multiple interrupt requests assigned to one interrupt vector number in a batch. Interrupt requests that have been generated can be identified by using the bit search instruction of the FR80-family CPU.

16.2 Features

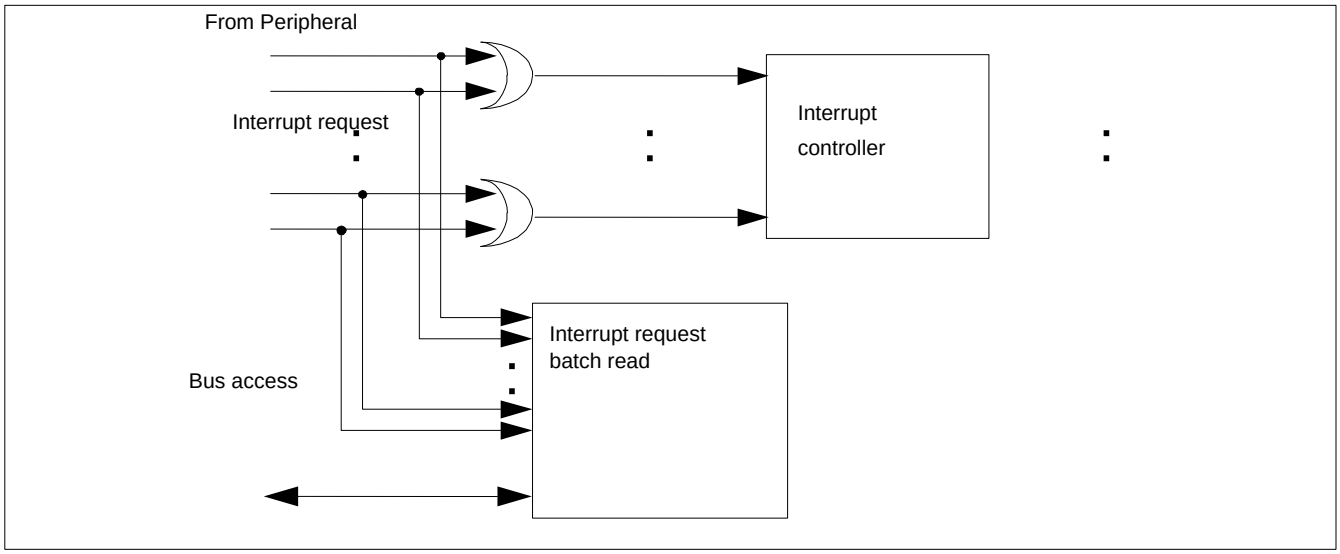
This section shows features of the interrupt request batch read.

Using this module, you can easily check whether interrupts have been generated.

16.3 Configuration

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

Figure 16-1. Block Diagram



16.4 Registers

This section explains the registers of the interrupt request batch read.

Table 16-1. Registers Map

Address	Registers				Register Functions
	+0	+1	+2	+3	
0x0418	IRPR0H	IRPR0L	IRPR1H	IRPR1L	Interrupt request batch read register 0 upper-order (#18) Interrupt request batch read register 0 lower-order (#19) Interrupt request batch read register 1 upper-order (#20) Interrupt request batch read register 1 lower-order (#22)
0x041C	IRPR2H	IRPR2L	IRPR3H	IRPR3L	Interrupt request batch read register 2 upper-order (#38) Interrupt request batch read register 2 lower-order (#39) Interrupt request batch read register 3 upper-order (#40) Interrupt request batch read register 3 lower-order (#41)
0x0420	IRPR4H	IRPR4L	IRPR5H	IRPR5L	Interrupt request batch read register 4 upper-order (#42) Interrupt request batch read register 4 lower-order (#43) Interrupt request batch read register 5 upper-order (#44) Interrupt request batch read register 5 lower-order (#24)
0x0424	IRPR6H	IRPR6L	IRPR7H	IRPR7L	Interrupt request batch read register 6 upper-order (#50) Interrupt request batch read register 6 lower-order (#46) Interrupt request batch read register 7 upper-order (#47) Interrupt request batch read register 7 lower-order (#48)
0x0428	IRPR8H	IRPR8L	IRPR9H	IRPR9L	Interrupt request batch read register 8 upper-order (#52) Interrupt request batch read register 8 lower-order (#53) Interrupt request batch read register 9 upper-order (#54) Interrupt request batch read register 9 lower-order (#54)
0x042C	IRPR10H	IRPR10L	IRPR11H	IRPR11L	Interrupt request batch read register 10 upper-order (#56) Interrupt request batch read register 10 lower-order (#56) Interrupt request batch read register 11 upper-order (#55) Interrupt request batch read register 11 lower-order (#57)
0x0430	IRPR12H	IRPR12L	IRPR13H	IRPR13L	Interrupt request batch read register 12 upper-order (#58) Interrupt request batch read register 12 lower-order (#59) Interrupt request batch read register 13 upper-order (#60) Interrupt request batch read register 13 lower-order (#61)
0x0434	IRPR14H	Reserved	IRPR15H	IRPR15L	Interrupt request batch read register 14 upper-order (#62) Interrupt request batch read register 15 upper-order (#15) Interrupt request batch read register 15 upper-order (#35)

#nn: Interrupt vector number (decimal)

16.4.1 Interrupt Request Batch Read Register 0 Upper-order: IRPR0H (Interrupt Request Peripheral Read register 0H)

The bit configuration of the interrupt request batch read register 0 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #18)

IRPR0H: Address 0418_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RTIR0	RTIR1	RTIR4	Reserved				
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RTIR0 (Reload Timer Interrupt Request 0): Reload timer 0 interrupt request

[bit6] RTIR1 (Reload Timer Interrupt Request 1): Reload timer 1 interrupt request

[bit5] RTIR4 (Reload Timer Interrupt Request 4): Reload timer 4 interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.2 Interrupt Request Batch Read Register 0 Lower-order: IRPR0L (Interrupt Request Peripheral Read register 0L)

The bit configuration of the interrupt request batch read register 0 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #19)

IRPR0L: Address 0419_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RTIR2	RTIR3	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RTIR2 (Reload Timer Interrupt Request 2): Reload timer 2 interrupt request

[bit6] RTIR3 (Reload Timer Interrupt Request 3): Reload timer 3 interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.3 Interrupt Request Batch Read Register 1 Upper-order: IRPR1H (Interrupt Request Peripheral Read register 1H)

The bit configuration of the interrupt request batch read register 1 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #20)

IRPR1H: Address $041A_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RXIR0	ISIR0	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RXIR0 (Multi-Function-Serial-Interface RX Interrupt Request 0): Multifunction serial interface ch.0 reception completion interrupt request

[bit6] ISIR0 (Multi-Function-Serial-Interface Status Interrupt Request 0): Multifunction serial interface ch.0 status interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.4 Interrupt Request Batch Read Register 1 Lower-order: IRPR1L (Interrupt Request Peripheral Read register 1L)

The bit configuration of the interrupt request batch read register 1 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #22)

IRPR1L: Address $041B_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RXIR1	ISIR1	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RXIR1 (Multi-Function-Serial-Interface RX Interrupt Request 1): Multi-function serial interface ch.1 reception completion interrupt request

[bit6] ISIR1 (Multi-Function-Serial-Interface Status Interrupt Request 1): Multi-function serial interface ch.1 status interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.5 Interrupt Request Batch Read Register 2 Upper-order: IRPR2H (Interrupt Request Peripheral Read register 2H)

The bit configuration of the interrupt request batch read register 2 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #38)

IRPR2H: Address $041C_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWMSEIR0	PWMSEIR1	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWMSEIR0 (PWM Special Event Interrupt Request 0): PWM special event interrupt request 0

[bit6] PWMSEIR1 (PWM Special Event Interrupt Request 1): PWM special event interrupt request 1

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.6 Interrupt Request Batch Read Register 2 Lower-order: IRPR2L (Interrupt Request Peripheral Read register 2L)

The bit configuration of the interrupt request batch read register 2 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #39)

IRPR2L: Address 041D$_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FRTZIR	FRTMIR	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] FRTZIR (Free-run Timer Zero Interrupt Request): Free-run timer 0 detection interrupt request

[bit6] FRTMIR (Free-run Timer Compare Clear Interrupt Request): Free-run timer compare clear interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.7 Interrupt Request Batch Read Register 3 Upper-order: IRPR3H (Interrupt Request Peripheral Read register 3H)

The bit configuration of the interrupt request batch read register 3 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #40)

IRPR3H: Address $041E_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWMZIR0	PWMMIR0	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWMZIR0 (PWM Zero Interrupt Request 0): PWM 0 detection interrupt request 0

[bit6] PWMMIR0 (PWM Compare Match Interrupt Request 0): PWM compare clear interrupt request 0

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.8 Interrupt Request Batch Read Register 3 Lower-order: IRPR3L (Interrupt Request Peripheral Read register 3L)

The bit configuration of the interrupt request batch read register 3 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #41)

IRPR3L: Address 041F$_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWMZIR1	PWMMIR1	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWMZIR1 (PWM Zero Interrupt Request 1): PWM 0 detection interrupt request 1

[bit6] PWMMIR1 (PWM Compare Match Interrupt Request 1): PWM compare clear interrupt request 1

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.9 Interrupt Request Batch Read Register 4 Upper-order: IRPR4H (Interrupt Request Peripheral Read register 4H)

The bit configuration of the interrupt request batch read register 4 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #42)

IRPR4H: Address 0420_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWMSIR0	PWMFIR0	PWMFIR1	PWMCIR0	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWMSIR0 (PWM SOW Interrupt Request 0): PWM SOW interrupt request 0

[bit6] PWMFIR0 (PWM Fault Interrupt Request 0): PWM fault interrupt request 0

[bit5] PWMFIR1 (PWM Fault Interrupt Request 1): PWM fault interrupt request 1

[bit4] PWMCIR0 (PWM Capture Interrupt Request 0): PWM capture interrupt request 0

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.10 Interrupt Request Batch Read Register 4 Lower-order: IRPR4L (Interrupt Request Peripheral Read register 4L)

The bit configuration of the interrupt request batch read register 4 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #43)

IRPR4L: Address 0421_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWMSIR1	PWMFIR2	PWMFIR3	PWMCIR1	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWMSIR1 (PWM SOW Interrupt Request 1): PWM SOW interrupt request 1

[bit6] PWMFIR2 (PWM Fault Interrupt Request 2): PWM fault interrupt request 2

[bit5] PWMFIR3 (PWM Fault Interrupt Request 3): PWM fault interrupt request 3

[bit4] PWMCIR1 (PWM Capture Interrupt Request 1): PWM capture interrupt request 1

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.11 Interrupt Request Batch Read Register 5 Upper-order: IRPR5H (Interrupt Request Peripheral Read register 5H)

The bit configuration of the interrupt request batch read register 5 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #44)

IRPR5H: Address 0422_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWMSIR2	PWMFIR4	PWMFIR5	PWMCIR2	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWMSIR2 (PWM SOW Interrupt Request 2): PWM SOW interrupt request 2

[bit6] PWMFIR4 (PWM Fault Interrupt Request 4): PWM fault interrupt request 4

[bit5] PWMFIR5 (PWM Fault Interrupt Request 5): PWM fault interrupt request 5

[bit4] PWMCIR2 (PWM Capture Interrupt Request 2): PWM capture interrupt request 2

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.12 Interrupt Request Batch Read Register 5 Lower-order IRPR5L (Interrupt Request Peripheral Read register 5L)

The bit configuration of the interrupt request batch read register 5 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #24)

IRPR5L: Address 0423_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RXIR2	ISIR2	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] RXIR2 (Multi-Function-Serial-Interface RX Interrupt Request 2): Multi-function serial interface ch.2 reception completion interrupt request

[bit6] ISIR2 (Multi-Function-Serial-Interface Status Interrupt Request 2): Multi-function serial interface ch.2 status interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.13 Interrupt Request Batch Read Register 6 Upper-order: IRPR6H (Interrupt Request Peripheral Read register 6H)

The bit configuration of the interrupt request batch read register 6 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #50)

IRPR6H: Address 0424_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	OVF_IRQ	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit6] OVF_IRQ (PWM PLL Overflow Interrupt Request): PWM PLL alarm interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.14 Interrupt Request Batch Read Register 6 Lower-order: IRPR6L (Interrupt Request Peripheral Read register 6L)

The bit configuration of the interrupt request batch read register 6 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #46)

IRPR6L: Address 0425_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MTIR	Reserved	PTIR	Reserved	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R0,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] MTIR (Main Timer Interrupt Request): Main timer interrupt request

[bit5] PTIR (PLL Timer Interrupt Request): PLL timer interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.15 Interrupt Request Batch Read Register 7 Upper-order: IRPR7H (Interrupt Request Peripheral Read register 7H)

The bit configuration of the interrupt request batch read register 7 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #47)

IRPR7H: Address 0426_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWMTIR0	PWMTIR1	PWMTIR2	PWMTIR3	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWMTIR0 (PWM Trigger Interrupt Request 0): PWM trigger interrupt request 0

[bit6] PWMTIR1 (PWM Trigger Interrupt Request 1): PWM trigger interrupt request 1

[bit5] PWMTIR2 (PWM Trigger Interrupt Request 2): PWM trigger interrupt request 2

[bit4] PWMTIR3 (PWM Trigger Interrupt Request 3): PWM trigger interrupt request 3

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.16 Interrupt Request Batch Read Register 7 Lower-order: IRPR7L (Interrupt Request Peripheral Read register 7L)

The bit configuration of the interrupt request batch read register 7 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #48)

IRPR7L: Address 0427_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADCIR0	ADCIR1	ADCIR2	ADCIR3	ADCIR4	ADCIR5	ADCIR6	ADCIR7
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7] ADCIR0 (AD Converter Interrupt Request 0): A/D converter 0 conversion completion interrupt request

[bit6] ADCIR1 (AD Converter Interrupt Request 1): A/D converter 1 conversion completion interrupt request

[bit5] ADCIR2 (AD Converter Interrupt Request 2): A/D converter 2 conversion completion interrupt request

[bit4] ADCIR3 (AD Converter Interrupt Request 3): A/D converter 3 conversion completion interrupt request

[bit3] ADCIR4 (AD Converter Interrupt Request 4): A/D converter 4 conversion completion interrupt request

[bit2] ADCIR5 (AD Converter Interrupt Request 5): A/D converter 5 conversion completion interrupt request

[bit1] ADCIR6 (AD Converter Interrupt Request 6): A/D converter 6 conversion completion interrupt request

[bit0] ADCIR7 (AD Converter Interrupt Request 7): A/D converter 7 conversion completion interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.17 Interrupt Request Batch Read Register 8 Upper-order: IRPR8H (Interrupt Request Peripheral Read register 8H)

The bit configuration of the interrupt request batch read register 8 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #52)

IRPR8H: Address 0428_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CMPIR0	CMPIR1	CMPIR2	Reserved				
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] CMPIR0 (CMP Output Interrupt Request 0): Comparator output detection interrupt 0

[bit6] CMPIR1 (CMP Output Interrupt Request 1): Comparator output detection interrupt 1

[bit5] CMPIR2 (CMP Output Interrupt Request 2): Comparator output detection interrupt 2

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.18 Interrupt Request Batch Read Register 8 Lower-order: IRPR8L (Interrupt Request Peripheral Read register 8L)

The bit configuration of the interrupt request batch read register 8 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #53)

IRPR8L: Address 0429_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWCZIR 00	PWCZIR 10	PWCMIR 00	PWCMIR 10	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWCZIR00 (PWC Zero Interrupt Request 00): PWC 0 detection interrupt request 00

[bit6] PWCZIR10 (PWC Zero Interrupt Request 10): PWC 0 detection interrupt request 10

[bit5] PWCMIR00 (PWC Compare Match Interrupt Request 00): PWC compare clear interrupt request 00

[bit4] PWCMIR10 (PWC Compare Match Interrupt Request 10): PWC compare clear interrupt request 10

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.19 Interrupt Request Batch Read Register 9 Upper-order: IRPR9H (Interrupt Request Peripheral Read register 9H)

The bit configuration of the interrupt request batch read register 9 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #54)

IRPR9H: Address 042A$_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWCCUIR00	PWCCUIR10	PWCCUIR20	PWCCUIR30	PWCCLIR00	PWCCLIR10	PWCCLIR20	PWCCLIR30
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7] PWCCUIR00 (PWC Capture Data Upper Limit Interrupt Request 00): PWC capture data upper limit interrupt request 00

[bit6] PWCCUIR10 (PWC Capture Data Upper Limit Interrupt Request 10): PWC capture data upper limit interrupt request 10

[bit5] PWCCUIR20 (PWC Capture Data Upper Limit Interrupt Request 20): PWC capture data upper limit interrupt request 20

[bit4] PWCCUIR30 (PWC Capture Data Upper Limit Interrupt Request 30): PWC capture data upper limit interrupt request 30

[bit3] PWCCLIR00 (PWC Capture Data Lower Limit Interrupt Request 00): PWC capture data lower limit interrupt request 00

[bit2] PWCCLIR10 (PWC Capture Data Lower Limit Interrupt Request 10): PWC capture data lower limit interrupt request 10

[bit1] PWCCLIR20 (PWC Capture Data Lower Limit Interrupt Request 20): PWC capture data lower limit interrupt request 20

[bit0] PWCCLIR30 (PWC Capture Data Lower Limit Interrupt Request 30): PWC capture data lower limit interrupt request 30

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.20 Interrupt Request Batch Read Register 9 Lower-order: IRPR9L (Interrupt Request Peripheral Read register 9L)

he bit configuration of the interrupt request batch read register 9 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #54)

IRPR9L: Address 042B$_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWCDB IR00	PWCDB IR10	PWCDB IR20	PWCDB IR30	PWCBO IR00	PWCBO IR10	PWCBO IR20	PWCBO IR30
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7] PWCDBIR00 (PWC Data Buffer Interrupt Request 00): PWC data buffer interrupt request 00

[bit6] PWCDBIR10 (PWC Data Buffer Interrupt Request 10): PWC data buffer interrupt request 10

[bit5] PWCDBIR20 (PWC Data Buffer Interrupt Request 00): PWC data buffer interrupt request 20

[bit4] PWCDBIR30 (PWC Data Buffer Interrupt Request 00): PWC data buffer interrupt request 30

[bit3] PWCBOIR00 (PWC Buffer Over Run Interrupt Request 00): PWC buffer over run interrupt request 00

[bit2] PWCBOIR10 (PWC Buffer Over Run Interrupt Request 10): PWC buffer over run interrupt request 10

[bit1] PWCBOIR20 (PWC Buffer Over Run Interrupt Request 20): PWC buffer over run interrupt request 20

[bit0] PWCBOIR30 (PWC Buffer Over Run Interrupt Request 30): PWC buffer over run interrupt request 30

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.21 Interrupt Request Batch Read Register 10 Upper-order: IRPR10H (Interrupt Request Peripheral Read register 10H)

The bit configuration of the interrupt request batch read register 10 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #56)

IRPR10H: Address 042C$_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWCCUIR01	PWCCUIR11	PWCCUIR21	PWCCUIR31	PWCCLIR01	PWCCLIR11	PWCCLIR21	PWCCLIR31
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7] PWCCUIR01 (PWC Capture Data Upper Limit Interrupt Request 01): PWC capture data upper limit interrupt request 01

[bit6] PWCCUIR11 (PWC Capture Data Upper Limit Interrupt Request 11): PWC capture data upper limit interrupt request 11

[bit5] PWCCUIR21 (PWC Capture Data Upper Limit Interrupt Request 21): PWC capture data upper limit interrupt request 21

[bit4] PWCCUIR31 (PWC Capture Data Upper Limit Interrupt Request 31): PWC capture data upper limit interrupt request 31

[bit3] PWCCLIR01 (PWC Capture Data Lower Limit Interrupt Request 01): PWC capture data lower limit interrupt request 01

[bit2] PWCCLIR11 (PWC Capture Data Lower Limit Interrupt Request 11): PWC capture data lower limit interrupt request 11

[bit1] PWCCLIR21 (PWC Capture Data Lower Limit Interrupt Request 21): PWC capture data lower limit interrupt Request 21

[bit0] PWCCLIR31 (PWC Capture Data Lower Limit Interrupt Request 31): PWC capture data Lower limit interrupt Request 31

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.22 Interrupt Request Batch Read Register 10 Lower-order: IRPR10L (Interrupt Request Peripheral Read register 10L)

The bit configuration of the interrupt request batch read register 10 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #56)

IRPR10L: Address 042D$_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWCDB IR01	PWCDB IR11	PWCDB IR21	PWCDB IR31	PWCBO IR01	PWCBO IR11	PWCBO IR21	PWCBO IR31
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7] PWCDBIR01 (PWC Data Buffer Interrupt Request 01): PWC data buffer interrupt request 01

[bit6] PWCDBIR11 (PWC Data Buffer Interrupt Request 11): PWC data buffer interrupt request 11

[bit5] PWCDBIR21 (PWC Data Buffer Interrupt Request 01): PWC data buffer interrupt request 21

[bit4] PWCDBIR31 (PWC Data Buffer Interrupt Request 01): PWC data buffer interrupt request 31

[bit3] PWCBOIR01 (PWC Buffer Over Run Interrupt Request 01): PWC buffer over run interrupt request 01

[bit2] PWCBOIR11 (PWC Buffer Over Run Interrupt Request 11): PWC buffer over run interrupt request 11

[bit1] PWCBOIR21 (PWC Buffer Over Run Interrupt Request 21): PWC buffer over run interrupt request 21

[bit0] PWCBOIR31 (PWC Buffer Over Run Interrupt Request 31): PWC buffer over run interrupt request 31

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.23 Interrupt Request Batch Read Register 11 Upper-order: IRPR11H (Interrupt Request Peripheral Read register 11H)

The bit configuration of the interrupt request batch read register 11 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #55)

IRPR11H: Address $042E_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWCZIR01	PWCZIR11	PWCMIR01	PWCMIR11	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] PWCZIR01 (PWC Zero Interrupt Request 01): PWC 0 detection interrupt request 01

[bit6] PWCZIR11 (PWC Zero Interrupt Request 11): PWC 0 detection interrupt request 11

[bit5] PWCMIR01 (PWC Compare Match Interrupt Request 01): PWC compare clear interrupt request 01

[bit4] PWCMIR11 (PWC Compare Match Interrupt Request 11): PWC compare clear interrupt request 11

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.24 Interrupt Request Batch Read Register 11 Lower-order: IRPR11L (Interrupt Request Peripheral Read register 11L)

The bit configuration of the interrupt request batch read register 11 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #57)

IRPR11L: Address 042F$_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADCIR8	ADCIR9	ADCIR10	ADCIR11	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] ADCIR8 (AD Converter Interrupt Request 8): A/D converter 8 conversion completion interrupt request

[bit6] ADCIR9 (AD Converter Interrupt Request 9): A/D converter 9 conversion completion interrupt request

[bit5] ADCIR10 (AD Converter Interrupt Request 10): A/D converter 10 conversion completion interrupt request

[bit4] ADCIR11 (AD Converter Interrupt Request 11): A/D converter 11 conversion completion interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.25 Interrupt Request Batch Read Register 12 Upper-order: IRPR12H (Interrupt Request Peripheral Read register 12H)

The bit configuration of the interrupt request batch read register 12 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #58)

IRPR12H: Address 0430_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BT2IR0	BT2IR1	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] BT2IR0 (BT2 Interrupt Request 0): Base timer ch.2 interrupt request 0

[bit6] BT2IR1 (BT2 Interrupt Request 1): Base timer ch.2 interrupt request 1

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.26 Interrupt Request Batch Read Register 12 Lower-order: IRPR12L (Interrupt Request Peripheral Read register 12L)

The bit configuration of the interrupt request batch read register 12 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #59)

IRPR12L: Address 0431_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BT3IR0	BT3IR1	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] BT3IR0 (BT3 Interrupt Request 0): Base timer ch.3 interrupt request 0

[bit6] BT3IR1 (BT3 Interrupt Request 1): Base timer ch.3 interrupt request 1

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.27 Interrupt Request Batch Read Register 13 Upper-order:IRPR13H (Interrupt Request Peripheral Read register 13H)

The bit configuration of the interrupt request batch read register 13 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #60)

IRPR13H: Address 0432_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BT0IR0	BT0IR1	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] BT0IR0 (BT0 Interrupt Request 0): Base timer ch.0 interrupt request 0

[bit6] BT0IR1 (BT0 Interrupt Request 1): Base timer ch.0 interrupt request 1

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.28 Interrupt Request Batch Read Register 13 Lower-order: IRPR13L (Interrupt Request Peripheral Read register 13L)

The bit configuration of the interrupt request batch read register 13 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #61)

IRPR13L: Address 0433_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BT1IR0	BT1IR1	Reserved	Reserved	Reserved	Reserved		
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] BT1IR0 (BT1 Interrupt Request 0): Base timer ch.1 interrupt request 0

[bit6] BT1IR1 (BT1 Interrupt Request 1): Base timer ch.1 interrupt request 1

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.29 Interrupt Request Batch Read Register 14 Upper-order: IRPR14H (Interrupt Request Peripheral Read register 14H)

The bit configuration of the interrupt request batch read register 14 upper-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #62)

IRPR14H: Address 0434_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMAC0IR	DMAC1IR	DMAC2IR	DMAC3IR	DMAC4IR	DMAC5IR	DMAC6IR	DMAC7IR
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7] DMAC0IR (DMAC 0 Interrupt Request): DMAC ch.0 interrupt request

[bit6] DMAC1IR (DMAC 1 Interrupt Request): DMAC ch.1 interrupt request

[bit5] DMAC2IR (DMAC 2 Interrupt Request): DMAC ch.2 interrupt request

[bit4] DMAC3IR (DMAC 3 Interrupt Request): DMAC ch.3 interrupt request

[bit3] DMAC4IR (DMAC 4 Interrupt Request): DMAC ch.4 interrupt request

[bit2] DMAC5IR (DMAC 5 Interrupt Request): DMAC ch.5 interrupt request

[bit1] DMAC6IR (DMAC 6 Interrupt Request): DMAC ch.6 interrupt request

[bit0] DMAC7IR (DMAC 7 Interrupt Request): DMAC ch.7 interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.30 Interrupt Request Batch Read Register 15 Upper-order: IRPR15H (Interrupt Request Peripheral Read register 15H)

The bit configuration of the interrupt request batch read register 15 upper-order is shown.

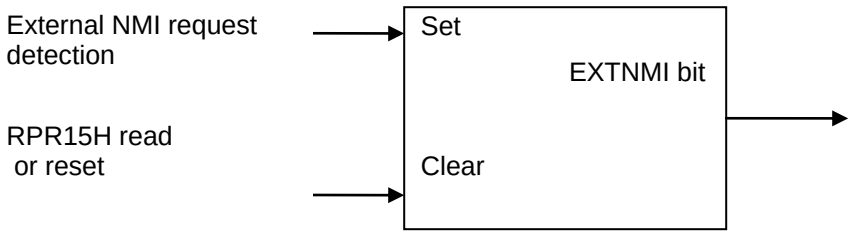
This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #15)

IRPR15H: Address 0436_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EXTNMI	XB_ECC_DE	BUS_NMI	TPU_VIO	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] EXTNMI: External NMI request

The EXTNMI bit is set by detecting external NMI request, and cleared by reading this register.



[bit6] XB_ECC_DE: XBS RAM double bit error generation interrupt request

[bit5] BUS_NMI: Bus test error interrupt request

[bit4] TPU_VIO: TPU violation interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.31 Interrupt Request Batch Read Register 15 Lower-order: IRPR15L (Interrupt Request Peripheral Read register 15L)

The bit configuration of the interrupt request batch read register 15 lower-order is shown.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #35)

IRPR15L: Address 0437_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	XBTC	XBIC	XBTE	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit6] XBTC (XBs ram Test Completed interrupt request): XBS RAM test completed interrupt request

[bit5] XBIC (XBs ram Initialization Completed interrupt request): XBS RAM initialization completed request interrupt request

[bit4] XBTE (XBs ram Test Error interrupt request): XBS RAM test error interrupt request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.5 Operation

This section explains the operation of the interrupt request batch read.

Within each interrupt handler, the pertinent register is read to determine what bits are set. As a consequence, what interrupt requests have been generated is found.

Notes:

- This register does not provide a function that can be used to input external interrupts. Read registers EIRR0, which is used to input external interrupts.
- This register does not have a function prepared that can be used for 12-bit A/D converters (4-channel simultaneous sampling). Read the AD4CS register for 12-bit A/D converters (4-channel simultaneous sampling).

17. Watchdog Timer

This chapter explains the watchdog timer.

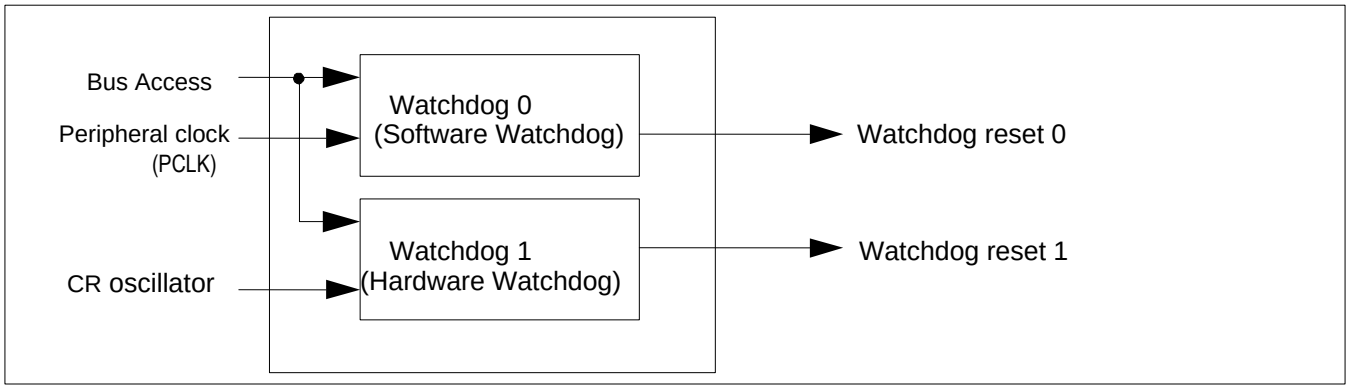
17.1 Overview
17.2 Features
17.3 Configuration
17.4 Registers
17.5 Operation
17.6 Usage Example

17.1 Overview

This section explains the overview of the watchdog timer.

This device has two watchdog timers that can detect both the states of software and hardware running out of control, and these watchdog timers can generate reset requests.

Figure 17-1. Block Diagram (Schematic)



17.2 Features

This section explains the features of the watchdog timer.

17.2.1 Watchdog Timer 0 (Software Watchdog)

17.2.2 Watchdog Timer 1 (Hardware Watchdog)

17.2.1 Watchdog Timer 0 (Software Watchdog)

This section explains the features of the watchdog timer 0.

- Stop mode detection function
  Able to detect the transition to watch mode or stop mode and generate a reset request.
- Watchdog timer clear
  The timer is cleared by operation initialization reset or by writing the inverse value of the value previously written to the clear register.
- Illegal write detection function
  If the incorrect value is written to the clear register, a reset request is generated.
- Watchdog timer period
  The period can be selected from among 16 choices of the peripheral clock (PCLK) x (2^9 to 2^{24}) cycles.
- Count stop conditions
  The count stops while the CPU is stopped.
- To set the lower limit value of the timer count of the watchdog timer.
  The value can be selected from among 16 choices of the peripheral clock (PCLK) x (2^8 to 2^{23}) cycles.
- Monitoring the watchdog timer window and generating a reset request.
  If the clear register is written below the lower limit value of the timer count of the watchdog timer, the watchdog timer generates a reset request.

17.2.2 Watchdog Timer 1 (Hardware Watchdog)

This section explains the features of the watchdog timer 1.

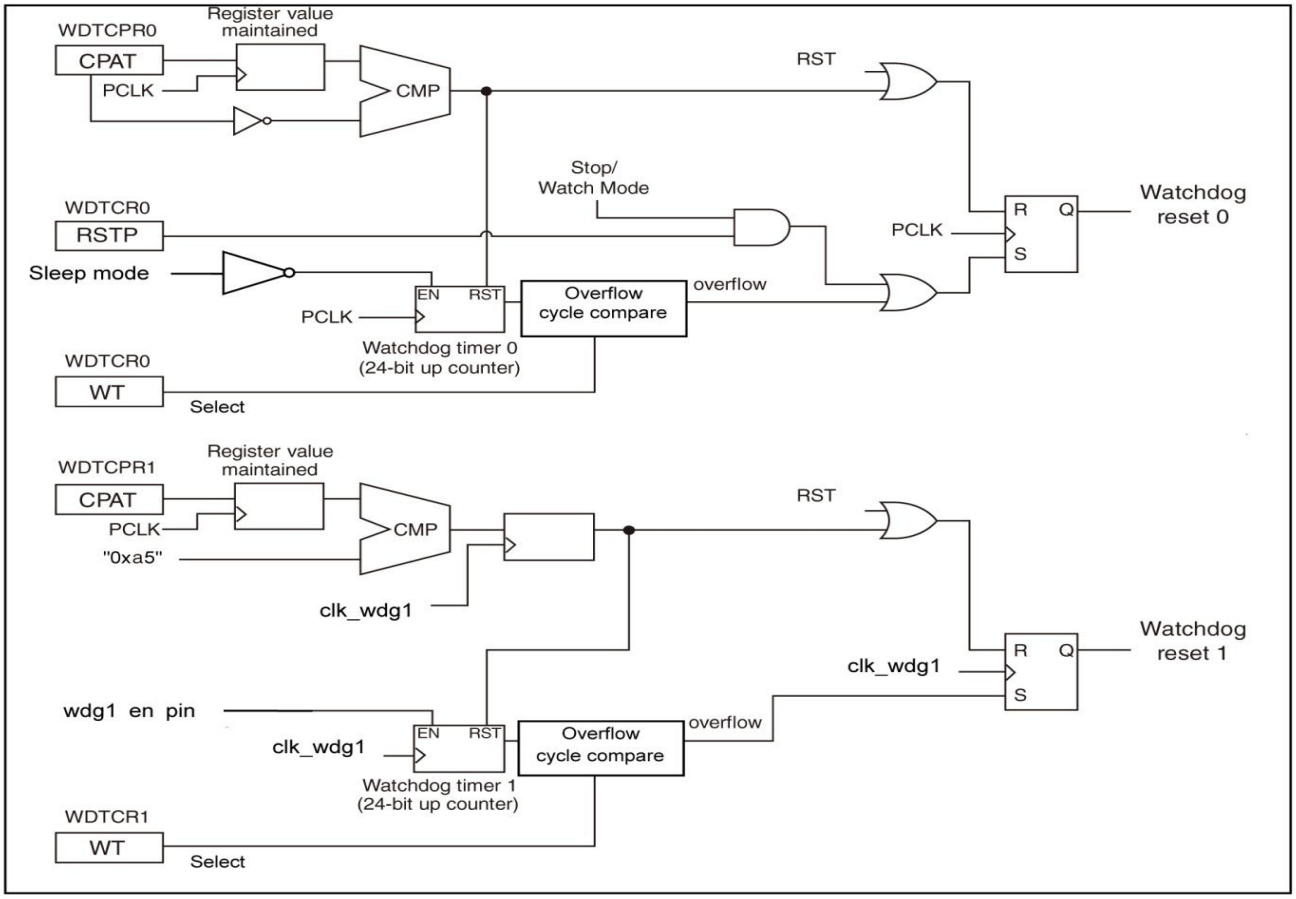
This timer is driven by the clock generated by the built-in CR oscillator circuit immediately after the reset is released. For information on settings (calibration) of the CR oscillator, see "Chapter: WDT1 Calibration".

- Watchdog timer clear
  The timer is cleared by the operation initialization reset or by writing "0xA5" to the clear register.
- Illegal write detection function
  If a value other than "0xA5" is written to the clear register, a reset request is generated.
- Watchdog timer period
  The period is fixed by the hardware at CR oscillator x 2^{15} cycles.
- Count stop conditions
  The count stops when using ICE, during sleep mode, watch mode, stop mode, and when waiting for the oscillator to stabilize when recovering from standby mode.

17.3 Configuration

This section shows the configuration of the watchdog timer.

Figure 17-2. Block Diagram (Detailed)



17.4 Registers

This section explains the registers of the watchdog timer.

Table 17-1. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0038	WDTECR0	Reserved			Watchdog timer 0 extended control register
0x003C	WDTCR0	WDTCPR0	WDTCR1	WDTCPR1	Watchdog timer 0 control register Watchdog timer 0 clear register Watchdog timer 1 cycle information register Watchdog timer 1 clear register

17.4.1 Watchdog Timer 0 Control Register: WDTCR0 (WatchDog Timer Configuration Register 0)

The bit configuration of the watchdog timer 0 control register is shown.

This register configures each of the settings of the watchdog timer 0.
Writing to this register is invalid after the watchdog timer 0 is activated.

WDTCR0: Address 003C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	RSTP	Reserved		WT[3:0]			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit7] (Reserved): (Reserved bit)

Be sure to write "0" to this bit. The read value is "0".

[bit6] RSTP (Reset by SToP): Stop mode detection reset enable

This bit configures whether a reset signal is generated or not when a transition to watch mode or stop mode is detected while the watchdog timer 0 is operating. When this bit is enabled, the watchdog timer reset 0 occurs when the CPU switches to watch mode or stop mode. When this bit is not enabled, the watchdog timer 0 is paused when the CPU switches to watch mode or stop mode, and the count stops until the CPU recovers from watch mode or stop mode.

RSTP	Stop Mode Detection
0	Not detected (initial value)
1	Generates a reset signal when detected

Writing to this bit after the watchdog timer 0 is activated is invalid.

[bit5, bit4] Reserved: (Reserved bits)

Be sure to write "0" to this bit. The read value is "0".

[bit3 to bit0] WT[3:0] (Watchdog Timer interval): Watchdog timer cycle selection

These bits configure the number of cycles of timer interval starting from when the watchdog timer 0 was last cleared to when a watchdog reset 0 is issued.

WT[3:0]	The Watchdog Timer 0 Cycle
0000	PCLK (peripheral clock) x 2^9 cycles
0001	PCLK x 2^{10} cycles
0010	PCLK x 2^{11} cycles
0011	PCLK x 2^{12} cycles
0100	PCLK x 2^{13} cycles
0101	PCLK x 2^{14} cycles
0110	PCLK x 2^{15} cycles
0111	PCLK x 2^{16} cycles
1000	PCLK x 2^{17} cycles
1001	PCLK x 2^{18} cycles
1010	PCLK x 2^{19} cycles
1011	PCLK x 2^{20} cycles
1100	PCLK x 2^{21} cycles
1101	PCLK x 2^{22} cycles
1110	PCLK x 2^{23} cycles
1111	PCLK x 2^{24} cycles

Writing to this bit after the watchdog timer 0 is activated is invalid.
The watchdog timer 0 does not count while the CPU is not operating.
Counting is performed while the CPU is operating even if DMA transfer is being performed.

17.4.2 Watchdog Timer 0 Clear Register: WDTCPR0 (WatchDog Timer Clear Pattern Register 0)

The bit configuration of the watchdog timer 0 clear register is shown.

This register activates or clears (delays issue of a reset signal) the watchdog timer 0.

WDTCPR0: Address $003D_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CPAT[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

[bit7 to bit0] CPAT[7:0] (Clear PATtern): Watchdog timer 0 clear

The watchdog timer 0 is activated by the first write to this register after the reset is released. The watchdog timer is cleared after being activated by writing a value with all of the bits inverted from the previously written value. If a value other than the inverse value of the previously written value is written, the watchdog reset 0 is issued at that time.
The value read out from this register is always "0x00" regardless of the value written.

17.4.3 Watchdog Timer 0 Extended Control Register: WDTECR0 (Watchdog Timer Extended Configuration Register 0)

The bit configuration of the watchdog timer 0 extended control register is shown.

This register configures the settings for window watching function of the watchdog timer 0.

Writing to this register is invalid after the watchdog timer 0 is activated.

WDTECR0: Address 0038_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			WTWE	WTLI[3:0]			
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W

[bit7 to bit5] Reserved: (Reserved bits)

Be sure to write "0" to this bit. The read value is "0".

[bit4] WTWE (Watchdog Timer Window Enable): Watchdog timer window function enable

This bit controls the window function of the watchdog timer 0. When the bit WTWE is set to "1" the window function becomes enabled.
The initial value of this bit is "0". (The window function is invalid.)

WTWE	Window Function Enabled
0	Window function is invalid (initial value)
1	Window function is valid

[bit3 to bit0] WTLI[3:0] (Watchdog Timer Lower Interval): Selection of the lower limit of watchdog timer

These bits configure the lower limit of the interval starting from when the watchdog timer 0 is cleared to when it is cleared next time. When the window function is valid, if a request for clearing the watchdog timer 0 comes before a lower limit of timer shown below, a watchdog reset signal is issued.

WTLI[3:0]	The Lower Limit of the Watchdog Timer
0000	PCLK (peripheral clock) x 2^8 cycles
0001	PCLK x 2^9 cycles
0010	PCLK x 2^{10} cycles
0011	PCLK x 2^{11} cycles
0100	PCLK x 2^{12} cycles
0101	PCLK x 2^{13} cycles
0110	PCLK x 2^{14} cycles
0111	PCLK x 2^{15} cycles
1000	PCLK x 2^{16} cycles
1001	PCLK x 2^{17} cycles
1010	PCLK x 2^{18} cycles
1011	PCLK x 2^{19} cycles
1100	PCLK x 2^{20} cycles
1101	PCLK x 2^{21} cycles
1110	PCLK x 2^{22} cycles
1111	PCLK x 2^{23} cycles

Set the watchdog timer below the period specified with WTCR0.WT[3:0]. If a period larger than that specified with WTCR0.WT[3:0] is set, a reset signal is generated. This is because the watchdog timer is satisfied with the condition to be cleared below the lower limit of the window even though the timer is cleared before overflow.

17.4.4 Watchdog Timer 1 Cycle Information Register: WDTCR1 (WatchDog Timer Cycle information Register 1)

The bit configuration of the watchdog timer 1 cycle information register is shown.

This register configures each of the settings of the watchdog timer 1.

WDTCR1: Address $003E_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				WT[3:0]			
Initial value	0	0	0	0	0	1	1	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R1,WX	R1,WX	R0,WX

This register cannot be written again.

[bit7 to bit4] Reserved: (Reserved bits)

The value "0" is always read. Writing to these bits has no meaning.

[bit3 to bit0] WT[3:0] (Watchdog Timer interval): Watchdog timer cycle selection

These bits configure the number of cycles of timer interval starting from when the watchdog timer 1 was last cleared to when a watchdog reset 1 is issued. The cycle is fixed to 2^{15} cycles. Writing to these bits are invalid.

WT[3:0]	The Watchdog Timer 1 Cycle
0110	CR oscillator x 2^{15} cycles (initial value, fixed)

17.4.5 Watchdog Timer 1 Clear Register: WDTCPR1 (WatchDog Timer Clear Pattern Register 1)

The bit configuration of the watchdog timer 1 clear register is shown.

This register clears watchdog timer 1 (delays issue of a reset signal).

WDTCPR1: Address 003F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CPAT[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W

[bit7 to bit0] CPAT[7:0] (Clear PATtern): Watchdog timer 1 clear

Watchdog timer 1 activates after the reset is released. The watchdog timer is cleared after being activated by writing "0xA5". When a value other than "0xA5" is written, the watchdog reset 1 is issued at that time. The value read out from this register is always "0x00" regardless of the value written.

17.5 Operation

This section explains the operation of the watchdog timer.

17.5.1 Software Watchdog Function

17.5.2 Hardware Watchdog Function

17.5.1 Software Watchdog Function

This section explains the software watchdog function.

17.5.1.1 Settings

17.5.1.2 Activation

17.5.1.3 Operation

17.5.1.1 Settings

This section explains the settings of the software watchdog function.

Before activating the watchdog timer 0, set bit3 to 0:WT[3:0] of the WDTCR0 register in order to select the period starting from clearing the watchdog timer to issuing the reset request.

Since the watchdog timer 0 counts only when the CPU is operating, set the period on the basis of the number of program steps and the clock division setting.

Before activating the watchdog timer 0, set bit6:RSTP of the WDTCR0 register in order to select whether or not to generate a reset signal when a transition to watch mode or stop mode is detected.

- When RSTP="0", the timer stops in watch mode or stop mode.
- When RSTP="1", a reset signal is generated as soon as the CPU enters watch mode or stop mode.

If the device is used in watch mode or stop mode, set RSTP="0". Writing to the RSTP bit is invalid after the watchdog timer 0 is activated.

17.5.1.2 Activation

This section explains the activation of the software watchdog function.

The watchdog timer 0 is activated by the first write of any data to the WDTCPR0 register after reset.

There is no restriction on the data written.

The value "0x00" is always read out from the WDTCPR0 register regardless of any data written.

17.5.1.3 Operation

This section explains the operation of the software watchdog function.

The operation of the watchdog timer 0 after activation is explained.

Counting Conditions

The watchdog timer 0 counts the rising edges of the peripheral clock (PCLK) while the CPU is operating.

DMA transfer does not influence the watchdog timer 0 to count.

As in sleep mode, the watchdog timer 0 stops counting only while the CPU is being stopped. Since sampling of operating state of the CPU is done by the peripheral clock (PCLK), a change in the operating state of the CPU occurring within the period of the peripheral clock is ignored.

When the watchdog timer 0 is connected with ICE, the timer stops counting under the following conditions:

- In emulator mode
- In the debug interface functions, if the watchdog reset suppression function is enabled

Under any conditions mentioned above, when the watchdog timer 0 stops counting it pauses without clearing the counter. Hence, when the watchdog timer 0 resumes counting the timer will continue counting from the previous count.

Because the peripheral clock stops during the oscillation stabilization wait time of the source clock, the watchdog timer 0 also stops counting.

Clearing the Timer

Once the watchdog timer 0 is activated, the timer must be cleared before the timer period has elapses.

Clearing the watchdog timer 0 is performed by writing data to the WDTCPR0 register. These data written must be the inverted values of all bits of the WDTCPR0 that was written previously.

When the watchdog timer 0 is activated with the set value "0x55", for example, written to the WDTCPR0 register, the timer is cleared in the following way.
After activation of the watchdog timer 0, the set value should be written alternately like "0xAA" then "0x55" then "0xAA" then "0x55".

Since the read value of the WDTCPR0 register is always "0x00", the previously written value cannot be determined by reading WDTCPR0. For this reason, if the previously written value cannot be stored in other location, write to the register two times consecutively in a single clear.

When the window function is effective during the watching period, clear the timer within a period of time while the counter can be cleared effectively.

Reset Request Generation

The watchdog timer 0 generates a watchdog reset request under the following conditions.

- An overflow of the configured watchdog timer cycle occurs.
- There is a transition to watch mode or to stop mode while stop mode detection reset is enabled.
- A value, other than the inverted value of the value which is previously written, is written to the clear register.
- Writing to the clear register within the lower limit of the watching period of the window function.

17.5.2 Hardware Watchdog Function

This section explains the operation of the hardware watchdog function.

17.5.1.1 Settings

17.5.1.2 Activation

17.5.1.3 Operation

17.5.2.1 Settings

This section explains the settings of the hardware watchdog function.

The values set to those bits from bit3 to bit0:WT[3:0] of the WDTCR1 register of the watchdog timer 1 are fixed with hardware.

17.5.2.2 Activation

This section explains the activation of the hardware watchdog function.

The watchdog timer 1 is activated immediately after the reset is released.

17.5.2.3 Operation

This section explains the operation of the hardware watchdog function.

The operation of the watchdog timer 1 after activation is explained.

Counting Conditions

The watchdog timer 1 counts the rising edges of the CR oscillation.

When the watchdog timer 1 is connected with ICE, the timer stops counting under the following conditions:

- In emulator mode
- In the debug interface functions, if the watchdog reset suppression function is enabled

The watchdog timer 1 stops counting in sleep mode, watch mode, stop mode, and during the oscillation stabilization wait time recovering from standby mode.

Clearing the Timer

Once the watchdog timer 1 is activated, the timer must be cleared before the timer period has elapses.

The watchdog timer 1 is cleared when the value "0xA5" is written to the WDTCPR1 register.

Reset Request Generation

The watchdog timer 1 generates a watchdog reset request under the following conditions.

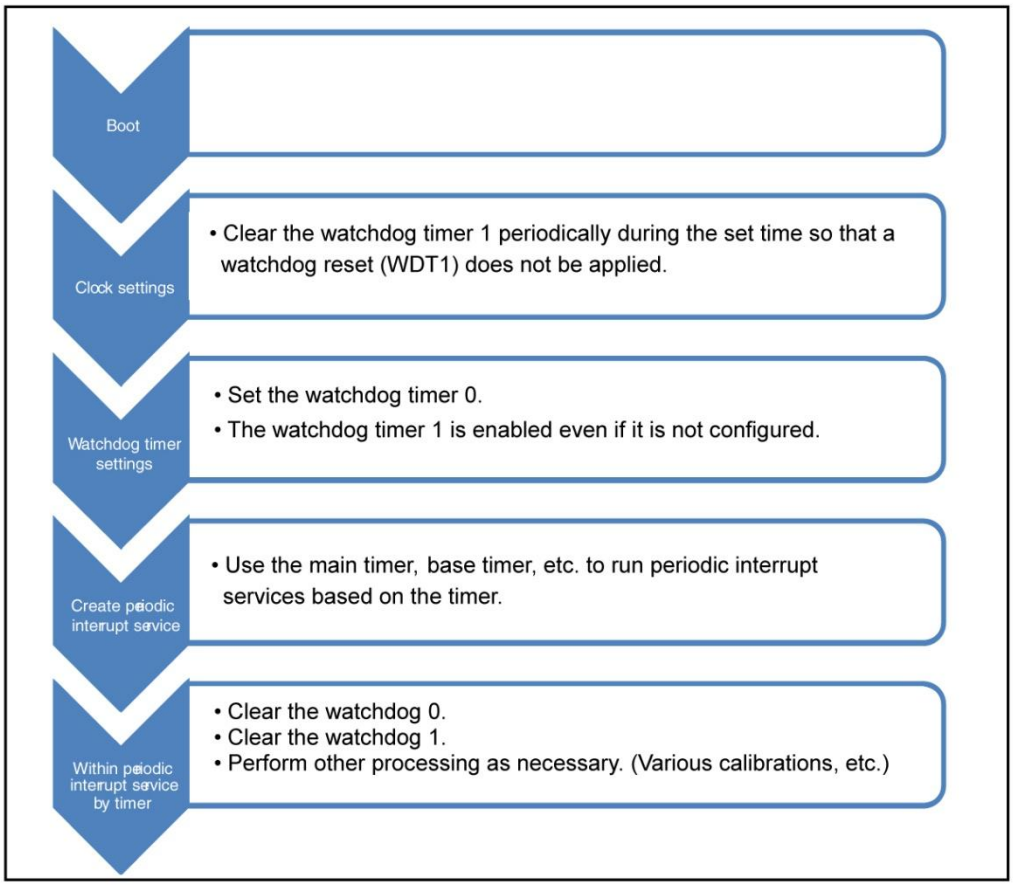
- An overflow of the watchdog timer cycle occurs.
- A value other than "0xA5" is written to the WDTCPR1 register.

17.6 Usage Example

This section gives an example of how the watchdog timer is used.

This example shows how to clear the watchdog timer.

Figure 17-3. Example of Clearing the Watchdog Timers



18. Base Timer

This chapter explains the base timer.

18.1 Overview

18.2 Features

18.3 Configuration

18.4 Registers

18.5 Operation

18.1 Overview

This section explains the overview of the base timer.

This series includes the base timer for max 4 channels. These base timers provide the following functions:

- 16/32-bit reload timer
- 16-bit PWM timer
- 16-bit PPG timer
- 16/32-bit PWC timer

18.2 Features

This section explains features of the base timer.

This series includes the base timer for 4 channels. Each channel selects and uses appropriate ones of the following functions:

18.2.1 16/32-bit Reload Timer

18.2.2 16-bit PWM Timer

18.2.3 16/32-bit PWC Timer

18.2.4 16-bit PPG Timer

18.2.1 16/32-bit Reload Timer

This section explains the 16/32-bit reload timer of the base timer.

A base timer can be used as a 16/32-bit reload timer. The 16/32-bit reload timer is a timer that decreases from a preset value.

I/O mode

You can select a signal (external clock, external activation trigger, waveform) I/O operation using the base timer I/O selection function.

Timer mode

You can run multiple timers for individual channels and can combine 16-bit reload timers for 2 channels into one 32-bit reload timer.

Operation mode

You can select one of the following two modes:

- Reload mode: In this mode, when the down counter underflows, the preset value (cycle) is reloaded to allow the timer to restart counting.
- One-shot mode: Once the down counter underflows, the counter will no longer count.

Count clock

You can select one of five internal (peripheral) clocks and three external clocks (ECK signals).

- Internal clock (peripheral clock): Clock obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, or 2048.
- External clock (ECK signal): Rising edges, falling edges, or both edges are detected.

Activation trigger

One of the following can be selected:

- Software trigger
- External event: Rising edge, falling edge, or both edges
- 16/32-bit reload timer reactivation: The 16/32-bit reload timer can be reactivated when an activation trigger is detected during counting.

Interrupt request

An interrupt request can be generated in one of the following events:

- IRQ0: When an underflow occurs
- IRQ1: When a 16/32-bit reload timer activation trigger is detected

18.2.2 16-bit PWM Timer

This section explains the 16-bit PWM timer of the base timer.

The 16-bit PWM timer, PWM standing for Pulse Width Modulator, produces a desired waveform at an external pin when a duty ratio of the pulse width is specified.

I/O mode

You can select a signal (external clock, external activation trigger, waveform) I/O operation using the base timer I/O selection function.

Operation mode

You can select one of the following two:

- Reload mode: In this mode, when the 16-bit down counter underflows, the preset cycle is reloaded to allow the timer to restart counting.
- One-shot mode: Once the 16-bit down counter underflows, the counter will no longer count.

Count clock

You can select one of five internal (peripheral) clocks and three external clocks (ECK signals).

- Internal clock (peripheral clock): Clock obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, or 2048.
- External clock (ECK signal): Rising edges, falling edges, or both edges are detected.

Activation trigger

One of the following can be selected:

- Software trigger
- Three external events: (Rising edge, falling edge, or both edges detection)

16-bit PWM timer reactivation

The 16-bit PWM timer can be reactivated when an activation trigger is detected during counting.

Output waveform

The output signal from the external pin can be fixed at the "L" or "H" level.

Interrupt request

An interrupt request can be generated in one of the following events:

- IRQ0: When an underflow occurs or counting is performed up to a preset value (duty)
- IRQ1: When a 16-bit PWM timer activation trigger is detected

18.2.3 16/32-bit PWC Timer

This section explains the 16/32-bit PWC timer of the base timer.

The 16/32-bit PWC timer, PWC standing for Pulse Width Counter, is used to measure pulse widths or cycles.

I/O mode

You can select a signal (waveform) I/O operation using the base timer I/O selection function.

Timer mode

You can run multiple timers for individual channels and can combine 16-bit PWC timers for 2 channels into one 32-bit PWC timer.

Operation mode

You can select one of the following two modes:

- Single measurement mode: In this mode, measurement is conducted only once.
- Continuous measurement mode: In this mode, after one sequence of measurement is conducted, the input of the next measurement start edge is awaited and the detection of the next measurement start edge triggers another sequence of measurement.

Count clock

You can select one of the internal (peripheral) clocks obtained by dividing the frequency of the peripheral clock (PCLK) by five types.

- Clocks obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, and 2048.

Measurement mode

You can select one of the following five options relating to the pulse width and cycle to be measured:

- "H" pulse width: Duration in which the input signal is maintained at the "H" level
- "L" pulse width: Duration in which the input signal is maintained at the "L" level
- Rising edge interval: Period from the detection of a rising edge to the detection of the next rising edge
- Falling edge interval: Period from the detection of a falling edge to the detection of the next falling edge
- Edge-to-edge pulse width: The width between consecutive input edges is one of the following periods:
  - ☐ Period from the detection of a rising edge to the detection of the falling edge
  - ☐ Period from the detection of a falling edge to the detection of the rising edge

16/32-bit PWC timer reactivation

The 16/32-bit PWC timer can be reactivated when an activation trigger is detected during counting.

Interrupt request

An interrupt request can be generated in one of the following events:

- IRQ0: When an overflow occurs
- IRQ1: When measurement ends

18.2.4 16-bit PPG Timer

This section explains the 16-bit PPG timer of the base timer.

The 16-bit PPG timer, PPG standing for Programmable Pulse Generator, is a timer that generates a waveform with a desired pulse width.

I/O mode

You can select a signal (external clock, external activation trigger, waveform) I/O operation using the base timer I/O selection function.

Operation mode

You can select one of the following two modes:

- Reload mode: A sequence of "L"-level and "H"-level signals (consecutive pulses) is output.
- One-shot mode: A string of one "L"-level signal and one "H"-level signal (single pulses) is output.

Count clock

You can select one of five internal (peripheral) clocks and three external clocks (ECK signals).

- Internal clock (peripheral clock): Clock obtained by dividing the frequency of the peripheral clock (PCLK) by 1, 4, 16, 128, 256, 512, 1024, or 2048.
- External clock (ECK signal): Rising edges, falling edges, or both edges are detected.

Activation trigger

One of the following can be selected:

- Software trigger
- Three external events: (Rising edge, falling edge, or both edges detection)

16-bit PPG timer reactivation

The 16-bit PPG timer can be reactivated when an activation trigger is detected during counting.

Interrupt request

An interrupt request can be generated in one of the following events:

- IRQ0: When an underflow occurs based on the value of the base timer x H width setting reload register (BTxPRLH).
- IRQ1: When a 16-bit PPG timer activation trigger is detected.

18.3 Configuration

This section explains the configuration of the base timer.

Figure 18-1. Block Diagram Channel 0 and Channel 1 (Overview)

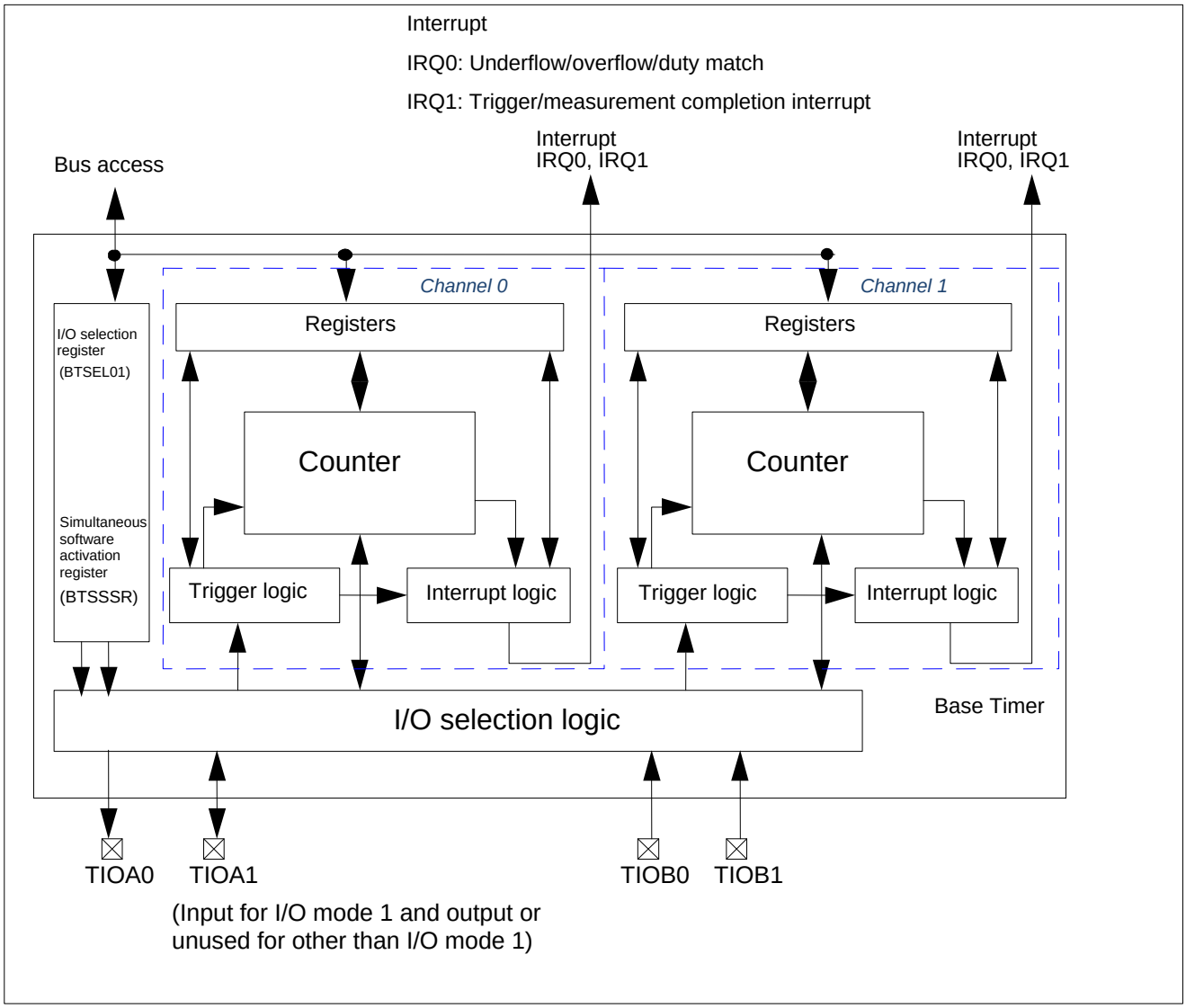
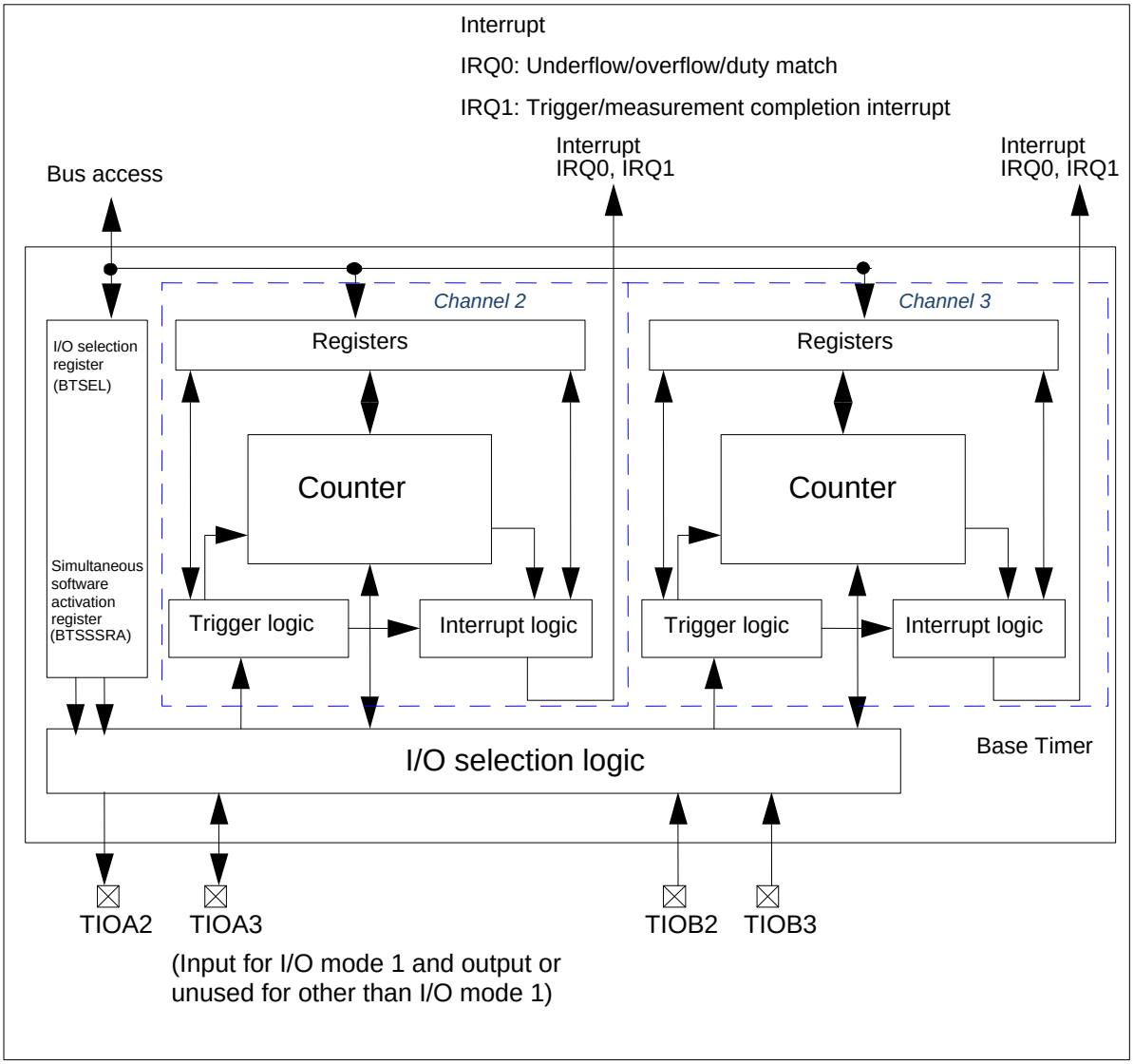


Figure 18-2. Block Diagram Channel 2 and Channel 3 (Overview)



18.4 Registers

This section explains registers of the base timer.

Table 18-1. Table of Base Addresses (Base_addr) and External Pins

Channel Number	Base Address	External Pin
0	0x0080	TIOA0, TIOA1, TIOB0, and TIOB1 are assigned based on the BTSEL01 register setting.
1	0x0090	
2	0x0118	TIOA2, TIOA3, TIOB2, and TIOB3 are assigned based on the BTSEL23 register setting.
3	0x0124	

Table 18-2. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0080	[Common] BT0TMR		[Common] BT0TMCR		[Common] Timer register 0 [Common] Control register 0
0x0084	[Common] BT0TMCR2	[Reload timer] BT0STC [PWM] BT0STC [PPG] BT0STC [PWC] BT0STC	Reserved		[Common] Control register 20 [Reload timer] Status control register 0 [PWM] Status control register 0 [PPG] Status control register 0 [PWC] Status control register 0
0x0088	[Reload timer] BT0PCSR [PWM] BT0PCSR [PPG] BT0PRLL [PWC] Reserved		[Reload timer] Reserved [PWM] BT0PDUT [PPG] BT0PRLH [PWC] BT0DTBF		[Reload timer] Cycle setting register 0 [PWM] Cycle setting register 0 [PPG] L width setting reload register 0 [PWM] Duty setting register 0 [PPG] H width setting reload register 0 [PWC] Data buffer register 0
0x008C	Reserved				
0x0090	[Common] BT1TMR		[Common] BT1TMCR		[Common] Timer register 1 [Common] Control register 1
0x0094	[Common] BT1TMCR2	[Reload Timer] BT1STC [PWM] BT1STC [PPG] BT1STC [PWC] BT1STC	Reserved		[Common] Control register 21 [Reload timer] Status control register 1 [PWM] Status control register 1 [PPG] Status control register 1 [PWC] Status control register 1
0x0098	[Reload Timer] BT1PCSR [PWM] BT1PCSR [PPG] BT1PRLL [PWC] Reserved		[Reload timer] Reserved [PWM] BT1PDUT [PPG] BT1PRLH [PWC] BT1DTBF		[Reload Timer] Cycle setting register 1 [PWM] Cycle setting register 1 [PPG] L width setting reload register 1 [PWM] Duty setting register 1 [PPG] H width setting reload register 1 [PWC] Data buffer register 1
0x009C	BTSEL01	Reserved	BTSSSR		I/O selection register Simultaneous software activation register

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0118	[Common] BT2TMR		[Common] BT2TMCR		[Common] Timer register 2 [Common] Control register 2
0x011C	[Common] BT2TMCR2	[Reload Timer] BT2STC [PWM] BT2STC [PPG] BT2STC [PWC] BT2STC	Reserved		[Common] Control register 22 [Reload timer] Status control register 2 [PWM] Status control register 2 [PPG] Status control register 2 [PWC] Status control register 2
0x0120	[Reload timer] BT2PCSR [PWM] BT2PCSR [PPG] BT2PRLL [PWC] Reserved		[Reload timer] Reserved [PWM] BT2PDUT [PPG] BT2PRLH [PWC] BT2DTBF		[Reload timer] Cycle setting register 2 [PWM] Cycle setting register 2 [PPG] L width setting reload register 2 [PWM] Duty setting register 2 [PPG] H width setting reload register 2 [PWC] Data buffer register 2
0x0124	[Common] BT3TMR		[Common] BT3TMCR		[Common] Timer register 3 [Common] Control register 3
0x0128	[Common] BT3TMCR2	[Reload timer] BT3STC [PWM] BT3STC [PPG] BT3STC [PWC] BT3STC	Reserved		[Common] Control register 23 [Reload timer] Status control register 3 [PWM] Status control register 3 [PPG] Status control register 3 [PWC] Status control register 3
0x012C	[Reload timer] BT3PCSR [PWM] BT3PCSR [PPG] BT3PRLL [PWC] Reserved		[Reload timer] Reserved [PWM] BT3PDUT [PPG] BT3PRLH [PWC] BT3DTBF		[Reload timer] Cycle setting register 3 [PWM] Cycle setting register 3 [PPG] L width setting reload register 3 [PWM] Duty setting register 3 [PPG] H width setting reload register 3 [PWC] Data buffer register 3
0x0130	BTSEL23	Reserved	BTSSSRA		I/O selection register Simultaneous software activation register

18.4.1 Common Registers

This section explains the common registers of the base timer.

The registers described here are common to various operations.

18.4.1.1 Timer Registers 0 to 3: BTxTMR (Base Timer 0/1/2/3 TiMer Register)

The bit configuration of timer registers 0 to 3 is shown below.

These registers read the counter value on the timer. The registers are only valid when its content represents a reload, PWM, or PPG timer. The value read from the registers is undefined if a PWC timer is read. For information on the values that will be read, see the section of "Operation."

BTxTMR: Address Base_addr + 00_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R,WX	R,WX	- - -	R,WX	R,WX	R,WX

Note:

These registers must be accessed in 16-bit mode.

18.4.1.2 Timer Control Registers 0 to 3: BTxTMCR (Base Timer 0/1/2/3 TiMer Control Register)

The bit configuration of timer control registers 0 to 3 is shown below.

These registers variously configure and stop the base timer and issue software triggers.

BTxTMCR: Address Base_addr + 02_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	CKS[2:0]			[PWM and PPG] RTGEN [Others] Reserved	[PWM and PPG] PMSK [PWC] EGS[2] [Others] Reserved	EGS[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W0 R0, W0 *3	R/W	R/W	R/W	R/W R0, WX *1	R/W R0, WX *1	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	[Reload timer/PWC] T32 [Others] Reserved	FMD[2:0]			[Reload timer/PWM/PPG] OSEL [Others] Reserved	MDSE	CTEN	STRG
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W R0, W0 *1 R0, W0 *2	R/W	R/W	R/W	R/W R/W0 *1	R/W	R,W	R0,W R0,W0 *1

*1: Attribute assumed for "Reserved"
*2: Attribute assumed for a 32-bit timer serving an odd-number channel
*3: Attribute assumed for a 32-bit timer serving an odd-number channel or for a 16/32-bit PWC timer

BTxTMCR2: Address Base_addr + 04_H (Access: Byte)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							CKS3
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

Notes:

- If you need to change the FMD[2:0] setting, once reset it to FMD[2:0] = 000, and then set FMD[2:0] to the desired value.
- Reserved bits must be set to "0".
- If you want to set bits of these registers except for the software trigger (STRG) bit, proceed as follows:
  1. Once stop operation by writing FMD[2:0] = 000 or CTEN = 0.
  2. Write desired values to the timer function selection bits (FMD[2:0]) and other bits.
- When writing to the software trigger bit (STRG), be careful not to clear other bits.
- Since FMD[2:0] = 000 specifies reset mode, you cannot set other bits when setting FMD[2:0] = 000.
- These registers must be accessed in 16-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15] Reserved

Always write "0" to this bit.

BTxTMCR2: bit8 BTxTMCR: [bit14 to bit12] CKS[3:0] (ClocK Select): Count clock selection bits

This bit selects a count clock.

CKS[3:0]	Description	
	Clock Source	Description
0000	Internal clock (Peripheral clock (PCLK))	1 division
0001		4 division
0010		16 division
0011		128 division
0100		256 division
0101	[Reload timer/PWM/PPG] External clock (ECK signal) [PWC] Setting is prohibited.	Rising edge
0110		Falling edge
0111		Both edges
1000	Internal clock (Peripheral clock (PCLK))	512 division
1001		1024 division
1010		2048 division
Other	Setting is prohibited.	

In the PWC mode, settings of 0101, 0110, and 0111 are prohibited.

[PWM/PPG] [bit11] RTGEN (Restart by TriGger ENable): Restart enable bit

If "1" is written to the STRG bit or an external activation trigger (TGIN signal) is detected, this bit sets whether or not to recount the value of cycle setting register (BTxPCSR)/L width setting reload register (BTxPRLL) by reloading it to the 16-bit down counter.

RTGEN	Description of Operation
0	Do not reactivate
1	Reactivate

[PWM/PPG] [bit10] PMSK (Pulse MaSK): Pulse output mask bit

This bit selects a level of waveform to output (TOUT signal) from the followings:

- Normal output: Output the waveform output from the 16-bit PWM/PPG timer without modification.
- Fixed output: Output a sequence of "L" level or "H" level signals regardless of the settings of cycle or duty.

PMSK	Description
0	Normal output
1	Fixed output

If the fixed output is selected by writing "1" to this bit, the level being output will vary depending on the settings of the OSEL bit.

- If OSEL=0: "L" level will be output.
- If OSEL=1: "H" level will be output.

[Reload timer/PWM/PPG] [bit9, bit8] EGS[1:0] (EdGe Select): Trigger input selection bits

These bits select an effective edge for the external activation trigger (TGIN) signal.

EGS[1:0]	Description
00	Trigger input has no effect on the operation
01	Rising edge
10	Falling edge
11	Both edges

[PWC] [bit10 to bit8] EGS[2:0] (EdGe Select): Measurement mode selection bits

These bits select a measurement mode.

EGS[2:0]	Description
000	"H" pulse width measurement: Duration in which the input signal is maintained at the "H" level
001	Rising edge interval measurement: Time from the detection of a rising edge to the detection of the next rising edge
010	Falling edge interval measurement: Time from the detection of a falling edge to the detection of the next falling edge
011	Edge-to-edge pulse width measurement: The width between consecutive input edges is either (1) or (2). (1) Time from the detection of a rising edge to the detection of the falling edge (2) Time from the detection of a falling edge to the detection of the rising edge
100	"L" pulse width measurement: Duration in which the input signal is maintained at the "L" level (Time from the detection of a falling edge to the detection of the rising edge)
101 to 111	Setting is prohibited.

[Reload timer/PWC] [bit7] T32 (Timer 32bit): 32-bit timer selection bit

This bit selects whether to run the 16/32-bit timer individually by each channel or use the 2 channels as 32-bit timer through a cascade connection. Set this bit for both channel 0 and channel 1 (channel 2 and channel 3).

T32 (channel 0/2)	T32 (channel 1/3)	Description
0	0	16-bit timer independent operation respectively
0	1	Setting is prohibited.
1	0	32-bit timer
1	1	Setting is prohibited.

Note:

Change this bit after changing the FMD[2:0] to "000". (Once you have changed the FMD[2:0] to "000", set the T32 bit and FMD[2:0] to a required value at the same time.)

[bit6 to bit4] FMD[2:0] (Function MoDe): Timer function selection bits

These bits select a function of base timer. To change these bits, go to "000" (reset mode) first, and set it to another mode.

FMD[2:0]	Description
000	Reset mode (Writing FMD = 000 will reverse the state of the base timer after the reset. Each register will be reset to the initial value.)
001	16-bit PWM timer
010	16-bit PPG timer
011	16/32-bit reload timer
100	16/32-bit PWC timer
101 to 111	Setting is prohibited.

[bit3] OSEL (Output SELect): Output polarity selection bit

When this bit is set, the signal level (H/L) output from TOUT will be inverted.

OSEL	Description
0	Normal output
1	Inverted output

[bit2] MDSE (MoDe Select): Mode selection bit

[Reload timer-PWM]

MDSE	Description
0	Reload mode: When the down counter underflows, the value of the base timer x cycle setting register (BTxPCSR) is reloaded to continue counting.
1	One-shot mode: Once the down counter underflows, the counter will no longer count.

[PPG]

MDSE	Description
0	Reload mode: A sequence of "L"-level and "H"-level signals (consecutive pulses) is output.
1	One-shot mode: A string of one "L"-level signal and one "H"-level signal (single pulses) is output.

[PWC]

MDSE	Description
0	Continuous measurement mode: In this mode, after one sequence of measurement is conducted, the input of the next measurement start edge is awaited and the detection of the next measurement start edge triggers another sequence of measurement.
1	Single measurement mode: In this mode, measurement is conducted only once.

[bit1] CTEN (CounT ENable): Counter operation enable bit

This bit enables/disables the counter operation.

CTEN	Description	
	Read	Write
0	Stopped	This bit becomes 0.
1	Operation enabled	This bit becomes 1.

Note:

When a falling edge is output from the even-number channel during timer operation in I/O mode 4 and I/O mode 6, this bit, which is an odd-number channel bit, is cleared to 0.

[bit0] STRG (Software TRiGger): Software trigger bit

Functions as a trigger for timer activation, etc.

For the PWC, the read value is "0". For the PWC, write "0" to this bit.

Notes:

- When writing to this bit, be careful not to clear other bits.
- When writing to CTEN and FMD[2:0] simultaneously, a trigger is issued as soon as operation is enabled.

STRG	Description
0	No effect on the operation
1	Issues a trigger.

18.4.1.3 I/O Selection Register: BTSEL01/23 (Base Timer SELect register ch.0 and ch.1/ch.2 and ch.3)

The bit configuration of the I/O selection register is shown below.

These bits set the I/O mode of ch.0 and ch.1 (ch.2 and ch.3) for the base timer.

BTSEL01: Address $009C_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SEL01[3:0]			
Initial value	1	1	1	1	0	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W

[bit3 to bit0] SEL01[3:0] (SELect): ch.0/ch.1 I/O selection bits

These bits set the I/O mode of ch.0 and ch.1 for the base timer.

BTSEL23: Address 0130_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				SEL23[3:0]			
Initial value	1	1	1	1	0	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W

[bit3 to bit0] SEL23[3:0] (SELect): ch.2/ch.3 I/O selection bits

These bits set the I/O mode of ch.2 and ch.3 for the base timer.

SELn[3:0] (n=01,23)	Description
0000	I/O mode 0 (16-bit timer standard mode)
0001	I/O mode 1 (32-bit timer full mode)
0010	I/O mode 2 (External trigger sharing mode)
0011	Setting is prohibited.
0100	I/O mode 4 (Timer activation/stop mode)
0101	I/O mode 5 (Simultaneous software activation mode)
0110	I/O mode 6 (Software activation timer activation/stop mode)
0111	I/O mode 7 (Timer activation mode)
1xxx	Setting is prohibited.

Notes:

- These registers must be accessed in 8-bit mode.
- Rewrite these registers only after setting base timer reset mode (FMD2 to FMD0=000) using the FMD2 to FMD0 bits of the base timer x timer control register (BTxTMCR).

18.4.1.4 Simultaneous Software Activation Register: BTSSSR/BTSSSRA (Base Timer Software Synchronous Start Register)

The bit configuration of the simultaneous software activation register is shown below.

This register is the input signal in the I/O modes 5 and 6. Trigger can be generated simultaneously for all channels with this register.

BTSSSR: Address 009EH (Access: Byte, Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						SSSR1	SSSR0
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,W	R1,W

[bit1] SSSR1 (Software Synchronous Start Register ch.1): Simultaneous software activation bit ch.1

[bit0] SSSR0 (Software Synchronous Start Register ch.0): Simultaneous software activation bit ch.0

These bits are the input signal in the I/O modes 5 and 6. For the connections, see "Figure 18-4. Wiring Diagram of Each I/O Mode (2)".

BTSSSRA: Address 0132_H (Access: Byte, Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						SSSR3	SSSR2
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,W	R1,W

[bit1] SSSR3 (Software Synchronous Start Register ch.3): Simultaneous software activation bit ch.3

[bit0] SSSR2 (Software Synchronous Start Register ch.2): Simultaneous software activation bit ch.2

These bits are the input signal in the I/O modes 5 and 6. For the connections, see "Figure 18-4. Wiring Diagram of Each I/O Mode (2)".

SSSRn (n=0 to 3)	Description
0	No effect on the operation.
1	"1" pulse is applied to the timer input, and then the corresponding channel is activated.

18.4.2 Registers for 16/32-bit Reload Timer

This section explains registers for 16/32-bit reload timer.

18.4.2.1 Status Control Registers 0 to 3: BTxSTC (Base Timer 0/1/2/3 STatus Control)

The bit configuration of status control registers 0 to 3 is shown below.

These registers control interrupt requests.

BTxSTC: Address Base_addr + 05_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R(RM1),W	R0,W0	R(RM1),W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR and UDIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit6] TGIE (TriGger Interrupt Enable): Trigger interrupt request enable bit

This bit sets whether or not to generate a trigger interrupt request when an activation trigger for 16/32-bit reload timer has been detected (TGIR = 1).

[bit4] UDIE (UnDerflow Interrupt Enable): Underflow interrupt request enable bit

This bit sets whether or not to generate an underflow interrupt request when the down counter underflows (UDIR = 1).

TGIE/UDIE	Description
0	Disabled
1	Enabled

[bit2] TGIR (TriGger Interrupt Register): Trigger interrupt request flag bit

This bit indicates that an activation trigger for the 16/32-bit reload timer has been detected. When the TGIE bit is set to "1" while this bit is "1", a trigger interrupt request will be generated.

[bit0] UDIR (UnDerflow Interrupt Register): Underflow interrupt request flag bit

This bit indicates that the down counter value has changed from "0000_H" to "$FFFF_H$" and an underflow occurred. When this bit is "1" and the UDIE bit is set to "1", an underflow interrupt request is generated.

TGIR/UDIR	Read	Write
0	No trigger detection/underflow occurred.	This bit is cleared.
1	Trigger detection/underflow occurred.	No effect on the operation.

18.4.2.2 Cycle Setting Registers 0 to 3: BTxPCSR (Base Timer 0/1/2/3 Pulse Counter Start Register)

The bit configuration of cycle setting registers 0 to 3 is shown below.

These registers with a buffer set the cycle for 16/32-bit reload timer. The down counter counts down from the value set to these registers.

BTxPCSR: Address Base_addr + 08_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the 16/32-bit reload timer (FMD2 to FMD0 = 011) using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data): Data bits

These registers with a buffer set the cycle for the 16/32-bit reload timer. The down counter counts down from the value set to these registers.

The value set to these registers is loaded to the 16-bit down counter in the following cases:

- When the 16/32-bit reload timer is started
- When the down counter underflows

The following values are set to these registers when 2 channels of a 16-bit reload timer are cascaded and used as the 32-bit reload timer.

- Value of even-number channel cycle setting register (BTxPCSR): Value of lower 16-bit
- Value of odd-number channel cycle setting register (BTxPCSR): Value of upper 16-bit

For this reason, in the 32-bit timer mode, write values into these registers in the following order.

1. Odd-number channel base timer x cycle setting register (BTxPCSR)
2. Even-number channel base timer x cycle setting register (BTxPCSR)

18.4.3 Registers for 16-bit PWM Timer

This section explains registers for 16-bit PWM timer.

18.4.3.1 Status Control Registers 0 to 3: BTxSTC (Base Timer 0/1/2/3 STatus Control)

The bit configuration of status control registers 0 to 3 is shown below.

These registers control interrupt requests.

BTxSTC: Address Base_addr + 05_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TGIE	DTIE	UDIE	Reserved	TGIR	DTIR	UDIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R0,W0	R(RM1),W	R(RM1),W	R(RM1),W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR, DTIR, and UDIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit6] TGIE (TriGger Interrupt Enable): Trigger interrupt request enable bit

This bit sets whether or not to generate a trigger interrupt request when a 16-bit PWM timer activation trigger is detected (TGIR = 1).

[bit5] DTIE (DuTy Interrupt Enable): Duty match interrupt request enable bit

This bit sets whether or not to generate a duty match interrupt request when the value of the 16-bit down counter matches the value of the base timer x duty setting register (BTxPDUT) (DTIR = 1).

[bit4] UDIE (UnDerflow Interrupt Enable): Underflow interrupt request enable bit

This bit sets whether or not to generate an underflow interrupt request when the down counter underflows (UDIR = 1).

TGIE/DTIE/UDIE	Description
0	Disabled
1	Enabled

[bit2] TGIR (TriGger Interrupt Register): Trigger interrupt request flag bit

This bit indicates that a 16-bit PWM timer activation trigger is detected. When this bit is "1" and the TGIE bit is set to "1", a trigger interrupt request is generated.

[bit1] DTIR (DuTy Interrupt Register): Duty match interrupt request flag bit

This bit indicates that the value of the 16-bit down counter matches the value of the duty setting register (BTxPDUT) (a duty matches). When this bit is "1" and the DTIE bit is set to "1", a duty match interrupt request is generated.

[bit0] UDIR (UnDerflow Interrupt Register): Underflow interrupt request flag bit

This bit indicates that the 16-bit down counter value changed from "0000_H" to "$FFFF_H$" and an underflow occurred. When this bit is "1" and the UDIE bit is set to "1", an underflow interrupt request is generated.

TGIR/DTIR/UDIR	Read	Write
0	A trigger detection, duty match and underflow did not occur.	This bit is cleared.
1	A trigger detection, duty match or underflow occurred.	No effect on the operation.

18.4.3.2 Cycle Setting Registers 0 to 3: BTxPCSR (Base Timer 0/1/2/3 Pulse Counter Start Register)

The bit configuration of cycle setting registers 0 to 3 is shown below.

These registers with a buffer set the cycle for the 16-bit PWM timer. The 16-bit down counter counts down from the value set to these registers. When the counter value matches the value set to these registers, the level of the output signal (TOUT) is inverted.

BTxPCSR: Address Base_addr + 08_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the 16-bit PWM timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- Be sure to rewrite the duty setting register (BTxPDUT) when these registers are rewritten.
- Do not set a value smaller than the value set to the duty setting register (BTxPDUT).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data): Data bits

These registers with a buffer set the cycle for the 16-bit PWM timer. The 16-bit down counter counts down from the value set to these registers. When the counter value matches the value set to these registers, the level of the output signal (TOUT) is inverted.

These registers have a buffer and thus can be rewritten during counting. The value set to these registers is loaded to the 16-bit down counter in the following cases:

- When the 16-bit PWM timer is activated
- When the down counter underflows

When the same value is set to these registers and the base timer x duty setting register (BTxPDUT), the level of the output signal (TOUT) can be fixed. The output signal level is as follows according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR):

- OSEL=0: "H" level
- OSEL=1: "L" level

18.4.3.3 Duty Setting Registers 0 to 3: BTxPDUT (Base Timer 0/1/2/3 Pulse DuTy register)

The bit configuration of duty setting registers 0 to 3 (BTxPDUT) is shown below.

These registers with a buffer set the duty for the 16-bit PWM timer. When the 16-bit down counter value matches the value set to these registers, the level of the output signal (TOUT) is inverted.

BTxPDUT: Address Base_addr + $0A_H$ (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the 16-bit PWM timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- Do not set the value higher than the value set to the cycle setting register (BTxPCSR) when these registers are rewritten.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data): Data bits

These registers with a buffer set the duty for the 16-bit PWM timer. When the 16-bit down counter value matches the value set to these registers, the level of the output signal (TOUT) is inverted.
These registers have a buffer and thus can be rewritten during counting.
If the 16-bit down counter underflows, the buffer value will be transferred.
When the same value is set to these registers and the base timer x cycle setting register (BTxPCSR), the level of the output signal (TOUT) can be fixed. The output signal level is as follows according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR):

- OSEL=0: All "H" level
- OSEL=1: All "L" level

18.4.4 Registers for 16-bit PPG Timer

This section explains registers for 16-bit PPG timer.

18.4.4.1 Status Control Registers 0 to 3: BTxSTC (Base Timer 0/1/2/3 STatus Control)

The bit configuration of status control registers 0 to 3 is shown below.

These registers control interrupt requests.

BTxSTC: Address Base_addr + 05_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	TGIE	Reserved	UDIE	Reserved	TGIR	Reserved	UDIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R0,W0	R/W	R0,W0	R(RM1),W	R0,W0	R(RM1),W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to TGIR and UDIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit6] TGIE (TriGger Interrupt Enable): Trigger interrupt request enable bit

This bit sets whether or not to generate a trigger interrupt request when a 16-bit PPG timer activation trigger is detected (TGIR = 1).

[bit4] UDIE (UnDerflow Interrupt Enable): Underflow interrupt request enable bit

This bit sets whether or not to generate an underflow interrupt request when the base timer x H width setting reload register (BTxPRLH) completed counting down and the counter underflows (UDIR = 1).

TGIE/UDIE	Description
0	Disabled
1	Enabled

[bit2] TGIR (TriGger Interrupt Register): Trigger interrupt request flag bit

This bit indicates that a 16-bit PPG timer activation trigger is detected. When this bit is "1" and the TGIE bit is set to "1", a trigger interrupt request is generated.

[bit0] UDIR (UnDerflow Interrupt Register): Underflow interrupt request flag bit

This bit indicates that the base timer x H width setting reload register (BTxPRLH) completed counting down and an underflow occurred. An underflow will occur if the register attempts counting down when the 16-bit down counter value is "0000_H". When this bit is "1" and the UDIE bit is set to "1", an underflow interrupt request is generated.

TGIR/UDIR	Read	Write
0	No trigger detection/underflow occurred.	This bit is cleared.
1	Trigger detection/underflow occurred.	No effect on the operation.

18.4.4.2 L Width Setting Registers 0 to 3: BTxPRLL (Base Timer 0/1/2/3 Pulse Length of "L" register)

The bit configuration of L width setting registers 0 to 3 is shown below.

These registers set the default level for the signal output from the 16-bit PPG timer.

BTxPRLL: Address Base_addr + 08_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the PPG timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data): Data bits

These registers set the default level for the signal output from the 16-bit PPG timer. When the 16-bit down counter completes counting down the value set to these registers, the level of the output waveform (TOUT) will be inverted. Setting these registers and the base timer x H width setting reload register (BTxPRLH) determines the widths of "L" level and "H" level for the output signal. The signal level width set to these registers depends on the setting of the OSEL bit of the timer control register (BTxTMCR) as follows:

- OSEL=0: "L" level width
- OSEL=1: "H" level width

The value set to these registers is loaded to the 16-bit down counter when a 16-bit PPG timer activation trigger is detected or when the base timer x H width setting reload register (BTxPRLH) completed counting values and underflows.

18.4.4.3 H Width Setting Registers 0 to 3: BTxPRLH (Base Timer 0/1/2/3 Pulse Length of "H" register)

The bit configuration of H width setting registers 0 to 3 (BTxPRLH) is shown below.

These registers with a buffer set the width of signal level output when the base timer x L width setting reload register (BTxPRLL) completes counting values.

BTxPRLH: Address Base_addr + $0A_H$ (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

Notes:

- These registers must be accessed in 16-bit mode.
- Set these registers after selecting a base timer function to the PPG timer using the FMD2 to FMD0 bits of the timer control register (BTxTMCR).
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit15 to bit0] D[15:0] (Data): Data bits

These registers with a buffer set the width of signal level output when the L width setting reload register (BTxPRLL) completes counting values. When the 16-bit down counter completes counting down the value set to these registers, the signal level of the output waveform (TOUT) will be inverted.

Setting these registers and the base timer x L width setting reload register (BTxPRLL) determines the widths of "L" level and "H" level for the output signal. The signal level width set to these registers depends on the setting of the OSEL bit of the base timer x timer control register (BTxTMCR) as follows:

- OSEL=0: "H" level width
- OSEL=1: "L" level width

These registers have a buffer and thus can be rewritten during counting. These registers transfer values at the following timing.

- Transfer to the buffer
- When a 16-bit PPG timer activation trigger is detected
- When the base timer x H width setting reload register (BTxPRLH) completes counting down values and underflows
- Transfer to the 16-bit down counter
- When counting down from the value of the base timer x L width setting reload register (BTxPRLL) is completed.

For rewriting timing, see "Write Timing".

18.4.5 16/32-bit PWC Timer Register

This section explains registers for 16/32-bit PWC timer.

18.4.5.1 Status Control Registers 0 to 3: BTxSTC (Base Timer 0/1/2/3 STatus Control)

The bit configuration of status control registers 0 to 3 is shown below.

These registers control interrupt requests.

BTxSTC: Address Base_addr + 05_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ERR	EDIE	Reserved	OVIE	Reserved	EDIR	Reserved	OVIR
Initial value	0	0	0	0	0	0	0	0
Attribute	R,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R0,W0	R(RM1),W

Notes:

- Reserved bits must be set to "0".
- For the read-modify-write instruction to OVIR, "1" is read out.
- These registers must be accessed in 8-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

[bit7] ERR (ERRor): Error flag bit

This bit indicates that the next measurement is completed before the measurement result is read from the data buffer register (BTxDTBF) in the continuous measurement mode and the measurement result has been overwritten by the new value. The old value is discarded. This bit is cleared to "0" when a value is read from the data buffer register (BTxDTBF).

ERR	Description
0	The measurement result has not been overwritten.
1	The measurement result has been overwritten.

[bit6] EDIE (EnD Interrupt Enable): Measurement completion interrupt request enable bit

This bit sets whether or not to generate a measurement completion interrupt request when the measurement of the 16/32-bit PWC timer is completed (EDIR = 1).

[bit4] OVIE (OVerflow Interrupt Enable): Overflow interrupt request enable bit

This bit sets whether or not to generate an overflow interrupt request when the up counter overflows (OVIR = 1).

EDIE/OVIE	Description
0	Disabled
1	Enabled

[bit2] EDIR (EnD Interrupt Register): Measurement completion interrupt request flag bit

This bit indicates that the measurement of the 16/32-bit PWC timer is completed. When this bit is "1" and the EDIE bit is set to "1", a measurement completion interrupt request is generated. This bit is cleared when the measurement result (BTxDTBF) is read out.

[bit0] OVIR (OVerflow Interrupt Register): Overflow interrupt request flag bit

This bit indicates that the up counter value has changed from "$FFFF_H$" to "0000_H" and an overflow occurred. When this bit is "1" and the OVIE bit is set to "1", an overflow interrupt request is generated.

This bit is cleared when "0" is written.

EDIR/OVIR	Read	Write
0	No measurement completion/overflow occurred.	(EDIR) No effect on the operation. (OVIR) This bit is cleared.
1	Measurement completion/overflow occurred.	No effect on the operation.

18.4.5.2 Data Buffer Registers 0 to 3: BTxDTBF (Base Timer 0/1/2/3 DaTa BuFfer register)

The bit configuration of data buffer registers 0 to 3 is shown below.

These registers are used to read out the measurement value of the 16/32-bit PWC timer and the up counter value.

Notes:

- These registers must be accessed in 16-bit mode.
- These registers will also be initialized when reset mode is set (writing of BTxTMCR.FMD = 000).

BTxDTBF: Address Base_addr + $0A_H$ (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	D[15:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R,WX	R,WX	- - -	R,WX	R,WX	R,WX

[bit15 to bit0] D[15:0] (Data): Data bits

These registers are used to read out the measurement value of the 16/32-bit PWC timer and the up counter value. The value read from these registers is different in the single measurement mode and continuous measurement mode.

- Single measurement mode: The up counter value is read during counting and the measurement result is read after the measurement completion.
- Continuous measurement mode: The value measured previously is read both during counting and after the measurement completion. The up counter value cannot be read.

The following values are set to these registers when 2 channels of a 16-bit PWC timer are cascaded and used as the 32-bit PWC timer.

- Value of even-number channel data buffer register (BTxDTBF): Value of lower 16-bit
- Value of odd-number channel data buffer register (BTxDTBF): Value of upper 16-bit

In the 32-bit timer mode, read values from these registers in the following order.

1. Even-channel data buffer register (BTxDTBF)
2. Odd-channel data buffer register (BTxDTBF)

18.5 Operation

This section explains the operation of the base timer.

18.5.1 Selection of Timer Function

This section explains selection of the timer function.

Select the timer function for BTxTMCR.FMD[2:0].

18.5.2 I/O Allocation

This section explains I/O allocation.

Set I/O of the base timer for the BTSEL01/BTSEL23 register before using the timer. You can select one of the following seven modes:

I/O mode 0

16-bit timer standard mode

The base timer operates separately for each channel in this mode.

I/O mode 1

32-bit timer full mode

The even-number channel signals of the base timer are allocated to the external pin in this mode.

I/O mode 2

External trigger sharing mode

The external activation trigger can be input to 2 channels of base timer at the same time in this mode. Using this mode allows simultaneous activation of 2 channels of base timer.

I/O mode 4

Timer activation/stop mode

Activation/stop of the odd-number channel is controlled by the even-number channel in this mode. The odd-number channel is started with the rising edge(*) of the output signal from the even-number channel and stops with the falling edge(*).

I/O mode 5

Simultaneous software activation mode

More than one channels are started by the software at the same time in this mode.

I/O mode 6

Software activation timer activation/stop mode

Activation/stop of the odd-number channel is controlled by the even-number channel in this mode. The even-number channel is started by the software. The odd-number channel is started with the rising edge(*) of the output signal from the even-number channel and stops with the falling edge(*).

I/O mode 7

Timer activation mode

Activation of the odd-number channel is controlled by the even-number channel in this mode. The odd-number channel is started with the rising edge(*) of the output signal from the even-number channel.

(*) Make a setting using the trigger input selection bit (BTxTMCR.EGS).

Figure 18-3. Wiring Diagram of Each I/O Mode (1)

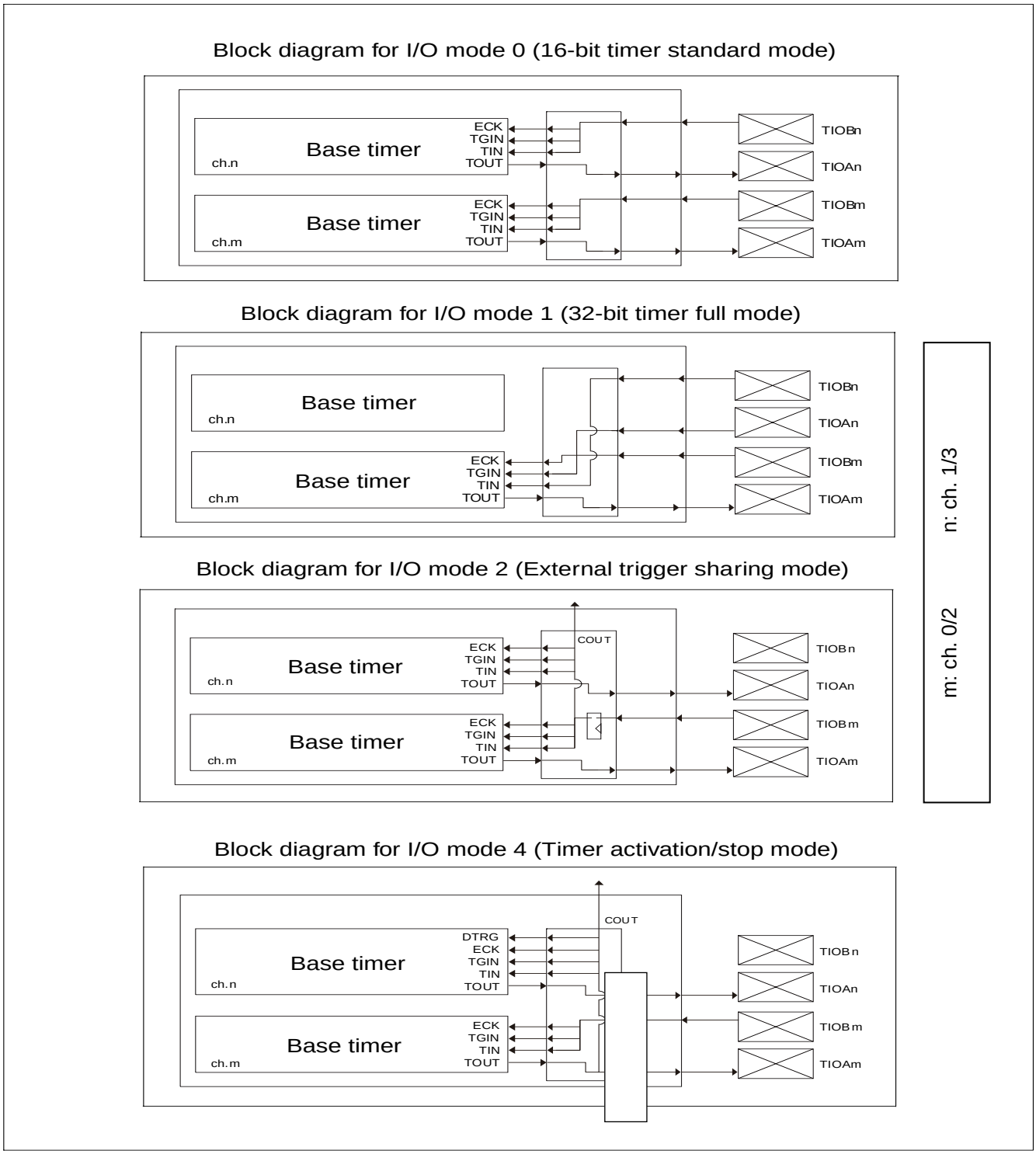
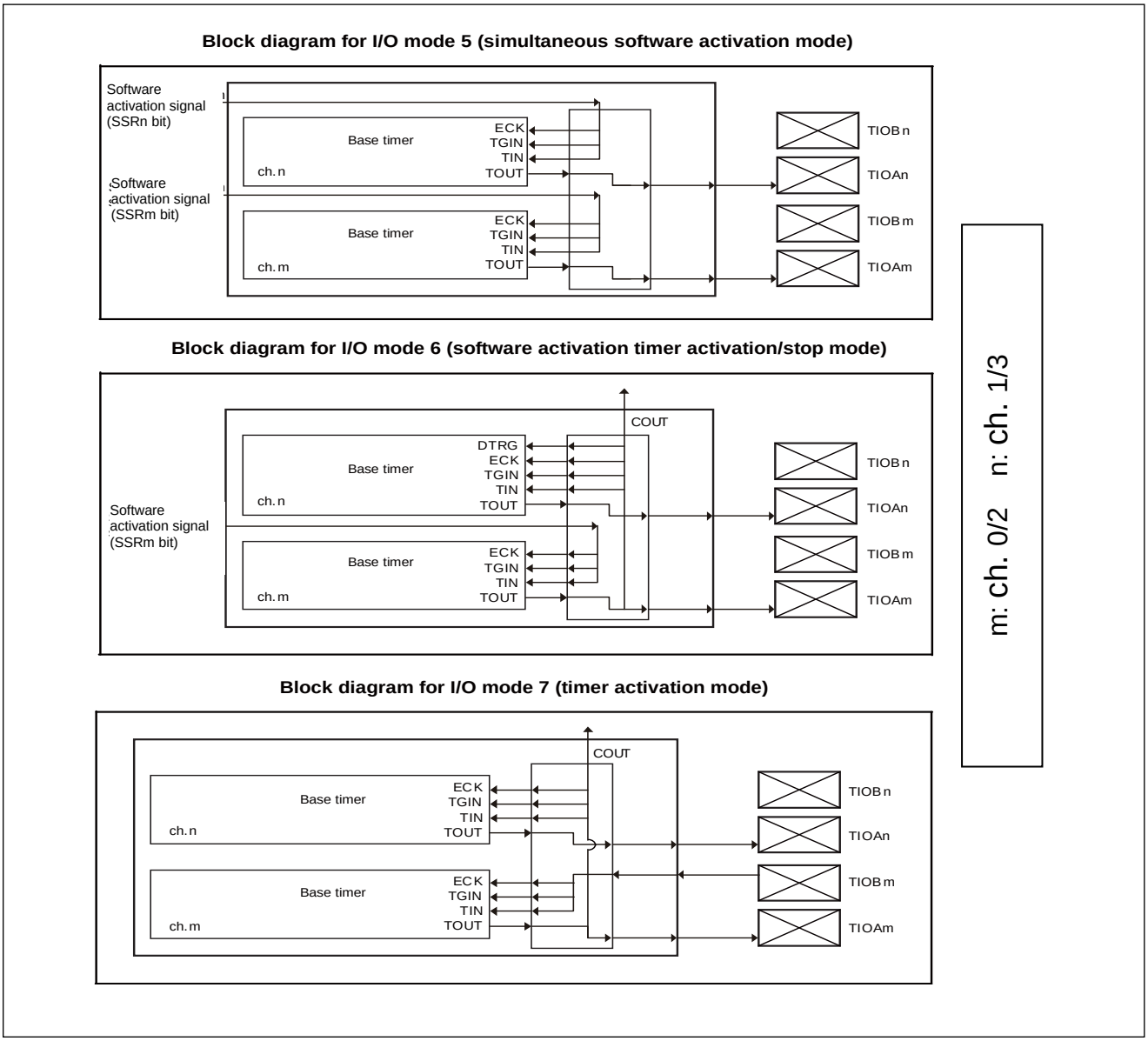


Figure 18-4. Wiring Diagram of Each I/O Mode (2)



18.5.3 32-bit Mode Operation

This section explains 32-bit mode operation.

The reload timer and PWC timer can be operated in the 32-bit mode using 2 channels. The basic function/operation in the 32-bit mode is shown below.

18.5.3.1 32-bit Mode Function

This section explains the 32-bit mode function.

This function realizes the operation of the 32-bit data reload timer or 32-bit data PWC timer by combining 2 channels of base timer. The upper 16-bit timer counter value of the odd-number channel is also loaded when the lower 16-bit timer counter value of the even-number channel is read. Thus, the timer counter value in operation can also be read.

18.5.3.2 32-bit Mode Setting

This section explains the 32-bit mode setting.

First, set "000" to the FMD bits of the BTxTMCR register of the even-number channel to reset to the reset mode, then select the reload timer or PWC timer and set the operation as in the 16-bit mode. While doing so, set to the 32-bit mode by writing "1" to the T32 bit of the BTxTMCR register. Leave the T32 bit of the odd-number channel "0". You do not have to set the reset mode.

For the reload timer, set the upper 16-bit reload values of the 32-bit to the cycle setting register of the odd-number channel, then set the lower 16-bit reload values to the cycle setting register of the even-number channel.

The transition to the 32-bit mode is reflected immediately after the writing to the T32 bit. Thus, setting change for both channels must be done when the counting is stopped.

To transit from the 32-bit mode to the 16-bit mode, set "000" to the FMD bits of the BTxTMCR register of the even-number channel to reset both the even-number and odd-number channels, and make a setting in the 16-bit mode for each channel.

18.5.3.3 32-bit Mode Operation

This section explains 32-bit mode operation.

After setting the 32-bit mode when the reload timer or PWC timer is started with the control of the even-number channel, the timer/counter of the even-number channel operates with lower 16-bit and the timer/counter of the odd-number channel operates with upper 16-bit.

The 32-bit mode operation depends on the setting of the even-number channel. Thus, the setting of the odd-number channel (excepting the cycle setting register for the reload timer) is ignored. Timer activation, waveform output and interrupt signal also apply the setting of the even-number channel. (The odd-number channel is masked with the value fixed to L.)

For the configuration, see "Figure 18-13. Configuration in 32-bit Timer Mode" and "Figure 18-31. Configuration in 32-bit Timer Mode".

18.5.4 16/32-bit Reload Timer Operation

This section explains the 16/32-bit reload timer operation.

This section explains the operation performed when the base timer included in this series is used as the 16/32-bit reload timer. An example is also given to set various operation conditions.

Figure 18-5. Block Diagram (16-bit Reload Timer Operation)

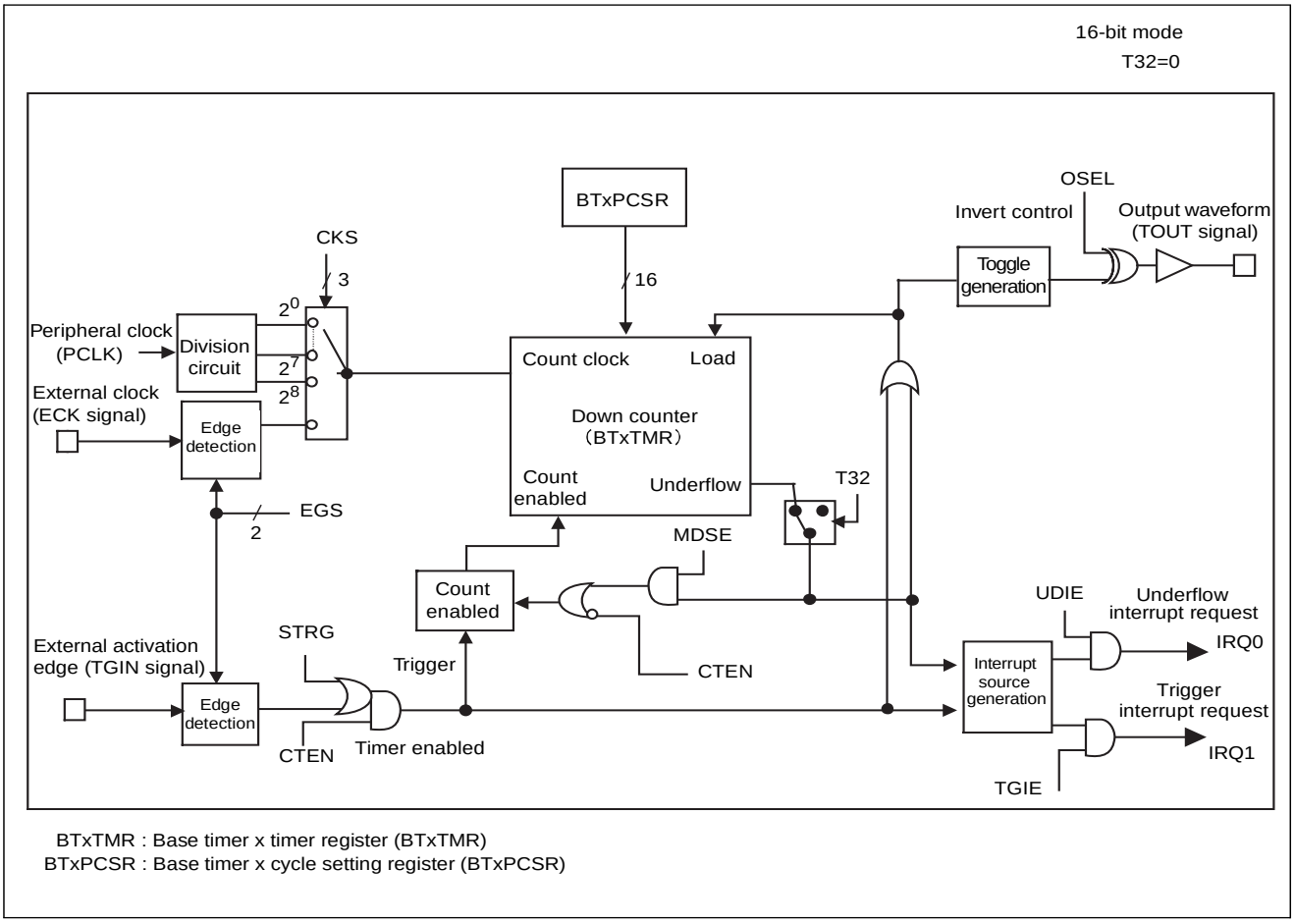
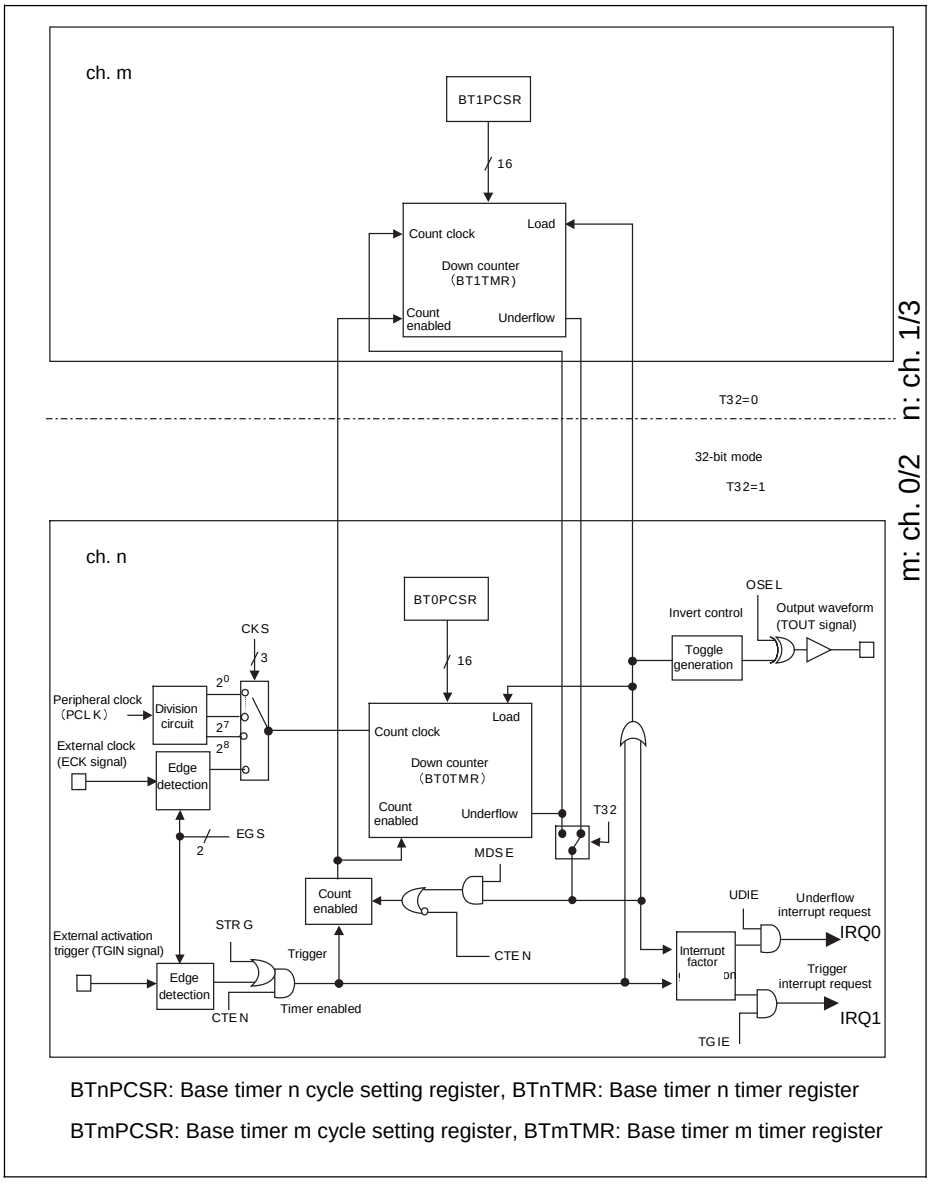


Figure 18-6. Block Diagram (32-bit Reload Timer Operation)



BTnPCSR: Base timer n cycle setting register, BTnTMR: Base timer n timer register

BTmPCSR: Base timer m cycle setting register, BTmTMR: Base timer m timer register

18.5.4.1 Overview

This section explains the overview of the 16/32-bit reload timer operation.

The 16/32-bit reload timer is a timer that decreases from the value set in the base timer x cycle setting register (BTxPCSR). This timer has a function of generating an underflow interrupt request when the down counter underflows.

The 16/32-bit reload timer has two modes: timer mode and operation mode. The operation of the timer varies in accordance with combinations of these modes.

- Timer mode: One of the following two modes can be selected using the T32 bit of the base timer x timer control register (BTxTMCR).
  - ☐ 16-bit timer mode (T32 = 0): 16-bit reload timer can operate individually for each of the channels.
  - ☐ 32-bit timer mode (T32 = 1): 2 channels can be cascaded and used as a 32-bit reload timer.
- Operation mode: One of the following two modes can be selected using the MDSE bit of the base timer x timer control register (BTxTMCR).
  - ☐ Reload mode (MDSE = 0): In this mode, when the down counter underflows, the preset value (cycle) is reloaded to allow the timer to restart counting.
  - ☐ One-shot mode (MDSE = 1): Once the down counter underflows, the counter will no longer count.

18.5.4.2 Operation in Reload Mode

This section explains the operation in reload mode.

This section explains the operation in reload mode.

Overview

In this mode, the value set in the base timer x cycle setting register (BTxPCSR) is reloaded every time an underflow occurs to ensure that countdown is continued. To use this mode, set reload mode by resetting the MDSE bit of the base timer x timer control register (BTxTMCR) to "0" (MDSE=0).

Operation

Activation

Activate the 16/32-bit reload timer with the following procedure:

1. Permit 16/32-bit reload timer operation by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "1" (CTEN=1).The 16/32-bit reload timer begins to wait for an activation trigger.
2. Enter an activation trigger by one of the following methods:
   - ☐ Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
   - ☐ Enter an effective edge (an edge set in the EGS1 and EGS0 bits) for an external activation trigger (TGIN signal).

Notes:

- The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01/BTSEL23). See "18.5.2 I/O Allocation".
- To start counting as soon as the operation is permitted, set both CTEN and STRG bits of the base timer x timer control register (BTxTMCR) to "1".

Counting operation

When an activation trigger is input, the value (cycle) set in the base timer x cycle setting register (BTxPCSR) is loaded to the down counter, which begins counting down, after one of the following lengths of time elapses:

- If a software trigger is input: 1T (T: Count clock cycle)
- If an external activation trigger (TGIN signal) is input: 2T to 3T (T: Count clock cycle)

Figure 18-7 and Figure 18-8 show the count start timing.

Figure 18-7. Count Start Timing (Software Trigger)

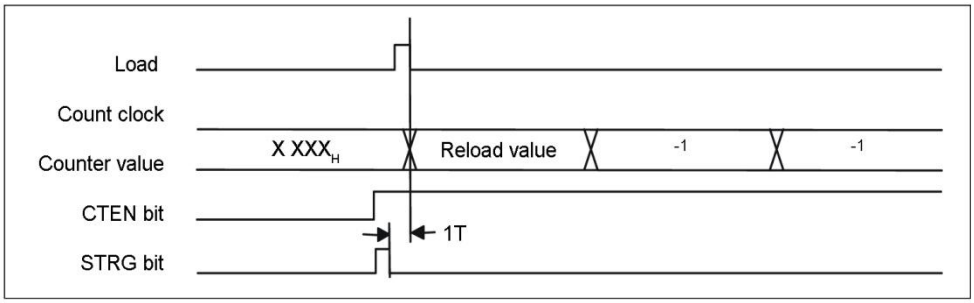
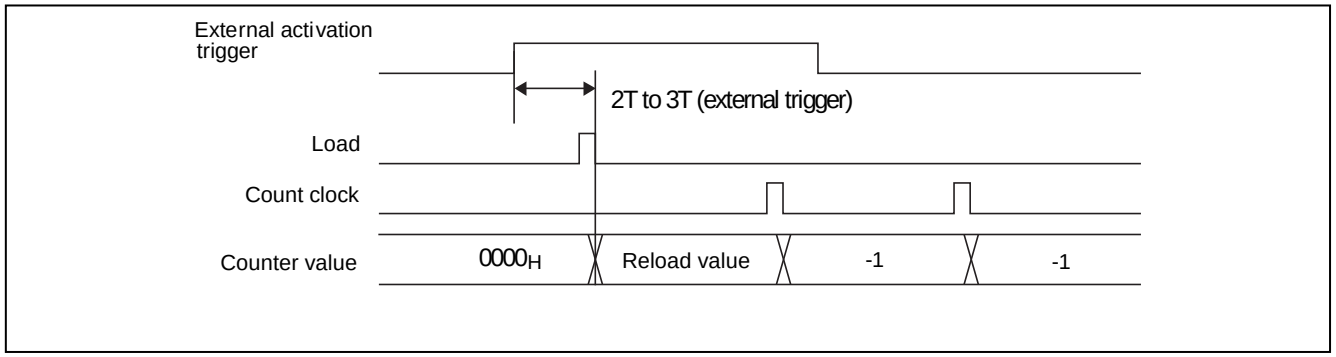


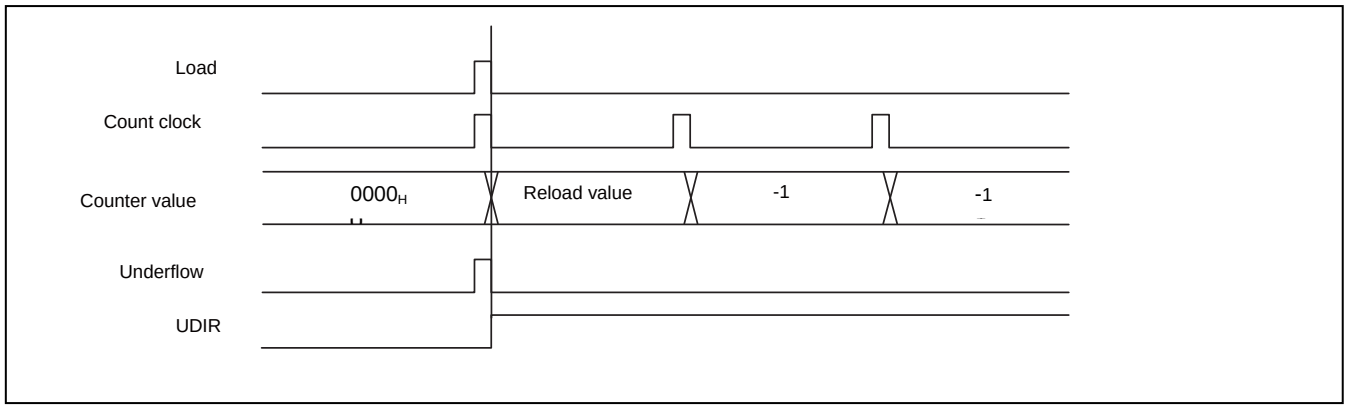
Figure 18-8. Count Start Timing (External Activation Trigger (TGIN Signal), Effective Edge = Rising Edge)



Note:

The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01/BTSEL23). See "18.5.2 I/O Allocation".
When the down counter underflows after attempting to count down further from the value of "0000_H", the value (cycle) set in the base timer x cycle setting register (BTxPCSR) is reloaded to the down counter, which continues to count down.
If an underflow occurs, the UDIR bit of the base timer x status control register (BTxSTC) changes to "1". At this time, an underflow interrupt request occurs if the UDIE bit is set to "1". The following figure shows the operation in case of an underflow.

Figure 18-9. Operation in Case of an Underflow



Output Waveform

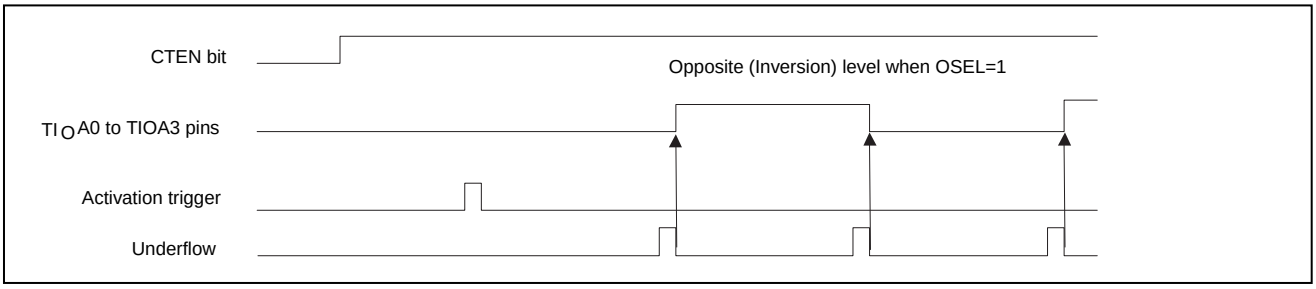
The waveform (TOUT signal) of the 16/32-bit reload timer can be output. The waveform (TOUT signal) to be output varies according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR).

Table 18-3. Correspondence between Output Polarities and Output Waveforms

Output Polarity	Output Waveform
Normal polarity (OSEL = 0)	"L" level pulse is output when counting starts. Thereafter, the output level is inverted every time an underflow occurs.
Inverted polarity (OSEL = 1)	"H" level pulse is output when counting starts. Thereafter, the output level is inverted every time an underflow occurs.

The following figure shows the output waveform in reload mode.

Figure 18-10. Output Waveform in Reload Mode (Normal Polarity)



18.5.4.3 Operation in One-Shot Mode

This section explains the operation in one-shot mode.

This section explains the operation in one-shot mode.

Overview

In this mode, the counter will no longer count down once an underflow occurs.
To use this mode, set one-shot mode by setting the MDSE bit of the base timer x timer control register (BTxTMCR) to "1" (MDSE=1).

Operation

Activation

The same operation as in reload mode. See "Operation" in "18.5.4.2 Operation in Reload Mode".

Counting operation

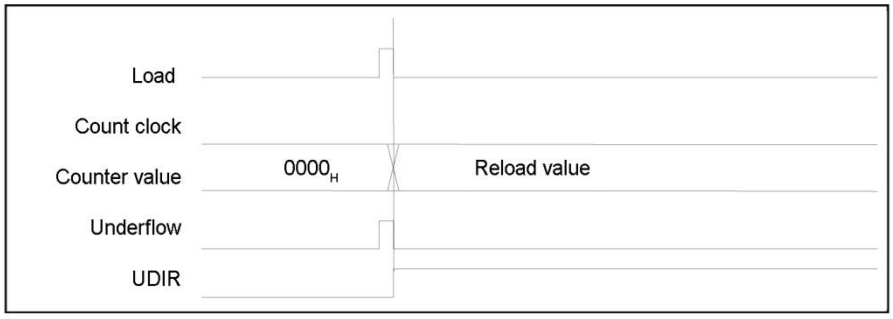
The operation is the same as in reload mode until an underflow occurs. See "Operation" in "18.5.4.2 Operation in Reload Mode".

When the down counter underflows, the value (cycle) set in the base timer x cycle setting register (BTxPCSR) is reloaded to the down counter. However, the down counter stops counting.

If an underflow occurs, the UDIR bit of the base timer x status control register (BTxSTC) changes to "1". At this time, an underflow interrupt request occurs if the UDIE bit of the base timer x status control register (BTxSTC) is set to "1".

The following figure shows the operation in case of an underflow.

Figure 18-11. Operation in Case of an Underflow



Output Waveform

The waveform (TOUT signal) of the 16/32-bit reload timer can be output. The waveform (TOUT signal) to be output varies according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR).

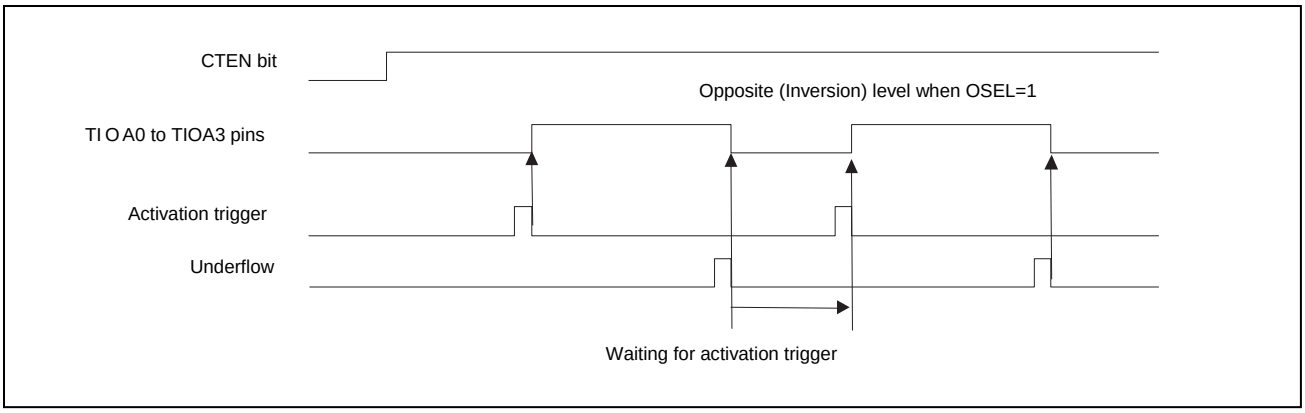
The following table shows the correspondence between output polarities and output waveforms.

Table 18-4. Correspondence between Output Polarities and Output Waveforms

Output Polarity	Output Waveform
Normal polarity (OSEL = 0)	When an activation trigger is input (counting in progress), "H" level pulse is output. "L" level pulse is output while the timer waits for an activation trigger.
Inverted polarity (OSEL = 1)	When an activation trigger is input (counting in progress), "L" level pulse is output. "H" level pulse is output while the timer waits for an activation trigger.

The following figure shows the output waveform in one-shot mode.

Figure 18-12. Output Waveform in One-shot Mode (Normal Polarity)



18.5.4.4 32-bit Timer Mode Operation

This section explains the 32-bit timer mode operation.

This section explains the setting and operation for cascading 2 channels of a 16-bit reload timer and using them as a 32-bit reload timer.

Overview

Using the T32 bit of the base timer x timer control register (BTxTMCR), 2 channels of a 16-bit reload timer can be cascaded and used as a 32-bit reload timer.

In this mode, the even-number channel corresponds to the lower 16-bit operation, and the odd-number channel corresponds to the upper 16-bit operation. Therefore, set the reload values in the order of the upper 16 bits (odd-number channels) -> the lower 16 bits (even-number channels) and read the down counter values in the order of the lower 16 bits (even-number channels) -> the upper 16 bits (odd-number channels).

Setting Procedure (Example)

To set 32-bit timer mode, set the T32 bit of the base timer x timer control register (BTxTMCR) of even-number channels to "1" and the T32 bit of the base timer x timer control register (BTxTMCR) of the odd-number channels to "0".

When setting 32-bit timer mode, set the registers using the procedure shown below. The register setting differs between even-number and odd-number channels. The following shows an example of using a cascade connection.

1. Specify ch.0 to reset mode by setting FMD2 to FMD0 bits of base timer 0 timer control register (BT0TMCR). (FMD2 to FMD0 = 000)
2. Select 16/32-bit reload timer for ch.0 and ch.1 by setting the FMD2 to FMD0 bits of the base timer x timer control register (BT0TMCR, BT1TMCR) of ch.0 and ch.1. (FMD2 to FMD0 = 011)
   At the same time, select 32-bit timer mode by setting the T32 bit of the base timer 0 timer control register (BT0TMCR). (T32=1)
3. Set a reload value in the upper 16 bits in the base timer 1 cycle setting register (BT1PCSR).
4. Set a reload value in the lower 16 bits in the base timer 0 cycle setting register (BT0PCSR).

Use the same procedure to set both ch.2 and ch.3.

Notes:

- Rewrite the T32 bit while the operation of both of the even-number and odd-number channels is stopped. Whether the counting operation is stopped can be checked by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "0" (CTEN=0).
- A reload value in the base timer x cycle setting register (BTxPCSR) must be set in the order of the odd-number -> even-number channels.

Operation

In 32-bit timer mode, the counting operation is basically the same as in 16-bit timer mode.
However, the counting operation conforms to the settings of the even-number channels, ignoring the settings of the following registers for the odd-number channels.

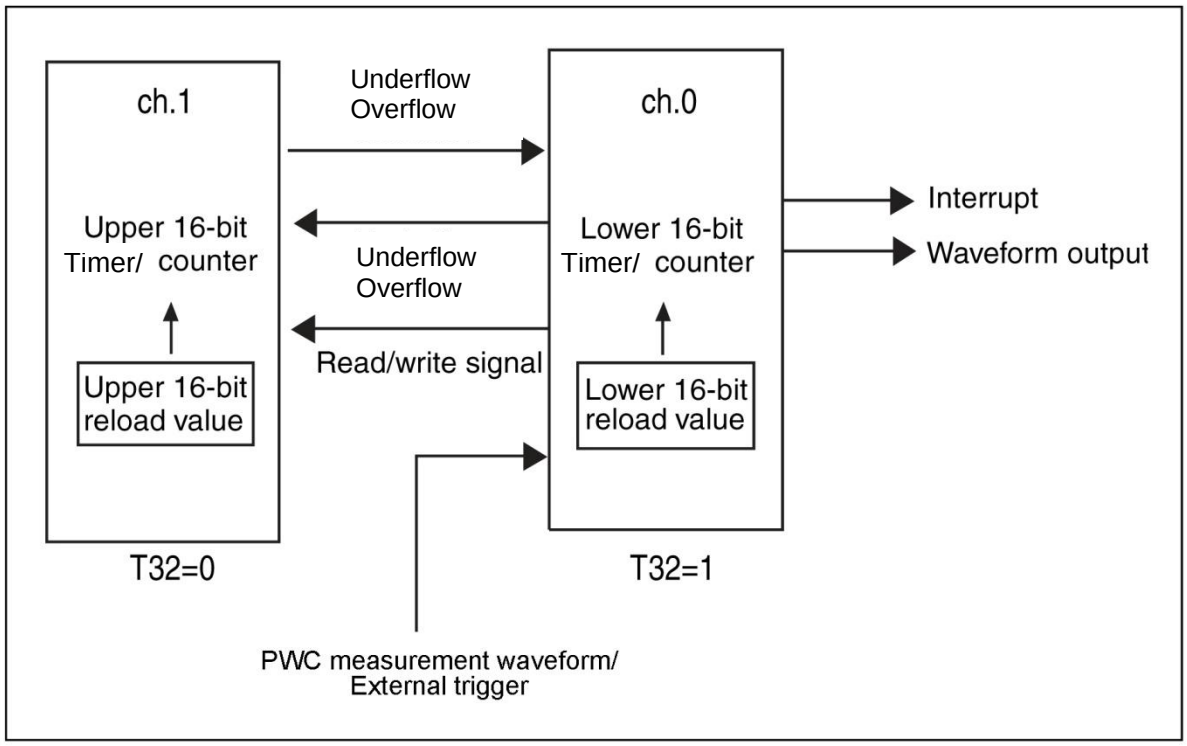
- Base timer x timer control register (BTxTMCR)
- Base timer x status control register (BTxSTC)

This section explains the counting in the 32-bit timer mode.

1. When the 32-bit reload timer activates, the values in the odd-number channel base timer x cycle setting register (BTxPCSR) and the even-number channel base timer x cycle setting register (BTxPCSR) (lower 16-bit) are loaded to the down counter.
2. The down counter starts counting as a 32-bit counter with the even-number channels serving as the lower 16-bit and the odd-number channels as the upper 16-bit.
3. When the down counter underflows, the UDIR bit of the base timer x timer control register (BTxTMCR) of the even-number channels changes to "1".

The following figure shows the channel configuration in 32-bit timer mode.

Figure 18-13. Configuration in 32-bit Timer Mode



Notes:

- The value of the down counter can be checked by reading the base timer x timer register (BTxTMR). In the 32-bit timer mode, it must be read in the order of the lower 16-bit (even-number channel) -> upper 16-bit (odd-number channel).
- In 32-bit timer mode, the operation of the 32-bit reload timer conforms to the settings of the even-number channels. Therefore, activation triggers and interrupt requests from even-number channels are valid. The output signal (TOUT) from an odd-number channel pin is fixed to "L" level.

18.5.4.5 Interrupts

This section explains interrupts of the base timer.

An interrupt request is generated in one of the following events:

- An activation trigger is detected. (trigger interrupt request)
- An underflow occurs. (underflow interrupt request)

Table 18-5. Interrupt Occurrence Conditions

Interrupt Request	Interrupt Request Flag	Permission of Interrupt Request	Interrupt Request Clear
Trigger interrupt request	BTxSTC: TGIR=1	BTxSTC: TGIE=1	Set the TGIR bit of BTxSTC to "0".
Underflow interrupt request	BTxSTC: UDIR=1	BTxSTC: UDIE=1	Set the UDIR bit of BTxSTC to "0".

Notes:

- Once the generation of an interrupt request is enabled while the interrupt request flag is "1", an interrupt request will be issued when the interrupt is enabled. To enable the generation of an interrupt request, perform one of the following operations:
  - ☐ Clear the current interrupt request before enabling the generation of an interrupt request.
  - ☐ Clear the current interrupt request when enabling the interrupt.
- Either clear the current interrupt request after disabling the generation of an interrupt request or clear the current interrupt request within the interrupt processing routine.
- For interrupt vector numbers used for issuing an interrupt request, see "List of Interrupt Vector" in "Appendix."
- Set the interrupt level corresponding to the interrupt vector number in one of the interrupt control registers (ICR00 to ICR47). For information on interrupt level setting, see "Chapter: Interrupt Control (Interrupt Controller)".

18.5.4.6 Precautions for Using this Device

This section explains precautions for using this device.

Note the following when using the 16/32-bit reload timer:

Notes on Program Setting

- Change the following bits of the base timer x timer control register (BTxTMCR) after stopping the 16-bit down counter by resetting CTEN bit to "0" (CTEN=0).
  - ☐ CKS2 to CKS0 bits
  - ☐ EGS1 and EGS0 bits
  - ☐ T32 bit
  - ☐ FMD2 to FMD0 bits
  - ☐ MDSE bit
- All registers are initialized when the FMD2 to FMD0 bits of the timer control register (BTxTMCR) are set to "000" to select reset mode.
  Before the base timer function or T32 bit can be changed, the base timer must be reset once. Except when rewriting the status of FMD2 to FMD0 bits or T32 bit of the timer control register (BTxTMCR) after a reset, be sure to set the FMD2 to FMD0 bits to "000" to select the reset mode. Then, rewrite the status of these bits.

Notes on Operations

- If the count timing of the down counter and the load timing occur at the same time, the load operation is given precedence.
- If a 16/32-bit reload timer activation trigger is detected when counting ends in one-shot mode, the value (cycle) set in the base timer x cycle setting register (BTxPCSR) is loaded to the 16-bit down counter, which begins counting.
  A different signal (external clock, external activation trigger, waveform) I/O operation can be selected using the base timer I/O selection function.

Note on Interrupts

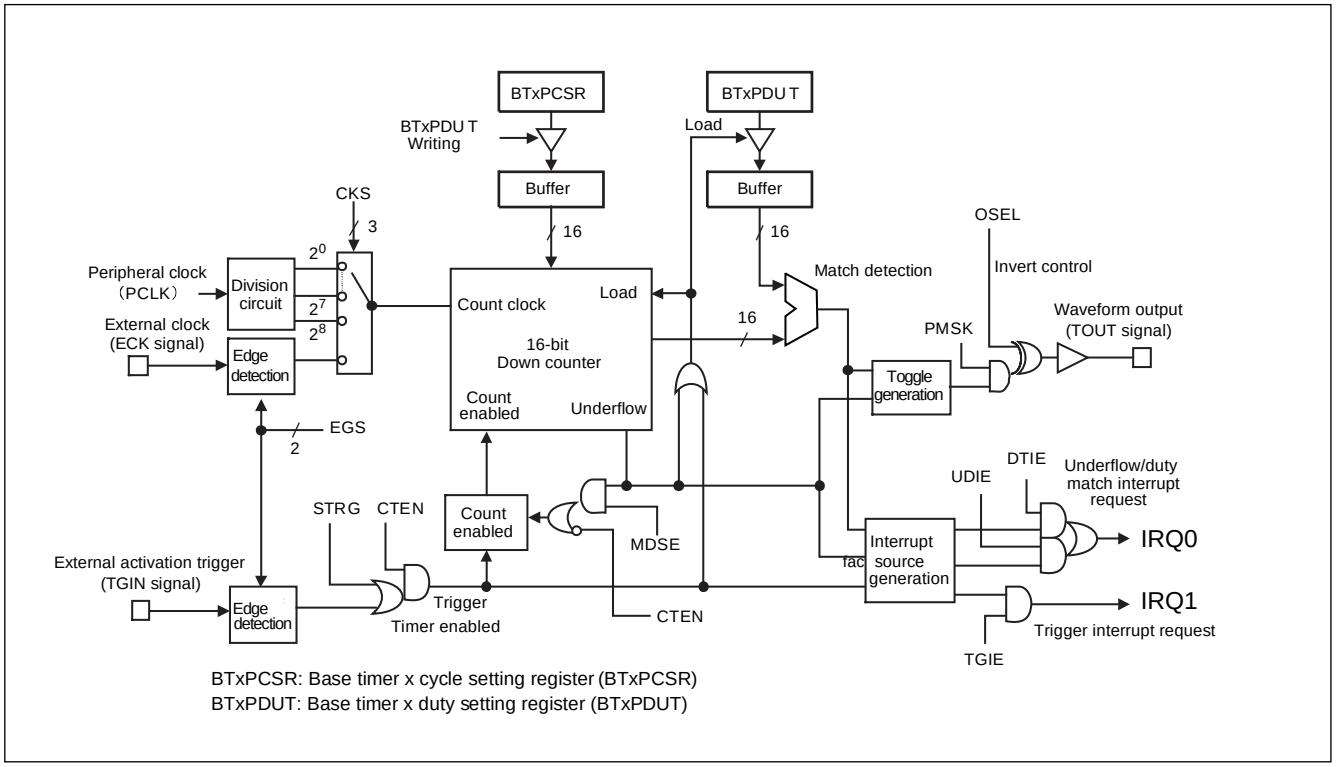
If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

18.5.5 16-bit PWM Timer Operation

This section explains the 16-bit PWM timer operation.

This section explains the operation performed when the base timer included in this series is used as the 16-bit PWM timer. An example is also given to set various operation conditions.

Figure 18-14. Block Diagram (16-bit PWM Timer Operation)



18.5.5.1 Overview

This section explains the overview of the 16-bit PWM timer operation.

The 16-bit PWM timer sets the cycle in the cycle setting register (BTxPCSR) and the duty in the duty setting register (BTxPDUT). A desired waveform (TOUT signal) can be output by setting values in these registers.

The 16-bit PWM timer starts decreasing from the value set in the base timer x cycle setting register (BTxPCSR). When the value of the down counter matches the value of the duty setting register (BTxPDUT), the output signal (TOUT) level is inverted. When the down counter underflows, the output level is inverted again. This method enables output of a desired waveform (TOUT signal) with a cycle and duty.

One of two 16-bit PWM timer operation modes can be selected using the MDSE bit of the timer control register (BTxTMCR) as follows:

- Reload mode (MDSE = 0): In this mode, when the 16-bit down counter underflows, the preset cycle is reloaded to allow the timer to restart counting.
- One-shot mode (MDSE = 1): Once the 16-bit down counter underflows, the counter will no longer count.

18.5.5.2 Operation in Reload Mode

This section explains the operation in reload mode.

This section explains the operation in reload mode.

Overview

In this mode, the value set in the base timer x cycle setting register (BTxPCSR) is reloaded every time an underflow occurs to ensure that countdown is continued. To use this mode, set reload mode by resetting the MDSE bit of the base timer x timer control register (BTxTMCR) to "0" (MDSE=0).

Operation

Activation

Activate the 16-bit PWM timer with the following procedure:

1. Permit the 16-bit PWM timer operation by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "1" (CTEN=1).
   - ☐ The 16-bit PWM timer begins to wait for an activation trigger.
2. Enter an activation trigger by one of the following methods:
   - ☐ Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
   - ☐ Enter an effective edge (an edge set in the EGS1 and EGS0 bits) for an external activation trigger (TGIN signal).

The 16-bit down counter starts decreasing from the value set in the base timer x cycle setting register (BTxPCSR).

Notes:

- The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01/BTSEL23).
- After a 16-bit PWM timer activation trigger is detected, the following time is required before the value set in the base timer x cycle setting register (BTxPCSR) can be loaded to the 16-bit down counter:
  - If a software trigger is input: 1T (T: Count clock cycle))
  - If an external event trigger is used: 2T to 3T (T: Count clock cycle)

Counting operation

When an activation trigger is input, the 16-bit down counter, in synchronization with the count clock, starts decreasing from the value set in the cycle setting register (BTxPCSR).
When the value of the 16-bit down counter matches the value of the duty setting register (BTxPDUT), the operation is performed as follows:

- The DTIR bit of the status control register (BTxSTC) changes to "1".
- The level of the output signal (TOUT) is inverted.
- Countdown is continued. Later, when the 16-bit down counter underflows, the operation is performed as follows:
- The UDIR bit of the status control register (BTxSTC) changes to "1" and the level of the output signal (TOUT) is inverted.
- The value of the cycle setting register (BTxPCSR) is reloaded to continue countdown.

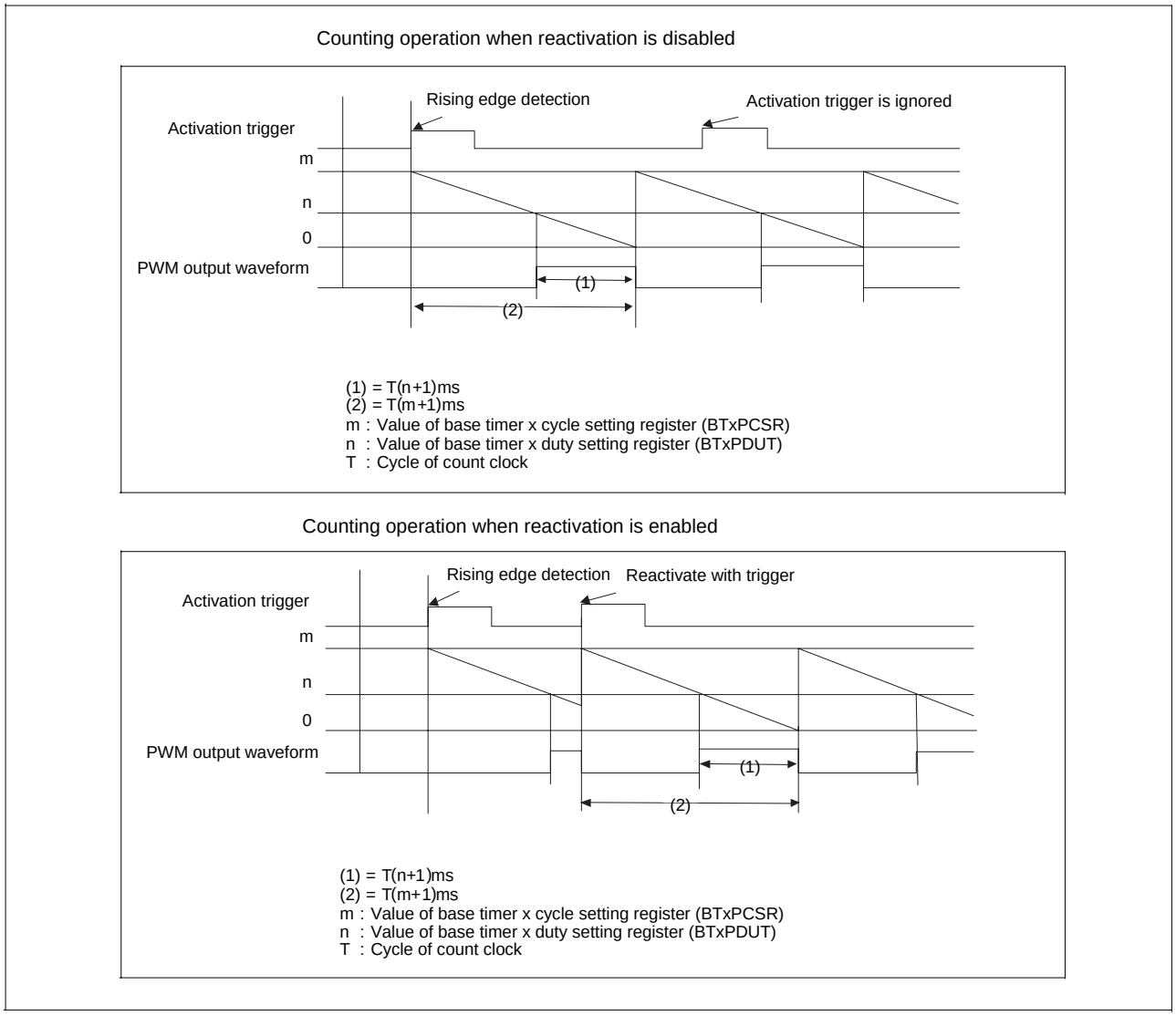
Every time an underflow occurs, the value of the cycle setting register (BTxPCSR) is reloaded to continue counting.
Operation to be performed when an activation trigger is input during counting depends on whether reactivation is permitted based on the RTGEN bit of the timer control register (BTxTMCR).

- If reactivation is not permitted (RTGEN = 0): Any activation trigger is ignored when it is entered during counting.

- If reactivation is permitted (RTGEN = 1): The TGIR bit of the base timer x status control register (BTxSTC) changes to "1". In addition, the value set in the base timer x cycle setting register (BTxPCSR) is reloaded to the 16-bit down counter, which begins counting.

These operations are shown below.

Figure 18-15. Counting Operation



Note:

If the count timing of the 16-bit down counter and the load timing occur at the same time, the load operation is given precedence.

Output Waveform

The waveform (TOUT signal) of the 16-bit PWM timer can be output. The waveform (TOUT signal) to be output varies according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR).

Normal polarity (OSEL = 0)

- When the 16-bit PWM timer is activated: "L" level
- When a duty match occurs: "H" level
- When an underflow occurs: "L" level

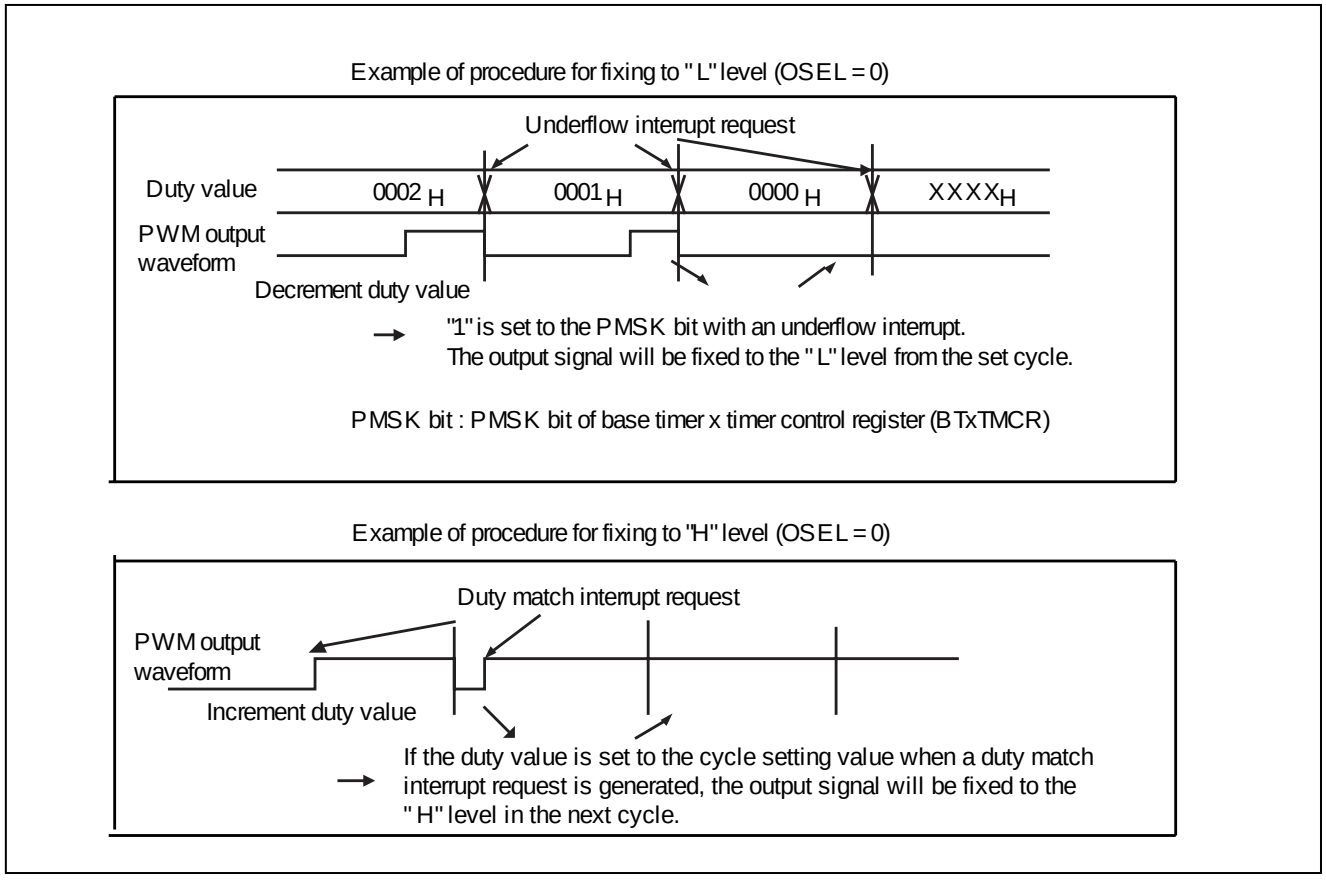
Inverted polarity (OSEL = 1)

- When the 16-bit PWM timer is activated: "H" level
- When a duty match occurs: "L" level
- When an underflow occurs: "H" level

The output (TOUT signal) can be fixed at the "L" or "H" level.

The output level varies depending on the setting of the OSEL bit of the base timer x timer control register (BTxTMCR). Examples of procedures are shown below.

Figure 18-16. Examples of Procedures for Fixing to "L" and "H" Levels



Note:

The output method and output destination of the waveform (TOUT signal) from the 16-bit PWM timer depend on the following settings:

- Base timer I/O mode
- TIOA0 to TIOA3 pin functions

Interrupt Generation Timing

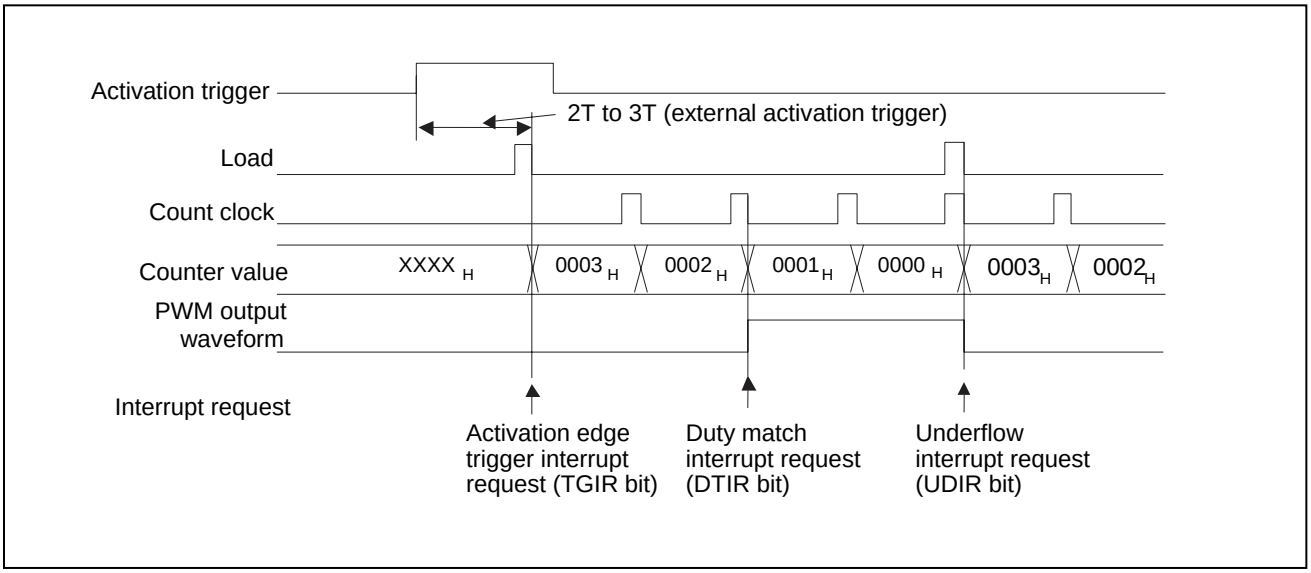
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- An activation trigger is detected.
- The value of the 16-bit down counter matches the value of the base timer x duty setting register (BTxPDUT)
- An underflow occurs.

An example of interrupt request generation timing using the following settings is shown below.

- Value of the cycle setting register (BTxPCSR) = 0003_H
- Value of the duty setting register (BTxPDUT) = 0001_H

Figure 18-17. Interrupt Request Generation Timing Chart



18.5.5.3 Operation in One-Shot Mode

This section explains the operation in one-shot mode.

This section explains the operation in one-shot mode.

Counting Operation

In this mode, counting stops if an underflow occurs when the value of the 16-bit down counter changes from the value set in the cycle setting register (BTxPCSR) to "$FFFF_H$".

To use this mode, set one-shot mode by setting the MDSE bit of the timer control register (BTxTMCR) to "1" (MDSE=1).

Activation

It is the same operation as in reload mode. See "Operation" in "18.5.5.2 Operation in Reload Mode".

Counting operation

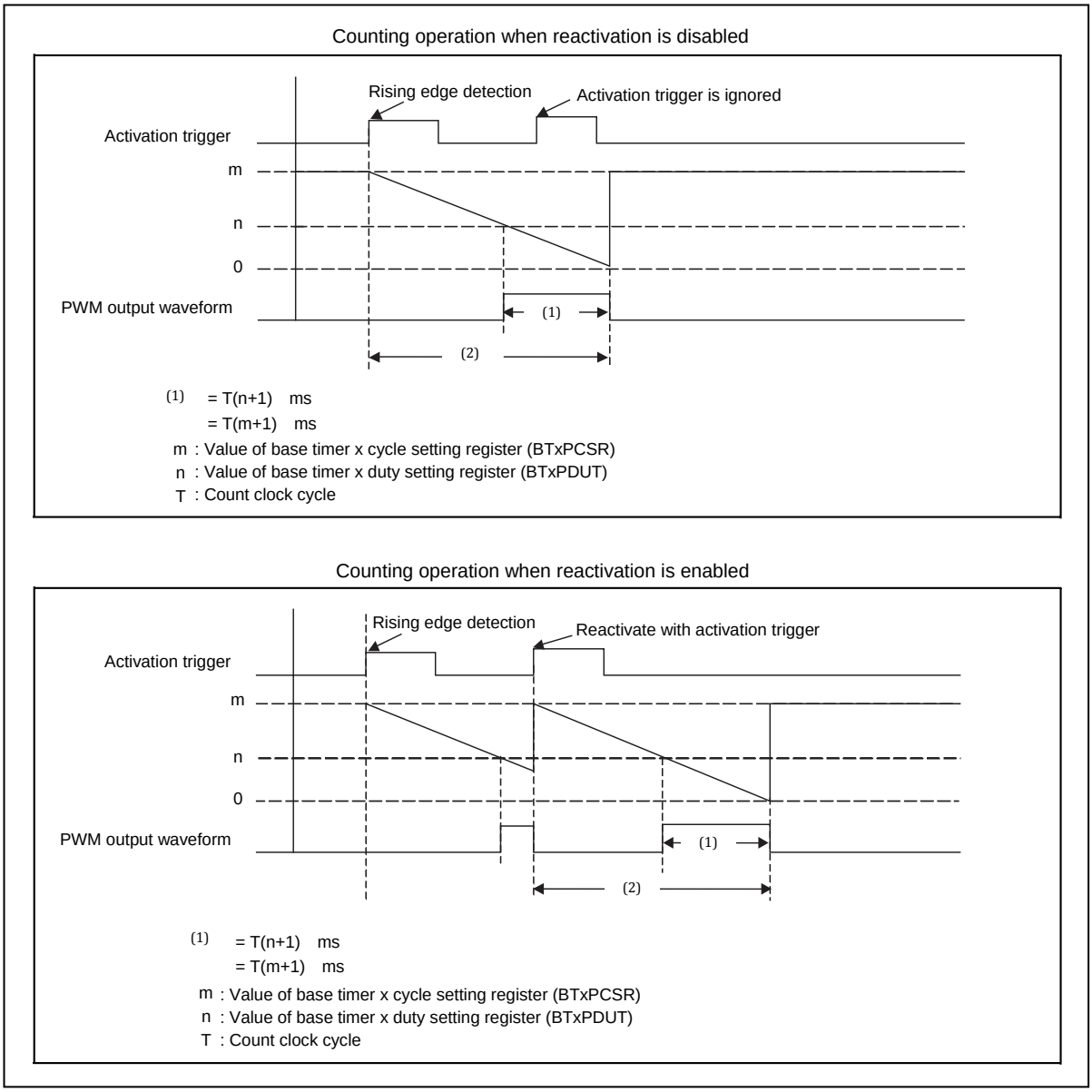
When an activation trigger is input, the 16-bit down counter, in synchronization with the count clock, starts decreasing from the value set in the cycle setting register (BTxPCSR). When the value of the 16-bit down counter matches the value of the duty setting register (BTxPDUT), the operation is performed as follows:

- The DTIR bit of the base timer x status control register (BTxSTC) changes to "1".
- The level of the output signal (TOUT signal) is inverted.
- Countdown is continued. Later, when the 16-bit down counter underflows, the operation is performed as follows:
  - ☐ The UDIR bit of the base timer x status control register (BTxSTC) changes to "1"
  - ☐ The level of the output signal (TOUT signal) is inverted.
  - ☐ Counting stops (The 16-bit down counter stops at the value "$FFFF_H$").

Operation to be performed when an activation trigger is input during counting depends on whether reactivation is permitted based on the RTGEN bit of the timer control register (BTxTMCR).

- If reactivation is not permitted (RTGEN = 0): Any activation trigger is ignored when it is entered during counting.
- If reactivation is permitted (RTGEN = 1): The TGIR bit of the base timer x status control register (BTxSTC) changes to "1". In addition, the value set in the base timer x cycle setting register (BTxPCSR) is reloaded to the 16-bit down counter, which begins counting.

Figure 18-18. Counting Operation



Note:

If a 16-bit PWM timer activation trigger is detected when counting ends, the value set in the cycle setting register (BTxPCSR) is loaded to the 16-bit down counter, which begins counting.

Output Waveform

It is the same operation as in reload mode. See "Output Waveform" in "18.5.5.2 Operation in Reload Mode".

Interrupt Generation Timing

It is the same operation as in reload mode. See "Interrupt Generation Timing" in "18.5.5.2 Operation in Reload Mode".

18.5.5.4 Interrupt

This section explains interrupts of the base timer.

An interrupt request is generated in one of the following events:

- An activation trigger is detected. (trigger interrupt request)
- The value of the 16-bit down counter matches the value of (the base timer x duty setting register (BTxPDUT)). (duty match interrupt request)
- An underflow occurs. (underflow interrupt request)

Table 18-6. Interrupt Occurrence Conditions

Interrupt Request	Interrupt Request Flag	Permission of Interrupt Request	Interrupt Request Clear
Trigger interrupt request	BTxSTC: TGIR=1	BTxSTC: TGIE=1	Set the TGIR bit of BTxSTC to "0".
Duty match interrupt request	BTxSTC: DTIR=1	BTxSTC: DTIE=1	Set the DTIR bit of BTxSTC to "0".
Underflow interrupt request	BTxSTC: UDIR=1	BTxSTC: UDIE=1	Set the UDIR bit of BTxSTC to "0".

Notes:

- Once the generation of an interrupt request is enabled while the interrupt request flag is "1", an interrupt request will be issued when the interrupt is enabled. To enable the generation of an interrupt request, perform one of the following operations:
  - ☐ Clear the current interrupt request before enabling the generation of an interrupt request.
  - ☐ Clear the current interrupt request when enabling the interrupt.
- Either clear the current interrupt request after disabling the generation of an interrupt request or clear the current interrupt request within the interrupt processing routine.
- For interrupt vector numbers used for issuing an interrupt request, see "List of Interrupt Vector" in "Appendix."
- Set the interrupt level corresponding to the interrupt vector number in one of the interrupt control registers (ICR00 to ICR47). For information on interrupt level setting, see "Chapter: Interrupt Control (Interrupt Controller)".

18.5.5.5 Precautions for Using this Device

This section explains precautions for using this device.

Note the following when using the 16-bit PWM timer:

Notes on Program Setting

- Change the following bits of the timer control register (BTxTMCR) only after stopping the 16-bit down counter by resetting the CTEN bit to "0" (CTEN=0).
  - ☐ CKS2 to CKS0 bits
  - ☐ EGS1 and EGS0 bits
  - ☐ FMD2 to FMD0 bits
  - ☐ MDSE bit
- All registers are initialized when the FMD2 to FMD0 bits of the base timer x timer control register (BTxTMCR) are set to "000" to select reset mode.
- Before the base timer function can be changed, the base timer must be reset once. Except when rewriting the FMD2 to FMD0 bits of the base timer x timer control register (BTxTMCR) after reset, be sure to clear FMD2 to FMD0 bits to "000" to select the reset mode, and then select a base timer function using the FMD2 to FMD0 bits again.
- To set 16-bit PWM timer cycles or duties, proceed as follows:
  1. Select the 16-bit PWM timer as the base timer function by setting the FMD2 to FMD0 bits of the base timer x timer control register (BTxTMCR) to "001" (FMD2 to FMD0=001).
  2. Set the cycle in the base timer x cycle setting register (BTxPCSR).
  3. Set the duty in the base timer x duty setting register (BTxPDUT).

Notes on Operations

- If the count timing of the 16-bit down counter and the load timing occur at the same time, the load operation is given precedence.
- When a 16-bit PWM timer reactivation trigger is detected when counting ends in one-shot mode, the value in the base timer x cycle setting register (BTxPCSR) is loaded to the 16-bit down counter, which then starts counting.
- A different signal (external clock, external activation trigger, waveform) I/O operation can be selected using the base timer I/O selection function.

Note on Interrupts

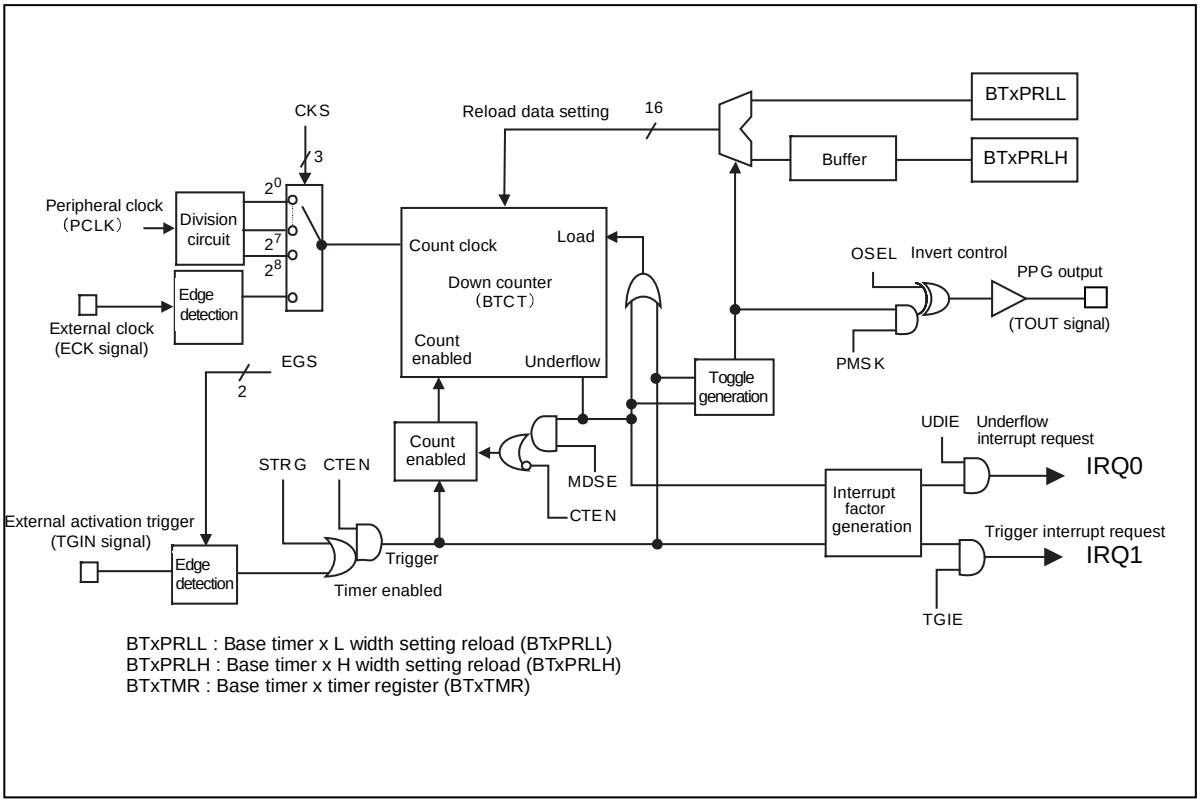
If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

18.5.6 16-bit PPG Timer Operation

This section explains the 16-bit PPG timer operation.

This section explains the operation performed when the base timer included in this series is used as the 16-bit PPG timer. Examples of procedures for setting various operating conditions are also provided.

Figure 18-19. Block Diagram (16-bit PPG Timer Operation)



18.5.6.1 Overview

This section explains the overview of the 16-bit PPG timer operation.

The 16-bit PPG timer, once activated, decreases from the value initially specified by the base timer x L width setting reload register (BTxPRLL). When counting down from the value set in the L width setting reload register (BTxPRLL) is completed, the timer begins counting down from the value set in the H width setting reload register (BTxPRLH).

When counting down from the value set in each register is completed, the output signal (TOUT) inverts its level. Therefore, by configuring the L width setting reload register (BTxPRLL) and H width setting reload register (BTxPRLH), you can arbitrarily set the widths of the "L" and "H" levels.

One of two 16-bit PPG timer operation modes can be selected using the MDSE bit of the timer control register (BTxTMCR) as follows:

- Reload mode (MDSE = 0): A sequence of "L"-level and "H"-level signals (consecutive pulses) is output.
- One-shot mode (MDSE = 1): A string of one "L"-level signal and one "H"-level signal (single pulses) is output.

18.5.6.2 Pulse Width Calculation Method

This section explains the pulse width calculation method.

When the 16-bit PPG timer has counted down by the value set in the L width setting reload register (BTxPRLL) or base timer x H width setting reload register (BTxPRLH) plus 1, the output signal (TOUT) inverts its level. Therefore, the pulse width of the signal to be output is obtained by the following formula:

Example: If the output polarity is normal:

"L" level pulse width = T x (L + 1)
"H" level pulse width = T x (H + 1)

T: Count clock cycle
L: Value set in the base timer x L width setting reload register (BTxPRLL)
H: Value set in the base timer x H width setting reload register (BTxPRLH)

This means that when the L width setting reload register (BTxPRLL) and H width setting reload register (BTxPRLH) are set to "0000_H", the pulse width will be equal to one cycle of the count clock. When they are set to "$FFFF_H$", the pulse width will be equal to 65536 cycles of the count clock.

18.5.6.3 Operation in Reload Mode

This section explains the operation in reload mode.

This section explains the operation in reload mode.

Overview

In this mode, the values set in the base timer x L width setting reload register (BTxPRLL) and base timer x H width setting reload register (BTxPRLH) are alternately reloaded to the down counter to ensure that the down counter continues to count down. A desired pulse width can be output continuously by rewriting the base timer x L width setting reload register (BTxPRLL) and base timer x H width setting reload register (BTxPRLH) each time an underflow interrupt request is issued. To use this mode, set reload mode by resetting the MDSE bit of the base timer x timer control register (BTxTMCR) to "0" (MDSE=0).

Operation

Activation

Activate the 16-bit PPG timer with the following procedure:

1. Permit the 16-bit PPG timer operation by setting the CTEN bit of the timer control register (BTxTMCR) to "1" (CTEN=1). The 16-bit PPG timer begins to wait for an activation trigger.
2. Enter an activation trigger by one of the following methods:
   - ☐ Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
   - ☐ Enter an effective edge (an edge set in the EGS1 and EGS0 bits) for an external activation trigger (TGIN signal).

Notes:

- The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01/BTSEL23).
- After a 16-bit PPG timer activation trigger is detected, the following time is required before the value (cycle) set in the L width setting reload register (BTxPRLL) can be loaded to the 16-bit down counter:
  - ☐ If a software trigger is input: 1T (T: Count clock cycle)
  - ☐ If an external event trigger is used: 2T to 3T (T: Count clock cycle)

Counting operation

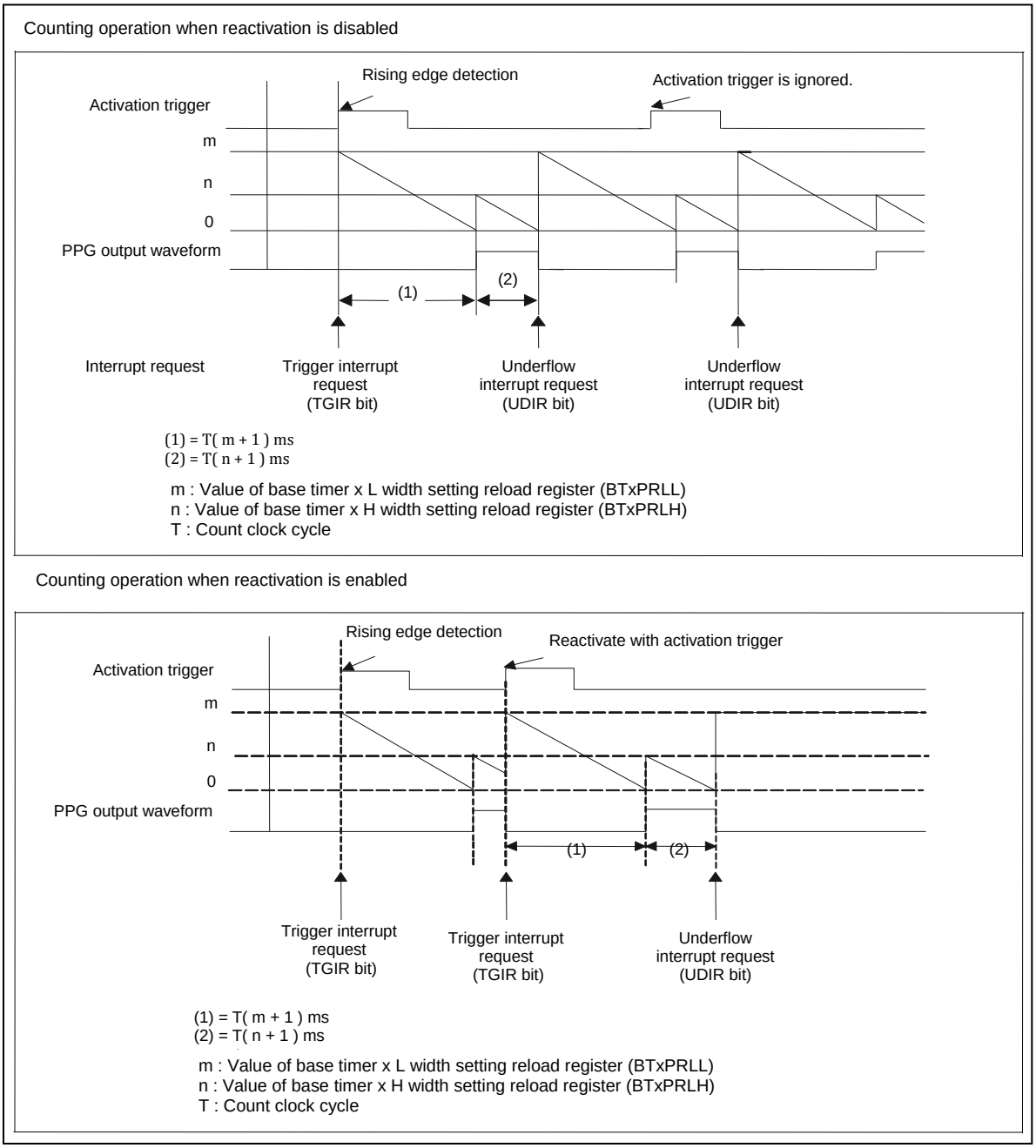
Counting operation initiated by the entry of an activation trigger is explained below, using an example where the OSEL bit of the timer control register (BTxTMCR) is set for normal polarity (OSEL = 0).

1. The value set in the L width setting reload register (BTxPRLL) is transferred to the 16-bit down counter and the value set in the base timer x H width setting reload register (BTxPRLH) is transferred to the buffer. The 16-bit down counter begins to count down from the value of the L width setting reload register (BTxPRLL). The output signal (TOUT) is at the "L" level.
2. The 16-bit down counter completes counting down from the value of L width setting reload register (BTxPRLL).
3. The buffered value of H width setting reload register (BTxPRLH) is reloaded to the 16-bit down counter, which continues counting down. The output signal (TOUT) is at the "H" level.
4. The 16-bit down counter completes counting down from the value of H width setting reload register (BTxPRLH), thus causing an underflow.
5. The value of L width setting reload register (BTxPRLL) is reloaded to the 16-bit down counter, which continues count down. The output signal (TOUT) is at the "L" level. In addition, the value of the H width setting reload register (BTxPRLH) is transferred to the buffer.
6. Steps 2 to 5 are repeated to continue counting.

Operation that is performed if reactivation is permitted or not during counting depends on whether reactivation is permitted based on the RTGEN bit of the timer control register (BTxTMCR).

- If reactivation is not permitted (RTGEN = 0): Any activation trigger is ignored when it is entered during counting.
- If reactivation is permitted (RTGEN = 1): The TGIR bit of the base timer x status control register (BTxSTC) changes to "1". In addition, the value of L width setting reload register (BTxPRLL) is reloaded to the 16-bit down counter, which starts counting.

Figure 18-20. Example of Counting Operation in Reload Mode



Notes:

- The output method and output destination of the output signal (TOUT) from the 16-bit PPG timer depend on the following settings:
  - ☐ Base timer I/O mode
  - ☐ TIOA0 to TIOA3 pin functions
- If the count timing of the 16-bit down counter and the load timing occur at the same time, the load operation is given precedence.

Write Timing

The values of the base timer x L width setting reload register (BTxPRLL) and base timer x H width setting reload register (BTxPRLH) are reloaded at the following timing:

The value set in the base timer x L width setting reload register (BTxPRLL)

It is loaded to the 16-bit down counter in one of the following events:

- An activation trigger is detected.
- An underflow occurs after counting down from the value of the base timer x H width setting reload register (BTxPRLH) is completed.

The value set in the base timer x H width setting reload register (BTxPRLH)

It is transferred to the buffer in one of the following events:

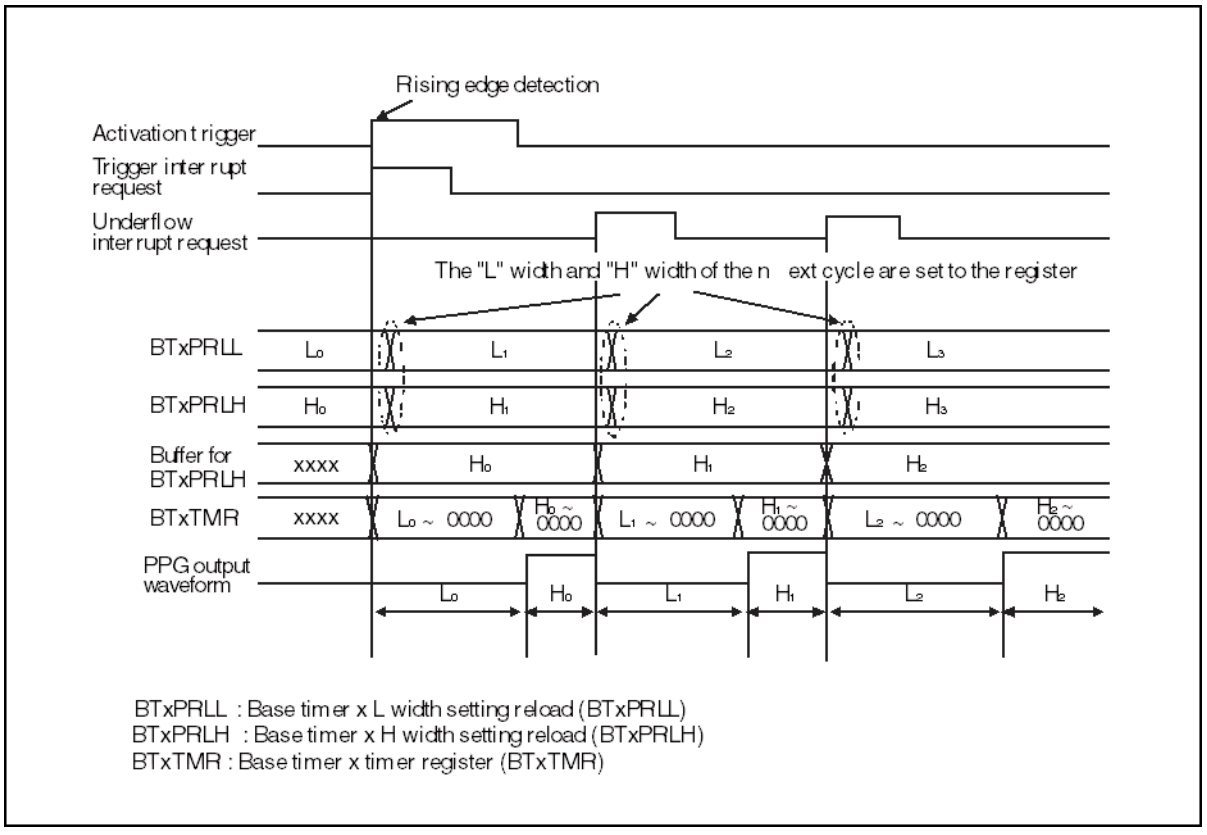
- An activation trigger is detected.
- An underflow occurs after counting down from the value of the base timer x H width setting reload register (BTxPRLH) is completed.

The content of the buffer is loaded to the 16-bit down counter in the following event:

- Counting down from the value of the base timer x L width setting reload register (BTxPRLL) is completed.

Therefore, rewrite the base timer x L width setting reload register (BTxPRLL) and base timer x H width setting reload register (BTxPRLH) during the period from the time an underflow occurs (the UDIR bit of the status control register (BTxSTC) changes to "1") to the time counting based on the next cycle begins. The new data will be effective as the next cycle.

Figure 18-21. Write Timing



Interrupt Generation Timing

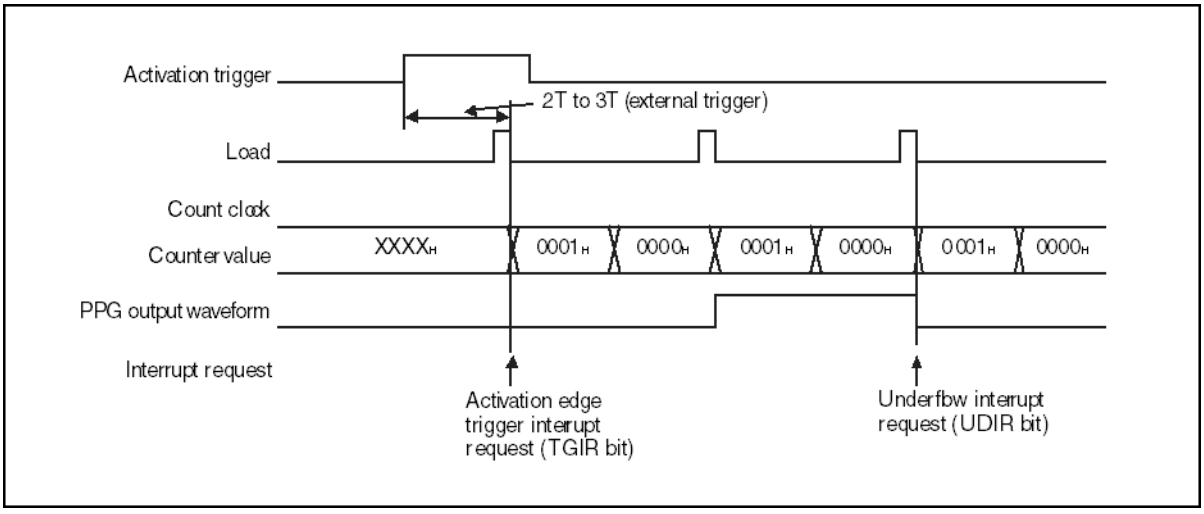
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- An activation trigger is detected.
- An underflow occurs based on the value of H width setting reload register (BTxPRLH).

Figure 18-22 shows an interrupt generation timing chart, as an example of interrupt request generation timing using the following settings:

- Value of L width setting reload register (BTxPRLL) = 0001_H
- Value of H width setting reload register (BTxPRLH) = 0001_H

Figure 18-22. Interrupt Request Generation Timing Chart



18.5.6.4 Operation in One-Shot Mode

This section explains the operation in one-shot mode.

This section explains the operation in one-shot mode.

Counting Operation

Activation

It is the same operation as in reload mode. See "Operation" in "18.5.6.3 Operation in Reload Mode".

Counting operation

Counting operation initiated by the entry of an activation trigger is explained below, using an example where the OSEL bit of the timer control register (BTxTMCR) is set for normal polarity (OSEL = 0).

1. The value set in the base timer x L width setting reload register (BTxPRLL) is transferred to the 16-bit down counter and the value set in the base timer x H width setting reload register (BTxPRLH) is transferred to the buffer. The 16-bit down counter begins to count down from the value of the L width setting reload register (BTxPRLL). The output signal (TOUT) is at the "L" level.
2. The 16-bit down counter completes counting down from the value of L width setting reload register (BTxPRLL).
3. The buffered value of H width setting reload register (BTxPRLH) is reloaded to the 16-bit down counter, which continues counting down. The output signal (TOUT) is at the "H" level.
4. The 16-bit down counter completes counting down from the value of H width setting reload register (BTxPRLH), thus causing an underflow.
5. The counting stops.

Operation that is performed if reactivation is permitted or not during counting depends on whether reactivation is permitted based on the RTGEN bit of the timer control register (BTxTMCR).

- If reactivation is not permitted (RTGEN = 0): Any activation trigger is ignored when it is entered during counting.
- If reactivation is permitted (RTGEN =1): The TGIR bit of the status control register (BTxSTC) changes to "1". In addition, the value of L width setting reload register (BTxPRLL) is reloaded to the 16-bit down counter, which starts counting.

Figure 18-23. Example of Counting Operation If Reactivation Is Not Enabled

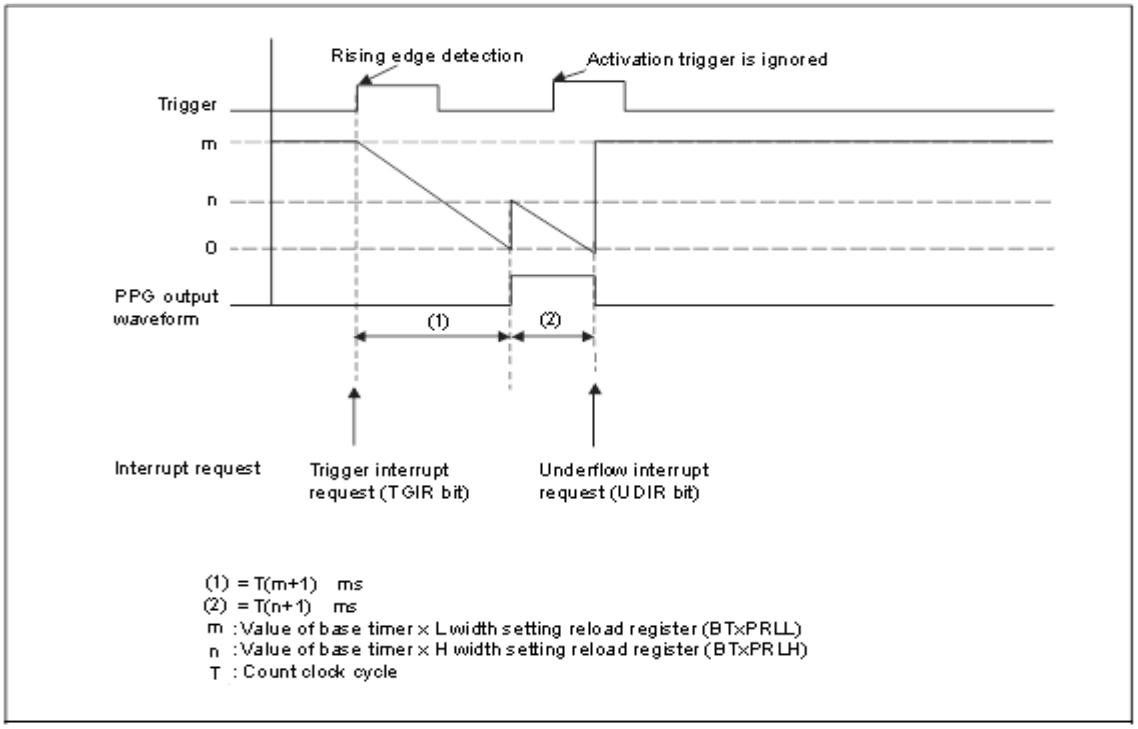
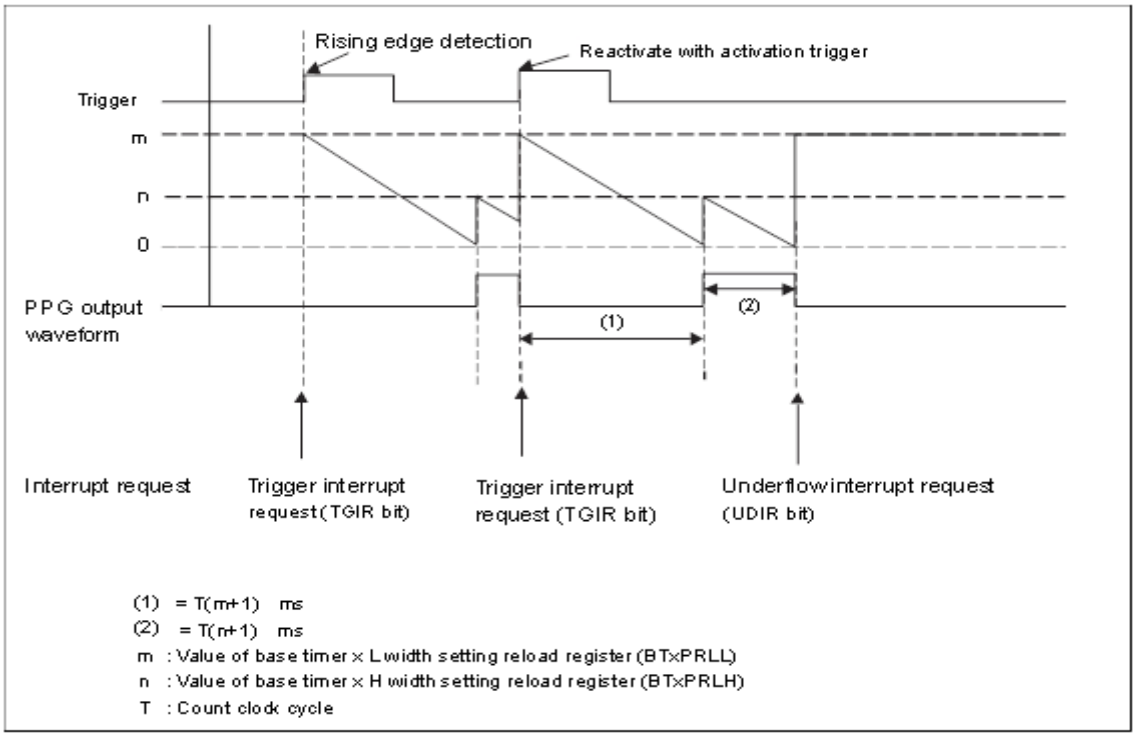


Figure 18-24. Example of Counting Operation If Reactivation Is Enabled



Notes:

- The output method and output destination of the output signal (TOUT) from the 16-bit PPG timer depend on the following settings:
  - Base timer I/O mode
  - TIOA0 to TIOA3 pin functions
- If a 16-bit PPG timer activation trigger is detected when counting ends, the value (cycle) of L width setting reload register (BTxPRLL) is loaded to the 16-bit down counter, which starts counting.

Interrupt Generation Timing

It is the same operation as in reload mode. See "Interrupt Generation Timing" in "18.5.6.3 Operation in Reload Mode".

18.5.6.5 Interrupt

This section explains interrupts of the base timer.

An interrupt request is generated in one of the following events:

- An activation trigger is detected. (trigger interrupt request)
- An underflow occurs based on the value of H width setting reload register (BTxPRLH). (underflow interrupt request)

Table 18-7. Interrupt Occurrence Conditions

Interrupt Request	Interrupt Request Flag	Permission of Interrupt Request	Interrupt Request Clear
Trigger interrupt request	BTxSTC: TGIR=1	BTxSTC: TGIE=1	Set the TGIR bit of BTxSTC to "0".
Underflow interrupt request	BTxSTC: UDIR=1	BTxSTC: UDIE=1	Set the UDIR bit of BTxSTC to "0".

Notes:

- Once the generation of an interrupt request is enabled while the interrupt request flag is "1", an interrupt request will be issued when the interrupt is enabled.
- To enable the generation of an interrupt request, perform one of the following operations:
  - ☐ Clear the current interrupt request before enabling the generation of an interrupt request.
  - ☐ Clear the current interrupt request when enabling the interrupt.
- Either clear the current interrupt request after disabling the generation of an interrupt request or clear the current interrupt request within the interrupt processing routine.
- Set an interrupt level corresponding to the interrupt vector number, using interrupt control registers (ICR00 to ICR47). For information on interrupt level setting, see "Chapter: Interrupt Control (Interrupt Controller)".

18.5.6.6 Precautions for Using this Device

This section explains precautions for using this device.

Note the following when using the 16-bit PPG timer:

Notes on Program Setting

- Change the following bits of the timer control register (BTxTMCR) only after stopping the 16-bit down counter by resetting the CTEN bit to "0" (CTEN=0).
  - ☐ CKS2 to CKS0 bits
  - ☐ EGS1 and EGS0 bits
  - ☐ FMD2 to FMD0 bits
  - ☐ MDSE bit
- All registers are initialized when the FMD2 to FMD0 bits of the timer control register (BTxTMCR) are set to "000" to select reset mode.
- Before the base timer function can be changed, the base timer must be reset once. Except when rewriting the FMD2 to FMD0 bits of timer control register (BTxTMCR) after reset, be sure to clear FMD2 to FMD0 bits to "000" to select the reset mode, and then select a base timer function using the FMD2 to FMD0 bits again.
- Set the 16-bit PPG timer in the following steps.
  1. Set the 16-bit PPG timer as the base timer function by setting the FMD2 to FMD0 bits of timer control register (BTxTMCR) to "010" (FMD2 to FMD0=010).
  2. Set the L width setting reload register (BTxPRLL).
  3. Set the H width setting reload register (BTxPRLH).

Notes on Operations

- If the count timing of the 16-bit down counter and the load timing occur at the same time, the load operation is given precedence.
- If a 16-bit PPG timer reactivation trigger is detected when counting ends in the one-shot mode, the value (cycle) of L width setting reload register (BTxPRLL) is loaded to the 16-bit down counter, which starts counting.
- A different signal (external clock, external activation trigger, waveform) I/O operation can be selected using the base timer I/O selection function.

Note on Interrupts

- If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

18.5.7 16/32-bit PWC Timer Operation

This section explains the 16/32-bit PWC timer operation.

This section explains the operation performed when the base timer included in this series is used as the 16/32-bit PWC timer. Examples of procedures for setting various operating conditions are also provided.

Figure 18-25. Block Diagram (16-bit PWC Timer Operation)

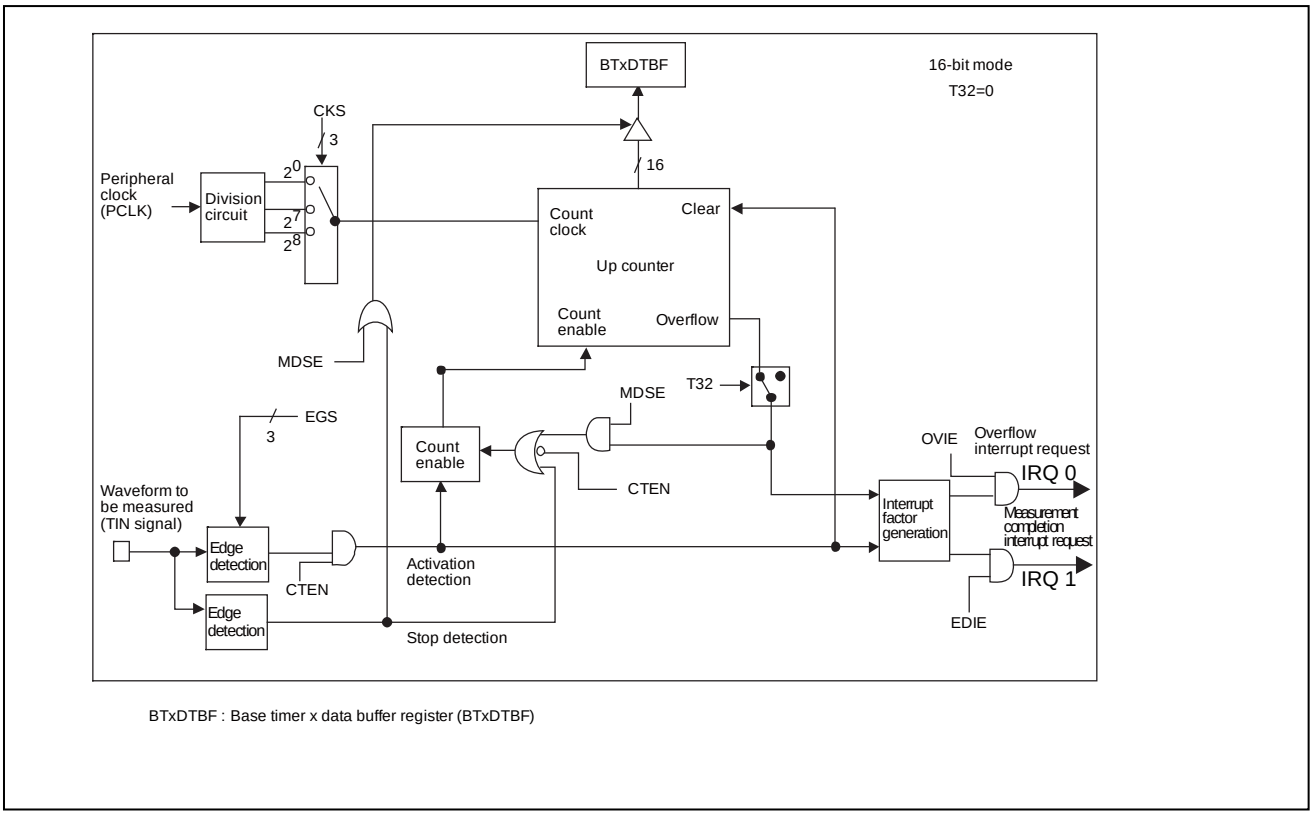
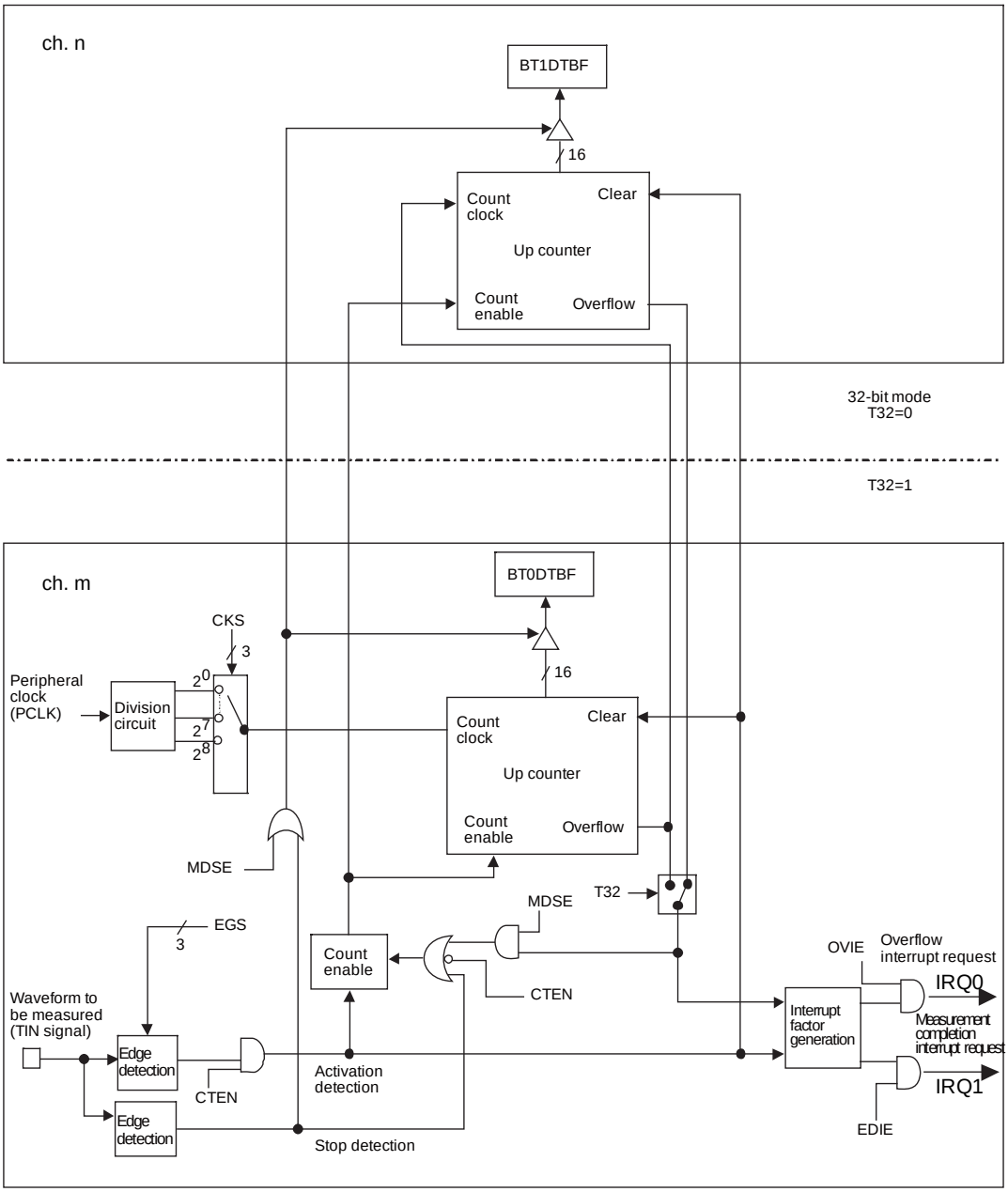


Figure 18-26. Block Diagram (32-bit PWC Timer Operation)



BTmDTBF: Base timer m data buffer register

BTnDTBF: Base timer n data buffer register

m: ch. 0/2 n: ch. 1/3

18.5.7.1 Overview

This section explains the overview of the 16/32-bit PWC timer operation.

The 16/32-bit PWC timer is used to measure the pulse width and cycle of input signals. When a measurement start edge is detected in an input signal (TIN), the counting up starts. This counting stops when a measurement end edge is detected. The counted value (that is, the measured result) is stored as the pulse width or cycles in the data buffer register (BTxDTBF).

The 16/32-bit PWC timer supports three modes: the timer mode, the operation mode, and measurement mode. The operation of the timer varies in accordance with a combination of these modes.

Note:

The input method of the TIN signal varies depending on the I/O mode that has been set by the I/O selection register (BTSEL01/BTSEL23). See "18.5.2 I/O Allocation".

Timer Mode

Either of the following timer modes can be selected using the T32 bit of the timer control register (BTxTMCR).

- 16-bit timer mode (T32 = 0): A 16-bit PWC timer can operate individually for each of the channels.
- 32-bit timer mode (T32 = 1): 2 channels can be cascaded and used as a 32-bit PWC timer.

See "18.5.7.3 32-bit Timer Mode Operation" for details on the operation in 32-bit timer mode.

Note:

The T32 bit setting differs between odd-number and even-number channels when the 32-bit timer mode is selected. For details, see "18.5.7.3 32-bit Timer Mode Operation".

Operation Mode

Either of the following two modes can be selected using the MDSE bit of the timer control register (BTxTMCR).

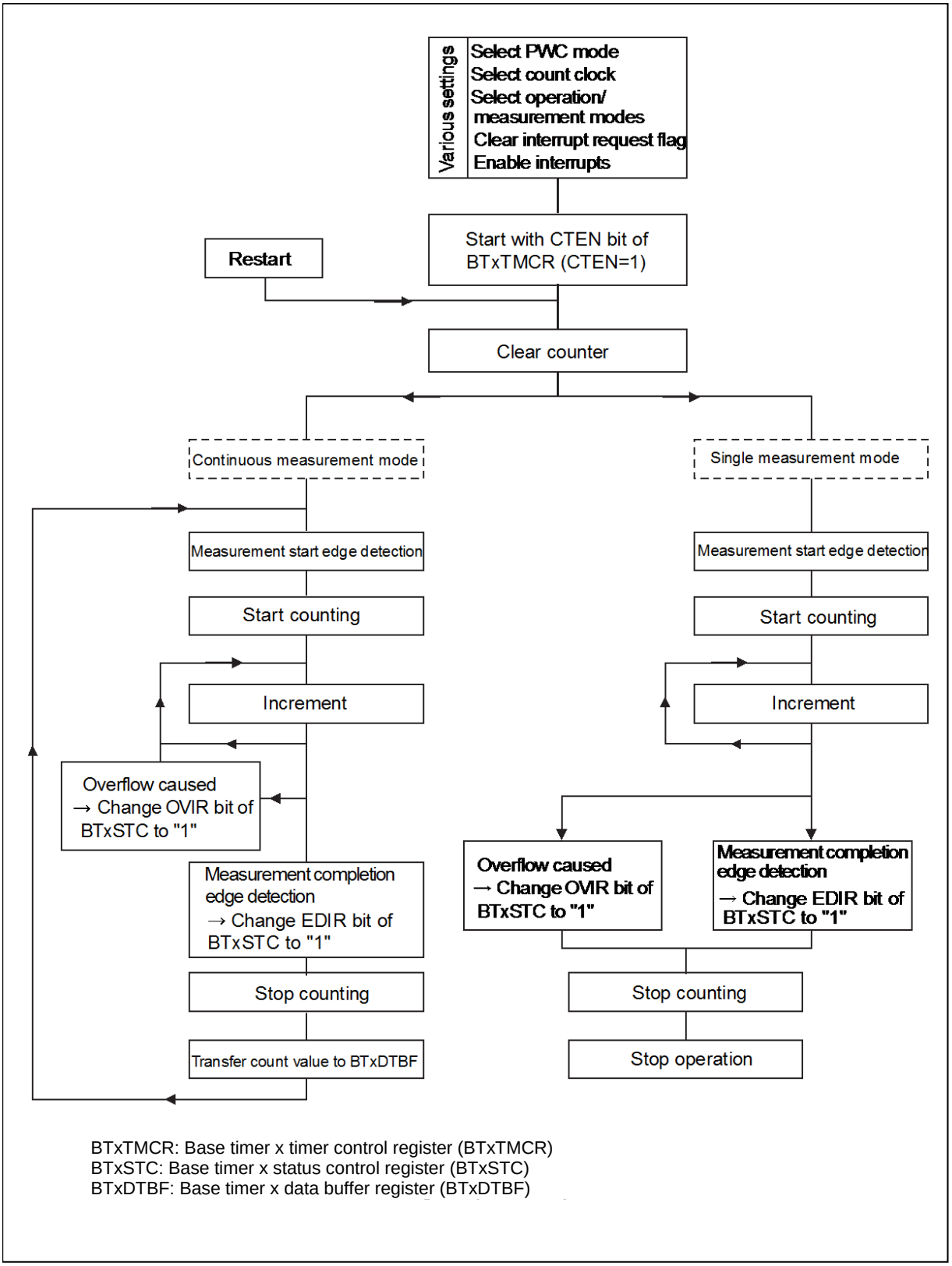
- Continuous measurement mode (MDSE = 0): In this mode, after one sequence of measurement is conducted, the input of the next measurement start edge is awaited and the detection of the next measurement start edge triggers another sequence of measurement.
- Single measurement mode (MDSE = 1): In this mode, measurement is conducted only once. Differences between the single and continuous measurement modes are listed on the table below.

Table 18-8. Differences between Single and Continuous Measurement Modes

	Single Measurement Mode	Continuous Measurement Mode
Measurement	Measurement stops when a measurement end edge is detected.	When a measurement end edge is detected, the measurement stops and the next measurement start edge is waited. When the next measurement start edge is detected, the measurement restarts.
BTxDTBF function	During measurement: The measured value is held. After measurement: The measurement result is held.	During measurement: The previous measurement result is held. After measurement: The measurement result is held.
During overflow	The measurement stops.	The measurement restarts from 0x0000

The following figure shows the standard operation flow.

Figure 18-27. Operation Flow



BTxTMCR: Base timer x timer control register (BTxTMCR)
BTxSTC: Base timer x status control register (BTxSTC)
BTxDTBF: Base timer x data buffer register (BTxDTBF)

Note:

In the continuous measurement mode, if the next measurement is completed before the measurement result has been read from the data buffer register (BTxDTBF), the value being held by the data buffer register (BTxDTBF) is overwritten by the new value. The old value is discarded. If it has occurred, the ERR bit of the status control register (BTxSTC) changes to "1". This ERR bit is cleared to "0" when a value is read from the base timer x data buffer register (BTxDTBF).

Measurement Mode

One of the following five modes can be selected using EGS2 to EGS0 bits of the timer control register (BTxTMCR).

Figure 18-28. Measurement Modes and their Explanation 1

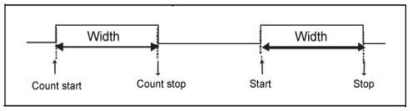
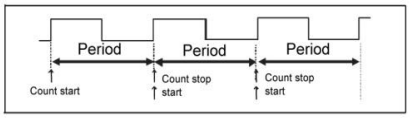
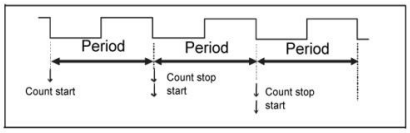
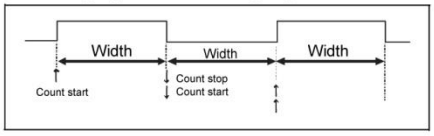
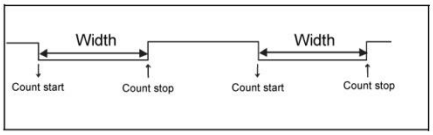
Measurement mode (EGS2 to EGS0)	Measurement description
Measurement of H pulse width (EGS2 to EGS0=000)	The width of the period which the "H" level signal is being input is measured.  Count (measurement) start: at rising edge detection Count (measurement) stop: at falling edge detection
Measurement of the cycle between rising edges (EGS2 to EGS0=001)	The cycle from the rising edge detection to the next rising edge detection is measured.  Count (measurement) start: at rising edge detection Count (measurement) stop: at rising edge detection
Measurement of the cycle between falling edges (EGS2 to EGS0=010)	The cycle from the falling edge detection to the next falling edge detection is measured.  Count (measurement) start: at falling edge detection Count (measurement) stop: at falling edge detection

Figure 18-29. Measurement Modes and their Explanation 2

Measurement mode (EGS2 to EGS0)	Measurement description
Measurement of the pulse width between all edges (EGS2 to EGS0=011)	The width between the edges input continuously is measured. •From rising edge detection to falling edge detection •From falling edge detection to rising edge detection  Count (measurement) start: at edge detection Count (measurement) stop: at edge detection
Measurement of L pulse width (EGS2 to EGS0=100)	The width of the period during which the "L" level signal being input is measured.  Count (measurement) start: at falling edge detection Count (measurement) stop: at rising edge detection

18.5.7.2 Operation during PWC Measurement

This section explains the operation during PWC measurement.

This section explains the operations during measurement. For explanation of "sensitive edges" (1) and (2) described below, see "Figure 18-28. Measurement Modes and their Explanation 1" and "Figure 18-29. Measurement Modes and their Explanation 2".

Activation

Activate the 16/32-bit PWC timer with the following procedure:

- Enable the 16/32-bit PWC timer operation by setting the CTEN bit of the timer control register (BTxTMCR) to "1" (CTEN=1).
  The counter value is cleared to "0000H" and the 16/32-bit PWC timer waits for an input of measurement start edge. (No counting occurs until an input of measurement start edge.)

Counting Operation

Operation in single measurement mode

If sensitive edge (1) is detected in the input signal (TIN) when a measurement start edge is waited, the up counter starts counting up from "0001_H" in synchronous with the count clock. If sensitive edge (2) is detected in the input signal (TIN), the up counter stops from operating.

During this time, the up counter value is stored in the data buffer register (BTxDTBF). An interrupt request can be generated at the end of measurement or at an occurrence of overflow.

Notes:

- In the single measurement mode, the counting stops if an overflow occurs.
- The input method of waveforms to be measured (TIN signal) varies depending on the I/O mode that has been set by the I/O selection register (BTSEL01/BTSEL23).

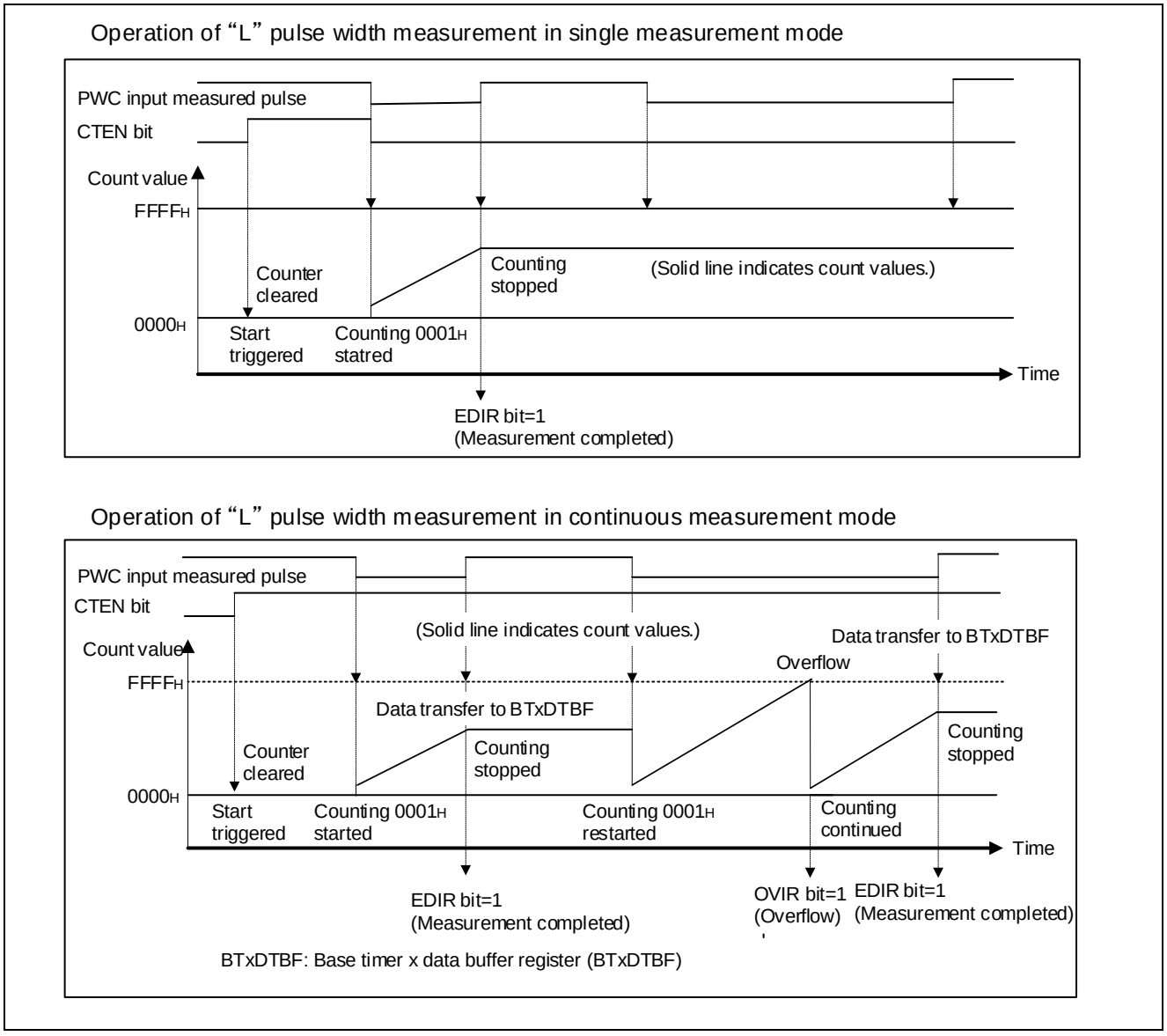
Operation in continuous measurement mode

If sensitive edge (1) is detected in the input signal (TIN) when a measurement start edge is waited, the up counter starts counting up from "0001_H" in synchronous with the count clock. If sensitive edge (2) is detected in the input signal (TIN), the up counter stops from operating and waits for an input of measurement start edge. During this time, the up counter value is stored in the data buffer register (BTxDTBF). If a rising edge of the input signal (TIN) is detected when a measurement start edge is waited, the up counter starts counting up from "0001_H" again. An interrupt request can be generated at the end of measurement or at an occurrence of overflow.

Note:

The input method of waveforms to be measured (TIN signal) varies depending on the I/O mode that has been set by the I/O selection register (BTSEL01/BTSEL23).

Figure 18-30. Operation Example



Reactivation

If the CTEN bit of the base timer x timer control register (BTxTMCR) is set to "1" during counting, the up counter reactivates and operates as follows.

If the counter is reactivated when a measurement start edge is waited:

The current status waiting for a measurement start edge is continued.

If the timer is reactivated during measurement:

The up counter value is cleared to "0000_H" and set to the measurement start edge waiting status.

Notes:

- If a detection of measurement end edge and a timer reactivation occur simultaneously, the following may result. In such case, set the interrupt control correctly by considering the operation of interrupt request flag.
  - ☐ Single measurement mode: The timer reactivates and waits for a measurement start edge.
    Also, the EDIR bit (the measurement end interrupt request flag) of the status control register(BTxSTC) is set to "1".
  - ☐ Continuous measurement mode: The timer reactivates and waits for a measurement start edge.
    Also, the EDIR bit (the measurement end interrupt request flag) of the status control register(BTxSTC) is set to "1"
    Also, the current measurement result is transferred to the data buffer register (BTxDTBF).
- If the 16/32-bit PWC timer is reactivated in the continuous measurement mode and if a measurement start edge is detected in the input signal (TIN) simultaneously, the timer immediately starts counting from the value "0001_H".

Calculating the Pulse Width

After the measurement, the measurement result can be read from the base timer x data buffer register (BTxDTBF) and the measured pulse width can be calculated using the following formula.

Pulse width = n x T

n: Data buffer register (BTxDTBF) value
T: Count clock cycle

18.5.7.3 32-bit Timer Mode Operation

This section explains the 32-bit timer mode operation.

This section explains the setting and operation for cascading 2 channels of a 16-bit PWC timer and using them as a 32-bit PWC timer.

Overview

Using the T32 bit of the timer control register (BTxTMCR), 2 channels of a 16-bit PWC timer can be cascaded and used as a 32-bit PWC timer.
In this mode, the even-number channel corresponds to the lower 16-bit operation, and the odd-number channel corresponds to the upper 16-bit operation. Therefore, the up counter must be read in the order of the lower 16 bits (even-number channel) -> the upper 16 bits (odd-number channel).

Setting Procedure (Example)

To select the 32-bit timer mode, set the T32 bit of the base timer x timer control register (BTxTMCR) of the even-number channel to "1". Also, set the T32 bit of the odd-number channel to "0". When setting 32-bit timer mode, set the registers using the procedure shown below.

The register setting differs between even-number and odd-number channels. In this example, channel 0 and channel 1 are connected by cascading.

1. Specify ch.0 to reset mode by setting the FMD2 to FMD0 bits of the base timer 0 timer control register (BT0TMCR). (FMD2 to FMD0 = 000)
2. Select 16/32-bit PWC timer for ch.0 and ch.1 by setting the FMD2 to FMD0 bits of the base timer x timer control register (BT0TMCR, BT1TMCR) of ch.0 and ch.1. (FMD2 to FMD0 = 100) At the same time, select the 32-bit timer mode by setting the T32 bit of the base timer 0 timer control register (BT0TMCR). (T32 = 1)

Use the same procedure to set both ch.2 and ch.3.

Note:

Rewrite the T32 bit while the operation of both of the even-number and odd-number channels is stopped. Whether the counting operation is stopped can be checked by setting the CTEN bit of the timer control register (BTxTMCR) to "0" (CTEN=0).

Operations

In 32-bit timer mode, the counting operation is basically the same as in 16-bit timer mode. However, the counting operation conforms to the settings of the even-number channels, ignoring the settings of the following registers for the odd-number channels.

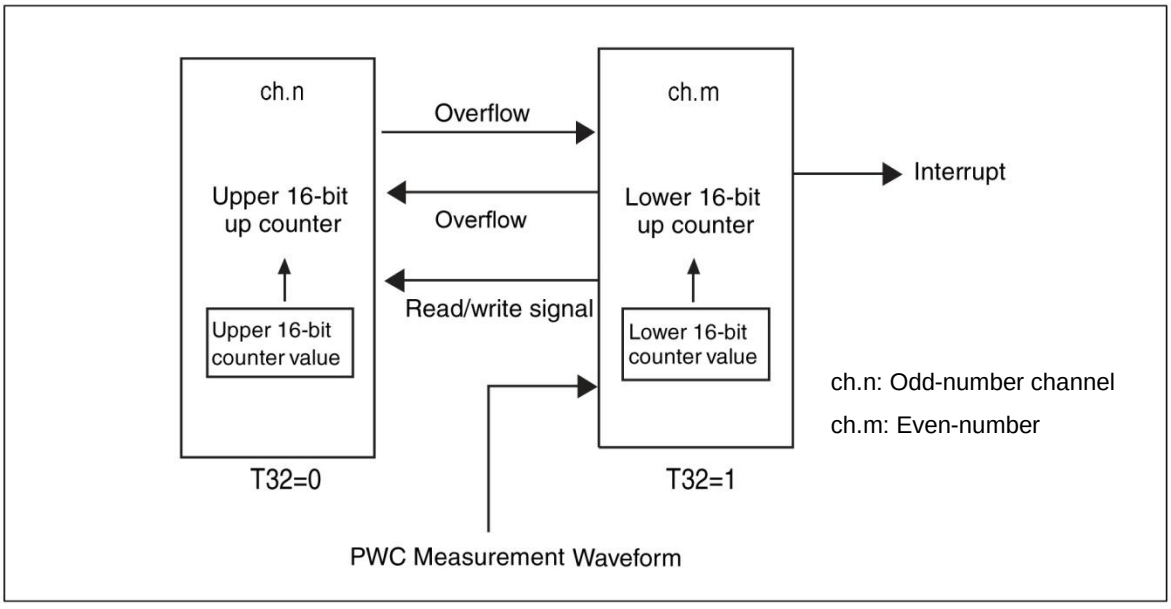
- Base timer x timer control register (BTxTMCR)
- Base timer x status control register (BTxSTC)

This section explains the counting in the 32-bit timer mode.

1. If the 16/32-bit PWC timer operation is enabled using the CTEN bit of the timer control register (BTxTMCR) (by setting CTEN = 1) of the even-number channel, the 32-bit PWC timer starts.
2. When a measurement start edge is detected in the input signal (TIN), the counting starts.
3. The up counter starts counting as a 32-bit counter with the even-number channel serving as the lower 16 bits and the odd-number channel as the upper 16 bits.
4. When a measurement end edge is detected in the input signal (TIN signal), the lower 16-bit data of the up counter value is stored in the data buffer register (BTxDTBF) of the even-number channel, and the upper 16-bit data of the up counter value is stored in the data buffer register (DTxDTBF) of the odd-number channel.

The channel configuration in 32-bit timer mode is shown below.

Figure 18-31. Configuration in 32-bit Timer Mode



Notes:

- The down counter value can be checked by reading the data buffer register (BTxDTBF). In the 32-bit timer mode, it must be read in the order of the lower 16-bit (even-number channel) -> upper 16-bit (odd-number channel).
- In 32-bit timer mode, the operation of the 32-bit PWC timer conforms to the settings of the even-number channel. Therefore, an interrupt request of the even-number channel is effective.

18.5.7.4 Interrupt

This section explains interrupts of the base timer.

An interrupt request is generated in one of the following events:

- An overflow occurs. (Overflow interrupt request)
- The measurement ends. (Measurement end interrupt request)

Table 18-9. Interrupt Occurrence Conditions

Interrupt Request	Interrupt Request Flag	Permission of Interrupt Request	Interrupt Request Clear
Overflow interrupt request	BTxSTC: OVIR=1	BTxSTC: OVIE=1	Set the OVIR bit of BTxSTC to "0".
Measurement end interrupt request	BTxSTC: EDIR=1	BTxSTC: EDIE=1	Read BTxDTBF

Notes:

- Once the generation of an interrupt request is enabled while the interrupt request flag is "1", an interrupt request will be issued when the interrupt is enabled.
- To enable the generation of an interrupt request, perform one of the following operations:
  - ☐ Clear the current interrupt request before enabling the generation of an interrupt request.
  - ☐ Clear the current interrupt request when enabling the interrupt.
- Either clear the current interrupt request after disabling the generation of an interrupt request or clear the current interrupt request within the interrupt processing routine.
- For interrupt vector numbers used for issuing an interrupt request, see "List of Interrupt Vector" in "Appendix."
- Set the interrupt level corresponding to the interrupt vector number in one of the interrupt control registers (ICR00 to ICR47). For information on interrupt level setting, see "Chapter: Interrupt Control (Interrupt Controller)".

18.5.7.5 Precautions for Using this Device

This section explains precautions for using this device.

Note the following when using the 16/32-bit PWC timer:

Notes on Program Setting

- Change the following bits of the base timer x timer control register (BTxTMCR) after stopping the up counter by resetting the CTEN bit to "0" (CTEN=0).
  - ☐ CKS2 to CKS0 bits
  - ☐ EGS2 to EGS0 bits
  - ☐ T32 bit
  - ☐ FMD2 to FMD0 bits
  - ☐ MDSE bit
- All registers are initialized when the FMD2 to FMD0 bits of the timer control register (BTxTMCR) are set to "000" to select reset mode.
- Before the base timer function or T32 bit can be changed, the base timer must be reset once. Except when rewriting the status of FMD2 to FMD0 bits or T32 bit of the timer control register (BTxTMCR) after a reset, be sure to set the FMD2 to FMD0 bits to "000" to select the reset mode. Then, rewrite the status of these bits.
- The timer may operate due to the status of previously measured signals if the followings are set simultaneously during system reset or during reset mode.
  - ☐ The base timer function is set for the 16/32-bit PWC timer by setting the FMD2 to FMD0 bits of the base timer x timer control register (BTxTMCR) to "100" (FMD2 to FMD0=100).
  - ☐ Enable 16/32-bit PWC timer operation by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "1" (CTEN=1).

Notes on Operations

- If the count timing of the up counter and the load timing occur at the same time, the load operation is given precedence.
- If the 16/32-bit PWC timer operation is enabled by setting the CTEN bit of the base timer x timer control register (BTxTMCR) to "1" (CTEN=1), the up counter value is cleared. Also, the up counter value is made invalid if it has been set before the operation is enabled.
- If the 16/32-bit PWC timer is reactivated in the continuous measurement mode and if a measurement start edge is detected in the input signal (TIN) simultaneously, the timer immediately starts counting from the value "0001_H".
- If 2 channels of PWC timers are used as a single 32-bit PWC timer, the 16-bit PWC timer setting of the even-number channel is made valid. The timer setting of odd-number channel is ignored.
- The input operation of measurement waveforms varies depending on the base timer I/O selection function.

Notes on Interrupts

- If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".
- If a detection of measurement end edge and a reactivation of 16/32-bit PWC timer occur simultaneously, the following may result. In such case, set the interrupt control correctly by considering the operation of the interrupt request flag.
  - ☐ Pulse width single measurement mode: The timer reactivates and waits for a measurement start edge. Also, the measurement end interrupt request flag (EDIR) is set to "1".
  - ☐ Pulse width continuous measurement mode: The timer reactivates and waits for a measurement start edge. The measurement end interrupt request flag (EDIR) is set to "1", and the currently measured result is transferred to the data buffer register (BTxDTBF).

19. Reload Timer

This chapter explains the reload timer.

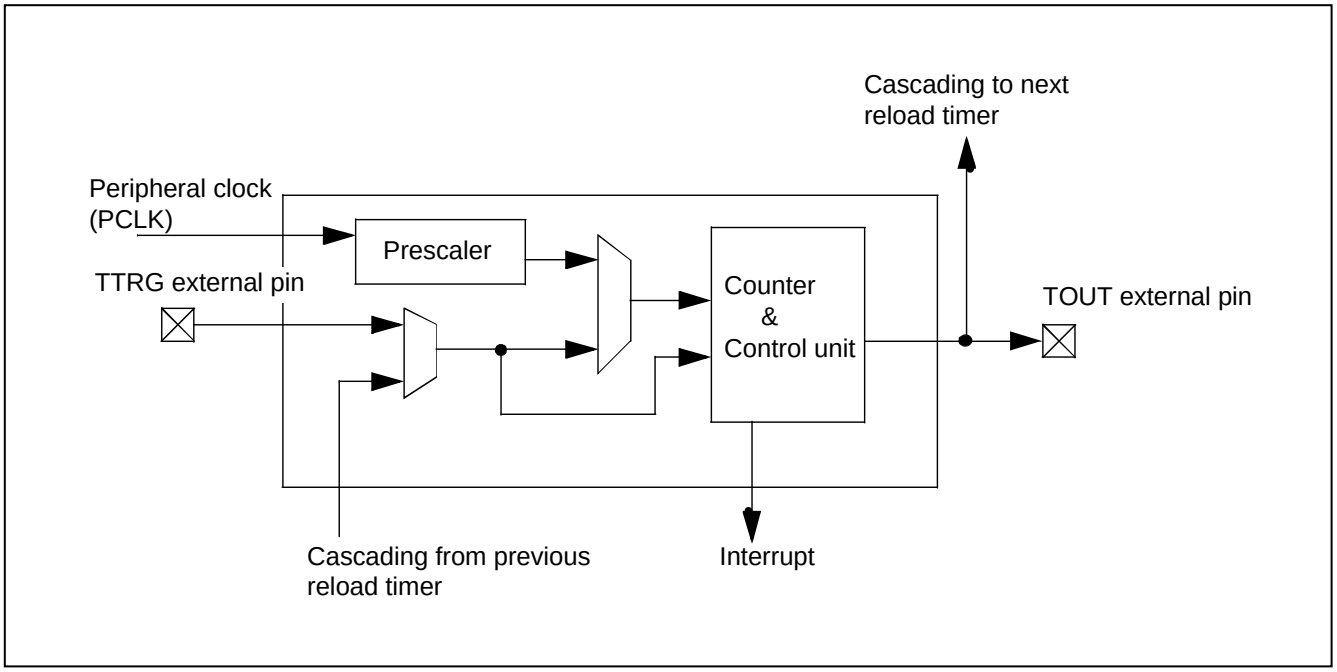
19.1 Overview
19.2 Features
19.3 Configuration
19.4 Registers
19.5 Operation
19.6 Application Note

19.1 Overview

This section explains the overview of the reload timer.

This module is a 16-bit reload down count timer with the interval timer mode, which counts the internal clock, and the event counter mode, which counts external events.

Figure 19-1. Block Diagram of Reload Timer (1 Channel, Overview)



19.2 Features

This section explains features of the reload timer.

A 5-channel reload timer is installed in this series.

Each channel is configured as follows.

- 16-bit down counter x 1
- 16-bit reload register x 1
- 16-bit reload/compare/capture register x 1
- Buffers described above x 1
- 6-bit prescaler for internal count clock creation x 1
- External trigger/event input (TIN) x 1
- External toggle output (TOUT) x 1
- Control register x 1
- Count comparator x 1

This timer, equipped with the interval timer mode/event counter mode described below, can be used for the following purposes and functions by setting the registers:

Interval timer mode

- Single one-shot operation >> Single-shot Timer
- Dual one-shot operation
- Single reload operation >> Reload Timer
- Dual reload operation >> PPG (Programmable Pulse Generator)
- Compare mode >> Output compare, PWM (Pulse Width Modulator)
- Capture mode (external trigger input/software trigger use) >> PWC (Pulse Width Counter)
- Underflow interrupt/capture interrupt
- 6 types of internal clocks (peripheral clock (PCLK) divided by 2/4/8/16/32/64)
- External trigger input (rising edge/falling edge/both edges)
- External gate input

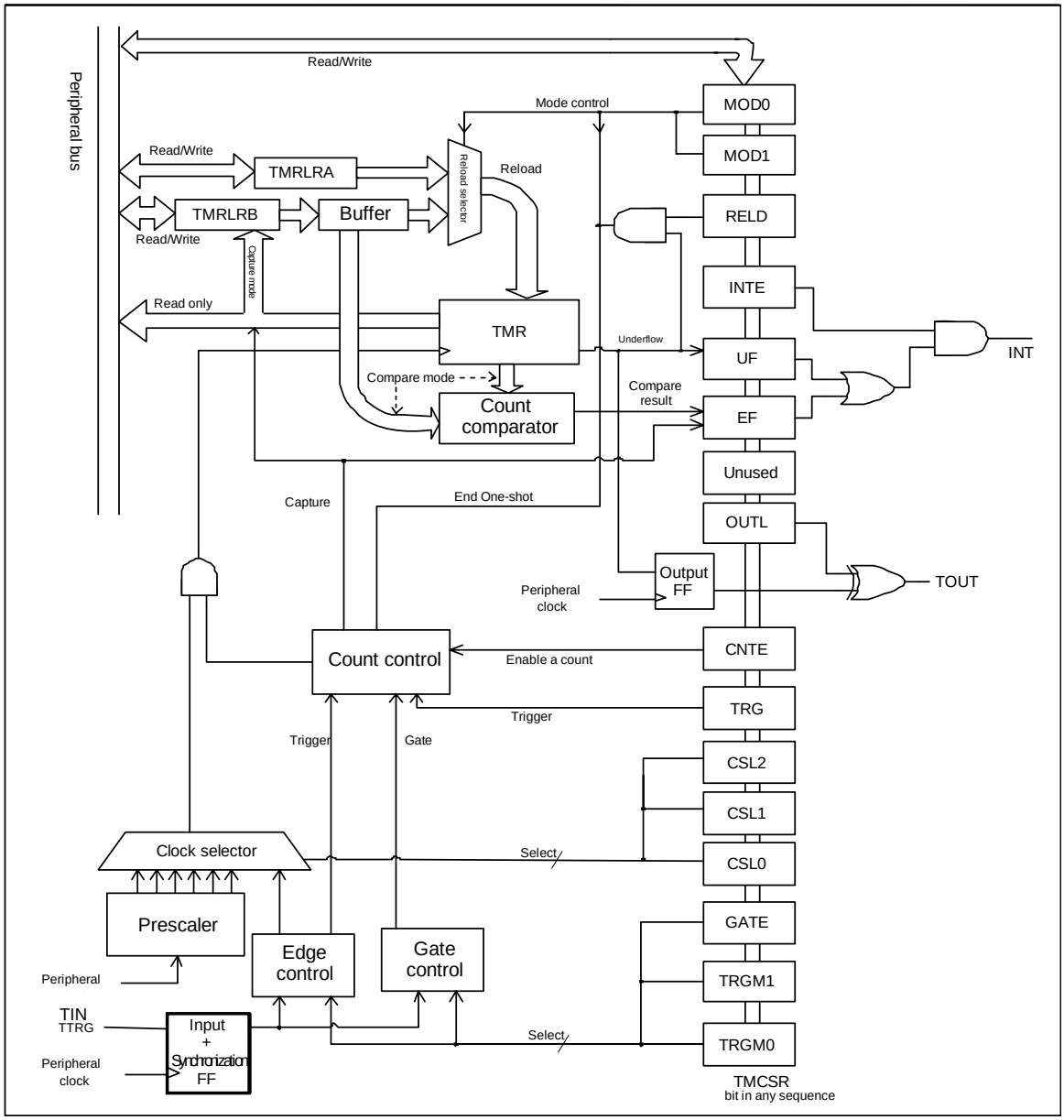
Event counter mode

- Single one-shot operation
- Dual one-shot operation
- Single reload operation
- Dual reload operation
- Compare mode
- Capture mode (only software trigger)
- Underflow interrupt/capture interrupt/compare interrupt
- External event input edge detection (rising edge detection/falling edge detection/both edge detection)
- Cascade mode
  - Use ch.0 output for ch.1 input. Use ch.1 output for ch.2 input. Use ch.2 output for ch.3 input.

19.3 Configuration

This section explains the configuration of the reload timer.

Figure 19-2. Block Diagram of Reload Timer (1 Channel, Details)



19.4 Registers

This section explains registers of the reload timer.

Table of Base Address (Base_addr), External Pins

Table 19-1. Table of Base Address (Base_addr), External Pins

Channel	Base_addr	External Pin	
		TOUT	TIN
0	0x0060	TOT0	TIN0
1	0x0100	TOT1	TIN1
2	0x0108	TOT2	TIN2
3	0x0110	TOT3	TIN3
4	0x01D8	TOT4	TIN4

Register Map

Table 19-2. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x01D8	TMRLRA4		TMR4		16-bit timer reload register A4 16-bit timer register 4
0x01DC	TMRLRB4		TMCSR4		16-bit timer reload register B4 Control status register 4
0x0060	TMRLRA0		TMR0		16-bit timer reload register A0 16-bit timer register 0
0x0064	TMRLRB0		TMCSR0		16-bit timer reload register B0 Control status register 0
0x0100	TMRLRA1		TMR1		16-bit timer reload register A1 16-bit timer register 1
0x0104	TMRLRB1		TMCSR1		16-bit timer reload register B1 Control status register 1
0x0108	TMRLRA2		TMR2		16-bit timer reload register A2 16-bit timer register 2
0x010C	TMRLRB2		TMCSR2		16-bit timer reload register B2 Control status register 2
0x0110	TMRLRA3		TMR3		16-bit timer reload register A3 16-bit timer register 3
0x0114	TMRLRB3		TMCSR3		16-bit timer reload register B3 Control status register 3

19.4.1 Control Status Register: TMCSR (TiMer Control and Status Register)

The bit configuration of the control status register is shown below.

This register controls the operating mode and interrupt.
It is not possible to rewrite any data other than bit7 and bit3 to bit0 when bit1:CNTE= "1".
It is possible to rewrite bit15 to bit8 and bit6 to bit4 and write counter operation enabling by writing CNTE= "1" simultaneously. It is also possible to rewrite bit15 to bit8, bit6 to bit4 and write operation disabling by writing CNTE= "0" simultaneously.

TMCSR: Address Base_addr + 06_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MOD[1:0]		TRGM[1:0]		CSL[2:0]			GATE
Initial value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EF	Reserved	OUTL	RELD	INTE	UF	CNTE	TRG
Initial value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R,W	R,W	R,W	R/W	R(RM1),W	R/W	R0,W

[bit15, bit14] MOD (MODe): Mode selection bits

MOD[1:0]	Operation Mode
00	Single mode (initial value)
01	Dual mode
10	Compare mode
11	Capture mode

[bit13, bit12] TRGM[1:0] (TRiGger input Mode select): TIN input mode selection bits

These bits control input pin functions. The functions of the interval timer mode differ from those of the event counter mode.

[Interval timer mode, trigger input (bit8:GATE bit= "0")]

Select an effective external edge which can be a reload trigger through TIN input in the following manner:

TRGM[1:0]	TIN Effective External Edge
00	No external trigger detection (initial value)
01	Rising edge
10	Falling edge
11	Both edges

[Interval timer mode, gate input (bit8:GATE bit= "1")]

Select the pin level which enables the counter during TIN input in the following manner:

TRGM[1:0]	TIN Effective Level
x0	Counted only during the input period for TIN pin "L" (initial value)
x1	Counted only during the input period for TIN pin "H"

[Effective edge setting at the event counter mode]

In the event counter mode, select an edge for external event detection in the following manner: Every time an external event is detected, the counter value is decreased. When an external event is selected, the setting of the bit8:GATE bit becomes invalid.

TRGM[1:0]	Count Target Edge
00	Reserved
01	Rising edge
10	Falling edge
11	Both edges

[bit11 to bit9] CSL[2:0] (Count source SeLect): Count source selection bits

These bits specify the count source. Select a count source from the internal clock (peripheral clock (PCLK)) and the external event (TIN input) specified following: When the event counter mode is set, set the count effective edge using bit13, bit12:TRGM[1:0].

CSL[2:0]	Count Source	Operation Mode
000	Division of the peripheral clock frequency by 2 (initial value)	Interval timer mode
001	Division of the peripheral clock frequency by 4	
010	Division of the peripheral clock frequency by 8	
011	Division of the peripheral clock frequency by 16	
100	Division of the peripheral clock frequency by 32	
101	Division of the peripheral clock frequency by 64	
110	Cascade mode (ch.0: TIN0, ch.1: TOUT0, ch.2: TOUT1, ch.3: TOUT2)	Event counter mode
111	External event (TIN input)	

[bit8] GATE (GATE input enable): Gate input enabling bit

This bit controls the functions of the input pin (TIN) of (bit11 to bit9:CSL[2:0]=000 to 101) at the interval timer mode specified following.

GATE	TIN Input Pin Function
0	Use as trigger input (initial value)
1	Use as gate input

This bit does not influence any operation at the event counter mode.

[bit7] EF (Extended Flag): Extended interrupt flag

This flag indicates that a compare match interrupt has occurred at the compare mode or a capture input interrupt has occurred at the capture mode.

Set factor	[Compare mode of the event counter mode] Count down occurs from compare match (TMR = TMRLRB) [Capture mode] Capture input (retrigger)
Clear factor	Writing "0" to this bit or reset.

Writing "1" to this bit will not be effective. In synchronization with the count clock, set operation or clear operation are performed in the compare mode. The values read with read-modify-write instructions will always be "1".

[bit6] Reserved

Reserved bit. Data writing is ineffective.

[bit5] OUTL (OUTput Level): Output polarity setting bit

This bit controls output polarity of the timer output pin (TOUT).

OUTL	TOUT Initial Value	TOUT Initial Output Level
0	Positive polarity (Initial value)	L level
1	Negative polarity	H level

[bit4] RELD (RELoaD enable): Reload operation enabling bit

This bit sets reload operation in case of underflow specified following:

RELD	Operation Mode	Description of Operation
0	One-shot mode	No sooner does a counter underflow occur, than the count operation stops. Reload is not performed until the next trigger is inputted. * (initial value)
1	Reload mode	Counter underflow occurs. At the same time, the contents of the reload register are loaded to the counter to continue count operation.

* However, the dual one-shot function reloads TMRLRB at the same time as TMRLRA underflow and continues counting. After that, count operation stops at the same time as TMRLRB underflow.

[bit3] INTE (INTerrupt Enable): Interrupt request enabling bit

This bit controls an interrupt request in case of underflow/compare match (event counter mode)/capture specified following:

INTE	Description of Operation
0	Interrupt disabled (no interrupt is generated even if the UF/EF bit is set.) (initial value)
1	Interrupt enabled (an interrupt request is generated if the UF/EF bit is set.)

[bit2] UF (Under flow Flag): Underflow flag

This flag indicates that underflow has occurred when the counter value is decreased from 0x0000.

Set factor	Counter underflow occurrence
Clear factor	Writing "0" to this bit or reset.

[bit1] CNTE (timer CouNTer Enable): Timer count enabling bit

This bit controls the operation of the timer as follows:

CNTE	Description of Operation
0	Operation disabled (initial value)
1	Operation enabled (waiting for activation trigger)

[bit0] TRG (software TRiGger): Software trigger bit

This bit generates a timer software trigger. If a software trigger is generated, the contents of the reload register are loaded to the counter to initiate count operation.

TRG	Description of Operation
Write "0"	No influence on the operation
Write "1"	A software trigger is generated.

When "0" is written into this bit, no influence on the operation. The read value is always "0".

Trigger input through this register is effective only when bit1:CNTE = "1".

Writing "1" into the TRG bit always generates an effective trigger if the timer is activated (bit1:CNTE= "1") in any operation mode.

19.4.2 16-bit Timer Register: TMR (16bit TiMer Register)

The bit configuration of the 16-bit timer register is shown below.

This register can read the timer count value.
Always perform 16-bit access to this register.

TMR: Address Base_addr + 02_H (Access: Half-word)

	bit15	bit14	-	-	-	bit2	bit1	bit0
	TMR[15:0]							
Initial value	X	X	-	-	-	X	X	X
Attribute	R,WX	R,WX	-	-	-	R,WX	R,WX	R,WX

[bit15 to bit0] TMR (TiMeR): 16-bit timer

This register can read the counter value of the 16-bit timer. The initial value is undefined.

19.4.3 16-bit Timer Reload Register A, 16-bit Timer Reload Register B: TMRLRA, TMRLRB (16bit TiMer ReLoad Register A/B)

The bit configuration of the control status register is shown below.

TMRLRA sets the count initial value.

TMRLRB functions differently depending on the operation mode.

Always perform 16-bit access to this register.

TMRLRA: Address Base_addr + 00_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	TMRLRA[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

TMRLRB: Address Base_addr + 04_H (Access: Half-word)

	bit15	bit14	- - -	bit2	bit1	bit0
	TMRLRB[15:0]					
Initial value	X	X	- - -	X	X	X
Attribute	R,W	R,W	- - -	R,W	R,W	R,W

[bit15 to bit0] TMRLRA (TiMer ReLoad Register A): 16-bit reload setting register A

[bit15 to bit0] TMRLRB (TiMer ReLoad Register B): 16-bit reload setting register B

The TMRLRA register holds the count initial value. The TMRLRA can be used in all modes regardless of the bit15, bit14:MOD[1:0] setting in the TMCSR register.

The TMRLRB is to be used based on the bit15, bit14:MOD[1:0] setting in the TMCSR register specified following:

Mode	MOD[1:0]	TMRLRB Function
Single mode	00	Not used
Dual mode	01	H width (when OUTL="0") counter value
Compare mode	10	Compare register (when H width setting is OUTL="0")
Capture mode	11	Capture register (TMR value upon retrigger input)

When using as a counter value, underflow is generated if 1 count is set when writing 0x0000 and 65,536 is set when writing 0xFFFF.
H width and L width of the timer output waveform (TOUT) are determined by the MOD[1:0] (bit15, bit14 of the TMCSR register), RELD (bit4 of the TMCSR register), and OUTL (bit5 of the TMCSR register) bit setting as well as the TMRLRA/B register value.

H width and L width setting of the waveform (TOUT) to be output is shown in the table below.

MOD[1:0]	Mode	RELD	OUTL	TOUT Output	
				H Width	L Width
00	Single	0	0	TMRLRA+1	---
			1	---	TMRLRA+1
		1	0	TMRLRA+1	
			1		
01	Dual	0	0	TMRLRB+1	TMRLRA+1
			1	TMRLRA+1	TMRLRB+1
		1	0	TMRLRB+1	TMRLRA+1
			1	TMRLRA+1	TMRLRB+1
10	Compare	0	0	See the explanation below.*	
			1		
		1	0		
			1		
11	Capture	0	0	TMRLRA+1	---
			1	---	TMRLRA+1
		1	0	TMRLRA+1	
			1		

* H width and L width are as follows in the compare mode:

- When TMRLRB < TMRLRA

  (OUTL=0) "L" width of TMRLRA-TMRLRB + 1, "H" width of TMRLRB
  (OUTL=1) "H" width of TMRLRA-TMRLRB + 1, "L" width of TMRLRB

- When TMRLRB = 0

  (OUTL=0) "L" output fixed
  (OUTL=1) "H" output fixed

- When TMRLRB > TMRLRA

  (OUTL=0) "H" output fixed
  (OUTL=1) "L" output fixed

- When TMRLRB = TMRLRA

  (OUTL=0) "L" output of 1 cycle, "H" width of TMRLRB
  (OUTL=1) "H" output of 1 cycle, "L" width of TMRLRB

The following formula represents the TOUT output time (TOUT) when the register is used as the single mode and dual mode in the interval timer mode:

TOUT = (Setting value of this register + 1) x count source cycle
(Note) The above formula is effective only in interval timer mode.

19.5 Operation

This section explains the operation of the reload timer.

19.5.1 Setting

Setting of the reload timer is shown below.

The operation of this timer is set based on the "count source" (select in the TMCSR.CSL[2:0]) and counter operation ({TMCSR.MOD[1:0], TMCSR.RELD}).

19.5.1.1 Count Source

The count source of the reload timer is shown below.

Select decrement conditions of the down counter in the TMCSR:CSL[2:0].

Table 19-3. List of Count Source

CSL[2:0]	Count Source	Operation Mode
000	Division of the peripheral clock frequency by 2 (initial value)	Interval timer mode
001	Division of the peripheral clock frequency by 4	
010	Division of the peripheral clock frequency by 8	
011	Division of the peripheral clock frequency by 16	
100	Division of the peripheral clock frequency by 32	
101	Division of the peripheral clock frequency by 64	
110	Cascade mode (ch.0: TIN0, ch.1: TOUT0, ch.2: TOUT1, ch.3: TOUT2)	Event counter mode
111	External event (TIN input)	

19.5.1.2 Timer Underflow Cycle

The timer underflow cycle is shown below.

Underflow is defined as counter down-counting from 0x0000. Set the time (cycle) to underflow occurrence since timer count operation start in the reload register (TMRLRA/TMRLRB). After loading to the reload register, underflow takes place if the count value reaches "reload register setting value + 1" count. The timer underflow cycle, TUF, in the interval timer mode can be represented as follows:

TUF = Peripheral clock (PCLK) cycle x prescaler division value (2 to 64) x
(Reload register value (TMRLRA/B) + 1)

19.5.1.3 Trigger

The trigger of the reload timer is shown below.

The trigger consists of the following two types:

- Software trigger ... Generated when writing "1" to the TMCSR.TRG
- External pin trigger ... Input from the TIN pin

The TIN pin is used as a count source in the event counter mode. Hence, a software trigger is always used. In the interval timer mode, settings are made in the TMCSR register.

19.5.1.4 Gate

The gate of the reload timer is shown below.

When configuring gate input (TMCSR.GATE = "1") in the interval timer mode, it is possible to stop counter down counting using the TIN external pin.

Table 19-4. TIN Effective Level

TRGM[0]	TIN Effective Level
0	Counted only during the input period for TIN pin "L" (initial value)
1	Counted only during the input period for TIN pin "H"

19.5.1.5 Counter Operation Selection

The counter operation selection is shown below.

Select the operation in case of counter underflow using the mode selection bits (bit15, bit14:MOD[1:0] of the TMCSR register) and the reload operation enabling bit (bit4:RELD of the TMCSR register). For details of operation in each mode, see the section of each counter operation.

Table 19-5. List of Counter Operation

MOD[1:0]	RELD	Operation in Case of Underflow	Counter Operation Name
00	0	Stop the counter with 0xFFFF	Single one-shot
	1	Reload TMRLRA	Single reload
01	0	(1) Reload TMRLRB (2) Stop the counter with 0xFFFF (See "19.5.3.3 Dual One-shot Operation").	Dual one-shot
	1	Reload TMRLRA and TMRLRB in turns	Dual reload
10	0	Stop the counter with 0xFFFF	Compare one-shot
	1	Reload TMRLRA	Compare reload
11	0	Stop the counter with 0xFFFF	Capture one-shot
	1	Reload TMRLRA	Capture reload

19.5.1.6 TOUT Pin Level Setting

The TOUT pin level setting is shown below.

Set pin output polarity using bit5:OUTL bit in the TMCSR register.

The relationships between events and the TOUT pin in each function are as follows:

A/B of the UF (underflow) section below indicates whether down counting underflow has occurred with a value when loading TMRLRA data or TMRLRB data. CMP (compare-match) shows the timing of down counting from TMRLRB = TMR.

Figure 19-3. TOUT Output Change in Each Event (1/3)

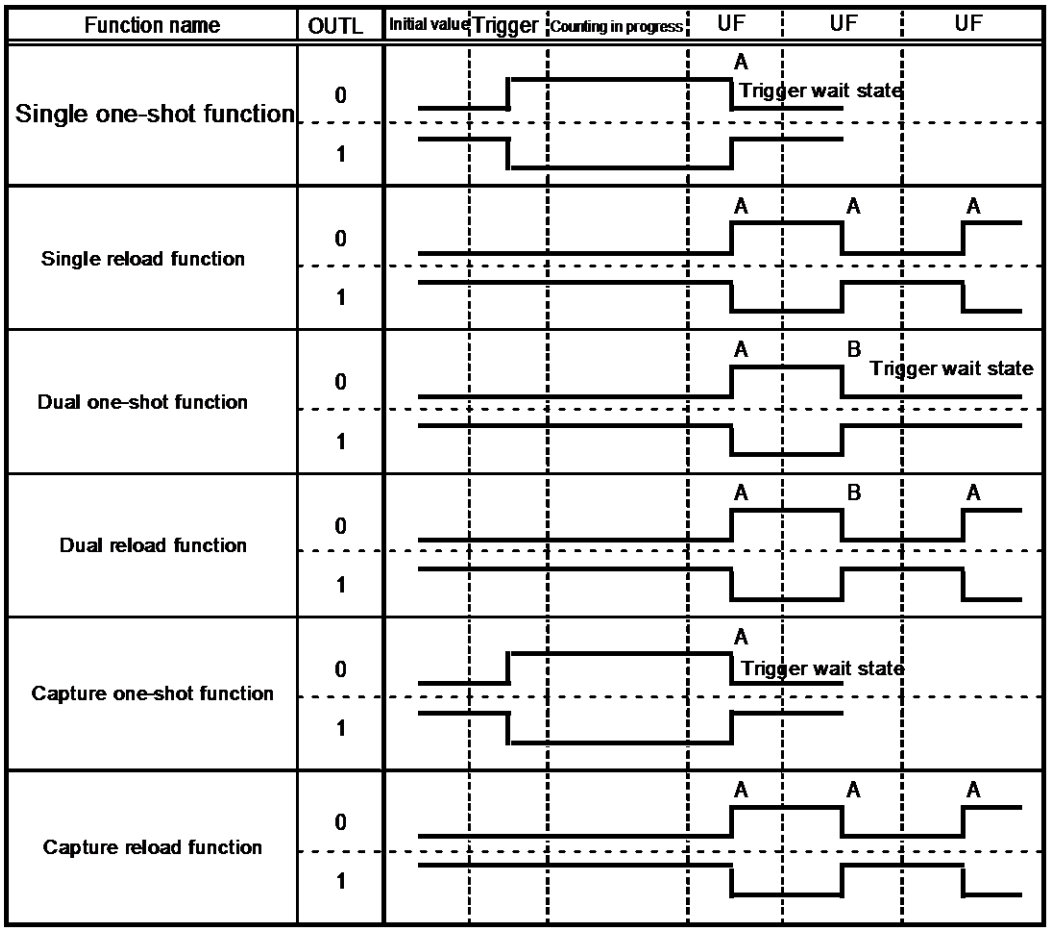


Figure 19-4. TOUT Output Change in Each Event (2/3)

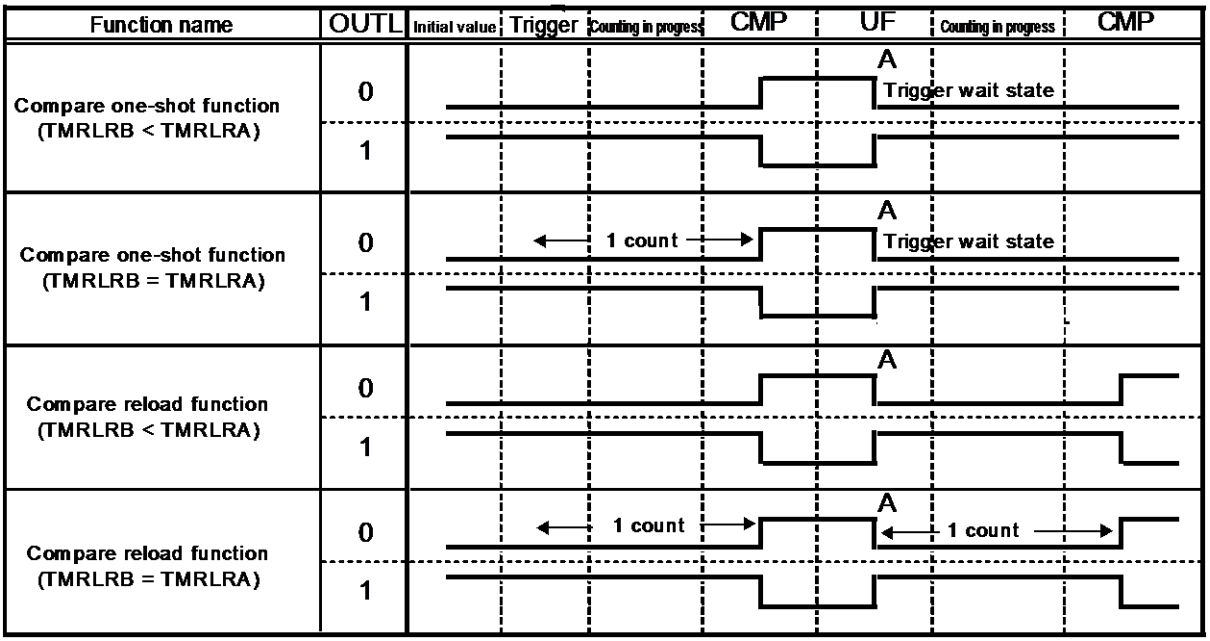
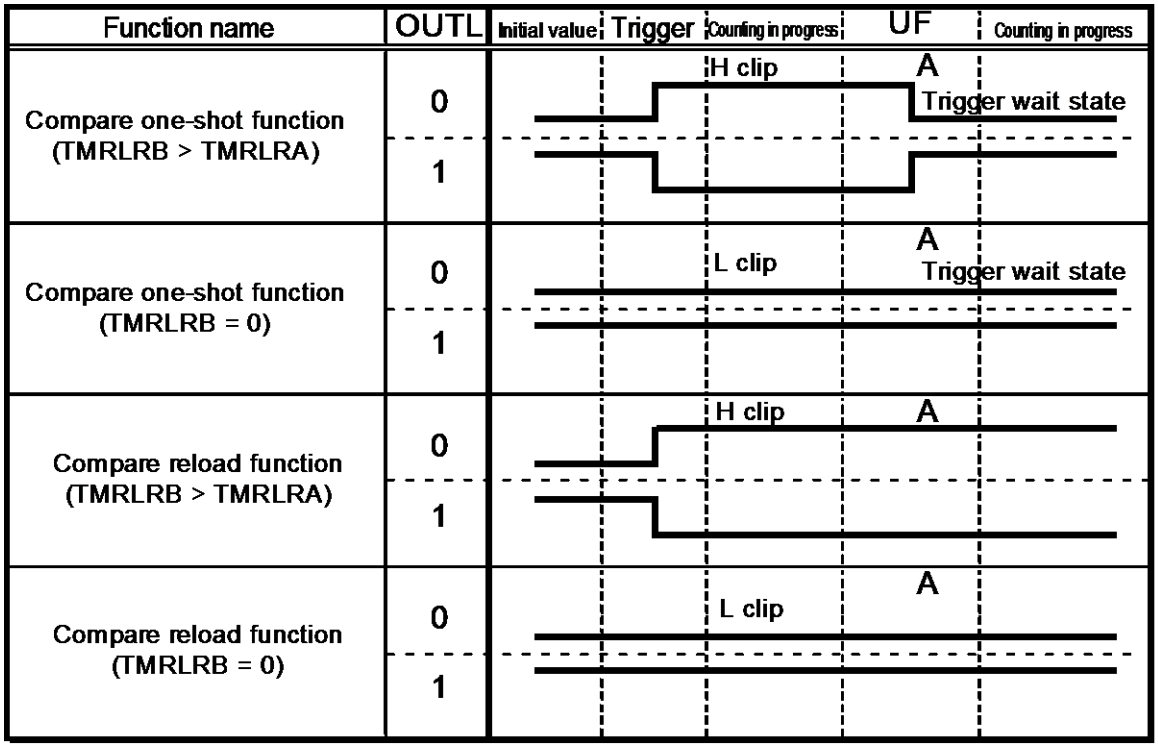


Figure 19-5. TOUT Output Change in Each Event (3/3)



19.5.2 Operation Procedure

Operation procedures are shown.

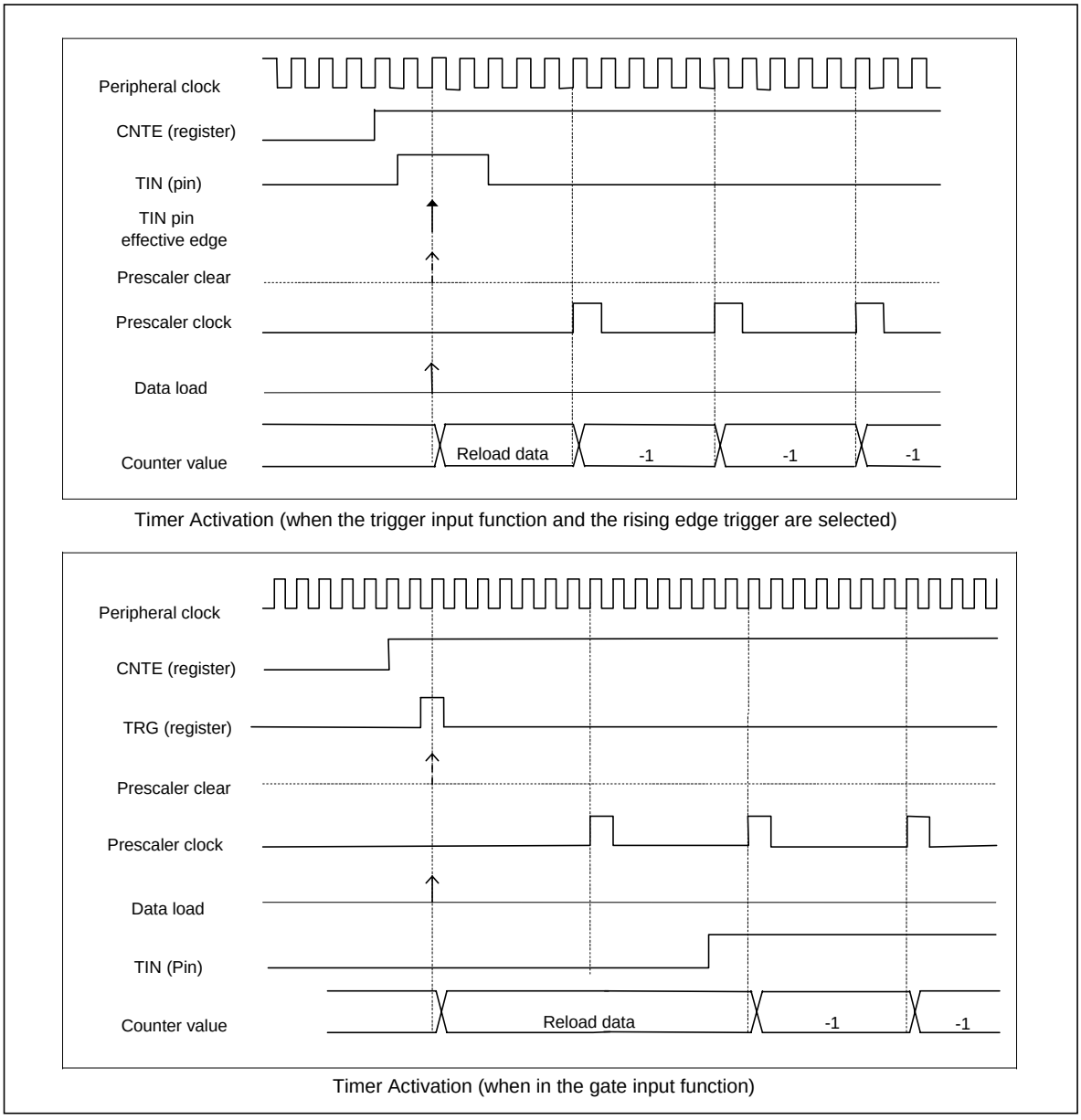
19.5.2.1 Activation

Activation is shown below.

Writing "1" into the bit1:CNTE bit of the TMCSR register changes the counter state to activation trigger waiting.

- TIN input during trigger input functioning
  If writing "1" to the bit0:TRG bit of the TMCSR register or inputting external trigger through TIN input takes place during activation trigger waiting, the prescaler will be cleared and the timer will load a value from the reload register to start down count operation. For TIN, input pulse of 2 x T (T indicates the peripheral clock (PCLK) cycle) or more.
- TIN input during gate input functioning
  If writing "1" to the bit0:TRG bit of the TMCSR register during activation trigger waiting, the prescaler will be cleared and the timer will load a value from the reload register and change the state to effective input polarity waiting. If there is any gate input with effective polarity from TIN input in the effective input polarity waiting, the timer initiates down count operation. For TIN, input pulse of 2 x T (T indicates the peripheral clock (PCLK) cycle) or more.

Figure 19-6. Timer Activation



19.5.2.2 Retrigger

The retrigger is explained.

The trigger which is generated during timer counting is called "retrigger." In this case, the following actions are taken:

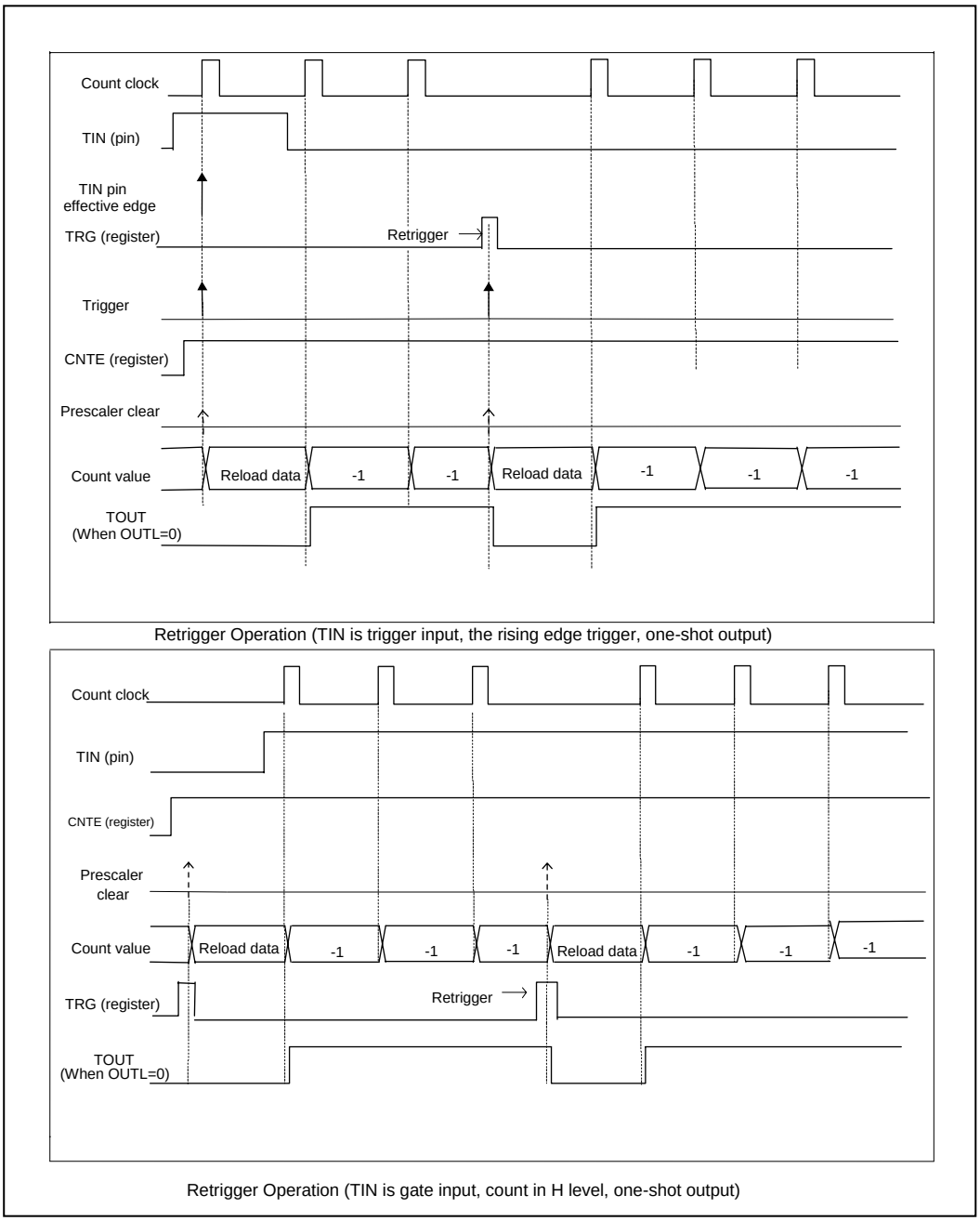
1. Initialize TOUT
2. Load the reload register value to the counter
3. Clear the 6-bit prescaler
4. Continue counting

Only in the capture mode, retrigger generation transfers a value being counted to the TMRLRB to set the EF bit of the TMCSR register.

Note:

TOUT is not initialized in the one shot mode at retrigger.

Figure 19-7. Retrigger Operation



Retrigger Operation (TIN is trigger input, the rising edge trigger, one-shot output)

Retrigger Operation (TIN is gate input, count in H level, one-shot output)

19.5.2.3 Underflow/Reload

Underflow/reload is shown below.

Underflow is defined as the timer down-counting from 0x0000. When underflow occurs, the bit2:UF bit of the TMCSR register is set. Underflow takes place in the timer if the count value reaches "reload register setting value + 1" count.

19.5.2.4 Generation of Interrupt Requests

Generation of interrupt requests is shown below.

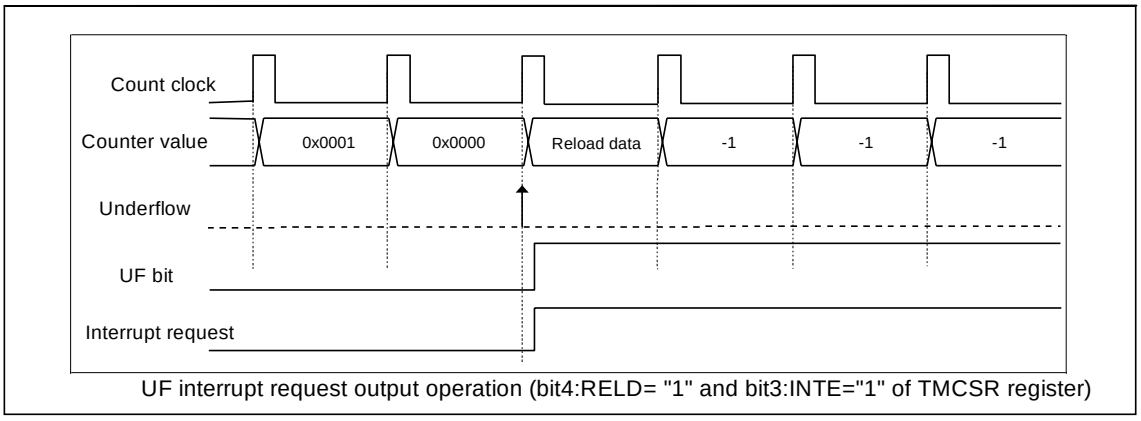
When bit3:INTE bit of the TMCSR register is "1", if bit2:UF bit/bit7:EF bit are set, an interrupt request is generated. In interval timer mode, the UF bit and the EF bit will be set under the following conditions.

- UF bit is set: A counter underflow occurred
- EF bit is set: A capture input occurred in capture mode

When a set of bit2:UF bit of the TMCSR register and a clear of the UF bit by writing "0" occurred concurrently, writing "0" to the UF bit will be invalid and the UF bit will be set. When a set of bit7:EF bit and a clear of the EF bit by writing "0" occurred concurrently, writing "0" to the EF bit will be invalid and the EF bit will be set.

The following is the example of generation of interrupt requests.

Figure 19-8. Example of UF Interrupt Request Output Operation



UF interrupt request output operation (bit4:RELD= "1" and bit3:INTE="1" of TMCSR register)

19.5.2.5 Concurrent Operation of Register Write and Timer Operation

The concurrent operation of register write and timer operation is shown below.

The following table shows the operation when a register write by a user and the timer operation occurred simultaneously.

Table 19-6. Concurrent Operation

Writing to Register	Operation of Timer	Operation to Execute
A clear of the UF bit by writing "0"	Setting of the UF bit	Setting of the UF bit (Writing "0" is ignored)
A clear of the EF bit by writing "0"	Setting of the EF bit	Setting of the EF bit (Writing "0" is ignored)
Writing to the reload register	Loading of timer by retrigger	Reloading old data (The written value will be loaded next time)

19.5.3 Operations of Each Counter

Operations of each counter are shown.

19.5.3.1 Single One-shot Operation

The single one-shot operation is shown below.

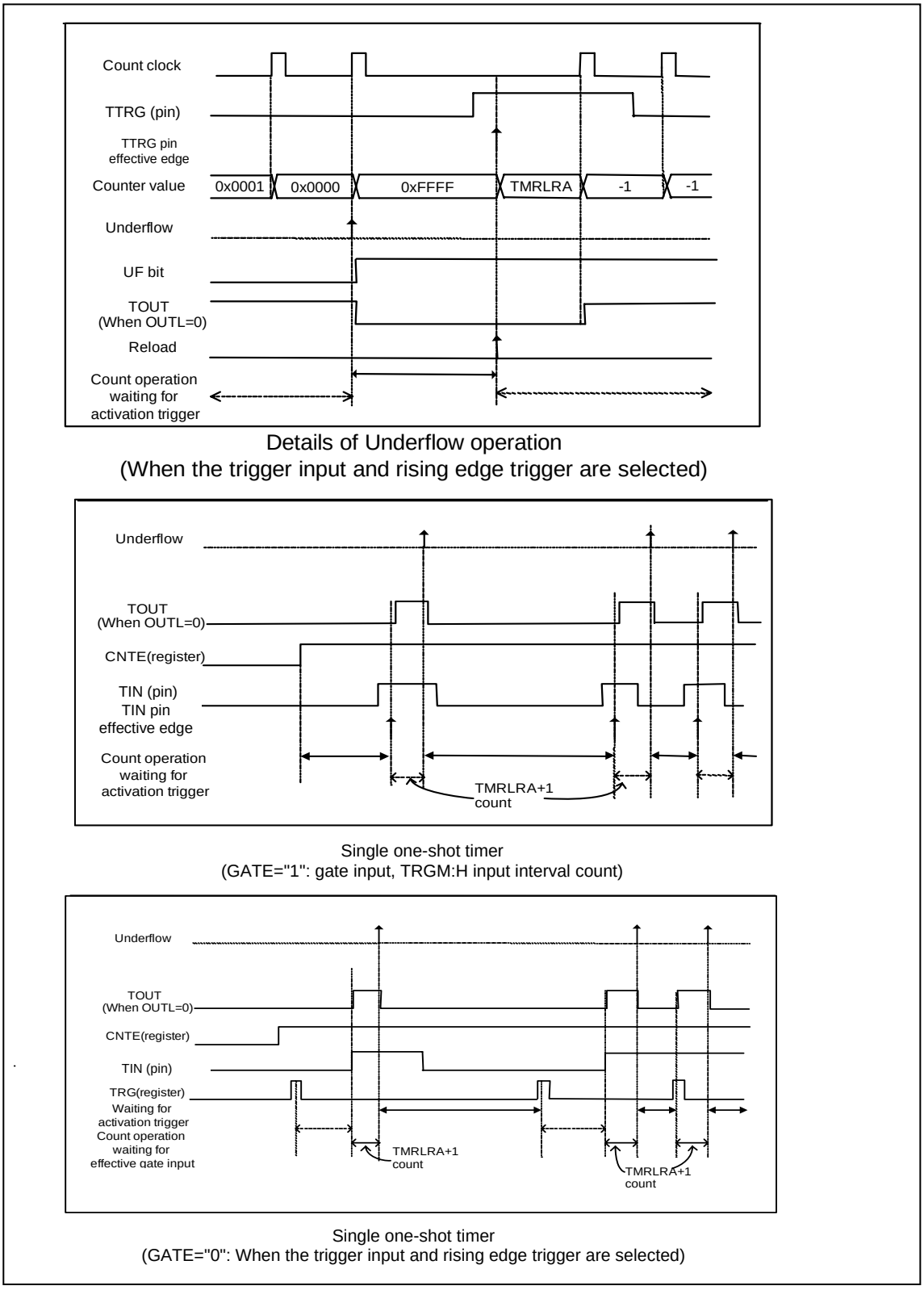
When bit15,14:MOD[1:0] = "00" and bit4:RELD = "0" of the TMCSR register, the timer performs single one-shot operation triggered by an underflow occurrence, that stops at 0xFFFF.

In the single one-shot configuration, if an underflow occurs, the following operation will be performed:

- Sets the UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Stops the count with 0xFFFF.
- Initializes TOUT output.
- Timer is waiting for a trigger.

For the single one-shot timer, TMRLRA turns to the initial value of the counter when a reload took place. TMRLRB is not used.

Figure 19-9. Single One-shot Operation



19.5.3.2 Single Reload Operation

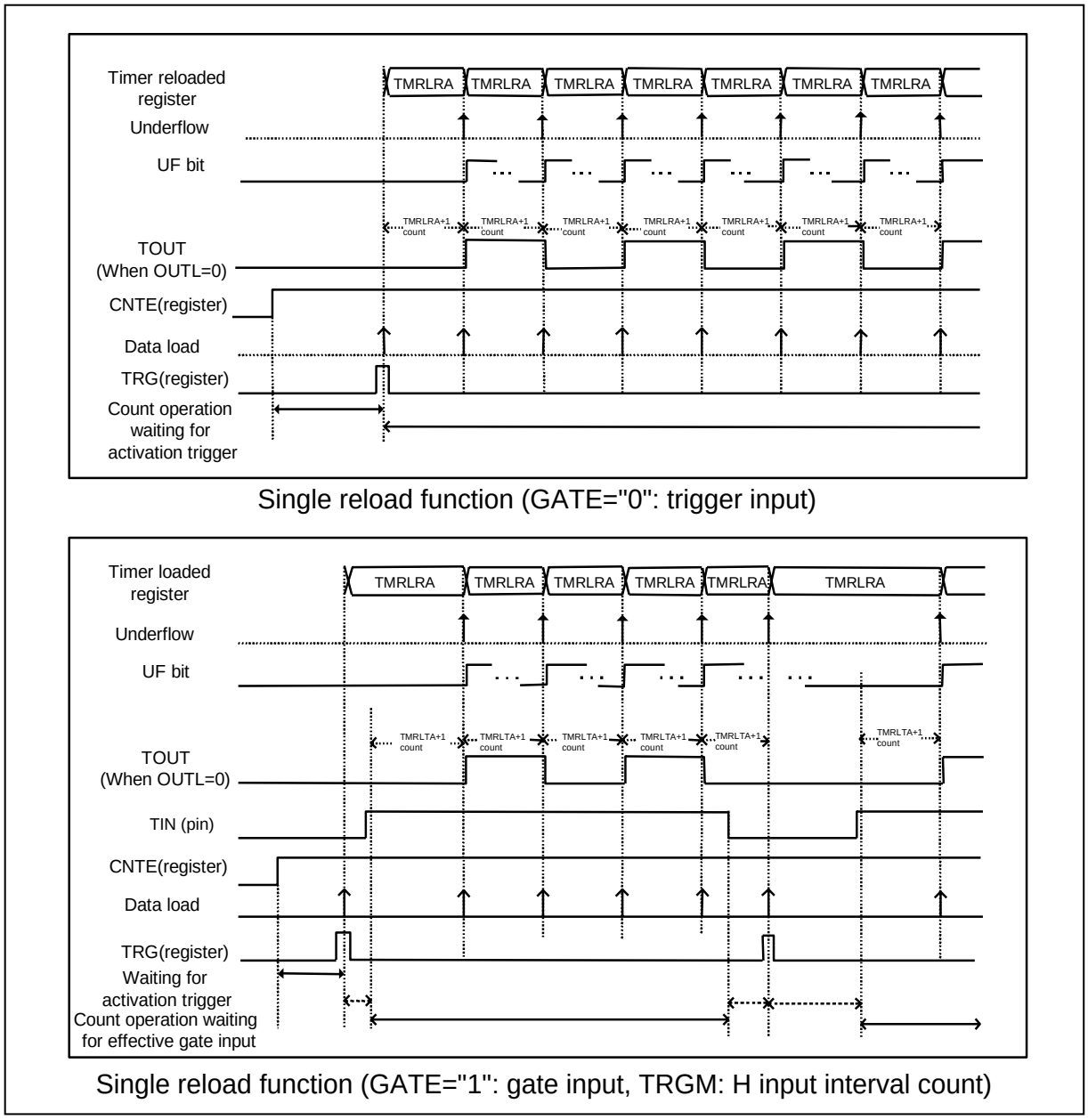
The single reload operation is shown below.

When bit15,14:MOD[1:0] = "00" and bit4:RELD = "1" of the TMCSR register, the single reload operation will be performed.

In single reload operation, a value will be loaded from TMRLRA to the timer by trigger input, a down count (decrementing the count) will start. When an underflow occurs, the value is reloaded from TMRLRA again and the down count operation continues. The value of TMRLRA represents the time the timer will reload. The TMRLRB register is not used. In single reload configuration, if an underflow occurs, the following operation will be performed:

- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Loads TMRLRA register onto the counter.
- Inverts TOUT output.
- Continues decrementing count.

Figure 19-10. Single Reload Operation



19.5.3.3 Dual One-shot Operation

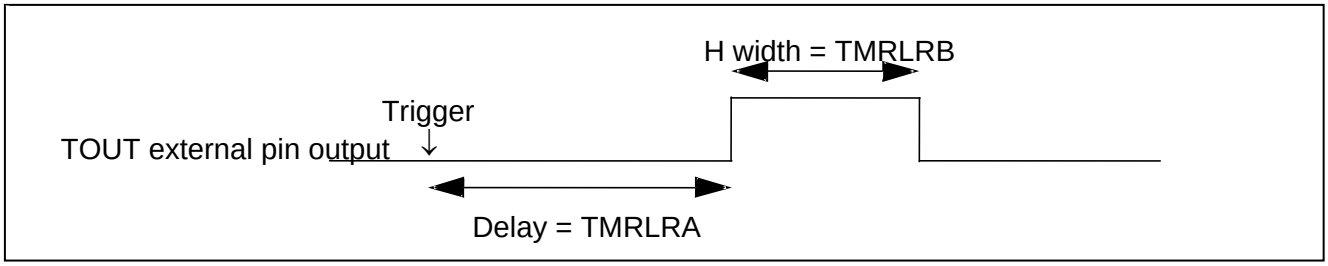
The dual one-shot operation is shown below.

When bit15,14:MOD[1:0] = "01" and bit4:RELD = "0" of the TMCSR register, the timer performs the dual one-shot operation. This can be used as a one-shot PPG.

In dual one-shot operation, values are loaded into the counter one by one in the order of TMRLRA then TMRLRB, and the down count is executed for each of the loaded values. The counter stops when the second underflow occurs.

When bit5:OUTL of the TMCSR register = "0", the value of TMRLRA represents the time interval between a timer activation (TOUT output is in L level) to a toggling of the TOUT output to "H", and the value of TMRLRB represents the time interval of H width of the TOUT output.

Figure 19-11. TOUT Pulse Width



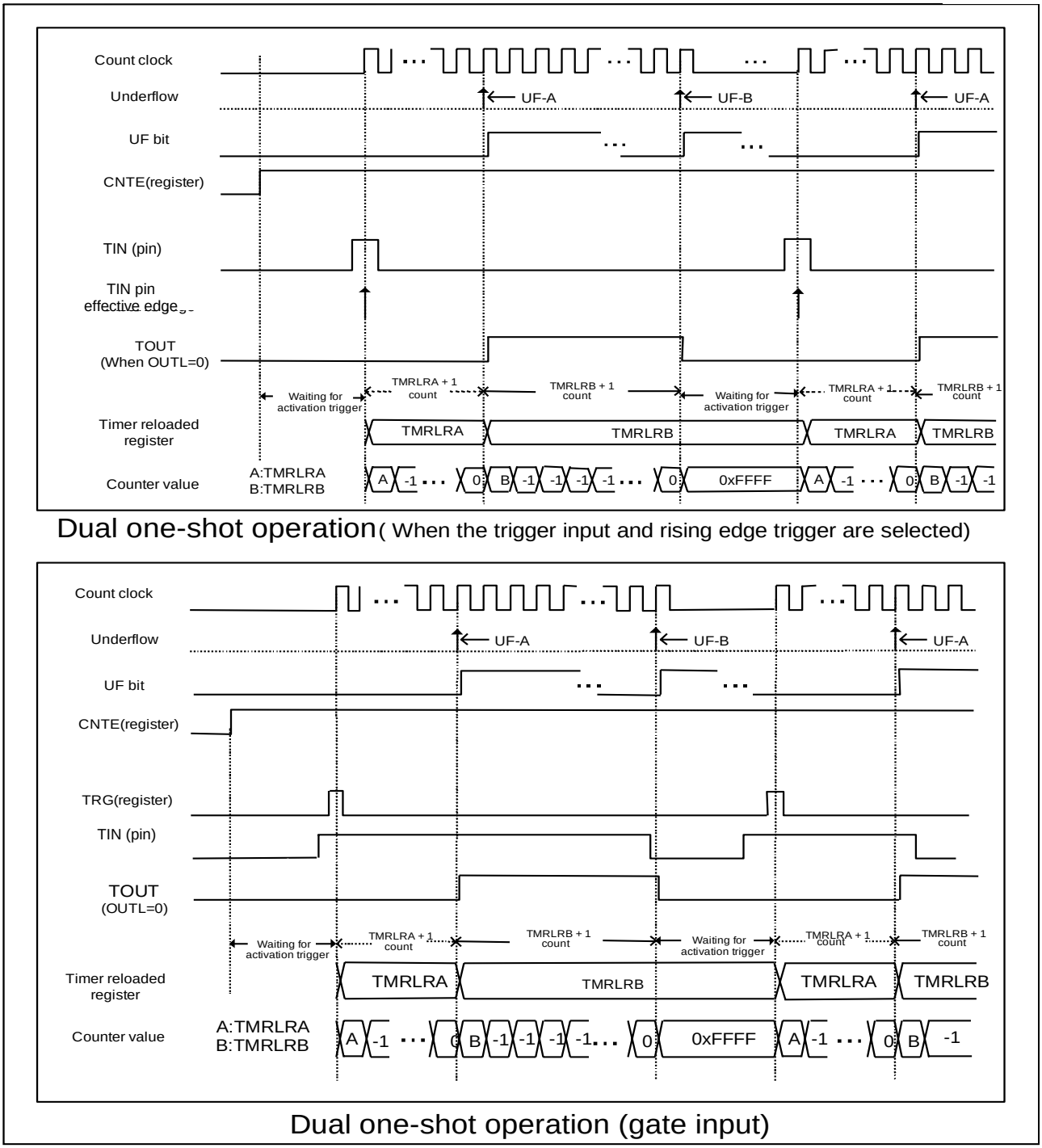
When the first underflow occurs (UF-A), the following operation will take place:

- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Loads TMRLRB to the counter.
- Inverts TOUT output.
- Starts a down count from TMRLRB.

When the second underflow (UF-B) occurs, the following operation will take place:

- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Stops the count with 0xFFFF.
- Initializes TOUT output.
- Timer is waiting for an activation trigger.

Figure 19-12. Dual One-shot Operation



19.5.3.4 Dual Reload Operation

The dual reload operation is shown below.

When bit15,14:MOD[1:0] = "01" and bit4:RELD = "1" of the TMCSR register, the timer performs the dual reload operation.

In dual reload operation, the values of TMRLRA and TMRLRB are loaded alternatively and decrement the counters for each load, that is, loads TMRLRA onto the counter and decrements the counter, and if an underflow occurs, loads TMRLRB onto the counter and decrement the counter, and if an another underflow occurs, loads TMRLRA onto the counter and decrements the counter, and so on.

When bit5:OUTL of the TMCSR register = "0", the value of TMRLRA represents the time interval between a timer activation (TOUT output is in L level) to a toggling of the TOUT output to "H", and the value of TMRLRB represents the time interval of H width of the TOUT output.

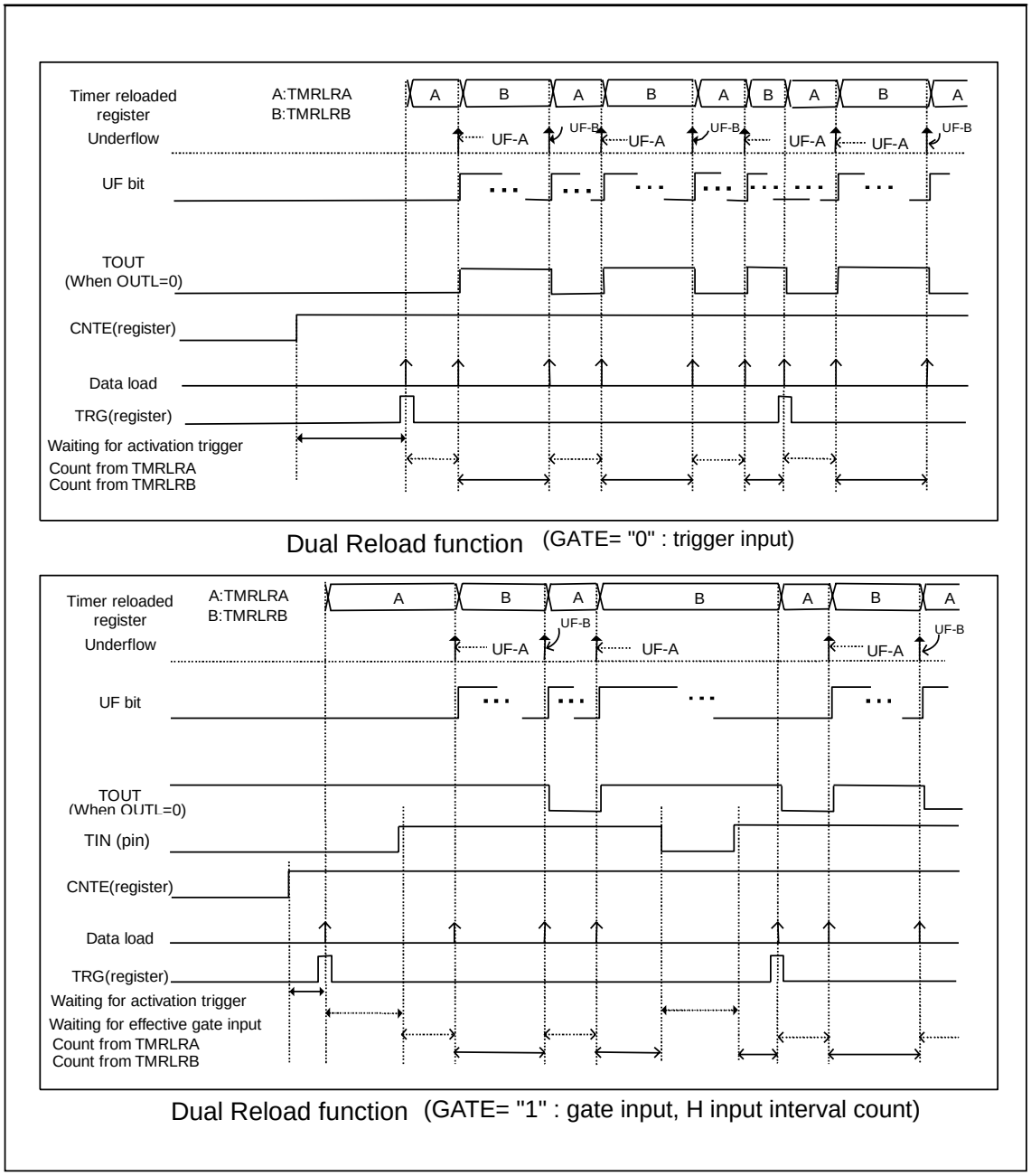
If an underflow (UF-A) occurs at the down count after loading a value from the TMRLRA, the following operation will be performed:

- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Loads TMRLRB to the counter.
- Inverts TOUT output.
- Starts a down count from TMRLRB.

If an underflow (UF-B) occurs at the down count after loading a value from the TMRLRB, the following operation will be performed:

- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Loads TMRLRA to the counter.
- Inverts TOUT output.
- Starts a down count from TMRLRA.

Figure 19-13. Dual Reload Operation



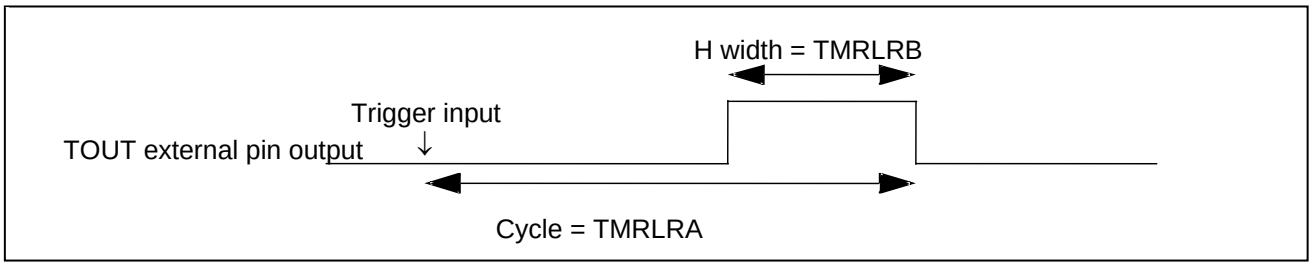
19.5.3.5 Compare One-shot Operation

The compare one-shot operation is shown below.

When bit15,14:MOD[1:0] = "10" and bit4:ELD = "0" of TMCSR register, the compare one-shot operation in which the counter value (TMR) and the value of TMRLRB register are compared for each down count will be performed. After accepting a trigger, the value of the TMRLRA register is loaded and the down count starts. When decrementing the count from the value of compare match (TMR = TMRLRB), the TOUT output will be inverted. When an underflow occurs, count operations stopped, TOUT output is initialized, and the timer go into the activation trigger wait state.

The value of TMRLRA indicates the time interval between the activation of a timer and the end of it and the value of TMRLRB indicates the counter value when an output of the H width of TOUT output starts. When OUTL="0" and TMR < TMRLRB, the TOUT output will become the "H level."

Figure 19-14. TOUT Interval, Pulse Width



From the start of a down count to TMR = TMRLRB (while TMR is greater than or equal to TMRLRB), the following operation will be performed:

- TOUT output continues to hold the initial value.
- The timer continues to count.

If a down count from TMR = TMRLRB occurs, the following operation will be performed:

- Inverts TOUT output.
- The timer continues to count.

(For the compare operation in interval timer mode, bit7:EF bit of TMCSR register will not be set.)

If an underflow occurs, the following operation will be performed:

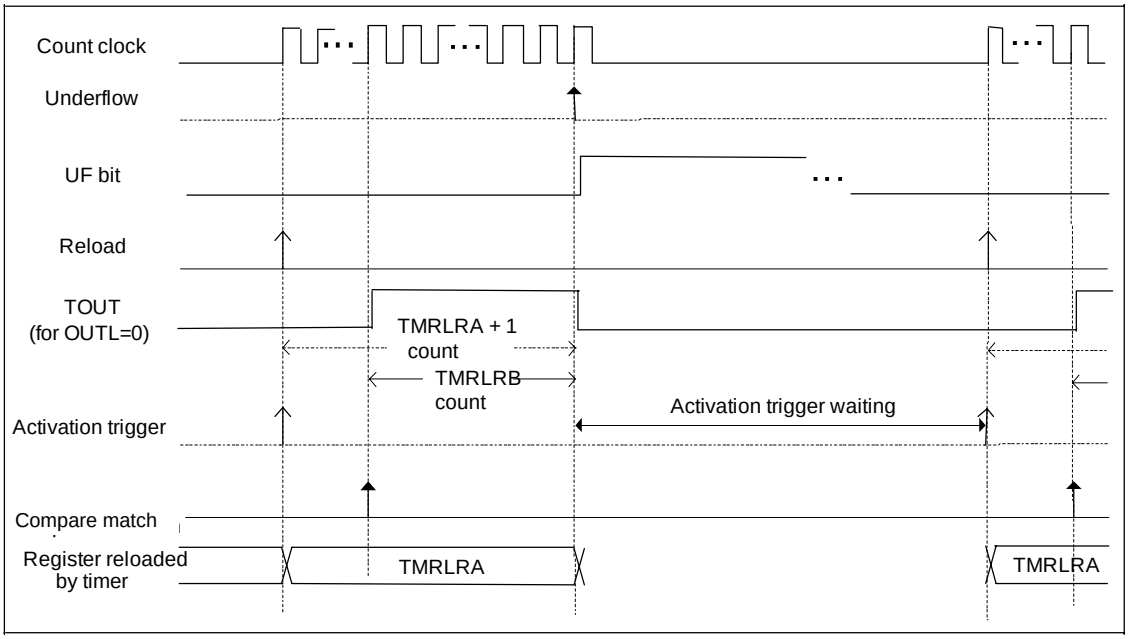
- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Initializes TOUT output.
- The timer stops with 0xFFFF.
- Timer is waiting for an activation trigger.

The operation of the compare function changes depending on the setting relation between TMRLRA and TMRLRB.

Figure 19-15. Compare One-shot Operation (1/2)

- Sets TMRLRB < TMRLRA

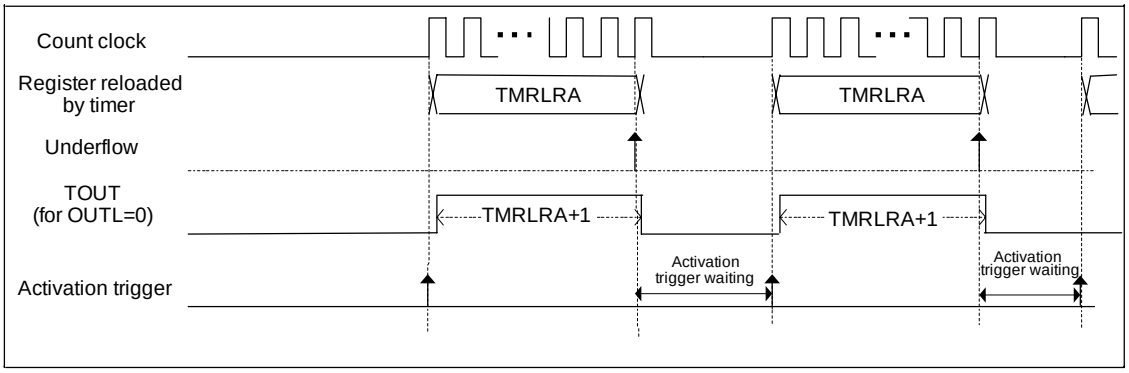
  When the register relation is as described above, the TOUT output is the "L" level until TMR and TMRLRB match after loading to the timer. When down counting from the compare match (TMR=TMRLRB), the TOUT output is "H" level until the TOUT output is inverted and an underflow occurs. When an underflow occurs the TOUT output will be initialized. Then, the timer will stop counting operation and turn into the activation trigger waiting state (for OUTL="0").



Compare one-shot function (TMRLRB < TMRLRA)

- Sets TMRLRB > TMRLRA

  When the register relation is as described above, the TOUT output is the "H" level between an activation trigger generation and an underflow occurrence because TMR is already smaller than TMRLRB after loading to the timer. When an underflow occurs, the timer will turn into the activation trigger waiting state and the TOUT output will be the "L" level (for OUTL="0").

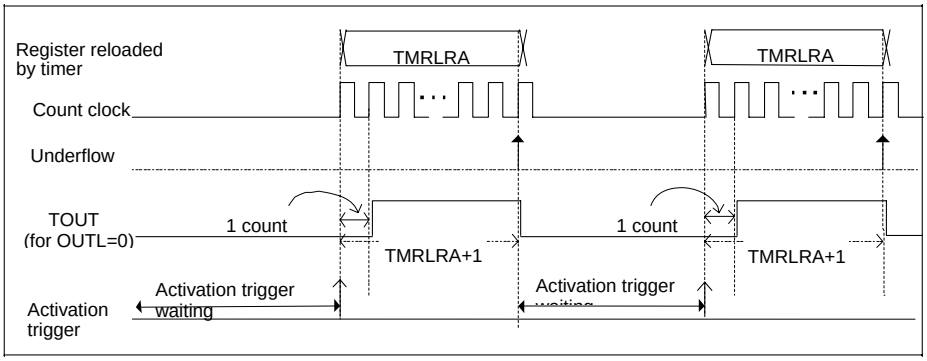


Compare one-shot function (TMRLRB > TMRLRA)

Figure 19-16. Compare One-shot Operation (2/2)

• Sets TMRLRB = TMRLRA

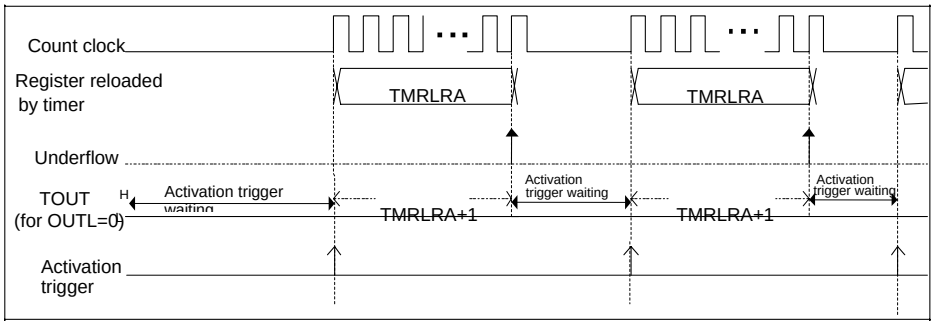
When the register relation is as described above, after loading to the timer, TMR will become smaller than TMRLRB after 1 count. Thus the TOUT output is the "L" level for 1 down count and then the "H" level until an underflow occurs. When an underflow occurs, the timer will turn into the activation trigger waiting state and the TOUT output will be the "L" level (for OUTL="0").



Compare one-shot function (TMRLRB=TMRLRA)

• Sets TMRLRB = 0

When the register relation is as described above, the TOUT output is the "L" level between down count start and an underflow occurrence because TMR will not become smaller than TMRLRB. The level will remain to be "L" even when an underflow occurs (for OUTL="0").



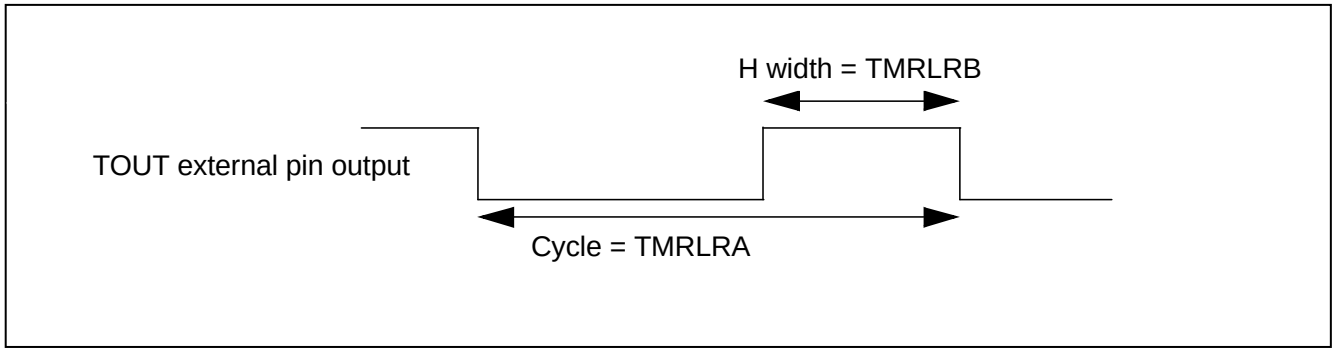
Compare one-shot function (TMRLRB="0")

19.5.3.6 Compare Reload Operation

The compare reload operation is shown below.

When bit15,14: MOD[1:0] = "10" and bit4:RELD = "1" of the TMCSR register, the timer compares a counter value (TMR) to the value of TMRLRB for each down count and if a compare match (TMR = TMRLRB) is detected, a down count starts and the TOUT output will be inverted. When an underflow occurs, the compare reload operation will be performed, in which a value is loaded from TMRLRA again and the down count operation starts. A load onto the counter starts from TMRLRA. The value of TMRLRA indicates the counter interval from a timer activation until a reload and the value of TMRLRB indicates the "H level width" after the TOUT output inverted from "L level output" to "H level output."
When TMR + 1 = TMRLRB, TOUT output will invert to the "H level" (when OUTL= "0").

Figure 19-17. TOUT Interval, Pulse Width



From the start of a down count to TMR = TMRLRB (while TMR is greater than or equal to TMRLRB), the following operation will be performed:

- TOUT output continues to hold the initial value.
- Count continues

When a down count starts from TMR = TMRLRB, the following operation will be performed:

- Inverts TOUT output.
- Count continues.
  (For the compare operation in interval timer mode, bit7:EF bit of TMCSR register will not be set.)

If an underflow occurs, the following operation will be performed:

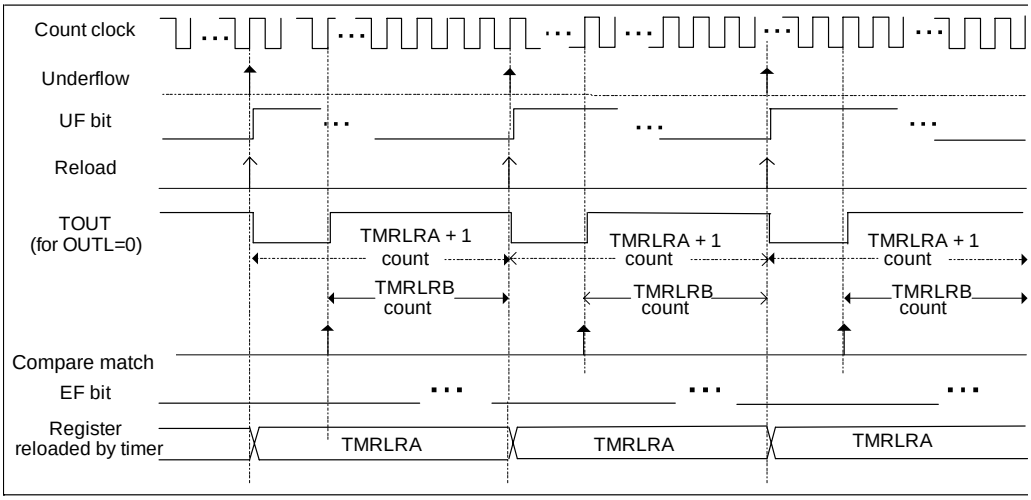
- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE= "1" of TMCSR register), an interrupt occurs.
- Initializes TOUT output.
- Reloads a value from TMRLRA.
- The timer continues to count.

The operation of a compare function depends on the setting relationship between TMRLRA and TMRLRB.

Figure 19-18. Compare Reload Operation (1/2)

• Sets TMRLRB < TMRLRA

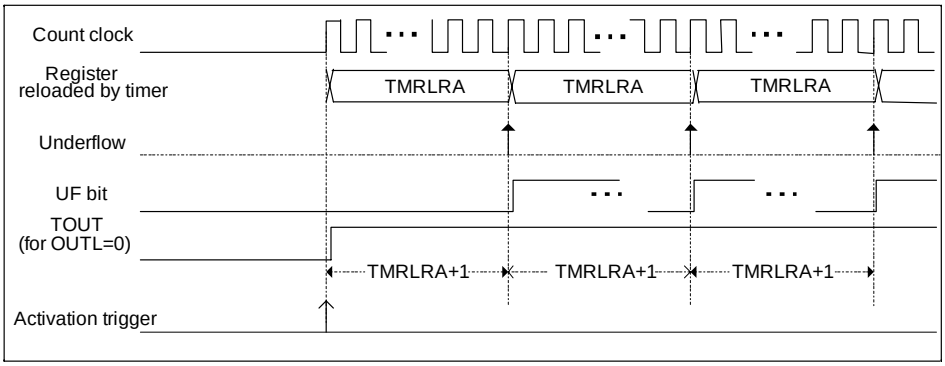
When the register relation is as described above, the TOUT output is the "L" level until TMR and TMRLRB match after loading to the timer. When down counting from the compare match (TMR=TMRLRB), the TOUT output is "H" level until the TOUT output is inverted and an underflow occurs. When an underflow occurs the TOUT output will be initialized. When an under flow occurs, the timer will reload from TMRLRA and continue counting operation (for OUTL="0").



Compare reload function (TMRLRB < TMRLRA) trigger input

• Sets TMRLRB > TMRLRA

When the register relation is as described above, the TOUT output is the "H" level after an activation trigger is generated and an underflow occurs because TMR is always smaller than TMRLRB. The level will remain to be "H" even when an underflow occurs. When an underflow occurs, the timer will load from TMRLRA and continue counting operation (for OUTL="0").

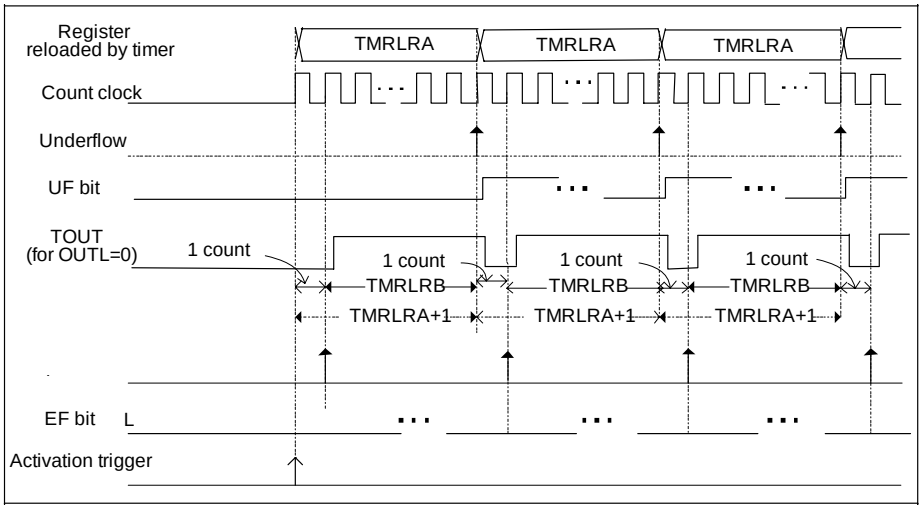


Compare reload function (TMRLRB > TMRLRA) trigger input

Figure 19-19. Compare Reload Operation (2/2)

• Sets TMRLRB = TMRLRA

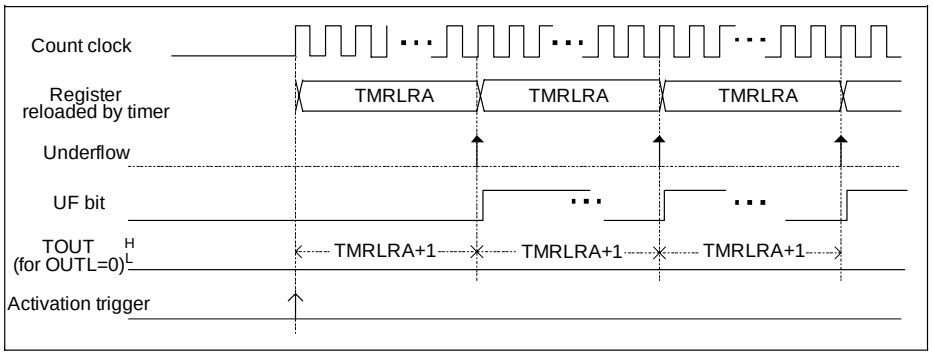
When the register relation is as described above, 1 count up after loading to the timer makes TMR become smaller than TMRLRB. Thus the TOUT output is the "L" level for 1 down count and then the "H" level until an underflow occurs. When an underflow occurs, the timer will reload from TMRLRA and continue counting operation. The TOUT output will remain to be the L level. (For OUTL= "0")



Compare reload function (TMRLRB = TMRLRA) trigger input

• Sets TMRLRB = 0

When the register relation is as described above, the TOUT output is the "L" level between down count start and an underflow occurrence after loading to the timer because TMR will not become smaller than TMRLRB. The level will remain to be "L" even when an underflow occurs.



Compare reload function (TMRLRB = "0") trigger input

19.5.3.7 Capture Mode

The capture mode is shown below.

When bit15,14:MOD[1:0] = "11" of the TMCSR register, the timer will perform capture operation. When a retrigger occurs, TMRLRB register captures the TMR value and sets bit7:EF of the TMCSR register.
When you use TIN input as the gate input (when bit8:GATE= "1" of the TMCSR register), generate a retrigger by bit0:TRG of the TMCSR register.
In a mode other than the capture mode, a capture will not be performed at a retrigger. The EF interrupt will also not be generated.
The timer operation and the TOUT output will be the same for the single one-shot function and the single reload function.

Note:

TOUT is not initialized in the one shot mode at retrigger.

Figure 19-20. Operation of Capture

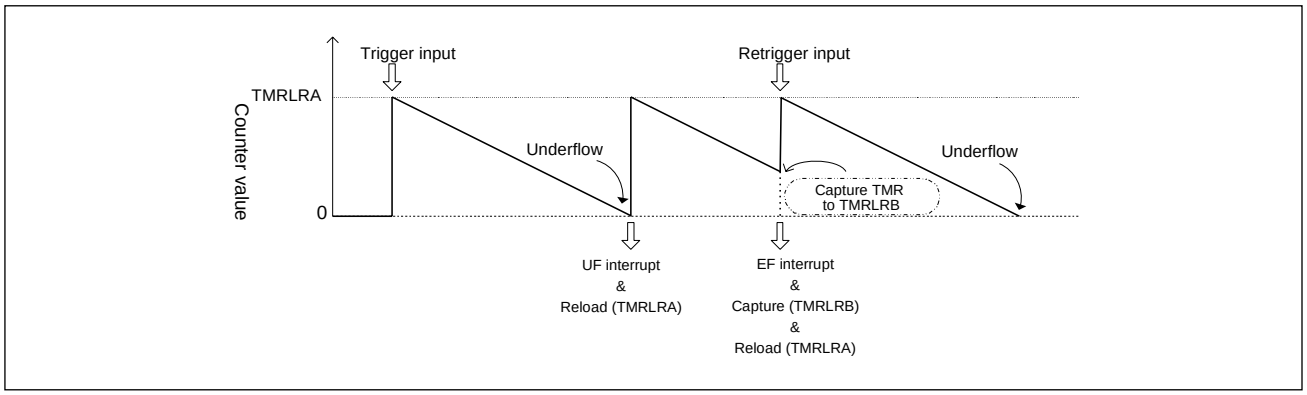


Figure 19-21. Flowchart of Trigger Input Functions in Interval Timer Mode

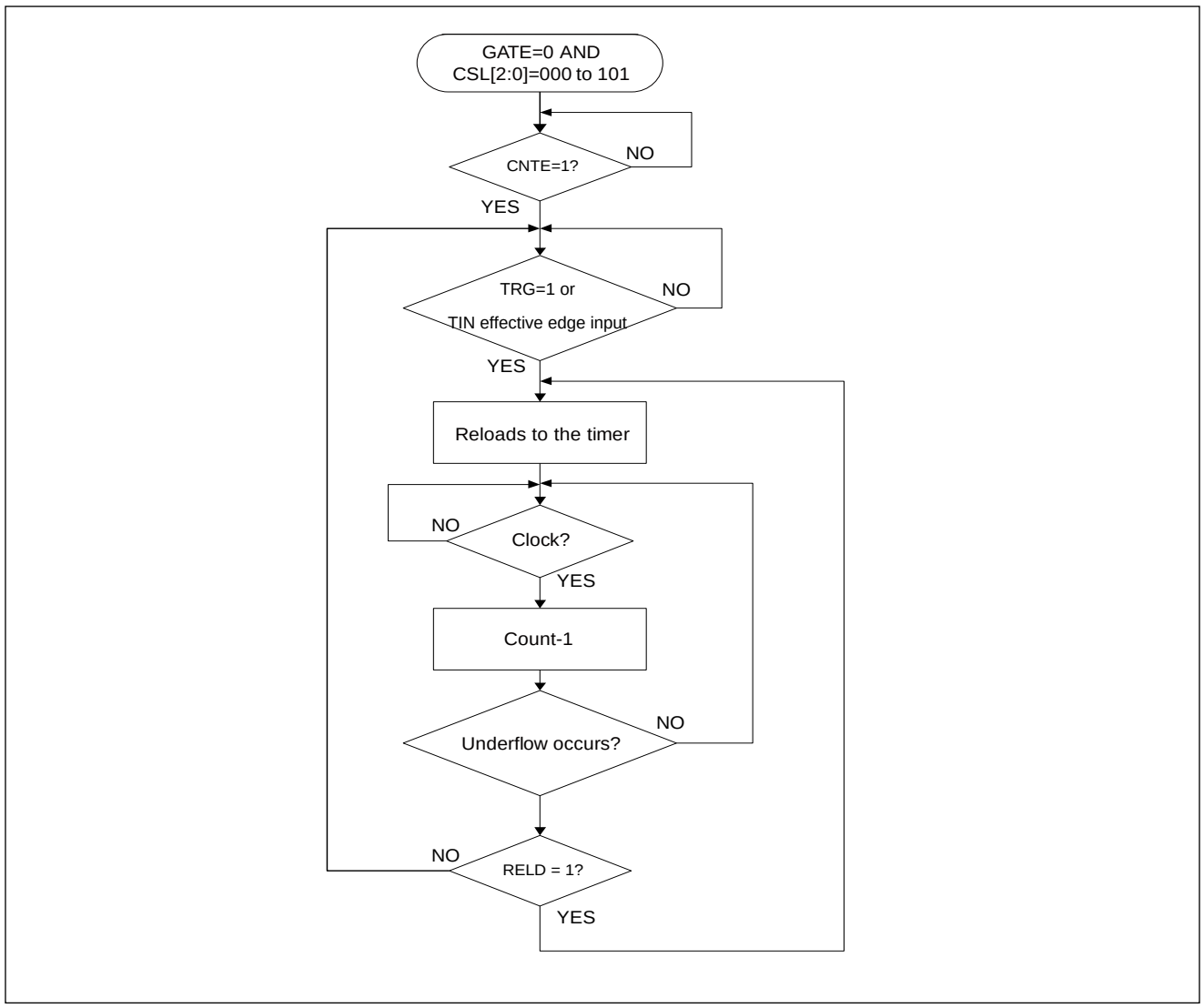
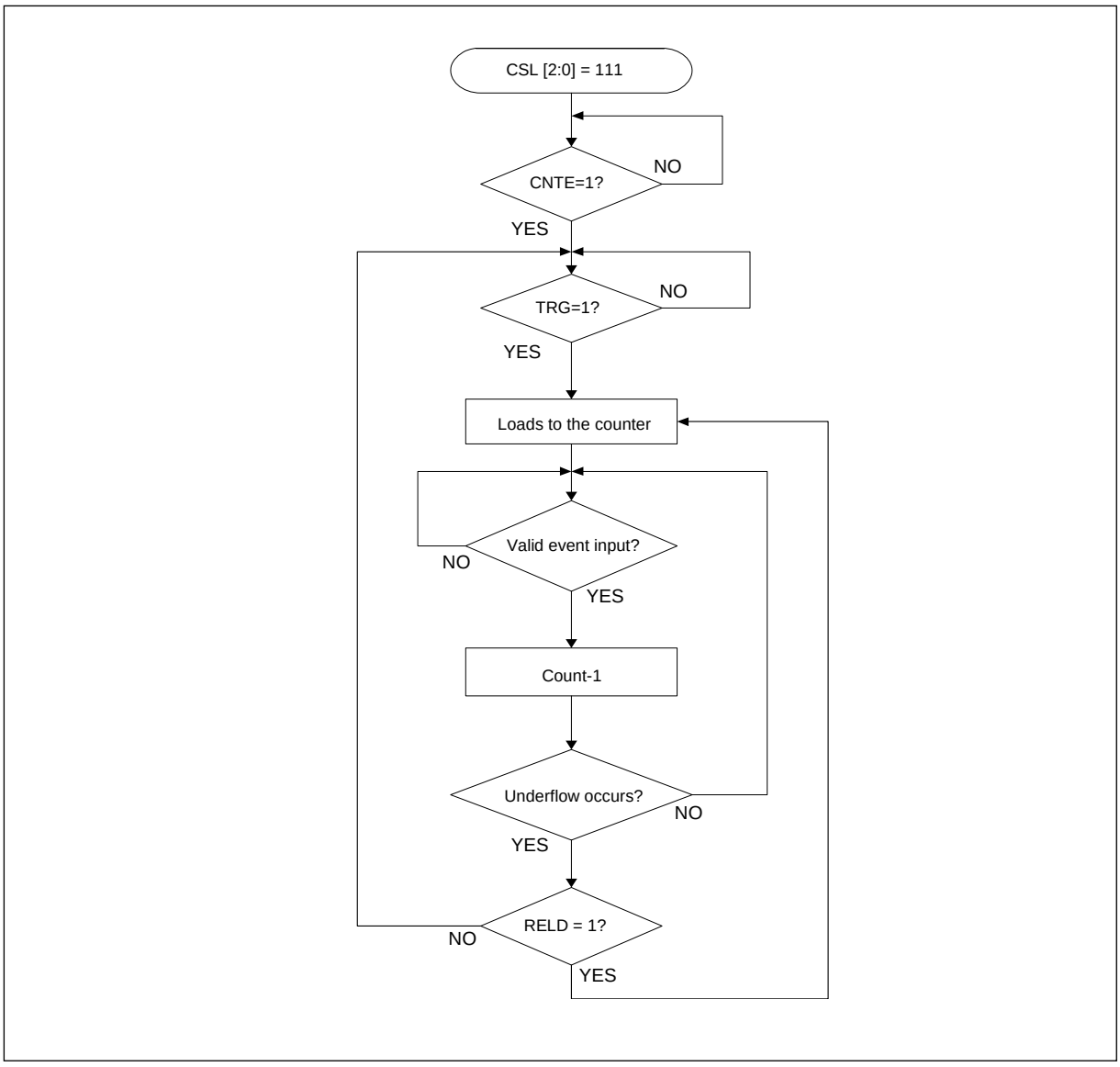


Figure 19-22. Flowchart in Event Counter Mode

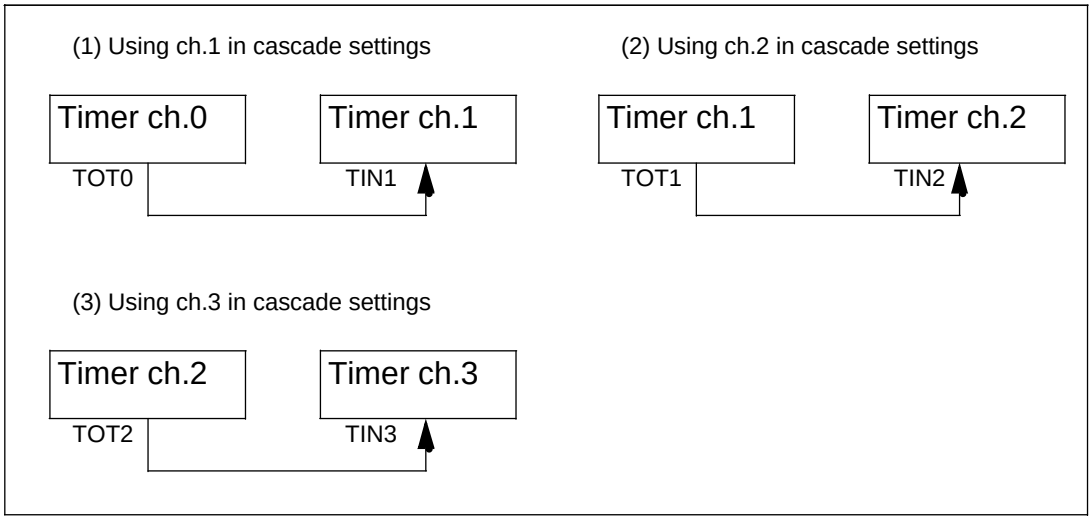


19.5.4 Cascade Input

Cascade input is shown below.

When you select cascade input (bit11 to bit9:CSL[2:0]="110" of TMCSR register), you can use the timer ch.0 output (TOUT0) for the input of ch.1 (TIN1), ch.1 output (TOUT1) for the input of ch.2 (TIN2), and ch.2 output (TOUT2) for the input of ch.3 (TIN3).

Figure 19-23. Timer Input/Output in Cascade Input Configuration



19.5.5 Priority of Concurrent Operations

The priority of concurrent operations is shown below.

When two events to decide the timer operation occur simultaneously, the priority of deciding the operating state is indicated.

1. Writing to register
2. Trigger input
3. Underflow
4. Clock input

When a set of each flag by the timer operation and a clear of a flag by register write occur concurrently, the priority of deciding the operation is indicated.

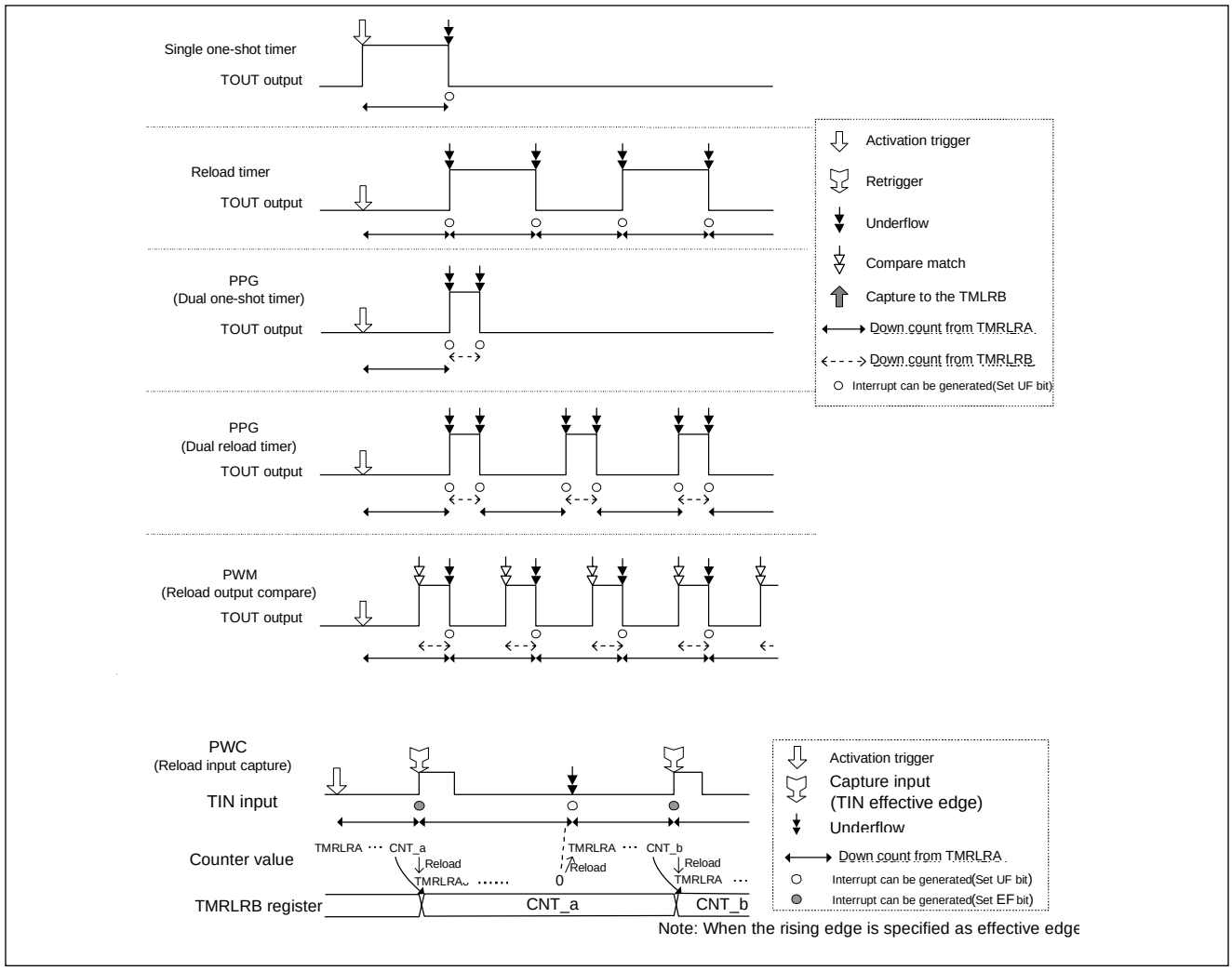
1. Setting flag by the timer operation
2. Writing to a register for a clear of flag to the UF bit/EF bit

19.6 Application Note

This section explains an application note concerning the register for the reload timer.

This section shows the typical functions which can be realized with this timer.

Figure 19-24. 1 Example



Following are some configurations for use of example figure above.

Table 19-7. Example of Configuration

Function	MOD[1:0]	RELD	TMRLRA	TMRLRB
Single one-shot timer	00 (Single mode)	0	Mandatory	---
Reload timer	00 (Single mode)	1	Mandatory	---
PPG (Programmable Pulse Generator)	01 (Dual mode)	0 or 1	Mandatory	Mandatory
PWM (Pulse Width Modulator)	10 (Compare mode)	1	Mandatory	Mandatory
PWC (Pulse Width Counter)	11 (Capture mode)	1	Mandatory	---

19.6.1 Single One-shot Timer

The single one-shot timer is shown below.

The single one-shot timer loads a value from the TMRLRA register onto the counter and starts to decrement the counter (down count operation) when a trigger is input. When an underflow occurs, the counting stops.
The TOUT pin outputs the "H level" in counting and when an underflow occurs it will output the "L level." (When OUTL= "0")

[Configuration] To use this timer as a single one-shot timer, configure as follows.

1. When TIN input is not used

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	00	*1	0	-	*2	0	*3	-	1	S	

S: Use at timer activation

-: Does not influence operation

*1: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*2: TOUT output polarity setting

OUTL=0------Initial value L >> Count starts H >> Underflow occurs L

OUTL=1------Initial value H >> Count starts L >> Underflow occurs H

*3: Interrupt request enable setting

INTE=0------Interrupt disabled

INTE=1------Interrupt enabled

2. When using TIN input as a gate input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	1	-	*3	0	*4	-	1	S	

S: Use at timer activation

-: Does not influence operation

*1: TIN effective level setting

TRGM[1:0]=x0------Count only for L input interval

TRGM[1:0]=x1------Count only for H input interval

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: TOUT output polarity setting

OUTL= 0------Initial value L >> Count starts H >> Underflow occurs L

OUTL= 1------Initial value H >> Count starts L >> Underflow occurs H

*4: Interrupt request enable setting

INTE= 0------Interrupt disabled

INTE= 1------Interrupt enabled

3. When using TIN input as a trigger input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	0	-	*3	0	*4	-	1	S	

S: Use at timer activation

-: Does not influence operation

*1: TIN effective level setting

TRGM[1:0]= 00------Does not detect external trigger edge

TRGM[1:0]= 01------Rising edge

TRGM[1:0]= 10------Falling edge

TRGM[1:0]= 11------Both edges

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: OUT output polarity setting

OUTL= 0------Initial value L >> Count starts H >> Underflow occurs L

OUTL= 1------Initial value H >> Count starts L >> Underflow occurs H

*4: Interrupt request enable setting

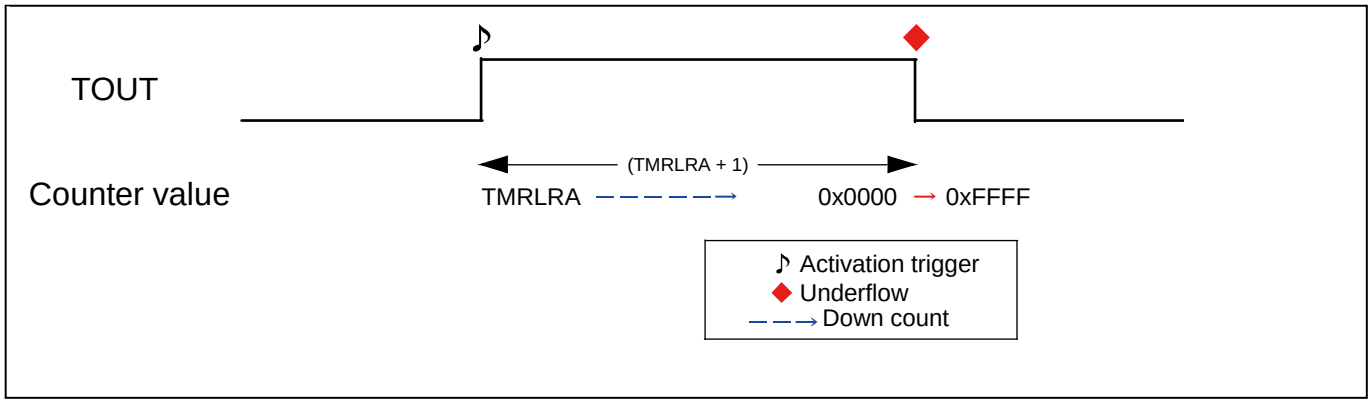
INTE= 0------Interrupt disabled

INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer:

- ☐ Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin)
- ☐ Input an effective level when you use TIN pin input as the gate input

Figure 19-25. Example of Operation (OUTL = 0)



19.6.2 Reload Timer

The reload timer is shown below.

The reload timer loads from the TMRLRA register onto the counter and repeats the down count operation each time underflow occurs. The TOUT outputs the "L level" while the count is ongoing from the activation trigger to the occurrence of the first underflow. The output is inverted each time an underflow occurs, and the TOUT outputs "H level" with the occurrence of the first underflow. When a retrigger occurs, TOUT output returns to its initial value. (When OUTL= "0")

[Configuration] To use the timer as the reload timer, configure as follows.

1. When TIN input is not used

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	00	*1	0	-	*2	1	*3	-	1	S	

S: Use at timer activation

-: Does not influence operation

*1: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*2: TOUT output polarity setting

OUTL=0------Initial value L >> Count starts L >> Invert whenever an underflow occurs

OUTL=1------Initial value H >> Count starts H >> Invert whenever an underflow occurs

*3: Interrupt request enable setting

INTE=0------Interrupt disabled

INTE=1------Interrupt enabled

2. When using TIN input as a gate input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	1	-	*3	1	*4	-	1	S	

S: Use at timer activation

-: Does not influence operation

*1: TIN effective level setting

TRGM[1:0]=x0------Count only for TIN=L input interval

TRGM[1:0]=x1------Count only for TIN=H input interval

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: OUT output polarity setting

OUTL=0------Initial value L >> Count starts L >> Invert whenever an underflow occurs

OUTL=1------Initial value H >> Count starts H >> Invert whenever an underflow occurs

*4: Interrupt request enable setting

INTE=0------Interrupt disabled

INTE=1------Interrupt enabled

3. When using TIN input as a trigger input

TMCSR											TMRLRA
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	Count initial value setting
00	*1	*2	0	-	*3	1	*4	-	1	S	

S: Use at timer activation

-: Does not influence operation

*1: TIN effective edge setting

TRGM[1:0]= 00------Does not detect external trigger edge

TRGM[1:0]= 01------Rising edge

TRGM[1:0]= 10------Falling edge

TRGM[1:0]= 11------Both edges

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: OUT output polarity setting

OUTL= 0------Initial value L >> Count starts L >> Invert whenever an underflow occurs

OUTL= 1------Initial value H >> Count starts H >> Invert whenever an underflow occurs

*4: Interrupt request enable setting

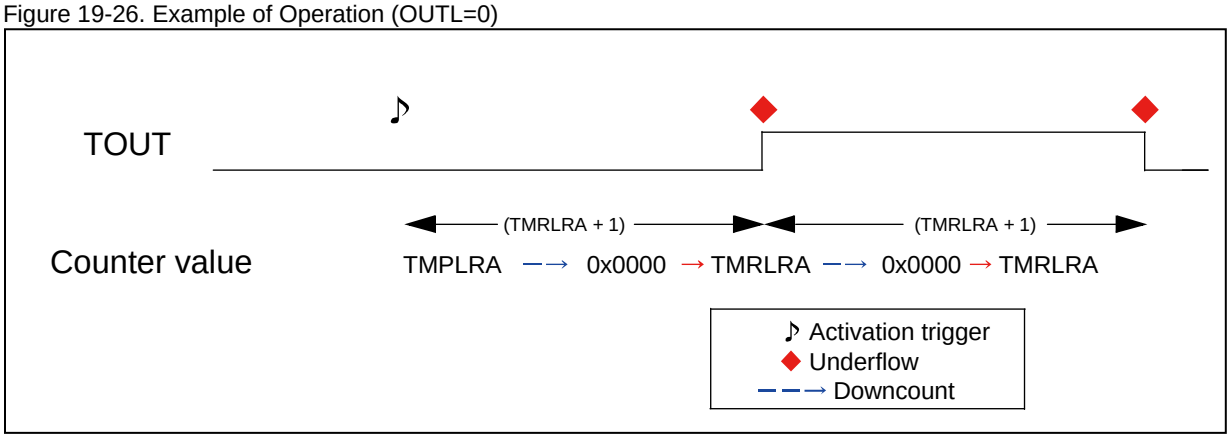
INTE= 0------Interrupt disabled

INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer:

- ☐ Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin)
- ☐ Input an effective level when you use TIN pin input as the gate input

Figure 19-26. Example of Operation (OUTL=0)



19.6.3 PPG

PPG is shown below.

PPG is the function which generates an output pulse by configuring L width/H width of the pulse. An activation trigger launches a load from TMRLRA to the counter and executes a down count. The operation switches to load the value from TMRLRB and execute a down count when an underflow occurs.
When RELD="0", "Activation trigger >> TMRLRA load >> Down count >> Underflow >> TMRLRB load >> Down count >> Underflow," the down count stops.
When RELD="1", counter is loaded with TMRLRA/TMRLRB alternatively and executes down count whenever an underflow occurs, such as Activation trigger >> TMRLRA load >> Down count >> Underflow >> TMRLRB load >> Down count >> Underflow >> TMRLRA load >> Down count >> Underflow >> TMRLRB load and so on.
The TOUT outputs the "L level" while counting until the occurrence of an underflow caused by the down count from TMRLRA, and outputs the "H level" while counting until the occurrence of an underflow caused by the down count from TMRLRB. When a retrigger occurs, TOUT output returns to its initial value.

Note:

TOUT is not initialized in the one shot mode at retrigger.

[Configuration] To use the timer as PPG, configure as follows.

1. When TIN input is not used

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
01	00	*1	0	-	*2	*3	*4	-	1	S		

(A): The count initial value at an activation trigger/The reload value at an underflow caused by the count from the TMRLRB value (when RELD=1)

(B): The reload value at an underflow caused by the count from the TMRLRA value

S: Use at timer activation

-: Does not influence operation

*1: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*2: TOUT output polarity setting

OUTL= 0------

Initial value L >> Count L from TMRLRA >> H when an underflow occurs >>

Count H from TMRLRB >> L when an underflow occurs

OUTL= 1------

Initial value H => Count H from TMRLRA => L when an underflow occurs =>

Count L from TMRLRB >> H when an underflow occurs

*3: Reload setting when an underflow occurs

RELD= 0------One-shot mode

RELD= 1------Reload mode

*4: Interrupt request enable setting

INTE= 0------Interrupt disabled

INTE= 1------Interrupt enabled

2. When using TIN input as a gate input

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
01	*1	*2	1	-	*3	*4	*5	-	1	S		

(A): The count initial value at an activation trigger/The reload value at an underflow caused by the count from the TMRLRB value (when RELD=1)

(B): The reload value at an underflow caused by the count from the TMRLRA value

S: Use at timer activation

-: Does not influence operation

*1: TIN effective level setting

TRGM[1:0]= x0------Count only for TIN=L input interval

TRGM[1:0]= x1------Count only for TIN=H input interval

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: TOUT output polarity setting

OUTL= 0------

Initial value L >> Count L from TMRLRA >> H when an underflow occurs >>

Count H from TMRLRB >> L when an underflow occurs

OUTL= 1------

Initial value H >> Count H from TMRLRA >> L when an underflow occurs >>

Count L from TMRLRB >> H when an underflow occurs

*4: Reload setting when an underflow occurs

RELD=0------One-shot mode

RELD=1------Reload mode

*5: Interrupt request enable setting

INTE=0------Interrupt disabled

INTE=1------Interrupt enabled

3. When using TIN input as a trigger input

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
01	*1	*2	0	-	*3	*4	*5	-	1	S		

(A): The count initial value at an activation trigger/The reload value at an underflow caused by the count from the TMRLRB value (when RELD=1)

(B): The reload value at an underflow caused by the count from the TMRLRA value

S: Use at timer activation

-: Does not influence operation

*1: TIN effective edge setting

TRGM[1:0]= 00------Does not detect external trigger edge

TRGM[1:0]= 01------Rising edge

TRGM[1:0]= 10------Falling edge

TRGM[1:0]= 11------Both edges

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: TOUT output polarity setting

OUTL= 0------

Initial value L >> Count L from TMRLRA >> Invert whenever an underflow occurs

OUTL= 1------

Initial value H >> Count H from TMRLRA >> Invert whenever an underflow occurs

*4: Reload setting when an underflow occurs

RELD=0------One-shot mode

RELD=1------Reload mode

*5: Interrupt request enable setting

INTE=0------Interrupt disabled

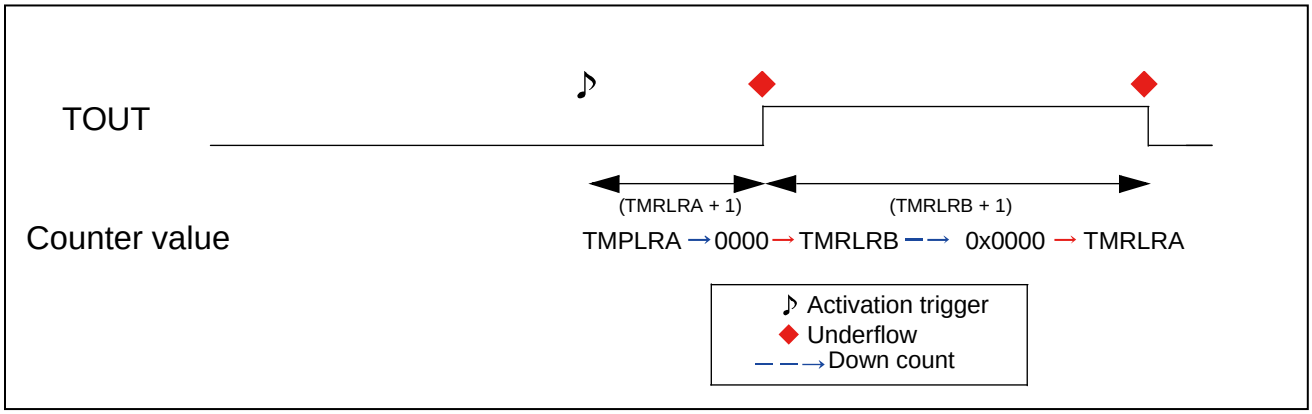
INTE=1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer:

☐ Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin)

☐ Input an effective level when you use TIN pin input as the gate input

Figure 19-27. Example of Operation (OUTL=0)



19.6.4 PWM

PWM is shown below.

PWM is the function which generates an output pulse by configuring the pulse interval and H width.
An activation trigger launches a load from TMRLRA to the counter and executes a down count.
TOUT outputs the "L level" after an activation trigger and then outputs the "H level" when the counter value becomes smaller than the TMRLRB value. When an underflow occurs, TOUT output returns to its initial value. (When OUTL= "0")

When RELD= "0", Activation trigger >> TMRLRA load >> Down count >> Underflow, then counter stops the down count.
When RELD= "1", counter is loaded with TMRLRA, and it is decremented for each load whenever an underflow occurs, such as Activation trigger >> TMRLRA load >> Down count >> Underflow >> TMRLRA load >> Down count, and so on.

[Configuration] To use the timer as PWM, configure as follows.

1. When TIN input is not used

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
10	0	*1	0	-	*2	*3	*4	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)

(B): Set the value to compare to the counter value (TMRLRB < TMRLRA) *5

S: Use at timer activation

-: Does not influence operation

*1: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*2: TOUT output polarity setting

OUTL= 0------

Initial value L >> Count L from TMRLRA >> H, the counter value is smaller than TMRLRB

OUTL= 1------

Initial value H >> Count H from TMRLRA >> L, the counter value is smaller than TMRLRB

*3: Reload setting when an underflow occurs

RELD= 0------One-shot mode

RELD= 1------Reload mode

*4: Interrupt request enable setting

INTE= 0------Interrupt disabled

INTE= 1------Interrupt enabled

*5: To use TOUT output with L clip output, set to TMRLRB = "0".

To use TOUT output with H clip output, set to TMRLRB = "TMRLRA + 1".

2. When using TIN input as a gate input

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
10	*1	*2	1	-	*3	*4	*5	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)

(B): Set the value to compare to the counter value (TMRLRB < TMRLRA) *6

S: Use at timer activation

-: Does not influence operation

*1: TIN effective level setting

TRGM[1:0]= x0------Count only for TRGM=L input interval

TRGM[1:0]= x1------Count only for TRGM=H input interval

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: TOUT output polarity setting

OUTL= 0------

Initial value L >> Count L from TMRLRA >> H, the counter value is smaller than TMRLRB

OUTL= 1------

Initial value H >> Count H from TMRLRA >> L, the counter value is smaller than TMRLRB

*4: Reload setting when an underflow occurs

RELD=0------One-shot mode

RELD=1------Reload mode

*5: Interrupt request enable setting

INTE=0------Interrupt disabled

INTE=1------Interrupt enabled

*6: To use TOUT output with L clip output, set to TMRLRB = "0".
To use TOUT output with H clip output, set to TMRLRB = "TMRLRA + 1".

3. When using TIN input as a trigger input

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
10	*1	*2	0	-	*3	*4	*5	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)

(B): Set the value to compare to the counter value (TMRLRB < TMRLRA) *6

S: Use at timer activation

-: Does not influence operation

*1: TIN effective edge setting

TRGM[1:0]= 00------Does not detect external trigger edge

TRGM[1:0]= 01------Rising edge

TRGM[1:0]= 10------Falling edge

TRGM[1:0]= 11------Both edges

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: TOUT output polarity setting

OUTL= 0------

Initial value L >> Count L from TMRLRA >> H, the counter value is smaller than TMRLRB

OUTL= 1------

Initial value H >> Count H from TMRLRA >> L, the counter value is smaller than TMRLRB

*4: Reload setting when an underflow occurs

RELD= 0------One-shot mode

RELD= 1------Reload mode

*5: Interrupt request enable setting

INTE= 0------Interrupt disabled

INTE= 1------Interrupt enabled

*6: To use TOUT output with L clip output, set to TMRLRB = "0".

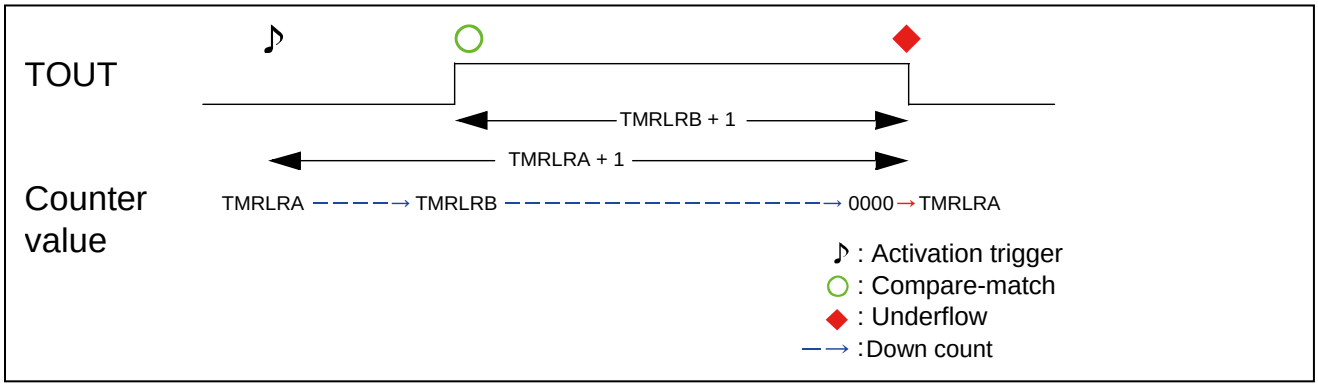
To use TOUT output with H clip output, set to TMRLRB = "TMRLRA + 1".

[Timer activation] Follow the steps below to activate the timer:

☐ Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin)

☐ Input an effective level when you use TIN pin input as the gate input

Figure 19-28. Example of Operation (OUTL=0)



19.6.5 PWC

PWC is shown below.

PWC is the function to measure the time interval between triggers to input.
An activation trigger launches a load of a value from TMRLRA onto the counter and executes a down count operation. A trigger input during a count enables the counter value at that time to be captured onto TMRLRB, which allows measuring the time interval between triggers to input.

[Configuration] To use the timer as PWC, configure as follows.

TMCSR											TMRLRA	TMRLRB
MOD [1:0]	TRGM [1:0]	CSL [2:0]	GATE	EF	OUTL	RELD	INTE	UF	CNTE	TRG	(A)	(B)
11	*1	*2	0	-	*3	*4	*5	-	1	S		

(A): The count initial value when activation trigger occurs/The reload value at an underflow (when RELD=1)

(B): Count value at trigger generation during count operation

S: Use at timer activation

-: Does not influence operation

*1: TIN effective edge setting

TRGM[1:0]= 00------Does not detect external trigger edge

TRGM[1:0]= 01------Rising edge

TRGM[1:0]= 10------Falling edge

TRGM[1:0]= 11------Both edges

*2: Count clock division setting

CSL[2:0]= 000------Division of peripheral clock (PCLK) by 2

CSL[2:0]= 010------Division of peripheral clock (PCLK) by 8

CSL[2:0]= 011------Division of peripheral clock (PCLK) by 16

CSL[2:0]= 100------Division of peripheral clock (PCLK) by 32

CSL[2:0]= 101------Division of peripheral clock (PCLK) by 64

*3: TOUT output polarity setting

OUTL= 0------

Initial value L >> Count L from TMRLRA >> Invert whenever an underflow occurs

OUTL= 1------

Initial value H >> Count H from TMRLRA >> Invert whenever an underflow occurs

*4: Reload setting when an underflow occurs

RELD= 0------One-shot mode

RELD= 1------Reload mode

*5: Interrupt request enable setting

INTE= 0------Interrupt disabled

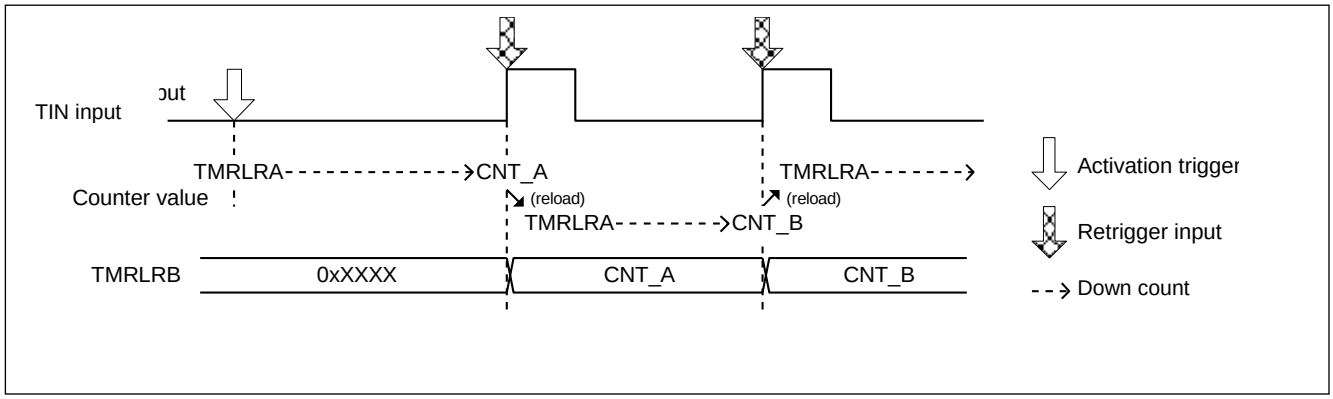
INTE= 1------Interrupt enabled

Timer activation] Follow the steps below to activate the timer:

- ☐ Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TIN pin) While down counting, the counter value will be captured onto the TMRLRB whenever a trigger input occurs. The time interval between edges of the triggers to input will be obtained by the following formula.

  T = (The set value for TMRLRA - The captured value for TMRLRB) x Peripheral clock (PCLK) cycle x Division ratio set with CSL

Figure 19-29. Example of Operation (TRGM=01)



20. 16-bit Free-Run Timer

This chapter explains the 16-bit free-run timer.

20.1 Overview
20.2 Features
20.3 Configuration
20.4 Registers
20.5 Operation

20.1 Overview

This section explains the overview of 16-bit free-run timers.

The free-run timers consist of 16-bit free-run timers.

20.2 Features

This section explains the features of 16-bit free-run timers.

Function of the 16-bit Free-run Timer

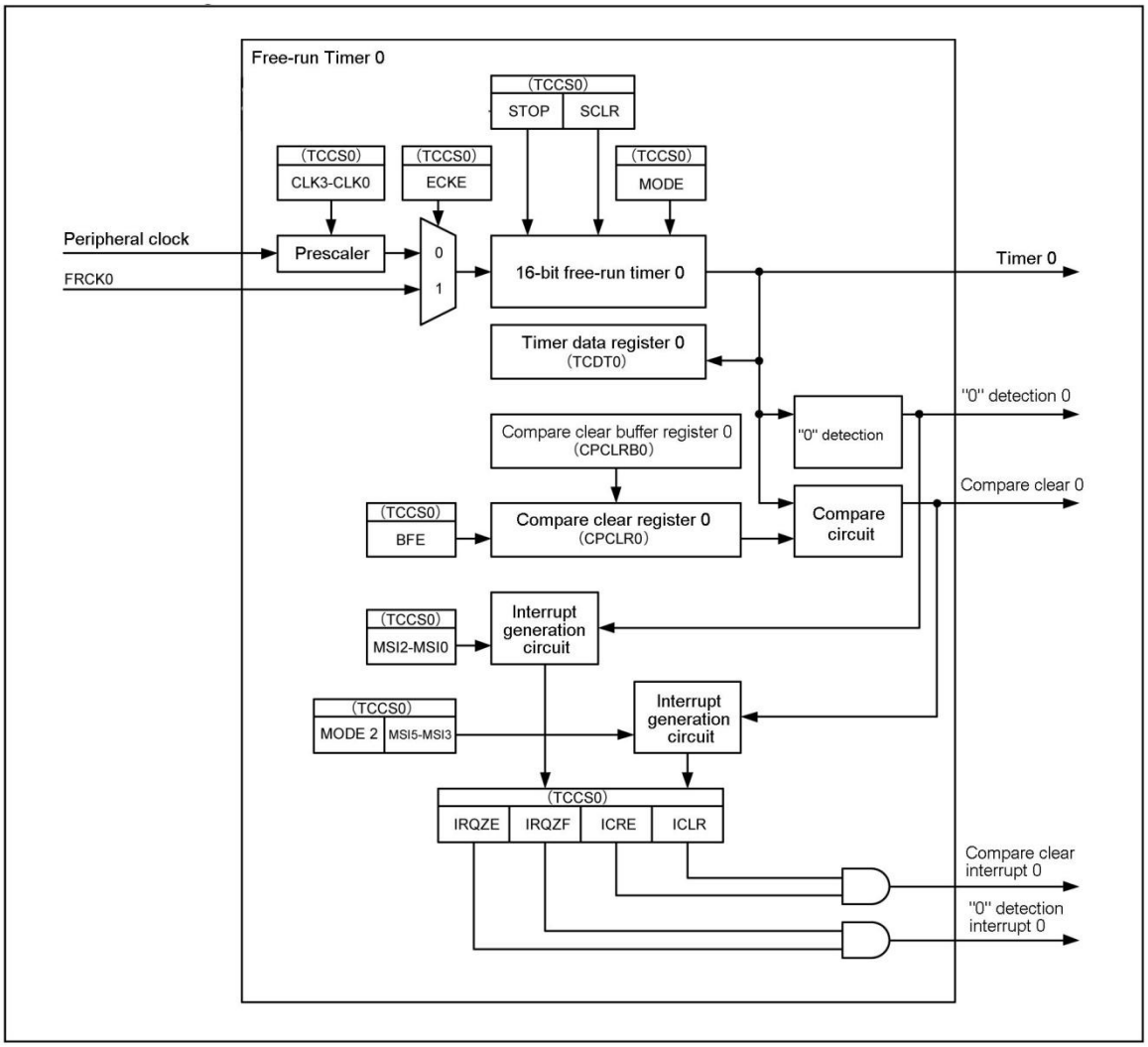
- The 16-bit free-run timer consists of 16-bit up/down counter, control register, 16-bit compare clear register (with buffer register), and prescaler.
- You can select one of the 9 counter operation clocks (φ, φ/2, φ/4, φ/8, φ/16, φ/32, φ/64, φ/128, φ/256). (φ: peripheral clock)
- A compare clear interrupt will be generated when a compare clear register matches the 16-bit free-run timer upon comparison of them. 0 detection interrupt will be generated while the 16-bit free-run timer is detecting the count value "0".
- The compare clear register comes with selectable buffer registers (Data written to this buffer register will be transferred to the compare clear register). Once data is written to the buffer after the 16-bit free-run timer has stopped, the transfer will be executed immediately. If the timer value "0" is detected while the 16-bit free-run timer is active, data will be transferred from the buffer.
- If there is a reset or if there is a compare match with the software clear or compare clear register, the counter value will be reset to "0x0000".
- This counter output value can be used as a clock count of the input capture.

20.3 Configuration

This section explains the configuration of 16-bit free-run timers.

20.3.1.1 Configuration of the 16-bit Free-run Timer

Figure 20-1. Configuration of the 16-bit Free-run Timer



20.4 Registers

This section explains the registers of 16-bit free-run timers.

External pin table

Table 20-1. 16-bit Free-run Timer External Pin

Channel	External Pin (FRCK)
0	FRCK

List of registers

Table 20-2. List of Registers for the 16-bit Free-run Timer

Address	+0	+1	+2	+3
0x1204	Compare clear buffer register 0 (CPCLRB0) Compare clear register 0 (CPCLR0)		Timer data register (TCDT0)	
0x1208	Timer state control register 0 (TCCS0)			Reserved

20.4.1 Registers for the 16-bit Free-run Timer

This section explains the registers of 16-bit free-run timers.

The 16-bit free-run timer consists of the compare clear buffer register, the compare clear register, the timer data register, and the timer state control register.

20.4.1.1 Compare Clear Buffer Register: CPCLRB/Compare Clear Register: CPCLR

This section explains the bit configuration of the compare clear buffer register and compare clear register.

The compare clear buffer register (CPCLRB) is a 16-bit buffer register contained in the compare clear register (CPCLR). The CPCLRB and CPCLR registers are located at the same address.

CPCLRB0: Address 1204H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial Value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial Value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

[bit15 to bit0] CL15 to CL00: Compare clear value buffer bits

	Function
CL15 to CL00	Compare clear value buffer

- The compare clear buffer register is a buffer register located at the same address of the compare clear register (CPCLR).
- If the buffer function is disabled (BFE:bit23 of timer state control register (TCCS) is 0) or the free-run timer stops, the value of the compare clear buffer register will be immediately transferred to the compare clear register.
- If the buffer function is enabled, the value will be transferred to the compare clear register when the count value "0" of the 16-bit free-run timer is detected.

Note:

The "0x0000" setting is prohibited in the compare clear buffer register.
When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction for access.

CPCLR0: Address 1204H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial Value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial Value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

[bit15 to bit0] CL15 to CL00: Compare clear value bits

	Function
CL15 to CL00	Compare clear value

- The compare clear register is used for comparison with the count value of the 16-bit free-run timer.
- In the up count mode, if this register matches the count value of the 16-bit free-run timer, the 16-bit free-run timer will be reset to "0x0000".
- In the up/down count mode, if this register matches the count value of the 16-bit free-run timer, the 16-bit free-run timer will be converted from up count to down count or it will be converted from down count to up count when "0" is detected.

Note:

When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction for access.

20.4.1.2 Timer Data Register: TCDT0

This section explains the bit configuration of the timer data register.

The timer data register (TCDT) is used to read the count value of the 16-bit free-run timer. It is also possible to set the count value of the 16-bit free-run timer.

TCDT0: Address 1206H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	T15	T14	T13	T12	T11	T10	T09	T08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	T07	T06	T05	T04	T03	T02	T01	T00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

[bit15 to bit0] T15 to T00: Count value bits

	Function
T15 to T00	Count value

- The timer data register is used to read the count value of the 16-bit free-run timer.
- The count value will be cleared to "0x0000" as soon as reset occurs.
- The timer value can be set by writing a value to this register. However, a value needs to be written while the timer is inactive (STOP:bit22 of timer state control register (TCCS) is 1).
- The 16-bit free-run timer will be initialized as soon as any of the following occurs.
  - ☐ Reset
  - ☐ While the 16-bit free-run timer is active (STOP:bit22 of timer state control register (TCCS) is 0), the clear bit (SCLR:bit20) of the timer state control register (TCCS) is 1.
  - ☐ The timer count value matches the compare clear register in the up count mode (MODE:bit21 of timer state control register (TCCS) is 0).

Note:

The 16-bit free-run timer will not be initialized even when the clear bit (SCLR:bit20) of the timer state control register (TCCS) is set to 1 while the 16-bit free-run timer is inactive (STOP:bit22=1 of timer state control register (TCCS)).
When accessing the timer data register, use a half-word or word access instruction.
If a count value is written during the up/down count mode (MODE:bit21=1 of timer state control register (TCCS)), an unintended counting may be performed.
To write a count value in the up/down count mode (MODE:bit21=1 of timer state control register (TCCS)), perform the following steps.

1. Stop counting the 16-bit free-run timer. (Writing "1" in STOP:bit21 of timer state control register (TCCS))
2. Set a count value for the timer data register.
3. Perform software clear. (Writing "1" in SCLR:bit20 of timer state control register (TCCS))
4. Start counting the 16-bit free-run timer.

20.4.1.3 Timer State Control Register: TCCS0

This section explains the bit configuration of timer state control register.

The timer state control register (TCCS) is used to control the operation of the 16-bit free-run timer.

TCCS0: Address 1208H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	ECKE	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R(RM1),W	R/W	R,W	R,W	R,W	R(RM1),W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0
Initial Value	0	1	0	0	0	0	0	0
Attribute	R/W	R,W	R/W	R0,W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved				MODE2	MSI5	MSI4	MSI3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1	R1,W1

[bit31] ECKE: Clock selection bit

ECKE	Function
0	Peripheral clock
1	External clock

- This bit is used to select a peripheral clock or external clock as the count clock of the 16-bit free-run timer.
- When this bit is set to "0":

  The peripheral clock is selected. To select the count clock frequency, you will also need to select the clock frequency selection bits (CLK3 to CLK0) of the TCCS register.
- When this bit is set to "1":

  The external clock is selected, and it is clock inputted from a pin named "external clock (FRCK)".

Note:

The count clock will be changed as soon as this bit is set. Therefore, this bit must be changed while input capture is inactive.

[bit30] IRQZF: 0 detection interrupt flag bit

IRQZF	Function	
	Read	Write
0	No 0 detected	This bit is cleared.
1	0 detected	This bit remains unaffected.

- When the count value of the 16-bit free-run timer is set to "0x0000", this bit will be set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- This bit is cleared when the 0 detection interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
This bit is not set by a software clear (writing "1" to SCLR:bit20 of the timer state control register (TCCS)) while the 16-bit free-run timer is active (STOP:bit22=0 in the timer state control register (TCCS)).
In the up/down count mode (MODE:bit21=1 in the timer state control register (TCCS)), this bit will be set to "1" when the interrupt configured by the interrupt mask selection bits (MSI2 to MSI0:bit28 to bit26 of the timer state control register (TCCS) is other than "000") is generated. If no interrupt occurs, this bit will not be set to "1".
In the up count mode (MODE:bit21=0), this bit will be set every time 0 detection occurs regardless of the value of MSI2 to MSI0:bit28 to bit26.
If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit29] IRQZE: 0 detection interrupt request enable bit

IRQZE	Function
0	Disable interrupt request.
1	Enable interrupt request.

- When this bit and interrupt flag bit (IRQZF:bit30) are set to "1", an interrupt request for CPU will be generated.

[bit28 to bit26] MSI2 to MSI0: Interrupt mask selection bits

MSI2	MSI1	MSI0	Function
0	0	0	An interrupt will be generated when there is a match for the 1st time.
0	0	1	An interrupt will be generated when there is a match for the 2nd time.
0	1	0	An interrupt will be generated when there is a match for the 3rd time.
0	1	1	An interrupt will be generated when there is a match for the 4th time.
1	0	0	An interrupt will be generated when there is a match for the 5th time.
1	0	1	An interrupt will be generated when there is a match for the 6th time.
1	1	0	An interrupt will be generated when there is a match for the 7th time.
1	1	1	An interrupt will be generated when there is a match for the 8th time.

- When MODE2:bit11 of the timer state control register (TCCS) is 0:
  - ☐ These bits are used to set the mask count for the compare clear interrupt in the up count mode (MODE:bit21=0 in the timer state control register (TCCS)). In the up/down count mode (MODE:bit21=1 in the timer state control register (TCCS)), they are used to set the mask count for the 0 detection interrupt.
  - ☐ When this bit is set to "0", the interrupt factor will not be masked.
- When MODE2:bit11 of the timer state control register (TCCS) is 1:
  - ☐ These bits are used to set the mask count for the 0 detection interrupt in the up/down count mode (MODE:bit21=1 in the timer state control register (TCCS)).
  - ☐ Settings of the up count mode (MODE:bit21 of the timer state control register (TCCS) is 0) are disabled.

Note:

The value read is a mask counter value.
If a read-modify-write instruction is executed, the value read is a mask register value.
The written data will be written to the mask register.
The written value to the mask register while the free-run timer is active (STOP:bit22 of the timer state control register (TCCS) is 0) will be reloaded to the counter only when the mask counter becomes "0".
The written value to the mask register while the free-run timer is inactive (STOP:bit22 of the timer state control register (TCCS) is 1) will be immediately reloaded to the counter.

[bit25] ICLR: Compare clear interrupt flag bit

ICLR	Function	
	Read	Write
0	No compare clear match	This bit is cleared.
1	Compare clear match	This bit remains unaffected.

- This bit will be set to "1" when the compare clear value matches the 16-bit free-run timer value.
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- This bit will be cleared when the compare clear match interrupt clear signal is "H".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
In the up count mode (MODE:bit21=0 in the timer state control register (TCCS)), this bit will be set to "1" when the interrupt configured by the interrupt mask selection bits is generated.
If no interrupt occurs, this bit will not be set to "1".
In the up/down count mode (MODE:bit21 of the timer state control register (TCCS) is 1), this bit will be set every time a compare clear occurs regardless of the value of the MSI2 to MSI0 bits.
If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit24] ICRE: Compare clear interrupt request enable bit

ICRE	Function
0	Disable interrupt request.
1	Enable interrupt request.

- When this bit and compare clear interrupt flag bit (ICLR:bit25) are set to "1", an interrupt request for CPU will be generated.

[bit23] BFE: Compare clear buffer enable bit

BFE	Function
0	Disable the compare clear buffer.
1	Enable the compare clear buffer.

- This bit is used to enable the compare clear buffer register (CPCLRB).
- When this bit is set to "0":

  Compare clear buffer register (CPCLRB) will be disabled. Thus, you can write to the compare clear register (CPCLR) directly.
- When this bit is set to "1":

  Compare clear buffer register (CPCLRB) will be enabled. Data written to and retained in the compare clear buffer register (CPCLRB) will be transferred to the compare clear register once the count value "0" from the 16-bit free-run timer has been detected.

[bit22] STOP: Timer enable bit

STOP	Function
0	Enable counting (Start the counting).
1	Disable counting (Stop the counting).

- This bit is used to stop/start the counting of the 16-bit free-run timer.
- When this bit is set to "0":

  Start counting the 16-bit free-run timer.
- When this bit is set to "1":

  Stop counting the 16-bit free-run timer.
- The free-run timer will not be initialized even when the SCLR:bit20 of the timer state control register (TCCS) is set to "1" while the free-run timer is inactive (this bit=1).

[bit21] MODE: Timer count mode bit

MODE	Function
0	Up count mode
1	Up/down count mode

- This bit is used to select the count mode of the 16-bit free-run timer.
- When this bit is set to "0":

  The up count mode is selected. The timer continues to count up until the count value matches the compare clear register to be reset to "0x0000". After that, it starts counting up again.
- When this bit is set to "1":

  The up/down count mode is selected. The timer continues to count up until the count value matches the compare clear register. After that, it will be converted to down count. Then, when the count value reaches to "0x0000", it will change to up count once again.
- You can write to this bit whether the timer is active or inactive. If the timer is active, the value written to this bit will be transferred to the buffer. Then, when the timer value becomes "0x0000", the count mode changes based on the buffer value.

[bit20] SCLR: Timer clear bit

SCLR	Function	
	Read	Write
0	Always read out "0".	Do not initialize the counter.
1		Initialize the counter to "0x0000".

- This bit is used to initialize the 16-bit free-run timer to "0x0000".
- Initialization of the 16-bit free-run timer:

  If this bit setting is "1" while the 16-bit free-run timer is active (STOP:bit22=0 in the timer state control register (TCCS)), the 16-bit free-run timer will be initialized to "0x0000" in the next set count clock. The 16-bit free-run timer will not be initialized when this bit is set to "1" while the 16-bit free-run timer is inactive (STOP:bit22 of the timer state control register (TCCS) is 1).
- The value read out is always "0".

Note:

Though writing "1" to this bit initializes the counter, it does not generate the 0 detection interrupt.
If you write "0" to this bit prior to the next count clock after setting "1", the timer clear will not be executed.

[bit19 to bit16] CLK3 to CLK0: Clock frequency selection bits

CLK3	CLK2	CLK1	CLK0	Function					
				Count Clock	φ=40 MHz	φ=20 MHz	φ=10 MHz	φ=5 MHz	φ=2.5 MHz
0	0	0	0	φ	25 ns	50 ns	100 ns	200 ns	400 ns
0	0	0	1	φ/2	50 ns	100 ns	200 ns	400 ns	800 ns
0	0	1	0	φ/4	100 ns	200 ns	400 ns	800 ns	1.6 µs
0	0	1	1	φ/8	200 ns	400 ns	800 ns	1.6 µs	3.2 µs
0	1	0	0	φ/16	400 ns	800 ns	1.6 µs	3.2 µs	6.4 µs
0	1	0	1	φ/32	800 ns	1.6 µs	3.2 µs	6.4 µs	12.8 µs
0	1	1	0	φ/64	1.6 µs	3.2 µs	6.4 µs	12.8 µs	25.6 µs
0	1	1	1	φ/128	3.2 µs	6.4 µs	12.8 µs	25.6 µs	51.2 µs
1	0	0	0	φ/256	6.4 µs	12.8 µs	25.6 µs	51.2 µs	102.4 µs
Other settings disabled				-	-	-	-	-	-

- These bits are used to select the count clock frequency of the 16-bit free-run timer.

Note:

When setting CLK3 to CLK0 bits, confirm that the free-run timer stops firmly.

[bit15 to bit12] Reserved: Reserved bit

The read value of these bits is always "0".

These bits always be written to "0".

[bit11] MODE2: Interrupt mask mode bit 2

MODE2	MODE	Function
0	0	Value set for MSI5 to MSI3 will be invalid.
0	1	Value set for MSI5 to MSI3 will be invalid.
1	0	Setting disabled (operation is not guaranteed.)
1	1	Value set for MSI5 to MSI3 will be valid

- This bit is used to mask the 0 detection interrupt and the compare clear interrupt independently in the up/down count mode (MODE:bit21=1 in the timer state control register (TCCS)) of the 16-bit free-run timer.
- During the MODE:bit21="1" of the timer state control register (TCCS) and if this bit is set to "1", the value configured at MSI5 to MSI3:bit10 to bit8 of this register becomes valid and the compare clear interrupt is masked for the number of times specified. For the mask count of 0 detection interrupt, the value configured at MSI2 to MSI0:bit28 to bit26 of the timer state control register (TCCS) becomes valid.

Note:

During MODE:bit21="0" of the timer state control register (TCCS) and if this bit is set to "1", the operation is not guaranteed.

[bit10 to bit8] MSI5 to MSI3: Compare clear interrupt mask selection bits

MSI5	MSI4	MSI3	Function
0	0	0	An interrupt occurs when there is a match for the 1st time.
0	0	1	An interrupt occurs when there is a match for the 2nd time.
0	1	0	An interrupt occurs when there is a match for the 3rd time.
0	1	1	An interrupt occurs when there is a match for the 4th time.
1	0	0	An interrupt occurs when there is a match for the 5th time.
1	0	1	An interrupt occurs when there is a match for the 6th time.
1	1	0	An interrupt occurs when there is a match for the 7th time.
1	1	1	An interrupt occurs when there is a match for the 8th time.

- These bits are used to set the mask count for the compare clear interrupt. They are valid only when MODE2:bit11=1 in this register and MODE:bit21=1 in the timer state control register (TCCS). MSI2 to MSI0:bit28 to bit26 of the timer state control register (TCCS) are used to set the mask count for the 0 detection interrupt.
- When these bits are set to "000", the compare clear interrupt factor will not be masked.

Note:

The value read is a mask counter value.
If a read-modify-write instruction is executed, the value read is a mask register value.
The written data will be written to the mask register.
The written value to the mask register while the free-run timer is active (STOP:bit22 of the timer state control register (TCCS) is 0) will be reloaded to the counter only when the mask counter becomes "0".
The written value to the mask register while the free-run timer is inactive (STOP:bit22 of the timer state control register (TCCS) is 1) will be immediately reloaded to the counter.

[bit7 to bit0] Reserved: Reserved bit

Always write "1" to these bits.

20.5 Operation

This section explains the operation of the 16-bit free-run timer.

16-bit free-run timer

The 16-bit free-run timer starts counting up from the set value in the timer data register (TCDT) when the count operation is enabled. The count value will be used as base time of the 16-bit input capture.

20.5.1 Interrupt for the 16-bit Free-run Timer

This section explains the interrupt for the 16-bit free-run timer.

Table 20-3 shows the interrupt control bits and interrupt factor of the 16-bit free-run timer.

Table 20-3 Interrupt Control Bits and Interrupt Factor of the 16-bit Free-run Timer

	16-bit Free-run Timer	
	Compare Clear	**0 Detection**
Interrupt Request Flag Bit	Timer state control register (TCCS), ICLR:bit25	Timer state control register (TCCS), IRQZF:bit30
Interrupt Request Enable Bit	Timer state control register (TCCS), ICRE:bit24	Timer state control register (TCCS), IRQZE:bit29
Interrupt Factor	The 16-bit free-run timer value matches the compare clear register (CPCLR).	The 16-bit free-run timer value becomes "0x0000".

When the value of the 16-bit free-run timer matches the compare clear register (CPCLR), ICLR:bit25 of the timer state control register (TCCS) will be set. If interrupt requests are enabled (ICRE:bit24 of TCCS is 1) while this bit is set, an interrupt request is output to the interrupt controller.

When the timer value becomes "0x0000", IRQZF:bit30 of the timer state control register (TCCS) will be set.

If interrupt requests are enabled (IRQZE:bit29 of TCCS is 1) while this bit is set, an interrupt request is output to the interrupt controller.

20.5.2 Operation of the 16-bit Free-run Timer

This section explains the operation of the 16-bit free-run timer.

The 16-bit free-run timer starts counting up from the value configured at the timer data register (TCDT) after reset. The count value will be used as base time of the 16-bit input capture.

20.5.2.1 Timer Clear

This section explains timer clear.

The count value of the 16-bit free-run timer will be cleared in any of the followings.

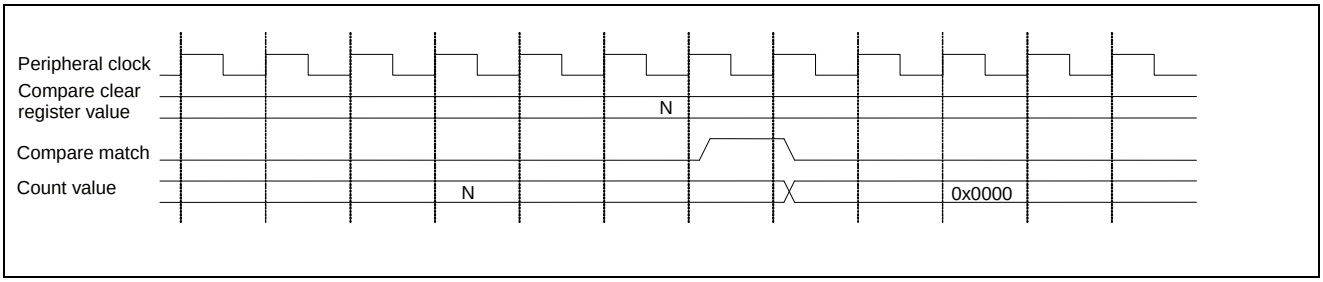
- When a match with the compare clear register by the up count mode (MODE:bit21 of TCCS register is 0) is detected.
- When "1" is written to SCLR:bit20 of the TCCS register while it is active.
- When "0x0000" is written to the TCDT register while it is inactive.
- When it has been reset.

The counter will be cleared as soon as it has been reset. In the case of a timer clear bit(SCLR:bit20) of timer state control register(TCCS) is written to "1", or when there is a match with the compare clear register, the counter will be cleared synchronously with the count timing.

Note:

Even when "1" is written to the SCLR:bit20 of the TCCS register while it is inactive, the count value of the 16-bit free-run timer will not be cleared.
If "0x0000" is written in TCDT register during the up/down count mode (MODE:bit21=1 of timer state control register (TCCS)), an unintended counting may be performed.
See Section "20.4.1.2.Timer Data Register: TCDT0" for the setting procedure of TCDT register during the up/down count mode (MODE:bit21=1 of timer state control register (TCCS)).

Figure 20-2. Clear Timing of the 16-bit Free-run Timer



20.5.2.2 Timer Mode

This section explains timer mode.

You can select either of the following modes for the 16-bit free-run timer.

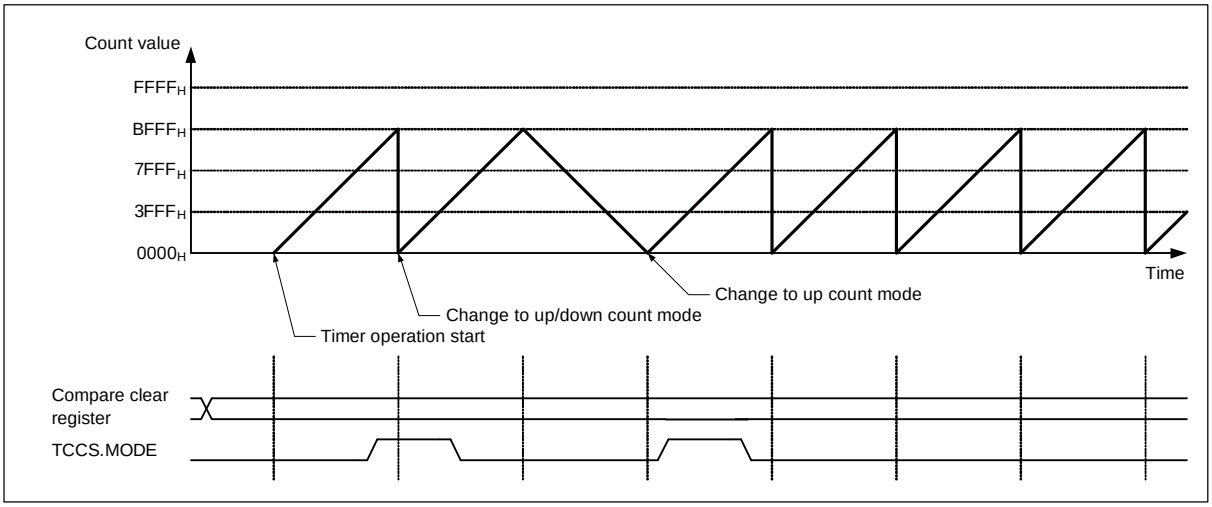
- Up count mode (MODE:bit21 of the TCCS register is 0)
- Up/down count mode (MODE:bit21 of the TCCS register is 1)

In the up count mode, the counter starts counting from the timer data register (TCDT) configured in advance. It continues to count up until the count value matches the value of the compare clear register (CPCLR). The counter will be cleared to "0x0000" and start counting up again.

In the up/down count mode, the counter starts counting from the timer data register (TCDT) configured in advance. It continues to count up until the count value matches the value of the compare clear register (CPCLR). Then, the counter changes counting method from up count to down count. The counter continues to count down until the counter value reaches "0x0000" and starts counting up again.

You will be able to write a value to the mode bit (MODE:bit21 of the TCCS register) whether the timer is active or inactive. If the timer is active, the value written to this bit will be transferred to the buffer. Then, when the timer value becomes "0x0000", the count mode changes.

Figure 20-3. Changing the Timer Mode While the Timer Is Active



20.5.2.3 Compare Clear Buffer

This section explains compare clear buffer.

The compare clear register (CPCLR) has a buffer function that can be enabled or disabled. When the buffer function is enabled (BFE:bit23 of the TCCS register is 1), data written to the compare clear buffer register (CPCLRB) will be transferred to the CPCLR register once the 16-bit free-run timer value "0" has been detected. When the buffer function is disabled (BFE:bit23 of the TCCS register is 0), you will be able to write data to the CPCLR register directly.

Figure 20-4. Operation in the Up Count Mode when the Compare Clear Buffer Is Disabled (BFE:bit23 of the TCCS register is 0)

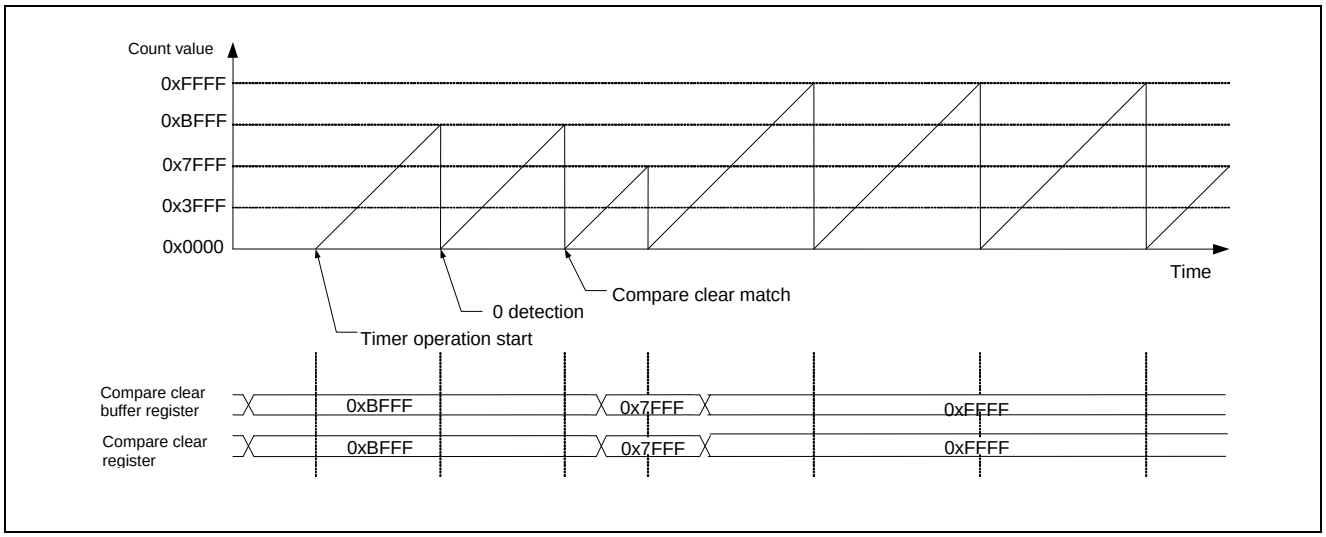


Figure 20-5. Operation in the Up Count Mode When the Compare Clear Buffer Is Enabled (BFE:bit23 of the TCCS Register Is 1)

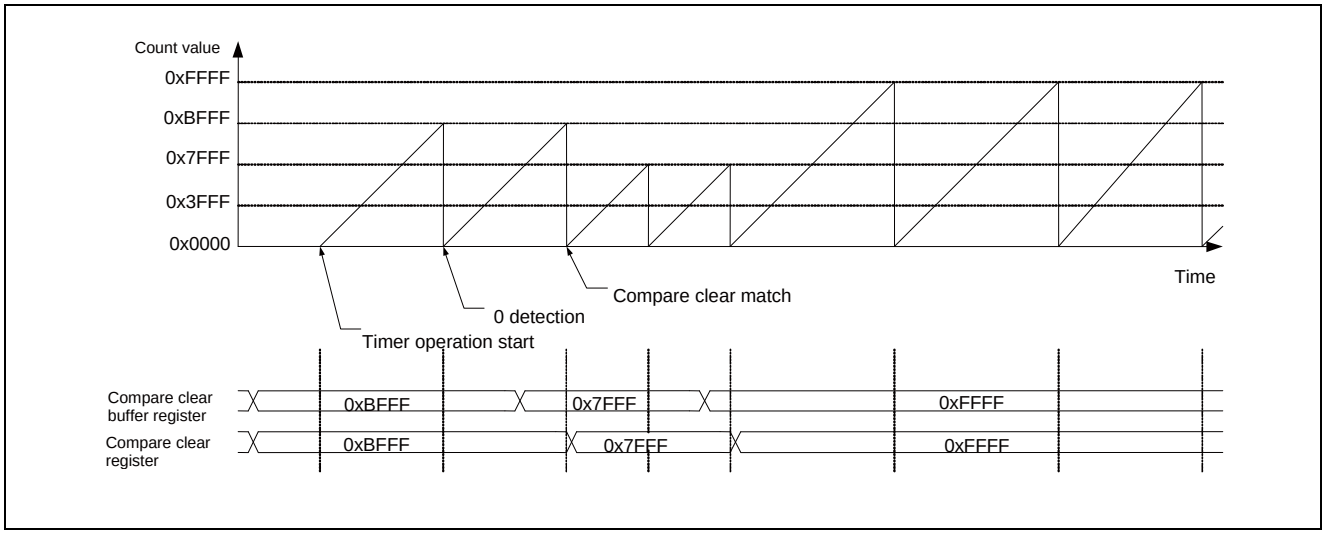
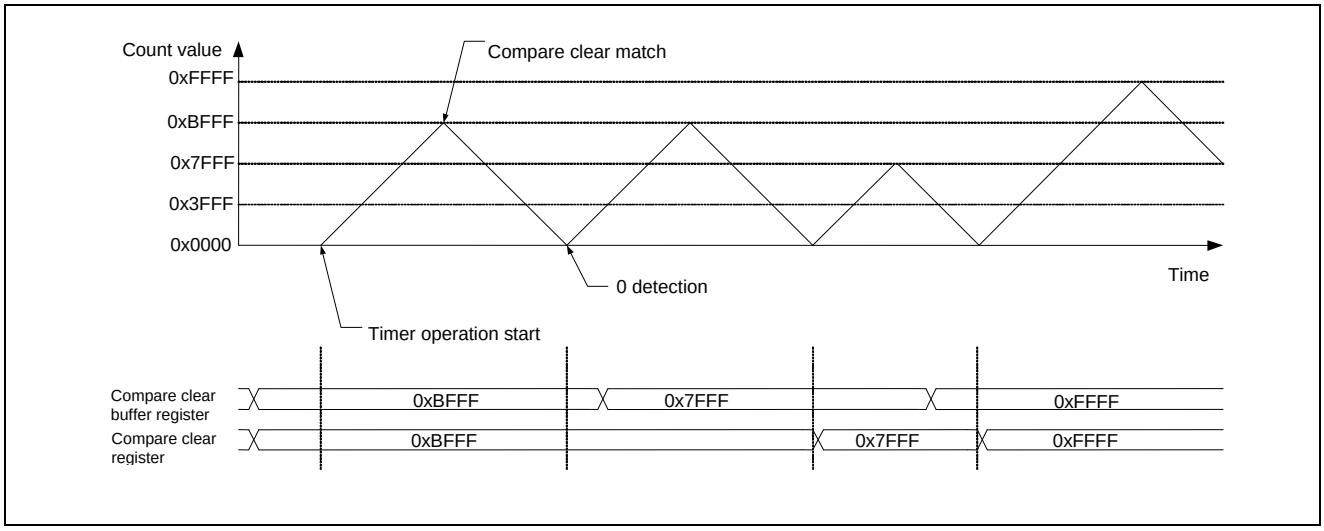


Figure 20-6. Operation in the Up/Down Count Mode When the Compare Clear Buffer Is Enabled (BFE:bit23 of the TCCS Register is 1)



20.5.2.4 Timer Interrupt

This section explains timer interrupt.

The 16-bit free-run timer can generate the following 2 types of interrupts.

- Compare clear interrupt
- 0 detection interrupt

The compare clear interrupt will be generated when the timer value matches the value of the compare clear register.

The 0 detection interrupt will be generated when the timer value reaches "0x0000".

Note:

Software clear (SCLR:bit20 of the TCCS register is 1) does not generate the 0 detection interrupt.

Figure 20-7. Interrupt Generated in the Up Count Mode (MODE:bit21 of the TCCS Register Is 0)

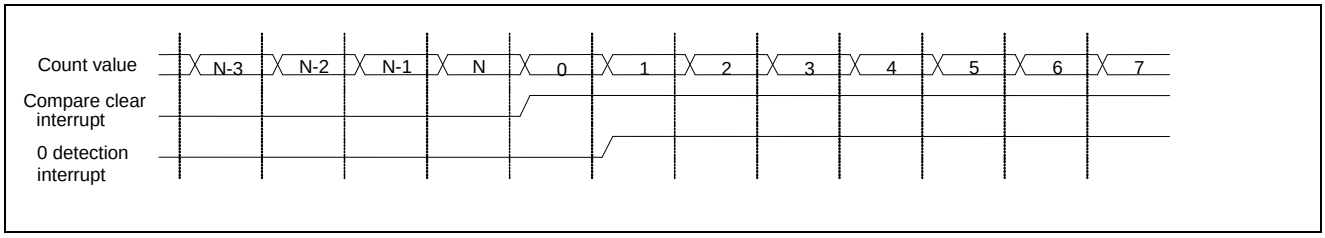
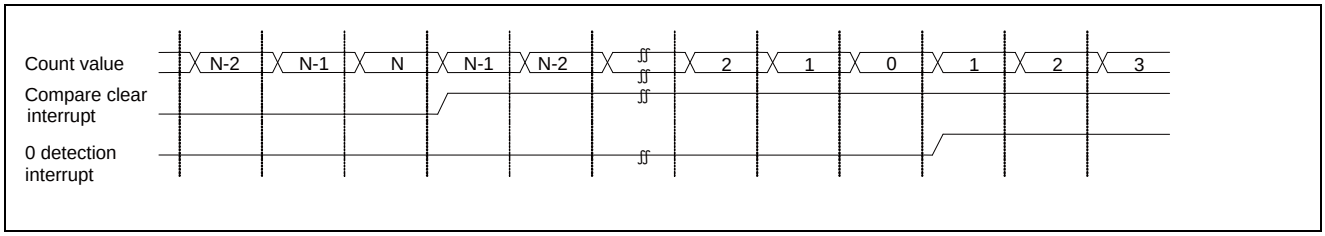


Figure 20-8. Interrupt Generated in the Up/Down Count Mode (MODE:bit21 of the TCCS Register is 1)



20.5.2.5 Interrupt Mask Function

This section explains the interrupt mask function.

You can mask either the 0 detection interrupt or the compare clear interrupt. Alternatively, you can mask both of them. The following explains how to mask either one of the interrupts.

- You will be able to mask the interrupt request by setting the MSI2 to MSI0:bit28 to bit26 of the TCCS register. MSI2 to MSI0 bits are 3-bit reload down register that reloads the value once the count value reaches "000". You can also load the count value by writing the value to the MSI2 to MSI0 bits directly. Mask count is the value configured at MSI2 to MSI0. When the MSI2 to MSI0 bits become "000", the interrupt request will not be masked.
- The interrupt request varies depending on the count mode (MODE:bit21 of the TCCS register). In the up count mode, you will be able to mask the compare clear interrupts only while the 0 detection interrupts will be generated every time "0" is detected. In the up/down count mode, you will be able to mask the 0 detection interrupts only.

The following explains how to mask both types of interrupt requests.

- Both types of interrupts can be masked only when the free-run timer is in the up/down count mode. To do so, set MODE2=1 in the TCCS register and MODE=1 in the TCCS register.

The MSI2 to MSI0 bits of the TCCS register are used to mask the "0" detection interrupt. The MSI5 to MSI3 bits of the TCCS register are used to mask the compare clear interrupt.

Note:

Software clear (SCLR:bit20 of the TCCS register is 1) does not generate the 0 detection interrupt.

Figure 20-9. Compare Clear Interrupt Masked in the Up Count Mode

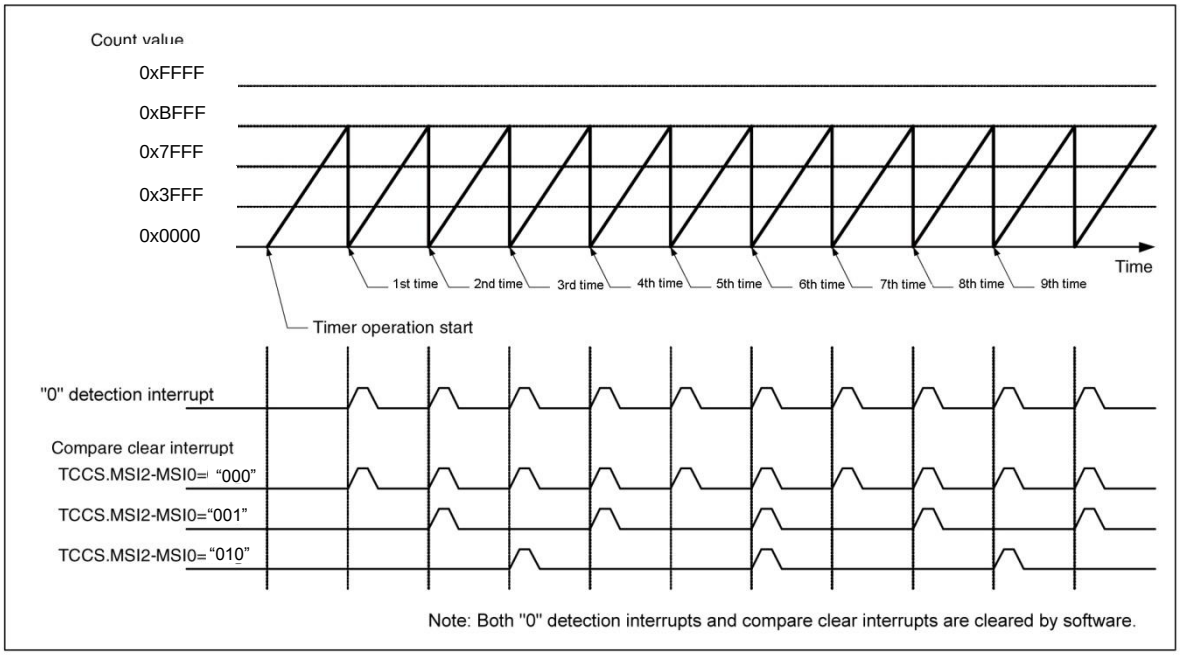
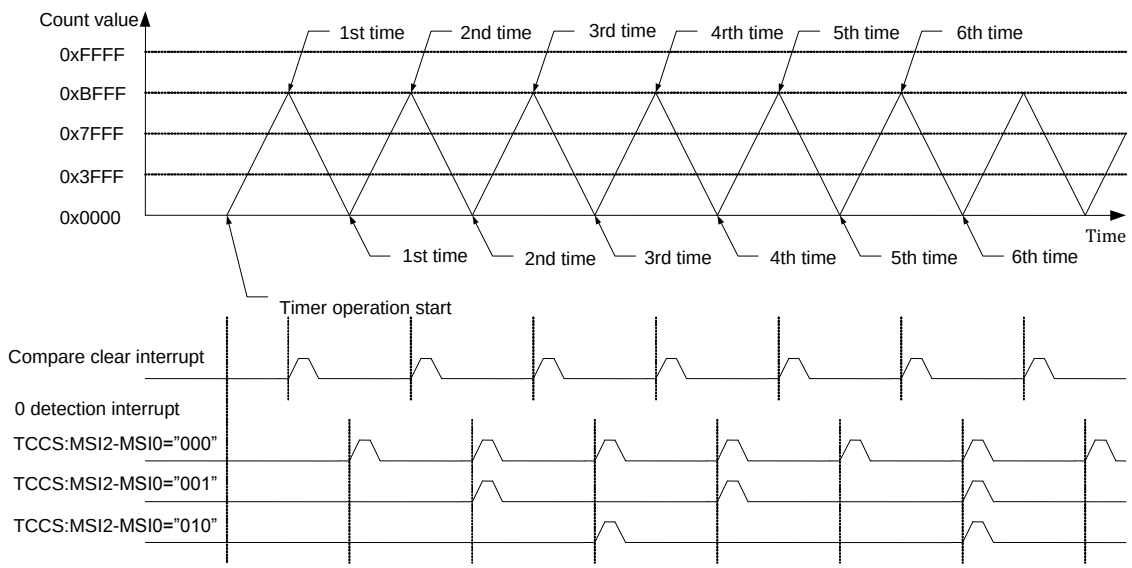
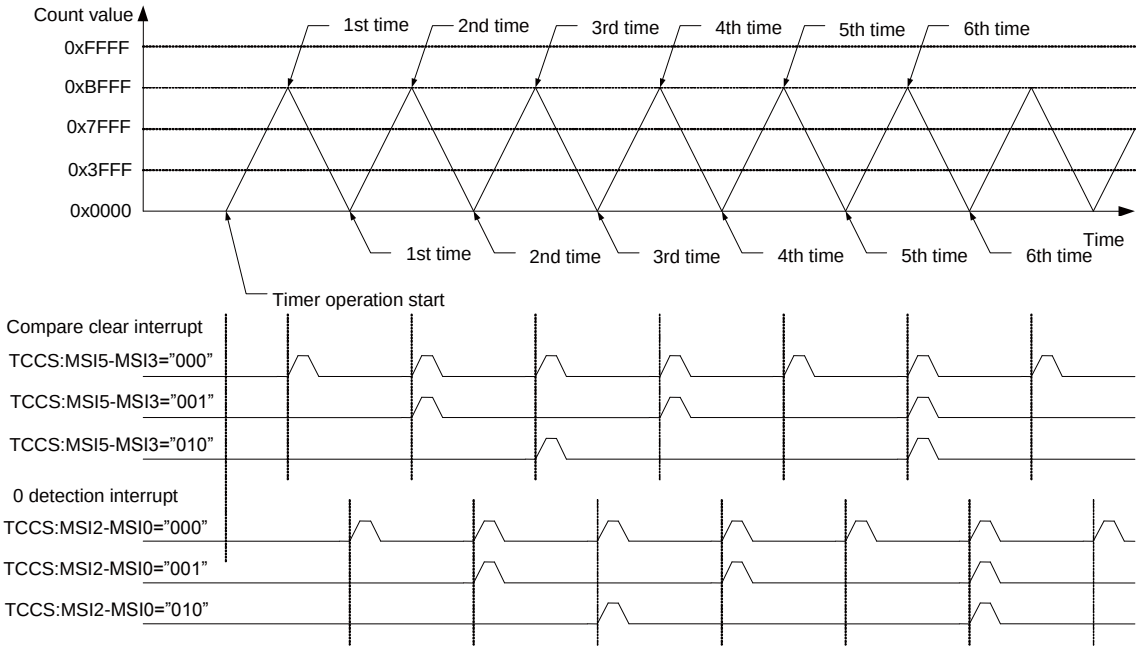


Figure 20-10. 0 Detection Interrupt Masked in the Up/Down Count Mode



Note: Both 0 detection interrupts and compare clear interrupts are cleared by software.

Figure 20-11. 0 Detection Interrupt and Compare Clear Interrupt Masked in the Up/Down Count Mode



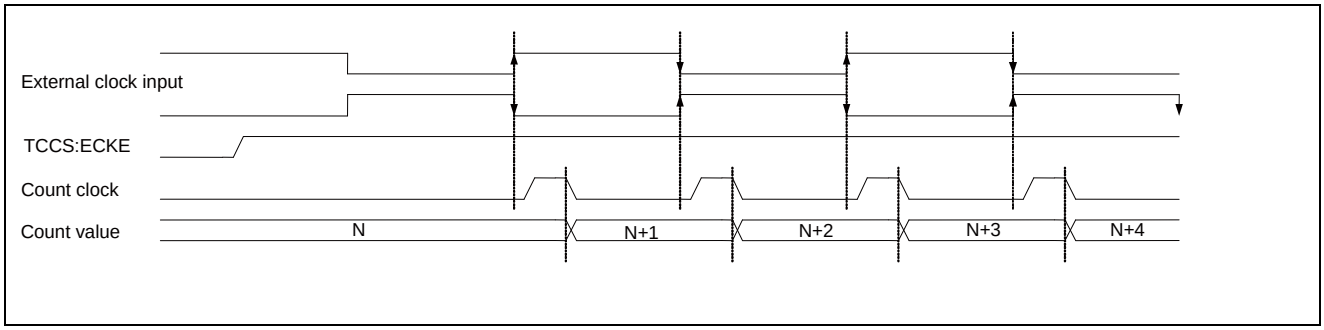
Note: Both 0 detection interrupts and compare clear interrupts are cleared by software.

20.5.2.6 Selected External Count Clock

This section explains the selected external count clock.

The 16-bit free-run timer will be incremented based on the input clock (peripheral clock or external clock). With an external clock selected, the 16-bit free-run timer starts counting up at a rising edge when the initial value of external clock input is "1" after the external clock mode is selected (ECKE:bit31=1 in the TCCS register). After that, the timer counts up at both edges. When the initial value of external clock input is "0", the timer starts counting up by a falling edge. After that, the timer counts up at both edges.

Figure 20-12. Count Timing of the 16-bit Free-run Timer



Note:

Counting of the external clock input uses both edges of the external clock.

20.5.3 Notes on Operating Specifications

This section explains notes on operating specifications.

20.5.3.1 Notes at Accessing the Buffer Registers

This section explains notes to observe when accessing the buffer registers.

The CPCLR register in the free-run timer has a buffer function. Do not use a read-modify-write instruction when accessing this register.

20.5.3.2 Notes on Using the 16-bit Free-run Timer

This section explains the notes on using the 16-bit free-run timer.

Notes on setting by a program

- When you execute reset, the timer value becomes "0x0000", however, the 0 detection interrupt flag will not be configured.
- Since the timer mode bit (MODE of the TCCS register) has a buffer, the timer mode changed after the 0 detection becomes valid.
- Software clear (SCLR of the TCCS register is 1) initializes the timer, but it does not generate the 0 detection interrupt.
- When you start counting while the compare value and count value match, the compare clear flag will not be configured.
- Set the value other than "0x0000" for the compare value. When setting the value, consider that the following operation will happen.
  - ☐ When the timer mode bit (MODE in TCCS register) is in the up count mode, the timer value is updated to "0x0000" and then is fixed to "0x0000". The 0 detection interrupt flag and the compare clear flag continue to be set every count clock.
  - ☐ When the timer mode bit (MODE in TCCS register) is in the up/down count mode, the timer value repeats the up count operation from "0x0000" to "$FFFF_H$". The "0" detection interrupt flag and the compare clear flag are set when the timer value and "0x0000" match.

Notes on accessing the TCCS register

- For the read-modify-write instruction, setting value will be read out from the MSI2 to MSI0/MSI5 to MSI3.
- For the normal reading mode, the counter value will be read out from the MSI2 to MSI0/MSI5 to MSI3.

21. 16-bit Input Capture

This chapter explains the 16-bit input capture.

21.1 Overview

21.2 Features

21.3 Configuration

21.4 Registers

21.5 Operation

21.1 Overview

This section explains the overview of the 16-bit input capture.

This product includes one 16-bit input capture channel.

21.2 Features

This section explains features of the 16-bit input capture.

Functions of 16-bit Input Capture

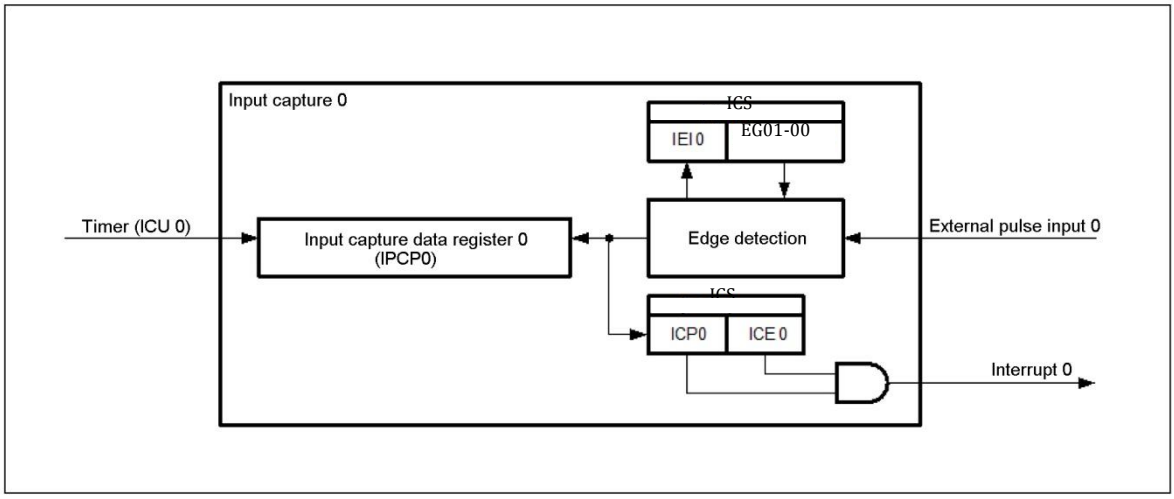
- The 16-bit input capture consists of 1 external input pin, capture registers corresponding to this pin, and capture control registers. When an edge of the input signal from the external pin is detected, the value of the 16-bit free-run timer can be stored in the capture register and an interrupt is generated simultaneously.
- 3 types of trigger edge (rising edge, falling edge, and both edges) of the external input signal can be selected and there is a register that indicates whether the trigger edge is rising or falling.
- An interrupt is generated when a valid edge from the external input signal is detected.
- Any desired free-run timer channel can be set for each compare unit.
- There is one input capture channel, input capture 0, for which free-run timer 0 can be selected as the input.

21.3 Configuration

This section explains the configuration of the 16-bit input capture.

Configuration of 16-bit Input Capture

Figure 21-1. Configuration of Input Capture



21.4 Registers

This section explains registers of the 16-bit input capture.

Table of External Pins

Table 21-1. Table of 16-bit Input Capture External Pins

Channels	External Pins (External Pulse Input)
0	ICU

List of 16-bit Input Capture Registers

Table 21-2. List of 16-bit Input Capture Registers

Address	+0	+1	+2	+3
0x0000127C	Input capture data register 0 (IPCP0)		Reserved	
0x00001280	Input capture state control register 01 (ICS)		Reserved	LIN SYNCH FIELD switching register (LSYNS)

21.4.1 16-bit Input Capture Registers

This section explains registers of the 16-bit input capture.

The 16-bit input capture consists of input capture data registers and input capture state control registers.

21.4.1.1 Input Capture Data Register: IPCP0

This section explains registers of the 16-bit input capture.

An input capture data register (IPCP) retains the count value of the free-run timer at the time of detection of an effective edge of the input waveform.

IPCP0: Address 127C$_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15 to bit0] CP15 to CP00: Free-run timer value

CP15 to CP00	**Function**
	Free-run timer value

- This register is used to store a free-run timer value at the time of detection of an effective edge of the corresponding external pin input waveform.
- The free-run timer value in the above explanation represents the operating state of a free-run timer for which the input capture has been selected.

Note:

When accessing this register, use a 16bit or 32bit access instruction. No data can be written to this register.

21.4.1.2 Input Capture State Control Register: ICS

The bit configuration for the input capture state control register is shown below.

An input capture state control register (ICS) is used to select an edge, enable interrupt request, and control an interrupt request flag. It is also used to indicate an effective edge detected by the input capture.

ICS: Address 1280H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						Reserved	IEI0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ICP0	Reserved	ICE0	Reserved	Reserved	EG01	EG00
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R(RM1),W	R0,W0	R/W	R0,W0	R0,W0	R/W	R/W

[bit15 to bit10] Reserved bits

Always write 0 to these bits.

[bit9] Reserved bit

"0" is always read. Writing does not affect the operation.

[bit8] IEI0: Effective edge indication bit

IEI0	Function
0	A falling edge is detected.
1	A rising edge is detected.

- This effective edge indication bit for the capture register (IPCP) indicates that a rising or falling edge has been detected.
- When a falling edge is detected, this bit is set to "0".
- When a rising edge is detected, this bit is set to "1".
- This bit is read-only.

Note:

If EG01, EG00:bit1, bit0 of the input capture state control register (ICS) are set to 00B, the value read from this register is meaningless.

[bit7] Reserved bit

Always write 0 to this bit.

[bit6] ICP0: Interrupt request flag bit

ICP0	Function	
	Read	Write
0	No effective edge is detected.	This bit is cleared.
1	An effective edge is detected.	This bit remains unaffected.

- This bit is used as an interrupt request flag for the input capture.
- This bit is immediately set to "1" when an effective edge from the external input pin is detected.
- An interrupt is immediately generated when an effective edge is detected while the interrupt request enable bit (ICE0:bit4) is set.
- When this bit is writing "0": This bit is cleared.
- When this bit is writing "1": This bit remains unaffected.
- If a read-modify-write (RMW) instruction is executed, "1" is always read.

Notes:

If a software clear (write of "0") or a clear due to an interrupt clear signal ("H") and a hardware set occur at the same time, the hardware set takes precedence.

[bit5] Reserved bit

Always write 0 to this bit.

[bit4] ICE0: Interrupt request enable bit

ICE0	Function
0	Interrupt request disabled
1	Interrupt request enabled

- This bit is used to enable an input capture interrupt request for the input capture.
- An input capture interrupt is generated when an interrupt request flag bit (ICP0:bit6) is set while this bit is set to "1".

[bit3, bit2] Reserved bits

Always write 0 to this bit.

[bit1, bit0] EG01, EG00: Edge selection bits

EG01	EG00	Function
0	0	No edge is detected (Stopped).
0	1	A rising edge is detected.
1	0	A falling edge is detected.
1	1	Both edges are detected.

- These bits are used to specify an effective edge polarity of the external input for the input capture.
- These bits are also used to enable the operation of input capture.

21.4.1.3 LIN SYNCH FIELD Switching Register: LSYNS

The bit configuration of the LIN SYNCH FIELD switching register is shown below.

The LIN SYNCH FIELD switching register (LSYNS) is used for LIN linkage control.

LSYNS: Address 1283_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					LSYN2	LSYN1	LSYN0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W

[bit7 to bit3] Reserved

Always write 0 to these bits.

[bit2 to bit0] LSYN2 to LSYN0: Input capture0 input selection

LSYN2	LSYN1	LSYN0	Function
0	0	0	External pin input (ICU)
0	0	1	Input Lin Synch Field detection signal from the multi-function serial interface ch.0.
0	1	0	Input Lin Synch Field detection signal from the multi-function serial interface ch.1.
0	1	1	Input Lin Synch Field detection signal from the multi-function serial interface ch.2.
Others			Settings are inhibited. (No operations are guaranteed.)

- These bits are used to enable Lin Synch Field from the multi-function serial interface ch.2 to ch.0.

21.5 Operation

This section explains the operation.

21.5.1 Interrupts for 16-bit Input Capture

21.5.2 Operation of 16-bit Input Capture

21.5.3 Notes on Using the 16-bit Input Capture

21.5.1 Interrupts for 16-bit Input Capture

This section explains the interrupts for 16-bit input capture.

Table 21-3 shows the interrupt control bits and interrupt factor of the 16-bit input capture.

Table 21-3. Interrupt Control Bits and Interrupt Factor of 16-bit Input Capture

	16-bit input capture
Interrupt request flag bit	Input capture state control register (ICS) ICP0:bit6
Interrupt request enable bit	Input capture state control register (ICS) ICE0:bit4
Interrupt factor	An effective edge is detected at the IN pin.

With 16-bit input capture, when an effective edge is detected at a pin, the input capture state control register (ICS) ICP0:bit6 is set to "1". If interrupt requests are enabled (ICE0:bit4 of ICS is 1) with this state, an interrupt request is output to the interrupt controller.

21.5.2 Operation of 16-bit Input Capture

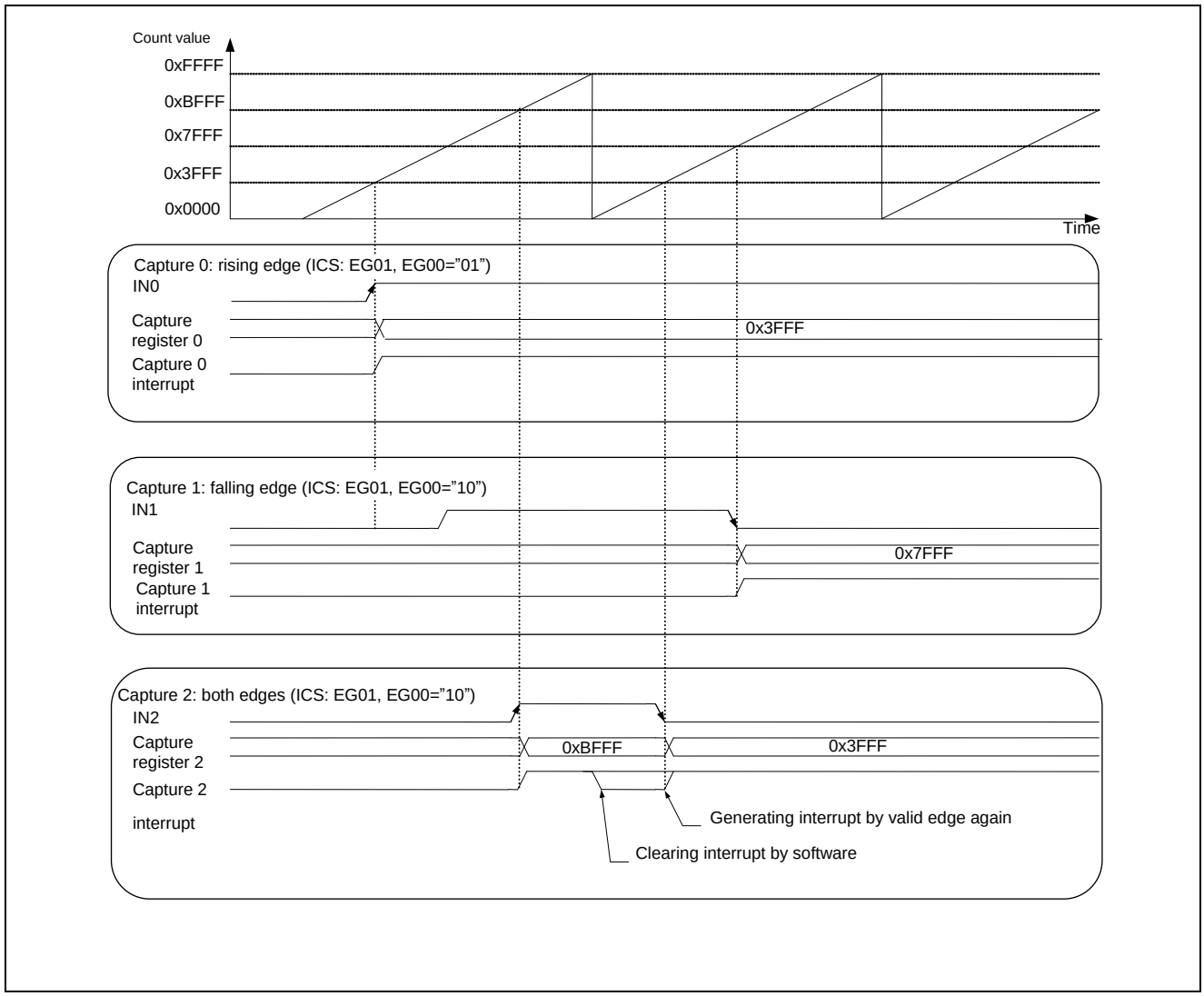
The operation of 16-bit input capture is shown below.

Input capture is used to detect a specified effective edge. When an effective edge is detected, an interrupt flag is set and the value of the 16-bit free-run timer is loaded to the capture register.

21.5.2.1 Operation of 16-bit Input Capture

The operation of 16-bit input capture is shown below.

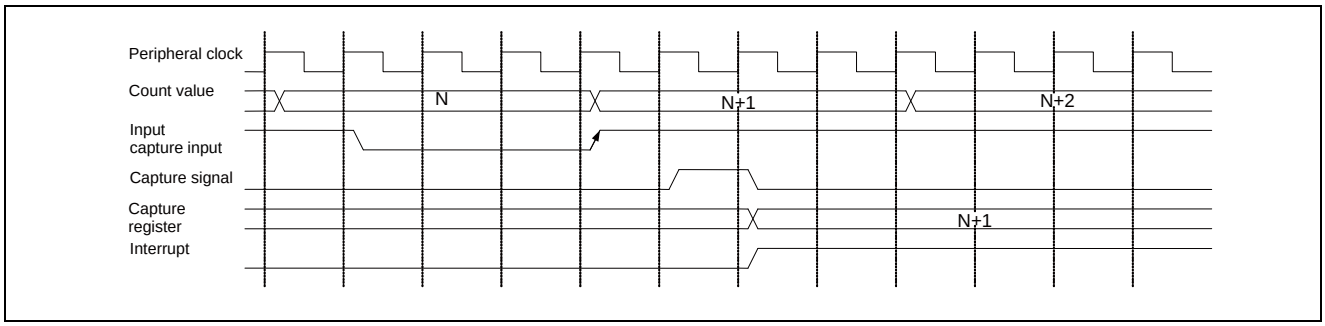
Figure 21-2. Example of Input Capture Timing



21.5.2.2 16-bit Input Capture Input Timing

The operation of 16-bit input capture input timing is shown below.

Figure 21-3. Example of 16-bit Input Capture Timing for Input Signals



21.5.3 Notes on Using the 16-bit Input Capture

This section explains the notes on using the 16-bit input capture.

If the input capture pin (IN) level is changed during the period from the bit setting of ICP0 of the input capture state control register (ICS) to the processing of an interrupt routine, the ICP0 effective edge indication bit (IEI0 of ICS register) indicates the latest edge detected.

Input capture register

Reading from the input capture register must be performed in 16-bit or 32-bit access.

Read-modify-write

When reading is performed using a read-modify-write instruction, ICP0 of the input capture state control register (ICS) is read as "1".

Note on interrupts

Before the input capture state control register (ICS) interrupt request enable bit (ICE0) is set to "1", be sure to clear the interrupt flag (ICP0).

22. WDT1 Calibration

This chapter explains the WDT1 calibration.

22.1 Overview

22.2 Features

22.3 Configuration

22.4 Registers

22.5 Operation

22.1 Overview

This section gives an overview of the WDT1 calibration.

This module calculates values to calibrate the frequency of the mounted CR oscillation circuit for WDT1.

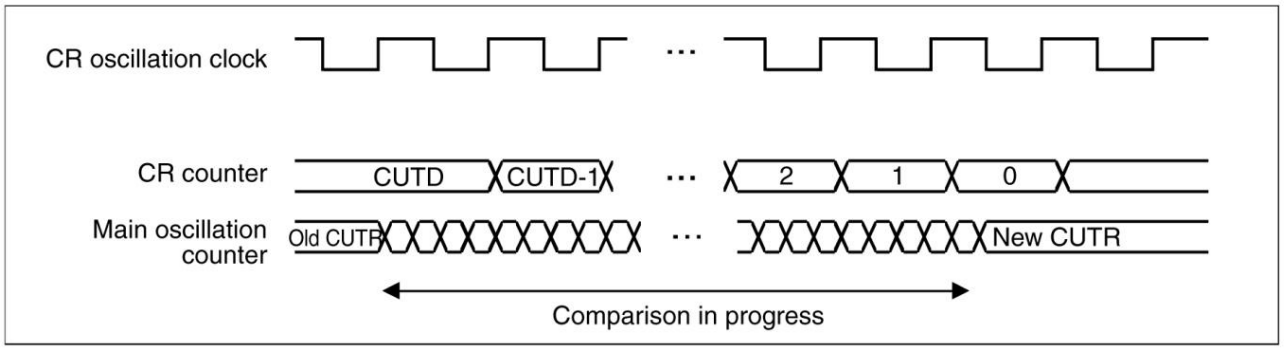
22.2 Features

This section explains features of the WDT1 calibration.

WDT1 (CR clock) calibration

This device has no CR clock calibration function. CR clock errors, however, can be measured by using the register of this module.

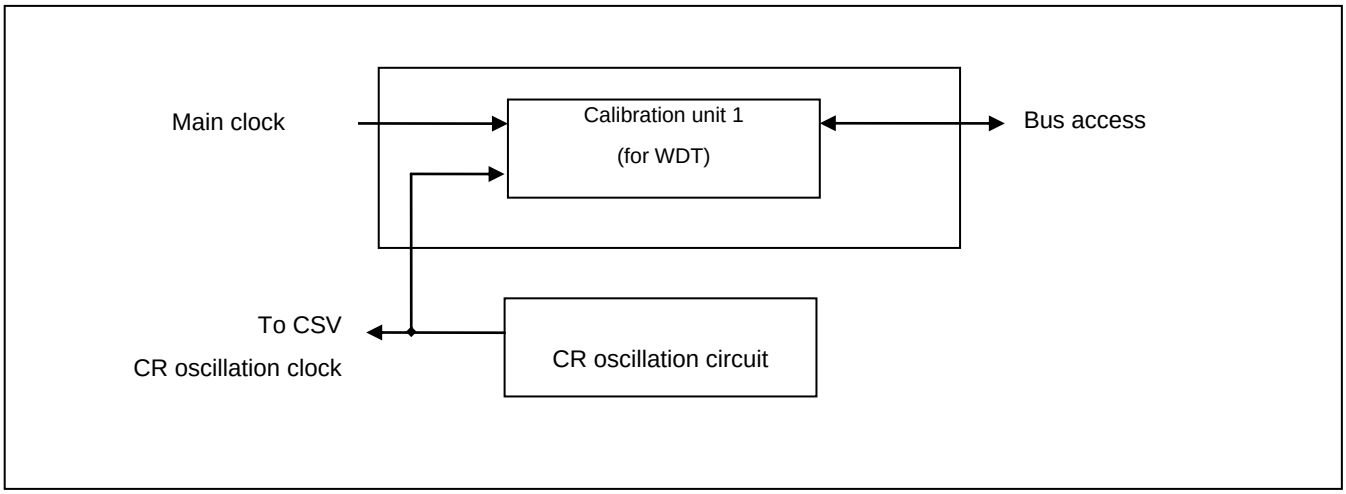
Figure 22-1. Comparison of Counters Driven by Different Clocks



22.3 Configuration

This section explains configuration of the WDT1 calibration.

Figure 22-2. Block Diagram



22.4 Registers

This section explains the registers of the WDT1 calibration.

Table 22-1. Register Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x04C0	Reserved	Reserved	Reserved	Reserved	Reserved
0x4C4	CUCR1		CUTD1		Calibration unit control register 1 CR oscillation timer data register
0x4C8	CUTR1				Main oscillation timer data register 1

22.4.1 Calibration Unit Control Register 1: CUCR1 (Calibration Unit Control Register 1)

The bit configuration of the calibration unit control register 1 is shown.

This register configures calibration start and interrupts for WDT calibration unit.

CUCR1: Address 04C4$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved		STRT	Reserved		INT	INTEN
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,WX	R0,WX	R,W	R0,WX	R0,WX	R,W0	R/W

[bit15 to bit8] Reserved

The read value of these bits is always "1". Writing to these bits does not influence other functions.

[bit7] Reserved

Be sure to write "0" to this bit.

[bit6, bit5, bit3, bit2] Reserved

The read value of these bits is always "0". Writing to these bits does not influence other functions.

[bit4] STRT (calibration STaRT): Calibration start

This bit starts counters driven by the main clock and CR clock. The INT bit will be set after the comparison is completed.

STRT	Function
"0" write	Stops comparison
"1" write	Starts comparison

Setting "0" to this bit stops comparison. While comparing, writing "1" to this bit will not take effect. This bit will be cleared to "0" after the comparison is completed.

[bit1] INT (calibration INTerrupt): Interrupt

The INT bit will be set to "1" after the comparison is completed. If the INTEN bit is set, an interrupt will occur. This bit is cleared by writing "0".

[bit0] INTEN (calibration INTerrupt Enable): Interrupt enable

This bit sets whether to generate an interrupt when the INT bit is set.

INTEN	Interrupt
0	Disabled
1	Enabled

22.4.2 CR Clock Timer Data Register: CUTD1 (Calibration Unit Timer Data register 1)

The bit configuration of the CR clock timer data register is shown.

This register configures a period of the time during which the CR clock driven counter operates.

CUTD1: Address $04C6_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TDD[15:8]							
Initial value	1	1	0	0	0	0	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TDD[7:0]							
Initial value	0	1	0	1	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] TDD[15:0] (Timer Data Data field): Timer data

These bits configure the comparison time interval in the number of the CR clock pulses.

22.4.3 Main Oscillation Timer Result Register 1: CUTR1 (Calibration Unit Timer Result register 1)

The bit configuration of the main oscillation timer result register 1 is shown.

This register indicates the number of the main clock pulses counted within the time interval set by CUTD1.

CUTR1: Address $04C8_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	TDR[23:16]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TDR[15:8]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TDR[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit24] Reserved

The reading value of these bits is always "0". Writing to these bits does not influence other functions.

[bit23 to bit0] TDR[23:0] (Timer Data Register): Timer data

These bits indicate the number of counts counted in the comparison interval. Read the results after the comparison is completed. The read value during comparison is undefined. Writing has no effect on operation.

22.5 Operation

This section explains operation.

22.5.1 Measurement of Errors in CR Clock

22.5.2 Note

22.5.1 Measurement of Errors in CR Clock

This section shows measurement of errors in the CR clock.

The procedure for measuring errors in the CR clock is as follows:

1. Setting CUTD1
2. Setting CUCR1.INTEN
3. Setting CUCR1.STRT
4. Loop waiting for interrupt
5. Occurrence of interrupt
6. Reading CUTR1
7. Comparison of CUTR1 and CUTD1 can be used to calculate the ratio between the main clock frequency and the CR clock frequency.

22.5.2 Note

This section gives a note.

The counter value will become invalid in such a case that transition to standby mode occurs. Write "0" to the STRT bit to stop, and then write "1" again to redo.

The following must be satisfied: $T_{OSC100} > 2 \times T_{OSC4} + 3 \times T_{CLKP}$.

T_{OSC4}: Main clock cycle
T_{OSC100}: CR oscillation circuit oscillation cycle
T_{CLKP}: Peripheral clock oscillation cycle

23. Power Consumption Control

This chapter explains the power consumption control.

23.1 Overview
23.2 Features
23.3 Configuration
23.4 Registers
23.5 Operation
23.6 Usage Example

23.1 Overview

This section gives an overview of the power consumption control.

This device has a variety of low-power consumption modes and can perform the power consumption control according to situations.

23.2 Features

This section explains features of the power consumption control.

Clock control

- Clock division
  By changing the division ratio for each running clock, the operating frequency can be lowered accordingly. See "Chapter: Clock."

Sleep mode

- CPU sleep mode
  In this mode, the only CPU core stops operating.
- Bus sleep mode
  In this mode, both the CPU core and on-chip buses stop operating.

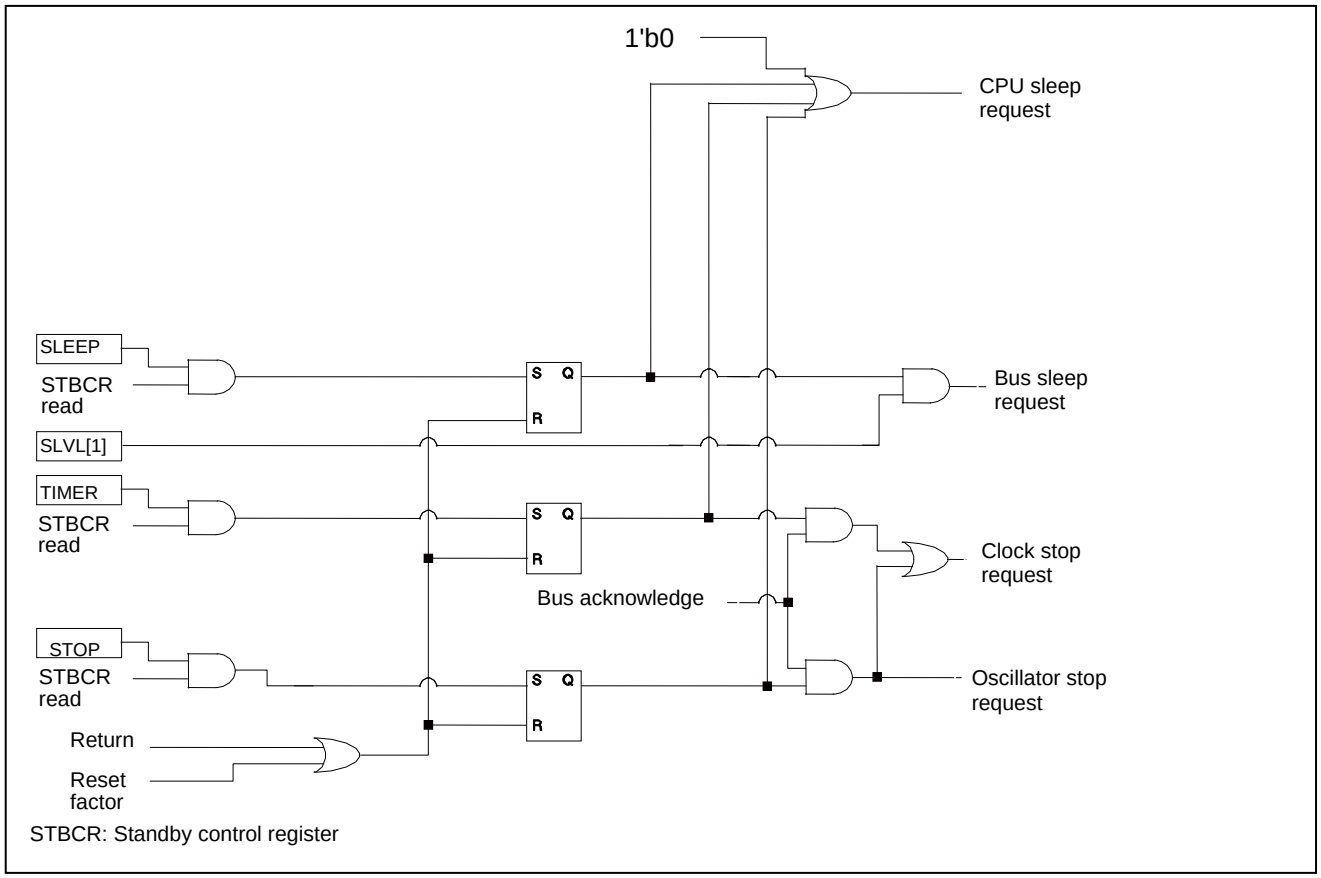
Standby mode

- Watch mode
  In this mode, all operations except some clock oscillations and the timer stop.
- Stop mode
  In this mode, all clock oscillations and operations stop.

23.3 Configuration

This section shows the configuration of the power consumption control.

Figure 23-1. Control Block Diagram



23.4 Registers

This section shows the registers of the power consumption control.

Table 23-1. Register Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0480	Reserved	Reserved	STBCR	Reserved	Standby control register

Note:

The addresses 0x0480 to 0x0481 are allocated for the register "Reset". (See "Chapter: Reset.")

23.4.1 Standby Control Register: STBCR (STandBy mode Control Register)

The bit configurations of the standby control register are shown below.

This register configures low-power consumption modes.

STBCR: Address 0482_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STOP	TIMER	SLEEP	Reserved	Reserved		SLVL[1:0]	
Initial value	0	0	0	0	0	0	1	1
Attribute	R,W	R,W	R,W	R0,W0	R0,W0	R0,W0	R/W	R/W

Note:

Writing to this register by DMA is prohibited.

[bit7] STOP (STOP mode): Stop mode enable

[bit6] TIMER (TIMER mode): Watch mode enable

[bit5] SLEEP (SLEEP mode): Sleep mode enable

Transitions to each standby mode (stop, watch, and sleep) are specified and enabled by these 3 bits. After writing the values shown below to these 3 bits and reading STBCR, the CPU goes into each standby mode.

STOP	TIMER	SLEEP	Enabled transition to each standby mode
0	0	0	No transition (initial value)
0	0	1	Transition to sleep mode by reading STBCR
0	1	X	Transition to watch mode by reading STBCR
1	X	X	Transition to stop mode by reading STBCR

The read value of each bit is as follows regardless of the writing value:

STOP	TIMER	SLEEP	Enabled transition to each standby mode
0	0	0	No transition
0	0	1	Transition to sleep mode
0	1	0	Transition to watch mode
1	0	0	Transition to stop mode

These bits are returned to their initial values by wake up factors arising from each low-power consumption mode.

[bit4] Reserved

The read value is always "0". Be sure to write "0" to this bit.

[bit3, bit2] Reserved

The read value is always "0". Be sure to write "0" to these bits.

[bit1, bit0] SLVL[1:0] (Standby LeVeL): Standby level setting

These bits control the operations in standby mode and sleep mode as shown below.

Mode	SLVL[1:0]	Operation control
Stop mode	0x	Does not make pins high impedance.
	1x	Makes pins high impedance.
Watch mode	0x	Does not make pins high impedance.
	1x	Makes pins high impedance.
Sleep mode	0x	CPU sleep mode (stop only CPU)
	1x	Bus sleep mode (stop CPU and on-chip bus) *

* On-chip bus will run only when DMA transfer is in progress.

For information on the pins with high impedance, see "Appendix."

23.5 Operation

Operations of the power consumption control are explained.

Features of the power consumption control of the device are explained in the following sections.

23.5.1 Clock Control

This section shows the clock control of the power consumption control.

By adjusting each operating clock of the device, its power consumption and processing capability can be optimized.

23.5.1.1 Division Setting

This section shows division setting of the clock.

See "Chapter: Clock."

23.5.2 List of Clocks Supplied in Low-power Consumption Mode

The list of clocks supplied in low-power consumption mode is shown below.

Table 23-2. List of Clocks Supplied in Low-power Consumption Mode

Clock	Standby		Sleep	
	Stop	Watch	Bus	CPU
CPU clock (CCLK)	○	○	○	×
CAN prescaler clock	○	○	*1	×
On-chip bus clock (HCLK)	○	○	○	×
Peripheral clock (PCLK)	○	○	×	×
PLL clock (PLLCLK)	○	○	×	×
Main clock (MCLK)	○	×	×	×
CR oscillation	○ *3	○ *3	× *2	× *2

○: Stops

×: Does not stop. (If the main clock/PLL clock are stopped by each clock setting register, supply of each clock stops, accordingly.)

*1 When on-chip bus clock (HCLK) is selected as CAN prescaler clock, this clock stops. When PLL clock is selected, whether CAN prescaler stops or not depends on PLL output. Otherwise, CAN prescaler clock does not stop.

*2 During sleep mode, the CR oscillation does not stop, but the watchdog timer 1 (HWWDT) stops.

*3 In order to stop the CR oscillation in standby mode, a setting is needed in advance. See the description of CSVCR.RCE in "Chapter: Clock Supervisor."

23.5.3 Sleep Mode

This section describes sleep mode.

Sleep mode is the mode in which CPU and on-chip bus are stopped and only the peripherals run. In sleep mode, there are the following modes according to the difference in the range of functional blocks to be stopped.

- CPU sleep mode: Only CPU is stopped.
- Bus sleep mode: Both CPU and on-chip bus are stopped.

The stop state continues until a wake up request occurs. It is possible to return to programmed operation within a few clock times by generating a wake up request.

Operations of each mode are explained in the following sections.

23.5.3.1 CPU Sleep Mode

This section describes CPU sleep mode.

CPU sleep mode is the mode to stop the CPU operating.

In this mode, the DMA controller and on-chip bus can continue operating, but more power will be consumed than that in bus sleep mode.

23.5.3.2 Bus Sleep Mode

This section describes bus sleep mode.

Bus sleep mode is the mode to stop CPU and on-chip bus operations. In this mode, the CPU clock (CCLK) and on-chip bus clock (HCLK) will stop.

When accepting a DMA transfer request in bus sleep mode, on-chip bus clock (HCLK) supply resumes temporarily and performs DMA transfers. After the DMA transfer, it stops the on-chip bus clock (HCLK) again.

In this mode, you can decrease the amount of power consumption more than that of CPU sleep mode, but the response time to the DMA transfer request will be somewhat degraded.

23.5.3.3 Configuration of Sleep Mode

The configuration of sleep mode is described below.

Before activating sleep mode, select the level of sleep mode with the values set to bit1:SLVL1 in the STBCR register.

- When setting bit1:SLVL1 = "0" in the STBCR register, CPU goes into CPU sleep mode.
- When setting bit1:SLVL1 = "1" in the STBCR register, CPU goes into bus sleep mode.

23.5.3.4 Activation of Sleep Mode

Activation of sleep mode is described below.

To activate sleep mode, follow the steps below.

- Write "001" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read STBCR.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering sleep mode.

[Example] Sample program of sleep mode activation

```
LDI     #value_of_sleep, R0     ; SLEEP bit = "1", SLVL setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; Write
LDUB    @R12, R0                ; Read (activation of sleep mode)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy processing for pipeline adjustment
```

23.5.3.5 Wake Up from the Sleep Mode

Wake up from the sleep mode is described below.

The sleep mode is terminated under the following conditions:

- Reset
- Generation of interrupt request whose value of corresponding ICR register is value other than "0x1F"
- Generation of NMI request
- Generation of tool break while connected to ICE

For the wake up caused by an interrupt request, the CPU does not necessarily have to be set so as to accept this interrupt request. When an interrupt request is not accepted, the program starts from the instruction next to the instruction which activated the sleep mode.

In the bus sleep mode, the on-chip bus clock (HCLK) is temporarily returned by generating the DMA transfer request and DMA transfer is performed. After the DMA transfer is ended, the on-chip bus clock (HCLK) is stopped again.

23.5.3.6 Effect of Sleep Mode

Effect of sleep mode is described below.

You can reduce power consumption on the peripheral or external input event wait state drastically by using sleep mode. This mode does not decrease power consumption as much as that of in watch mode or stop mode because the peripheral clock (PCLK) will continue to run. On the other hand, a return to the program operation within several clock times is possible by generating a wake up request.

23.5.4 Standby Mode: Watch Mode

This section describes standby mode: watch mode.

Watch mode is the mode to continue oscillation only for the specific clock and count the clock timer corresponding to that clock.

Notes:

- Enter the device into the standby mode only when main RUN is in progress. For the operation at a transition from the PLL-run state to its standby mode, see "23.5.7 Transition to Illegal Standby Mode."
- Transition to the standby mode while the FLASH memory is being programmed/erased is prohibited.

23.5.4.1 Configuration of Watch Mode

The configuration of watch mode is described below.

Before activating watch mode, set the state of external pins in watch mode with the bit1:SLVL1 in the STBCR register.

- When setting bit1:SLVL1 = "0" in the STBCR register, the external pins hold previous state.
- When setting bit1:SLVL1 = "1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. See "Appendix."

23.5.4.2 Activation of Watch Mode

Activation of watch mode is described below.

To activate watch mode, follow the steps below.

- When performing PLL RUN, CPU must go into main RUN state first.
- Write "010" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read the STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering watch mode.

[Example] Sample program of watch mode activation

```
LDI     #value_of_timer, R0     ; TIMER bit = "1", SLVL setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; Write
LDUB    @R12, R0                ; Read (activation of watch mode)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy processing for pipeline adjustment
```

23.5.4.3 Wake Up from the Watch Mode

Wake up from the watch mode is described below.

The watch mode is terminated under the following conditions:

- Reset
- Generation of interrupt request whose value of corresponding ICR register is value other than "0x1F" (see "Chapter: Interrupt Control (Interrupt Controller)")
- Generation of NMI request
- Generation of tool break while connected to ICE

For the wake up caused by an interrupt request, the CPU does not necessarily have to be set so as to accept this interrupt request. When an interrupt request is not accepted, the program continues to run from the instruction next to the instruction which activated the watch mode.

23.5.4.4 Effect of Watch Mode

The effect of watch mode is described below.

You can reduce power consumption on the external input event wait state drastically by using watch mode. This mode does not decrease power consumption as much as that of in stop mode because enabled clock oscillation will continue to run. On the other hand, a clock timer can continue to run and a return to the program operation is possible by generating a wake up request in a short time compared with the return from the stop mode.

23.5.5 Standby Mode: Stop Mode

This section describes standby mode: stop mode.

Stop mode is the mode to stop all clock oscillations and minimize power consumption of this device.

Notes:

- Enter the device into the standby mode only when main RUN is in progress. For the operation at a transition from the PLL-run state to its standby mode, see "23.5.7 Transition to Illegal Standby Mode."
- Transition to the standby mode while the FLASH memory is being programmed/erased is prohibited.

23.5.5.1 Configuration of Stop Mode

The configuration of stop mode is described below.

Before activating stop mode, set the state of external pins in stop mode with the bit1:SLVL1 in the STBCR register.

- When setting bit1:SLVL1 = "0" in the STBCR register, the external pins hold previous state.
- When setting bit1:SLVL1 = "1" in the STBCR register, the external pins become high impedance.

Pins whose state is controlled differ according to product types. See "Appendix."

23.5.5.2 Activation of Stop Mode

Activation of stop mode is described below.

To activate stop mode, follow the steps below.

- When performing PLL RUN, CPU must go into main RUN state first.
- Write "100" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
- Read the STBCR register.

In FR81S core, if the read value will not be used in the next instruction, that instruction is executed before the read is completed. Perform dummy processing to use the read value in the next instruction so as not to make the program progress before entering stop mode.

[Example] Sample program of stop mode activation

```
LDI     #value_of_stop, R0      ; STOP bit = "1", SLVL setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; Write
LDUB    @R12, R0                ; Read (activation of stop mode)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy processing for pipeline adjustment
```

23.5.5.3 Wake Up from the Stop Mode

Wake up from the stop mode is described below.

The stop mode is terminated under the following conditions:

- Reset
- Generation of interrupt request in which the value of corresponding ICR register is other than "0x1F" (see "Chapter: Interrupt Control (Interrupt Controller)")
- Generation of NMI request
- Generation of tool break while being connected to ICE

For the wake up caused by an interrupt request, the CPU does not necessarily have to be set so as to accept this interrupt request. When an interrupt request is not accepted, the program continues to run from the instruction next to the instruction which activated the stop mode.

23.5.5.4 Effect of Stop Mode

The effect of stop mode is described below.

You can minimize power consumption on the external input event wait state by using stop mode. On the other hand, a return to the program operation after generating a wake up request needs the oscillation stabilization wait time.

23.5.6 Stop State of Microcontroller

The stop state of the microcontroller is described below.

When the transition from the state of the standby mode (watch mode/stop mode) transition prohibition to the standby is controlled, the standby transition is not concluded.

<State of standby transition prohibition>

1. Connecting OCD
2. Operating PLL

<Standby control not done by microcontroller stop condition>

3. Flash memory power saving control
4. Oscillation stop (At the stop mode)

However, the oscillation stop operation is done detecting the illegal standby mode transition when the standby mode transition control is done while PLL is operating. For the illegal standby mode transition, see "23.5.7 Transition to Illegal Standby Mode."

23.5.7 Transition to Illegal Standby Mode

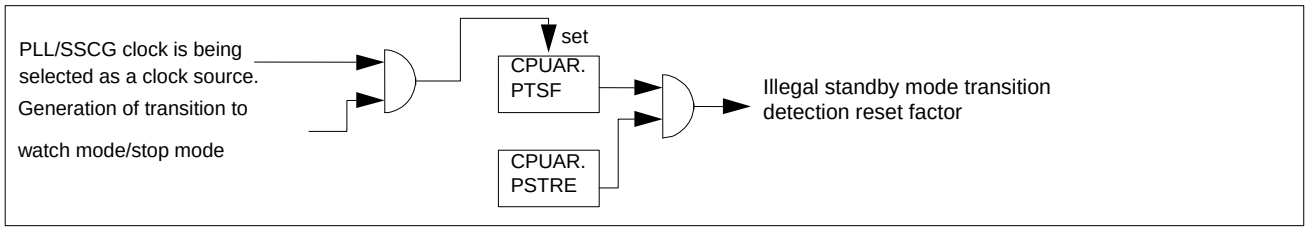
Transition to illegal standby mode is described below.

If the transition from PLL run state to standby mode (watch mode/stop mode) is made, standby mode is set and PLL oscillation stabilization is canceled. (Transition to illegal standby mode)

After returning from standby mode, CSELR.CKS[1:0] = 00 and CMONR.CKM[1:0] = 00 (divide-by-two output of the main clock).

The PSTF flag of the CPUAR register is set concurrently with the transition to standby mode. When the PSTRE bit in the CPUAR register is set, reset occurs by illegal standby mode transition detection reset factor. For the CPUAR register, see "CPU Abnormal Operation Register: CPUAR (CPU Abnormal operation Register)" in "Chapter: Reset."

Figure 23-2. Generation Diagram of Illegal Standby Mode Transition Detection Reset Factor

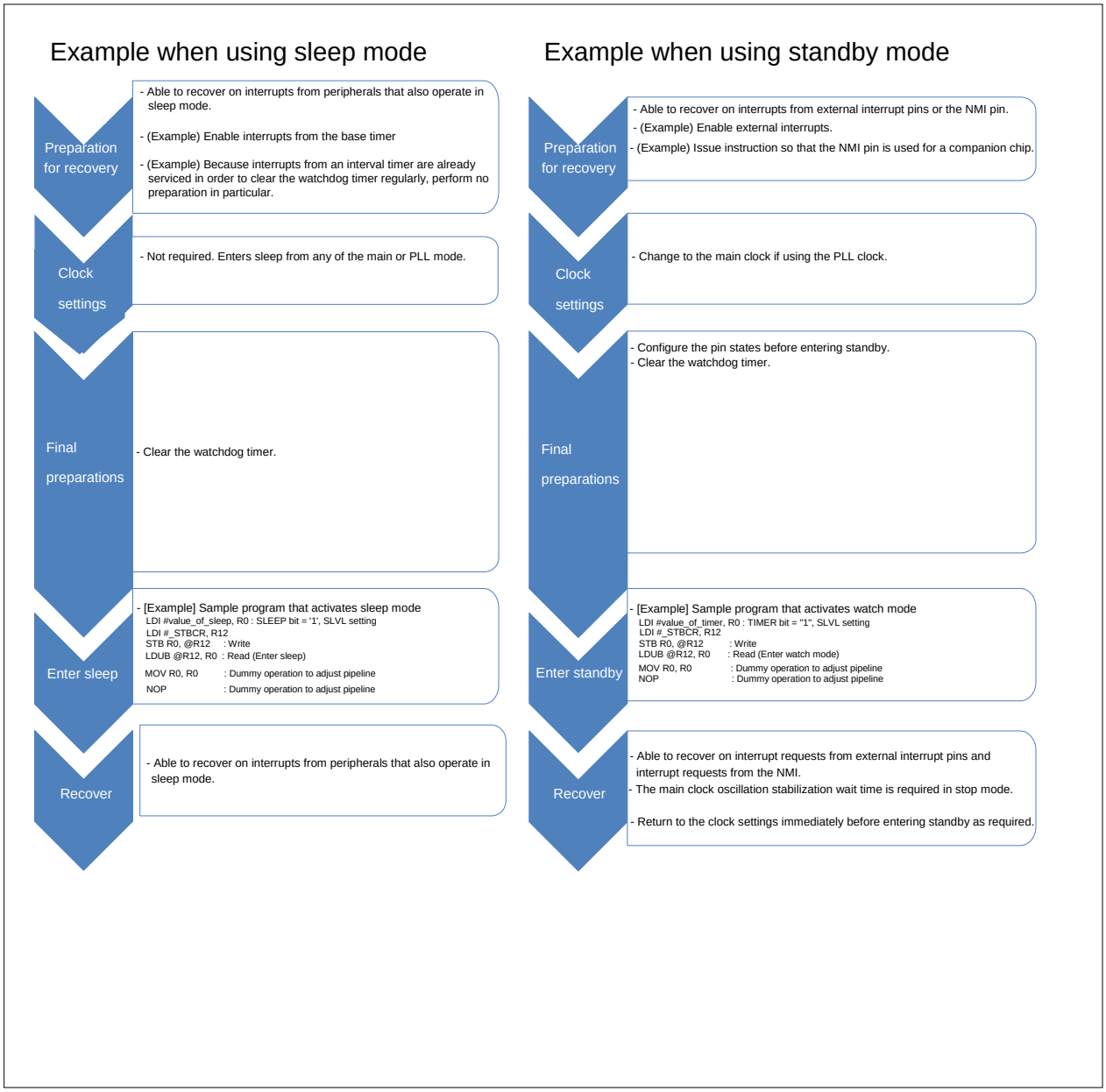


23.6 Usage Example

Power consumption control usage examples are shown below.

These are examples of activation of sleep mode and standby mode.

Figure 23-3. Examples of Activation of Sleep Mode and Standby Mode



24. Low-Voltage Detection (Internal Low-Voltage Detection)

This chapter explains the low-voltage detection (internal low-voltage detection).

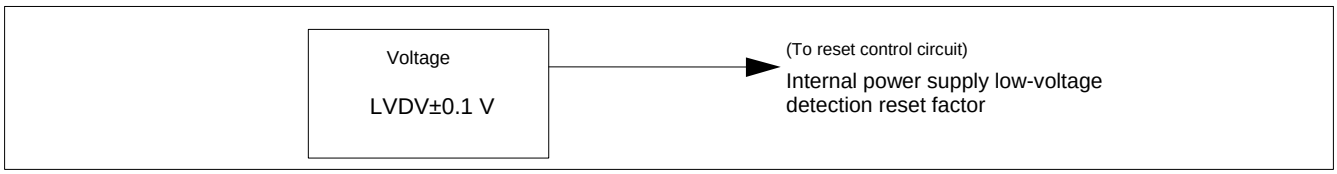
24.1 Overview
24.2 Features
24.3 Configuration
24.4 Register
24.5 Operation
24.6 Notes

24.1 Overview

This section gives an overview of the low-voltage detection (internal low-voltage detection).

The internal low-voltage detection is the function that monitors an internal power supply voltage and detects a fall of the power supply voltage below the low-voltage detection voltage level. When the internal low-voltage below the detection voltage level is detected, a detection flag is set and the device goes to the reset state by the low-voltage detection reset.

Figure 24-1. Block Diagram (Overview)



24.2 Features

This section explains features of the low-voltage detection (internal low-voltage detection).

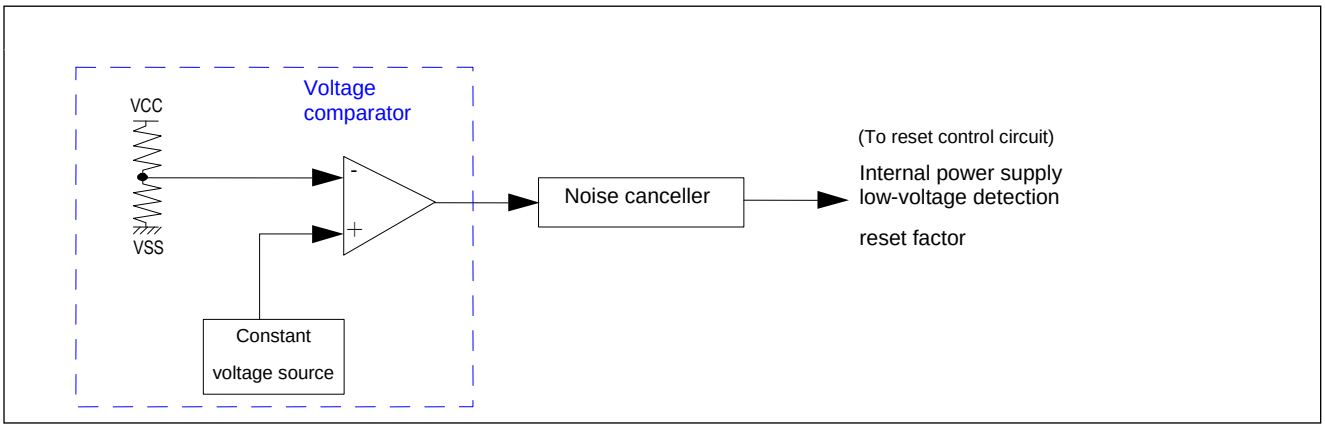
The internal low-voltage detection circuit

Function	: Generates a reset signal to initialize settings if a voltage LVDV ± 0.1V or less is detected. (LVDV : 0.9 V)
Number of units	: 1
Operation	: Continues to operate in sleep mode, stop mode, and watch mode.
Voltage comparator	: Compares the internal power supply voltage to the detection voltage level, and changes output from "H" to "L" if a low-voltage is detected. After the power is turned on the voltage comparator operates constantly.

24.3 Configuration

This section shows the configuration of the low-voltage detection (internal low-voltage detection).

Figure 24-2. Configuration Diagram



24.4 Register

This section shows the registers of the low-voltage detection (internal low-voltage detection).

Table 24-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0584	LVD5R	LVD5F	LVD	Reserved	Internal low-voltage detection register

24.4.1 Internal Low-Voltage Detection Register: LVD (Low-Voltage Detect internal power fall register)

The bit configuration of the internal low-voltage detection register is shown.

This register has the internal low-voltage detection flag (LVD_F) and the control bit.

LVD: Address 0586_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LVD_PD	LVD_SEL[2:0]			LVD_OE	Reserved		LVD_F
Initial Value	0	1	0	0	0	0	0	0
Attribute	R/W	R/W1	R/W0	R/W0	R/W	R0,WX	R0,WX	R(RM1),W

[bit7] LVD_PD (Low Voltage Detect fall Power Down)

This bit sets whether a fall of the internal power supply voltage in the microcontroller should be detected or not.

LVD_PD	Setting for Detection of Internal Power Supply Voltage Fall Power down in the Microcontroller
0	Disabled (Detection is executed.)
1	Enabled (Detection is stopped.)

(Note): This bit is initialized by only power-on reset.

Note:

Set detection enable (OE = 0) after 100 µs, if this bit sets the status of power-down enable to disable (operation start). If the detection enable is set before 100 µs, some detection flag setting can occur.

[bit6 to bit4] LVD_SEL[2:0] (Low Voltage Detect power fall SELect)

These bits select the detection level of a fall of the internal power supply voltage.

LVD_SEL[2:0]	Setting for Detection Level of Fall of Internal Power Supply Voltage
100	0.9V±0.1V
Other than those above	Setting is prohibited

(Note): These bits can be rewritten only when LVD_OE="1".

[bit3] LVD_OE (Low Voltage Detect power fall Output Enable)

This bit is the output enable signal for internal voltage fall detection.

LVD_OE	Internal Voltage Fall Detection Output Enable Setting
0	Enable
1	Disable

(Note): This bit is initialized by only power-on reset.

[bit2, bit1] Reserved

[bit0] LVD_F (Low Voltage Detect power fall Flag)

This bit indicates an internal power supply voltage fall detection flag.

LVD_F	Internal Power Supply Fall Detection Flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a fall in the internal power supply voltage is detected, the LVD_F bit is set to "1".
It will be initialized only when the external reset is input.

24.5 Operation

This section explains operations of the low-voltage detection (internal low-voltage detection).

24.5.1 Internal Low-voltage Detection

24.5.1 Internal Low-voltage Detection

The internal low-voltage detection is explained.

The internal low-voltage detection is the function that monitors an internal power supply voltage, detects it falling below the detection voltage level and sets the detection flag. When the internal low-voltage below the detection level is detected, a detection flag is set and a reset signal to initialize setting is generated.

If the internal voltage falls below the detection voltage level, it takes the oscillation stabilization wait time after the internal voltage is recovered. For details, see "Chapter: Reset."

Oscillation Stabilization Wait Time	2^{15} x Main clock cycle

24.6 Notes

This section provides notes on the low-voltage detection (internal low-voltage detection).

Operation of internal low-voltage detection

If the internal power supply voltage falls and the internal low-voltage detection flag in the microcontroller is set (LVD:LVD_F="1"), internal reset is generated by the function of low-voltage detection reset. Thus, writing and reading of the internal low-voltage detection register (LVD) in the microcontroller is not allowed. The internal low-voltage detection circuit can operate even though the device is in its sleep mode, stop mode, and watch mode, consuming a certain amount of current. The internal low-voltage detection circuit can be set to operate/stop by a user.

Initial value of internal low-voltage detection flag (LVD:LVD_F)

The internal low-voltage detection flag is set to "1" immediately after power-on. The internal low-voltage detection flag is cleared by external reset or by writing "0" to the LVD_F bit of the internal low-voltage detection register (LVD).

Oscillation stabilization wait time

If the internal voltage falls below the detection voltage level, it takes the oscillation stabilization wait time after the internal voltage recovers. For details, see "Chapter: Reset."

Hysteresis of detection/reset release voltage

Since the detection voltage and reset release voltage exhibit hysteresis of 0.1V, the reset release voltage becomes the set detection voltage + 0.1V. For example, when LVD: 0.9V ± 0.1V is set, the reset release voltage becomes 1.0V ± 0.1V.

25. Low-Voltage Detection (External Low-Voltage Detection)



This chapter explains the low-voltage detection (external low-voltage detection).

25.1 Overview

25.2 Features

25.3 Configuration

25.4 Registers

25.5 Operation

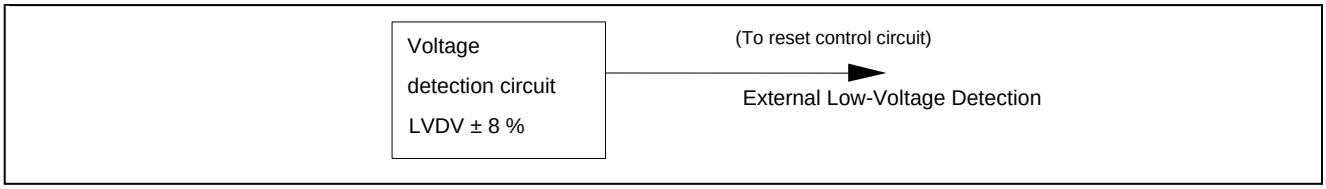
25.6 Notes

25.1 Overview

This section gives an overview of the low-voltage detection (external low-voltage detection).

The external low-voltage detection is the function that monitors external voltage and detects a fall of the power supply voltage below the low-voltage detection voltage level.

Figure 25-1. Block Diagram



Note: Rising LVDV: 2.3 V
Falling LVDV: 3.7 to 4.3 V (7 steps) variable

25.2 Features

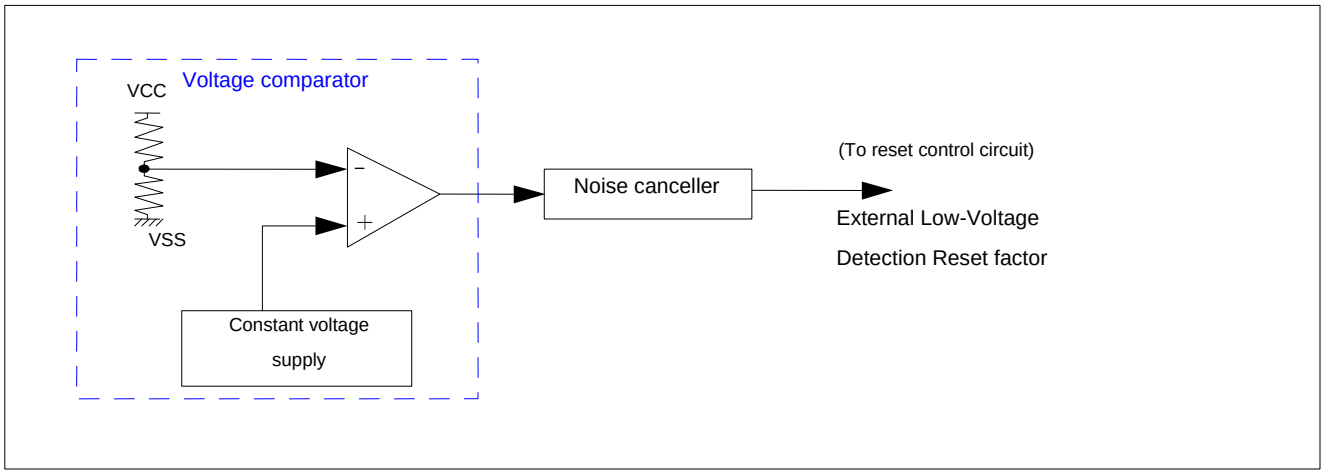
This section explains features of the low-voltage detection (external low-voltage detection).

- Function: Generates the reset signal to initialize settings if the voltage LVDV ±8% or less is detected.
  (Rising: LVDV: 2.3 V (fixed), falling LVDV: 3.7 to 4.3 V (variable))
- Number of units: One
- Operation: Switches operation/stop by user's settings.
  During writes to the internal RAM, the low-voltage reset occurs after the write has finished.
- Voltage comparator: Compares the detection voltage and the power supply voltage, outputting "L" if low-voltage is detected.
- Either to apply a reset or to generate an interrupt, when a low-voltage is detected, can be selected.

25.3 Configuration

This section explains the configuration of the low-voltage detection (external low-voltage detection).

Figure 25-2. Configuration Diagram



25.4 Registers

This section explains the registers of the low-voltage detection (external low-voltage detection).

Table 25-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0584	LVD5R	LVD5F	LVD	Reserved	External low-voltage detection rise detection register External low-voltage detection fall detection register

25.4.1 External Low-Voltage Detection Rise Detection Register: LVD5R (Low Voltage Detect external 5v Rise register)

The bit configuration of the external low-voltage detection rise detection register (LVD5R) is shown.

This register is the external power supply voltage rise detection flag.

LVD5R: Address 0584_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							LVD5R_F
Initial Value	0	0	0	0	0	0	0	1
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1),W

[bit7 to bit1] Reserved

[bit0] LVD5R_F (Low Voltage Detect external 5v Rise Flag): External voltage rise detection flag

This bit is an external voltage rise detection flag.

LVD5R_F	External Power Supply Rise Detection Flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a rise in external voltage is detected, the LVD5R_F bit is set to "1".
The bit will be cleared when external reset is input.

25.4.2 External Low-Voltage Detection Fall Detection Register: LVD5F (Low-Voltage Detect external 5v Fall register)

The bit configuration of the external low-voltage detection fall detection register (LVD5F) is shown.

This register is used in order to clear the low-voltage detection reset flag and set the low-voltage detection circuit

LVD5F: Address 0585_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LVD5F_PD	LVD5F_SEL[3: 1]			LVD5F_OE	LVD5F_SEL[0]	LVD5F_RI	LVD5F_F
Initial Value	0	0	1	1	0	0	0	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R(RM1),W

[bit7] LVD5F_PD (Low Voltage Detect external 5v Fall Power Down): External voltage fall power down setting

This bit is used in order to set whether to detect a fall in external voltage or not.

LVD5F_PD	External voltage fall power down setting
0	Invalid (Performs detection)
1	Valid (Stops detection)

This bit is initialized by only power-on reset.

Note:

When setting this bit from power down enable to disable (operation start), set to detection enable (LVD5F_OE=0) 100 µs after setting LVD5F_OE=1. If set before 100 µs, some detection flag setting can occur.

[bit6 to bit4, bit2] LVD5F_SEL (Low Voltage Detect external 5v Fall SELect): External fall detection voltage setting

These bits are the selection signal for a detection level of external voltage fall detection.

LVD5F_SEL[3: 0]	External Voltage Fall Detection Voltage Setting
0000	Setting prohibited
0001	Setting prohibited
0010	Setting prohibited
0011	Setting prohibited
0100	3.70 V ± 8%
0101	3.80 V ± 8%
0110	3.90 V ± 8%
0111	4.00 V ± 8%
1000	4.10 V ± 8%

LVD5F_SEL[3: 0]	External Voltage Fall Detection Voltage Setting
1001	4.20 V ± 8%
1010	4.30 V ± 8%
others	Setting prohibited

LVD5F_SEL[3:0] bits can be rewritten only when LVD5F_OE = "1".

[bit3] LVD5F_OE (Low Voltage Detect external 5v Fall Output Enable): External voltage fall detection output enable setting

This bit is the output enable signal for external voltage fall detection.

LVD5F_OE	External Power Voltage Detection Output Enable Setting
0	Enable
1	Stop

This bit is initialized by only power-on reset.

[bit1] LVD5F_RI (Low Voltage Detect external 5v Fall Reset Interrupt select)

This bit selects either low-voltage detection reset or interrupt.

LVD5F_RI	Low-voltage Detection Reset / Interrupt Selection Setting
0	Reset
1	Interrupt

[bit0] LVD5F_F (Low Voltage Detect external 5v Fall Flag): External voltage fall detection flag

This bit is an external voltage fall detection flag.

LVD5F_F	External Voltage Fall Detection Flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a fall in external voltage is detected, the LVD5F_F bit is set to "1".
This bit is cleared when an external reset is input.

25.5 Operation

This section explains operation of the low-voltage detection (external low-voltage detection).

The external low-voltage detection monitors the external voltage and generates an initialization reset or interrupt if the external voltage drops below the configured value.

Those values of this register cannot be guaranteed if a low-voltage is detected and a settings initialization reset occurs. After the low-voltage reset is released, the reset sequence will be executed without the oscillation stabilization wait time, and then the program is restarted from the address specified by the reset vector.

25.6 Notes

This section provides notes on the low-voltage detection (external low-voltage detection).

Notes on using the low-voltage detection reset circuit

Operation by program

- The low-voltage detection reset circuit operates in accordance with settings, except for the external low-voltage detection rise detection. The external low-voltage rise detection is used as power-on reset.
- Because the external low-voltage detection rise detection operates constantly, current is consumed even in sleep mode, stop mode, and watch mode.

Operation in stop mode

- The low-voltage detection reset can continue to operate even in stop mode by settings. If a low-voltage is then detected in stop mode, the settings initialization reset is generated and the stop mode is cleared.

Hysteresis of detection/reset release voltage

- Since the detection voltage and reset voltage exhibit hysteresis of 0.1 V, the reset release voltage becomes the set detection voltage + 0.1 V.

  For fall detection power supply voltage, the set detection voltage indicates the detection voltage.

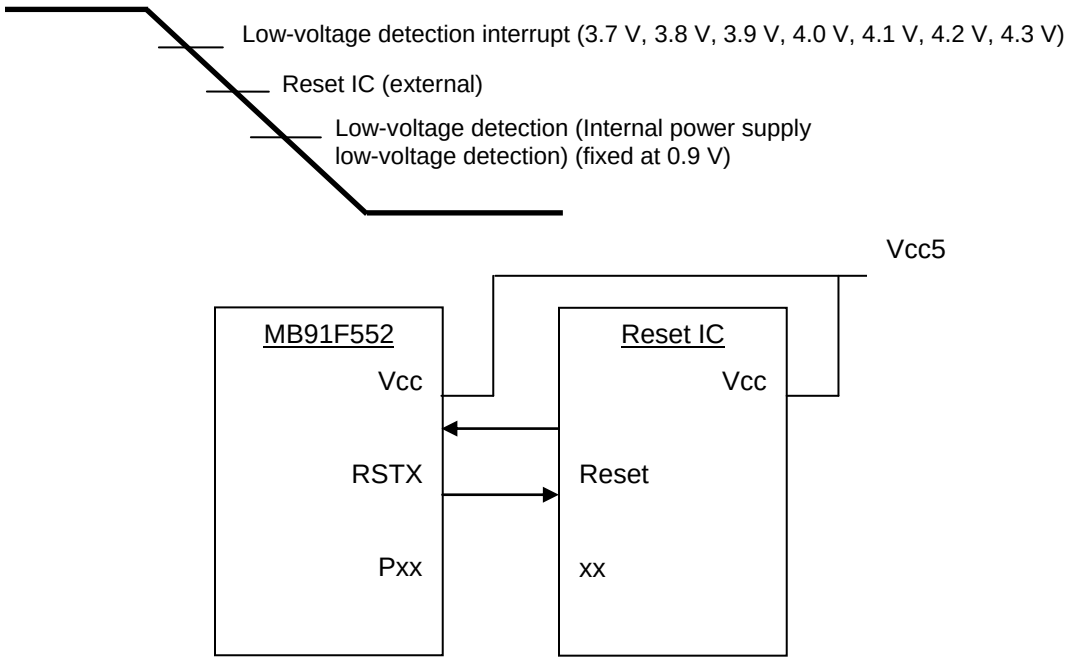
  For example, when 3.9 V ± 8 % is set, the release voltage becomes 4.0 V ± 8 %.

  For rise detection power supply voltage, the set detection voltage indicates the reset release voltage.

  For example, when 2.3 V ± 8 % is set, the detected voltage becomes 2.2 V ± 8 %.

Be sure to connect an external reset IC if an interrupt is generated when low-voltage is detected.

In addition, be sure to set voltage of the reset request signal 3.4 V or more at which operation of the CPU is assured.



26. Wild Register

This chapter explains the wild register.

26.1 Overview
26.2 Features
26.3 Configuration
26.4 Registers
26.5 Operation
26.6 Usage Example

26.1 Overview

This section explains the overview of the wild register.

The function of the wild register is to switch the patch target address data that has been set to the address register with the data that has been set to the data register.

26.2 Features

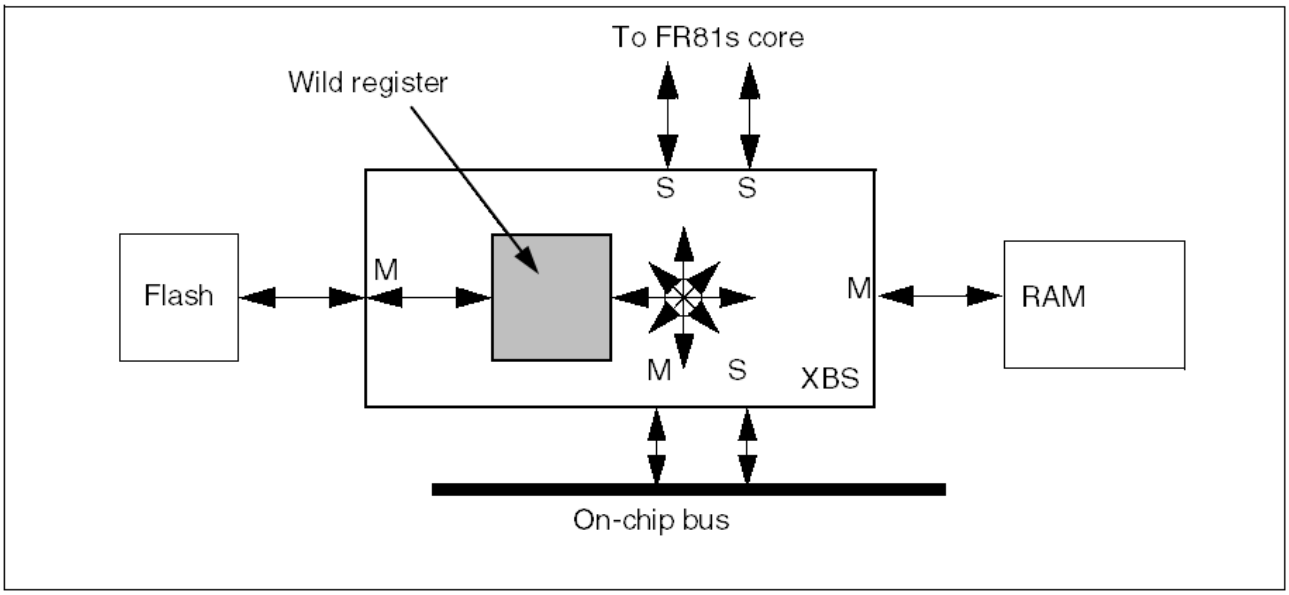
This section explains features of the wild register.

- Allows 16 locations of 1 word each to be patched.
- The target is only the flash area.
- One 16-bit control register
- Sixteen 32-bit address setting registers
- Sixteen 32-bit data setting registers

26.3 Configuration

This section explains the configuration of the wild register.

Figure 26-1. Configuration Diagram



Note:

When the access wait to the FLASH memory is set to 1 cycle, this function cannot be used.

26.4 Registers

This section explains registers of the wild register.

Table 26-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0858	Reserved		WREN		Wild register data enabled register
0x0880	WRAR00				Wild register address register 00
0x0884	WRDR00				Wild register data register 00
0x0888	WRAR01				Wild register address register 01
0x088C	WRDR01				Wild register data register 01
0x0890	WRAR02				Wild register address register 02
0x0894	WRDR02				Wild register data register 02
0x0898	WRAR03				Wild register address register 03
0x089C	WRDR03				Wild register data register 03
0x08A0	WRAR04				Wild register address register 04
0x08A4	WRDR04				Wild register data register 04
0x08A8	WRAR05				Wild register address register 05
0x08AC	WRDR05				Wild register data register 05
0x08B0	WRAR06				Wild register address register 06
0x08B4	WRDR06				Wild register data register 06
0x08B8	WRAR07				Wild register address register 07
0x08BC	WRDR07				Wild register data register 07
0x08C0	WRAR08				Wild register address register 08
0x08C4	WRDR08				Wild register data register 08
0x08C8	WRAR09				Wild register address register 09
0x08CC	WRDR09				Wild register data register 09
0x08D0	WRAR10				Wild register address register 10
0x08D4	WRDR10				Wild register data register 10
0x08D8	WRAR11				Wild register address register 11

Address	Registers				Register Function
	+0	+1	+2	+3	
0x08DC	WRDR11				Wild register data register 11
0x08E0	WRAR12				Wild register address register 12
0x08E4	WRDR12				Wild register data register 12
0x08E8	WRAR13				Wild register address register 13
0x08EC	WRDR13				Wild register data register 13
0x08F0	WRAR14				Wild register address register 14
0x08F4	WRDR14				Wild register data register 14
0x08F8	WRAR15				Wild register address register 15
0x08FC	WRDR15				Wild register data register 15

26.4.1 Wild Register Data Enable Register: WREN (Wild Register ENable register)

The bit configuration of the wild register data enable register is shown.

This register sets whether the wild register function is enabled or disabled on each channel.

WREN: Address 085A$_H$ (Access: Half-word)

	bit15	bit14	• • •	bit2	bit1	bit0
	WREN[15:0]					
Initial Value	0	0	• • •	0	0	0
Attribute	R/W	R/W	• • •	R/W	R/W	R/W

[bit15 to bit0] WREN[15:0] (Wild Register ENable): Enable bits

These bits set whether the wild register function is enabled or disabled on each channel.

WRENn (n = 0 to 15)	Function
0	Disables the wild register function of channel n.
1	Enables the wild register function of channel n.

26.4.2 Wild Register Address Register 00 to 15: WRAR00 to 15 (Wild Register Address Register 00 to 15)

The bit configuration of wild register address register 00 to 15 is shown.

These registers set the address to be amended by the wild register function. The read value is undefined when the wild register operation is enabled.
Always set these registers in units of 32 bits.

WRAR: Address 0880_H to $08F8_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		WRAR[21:16]					
Initial Value	0	0	X	X	X	X	X	X
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	WRAR[15:8]							
Initial Value	X	X	X	X	X	X	X	X
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	WRAR[7:2]						Reserved	
Initial Value	X	X	X	X	X	X	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R0,WX	R0,WX

[bit21 to bit2] WRAR[21:2] (Wild Register Address Register): Address register

These bits set the address to patch. The target address is (WRAR & 0x003FFFFC).
The read value is undefined when the wild register operation is enabled.

26.4.3 Wild Register Data Register 00 to 15: WRDR00 to 15 (Wild Register Data Register 00 to 15)

The bit configuration of wild register data register 00 to 15 is shown.

These registers set the replacement data. When the contents of the memory at the addresses specified by the wild register address registers (WRAR00 to WRAR15) are read, the value set in these registers is returned instead of the actual contents of the memory.
The read value of these registers is undefined while the wild register function is operating.
Always set these registers in units of 32 bits.

WRDR: Address 0884_H to $08FC_H$ (Access: Word)

	bit31	bit30	• • •	bit2	bit1	bit0
	WRDR[31:0]					
Initial Value	X	X	• • •	X	X	X
Attribute	R/W	R/W	• • •	R/W	R/W	R/W

[bit31 to bit0] WRDR[31:0] (Wild Register Data Register): Data register

These bits set the replacement value.
The read value of these registers is undefined while the wild register function is operating.

26.5 Operation

This section explains the operation of the wild register.

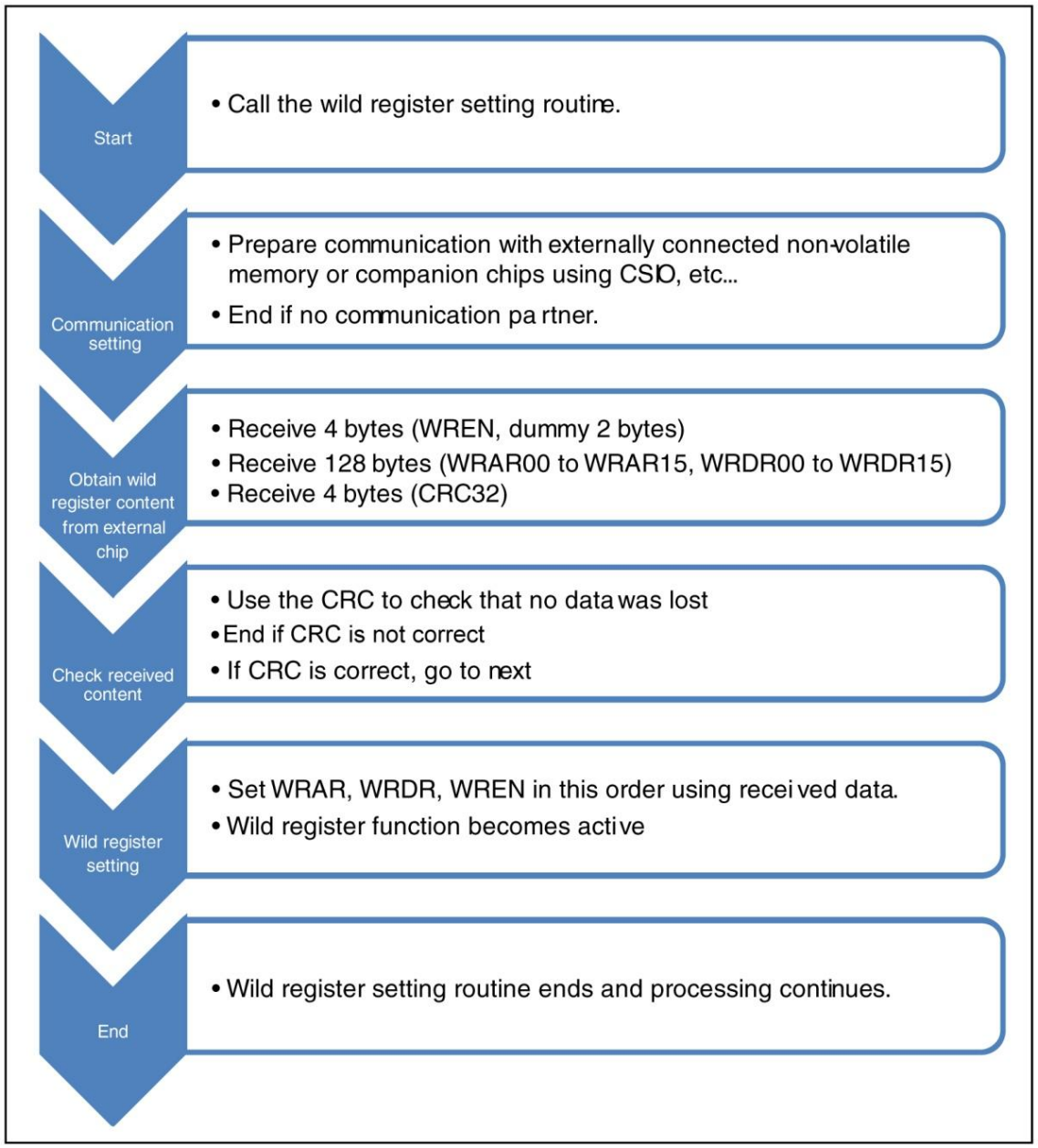
This function is used to patch the flash area. Because the enable register is initialized by reset, this register needs to be set on each reset when being used.
Addresses need to be set so that they do not overlap each other. When addresses overlap, the read value is undefined.
The data's byte line is the big endian.
The target area to replace is the flash area only.

26.6 Usage Example

This section explains a usage example of the wild register.

This section gives an example of using this function. In this example, the settings of this function are called from an externally attached device after reset is released.

Figure 26-2. Usage Example



27. Clock Supervisor

This chapter explains the clock supervisor.

27.1 Overview

27.2 Configuration

27.3 Register

27.4 Operation

27.1 Overview

This section explains the overview of the clock supervisor.

If some kind of problem occurs in the clock and it stops unintentionally, the built-in CR oscillator can substitute for the clock.

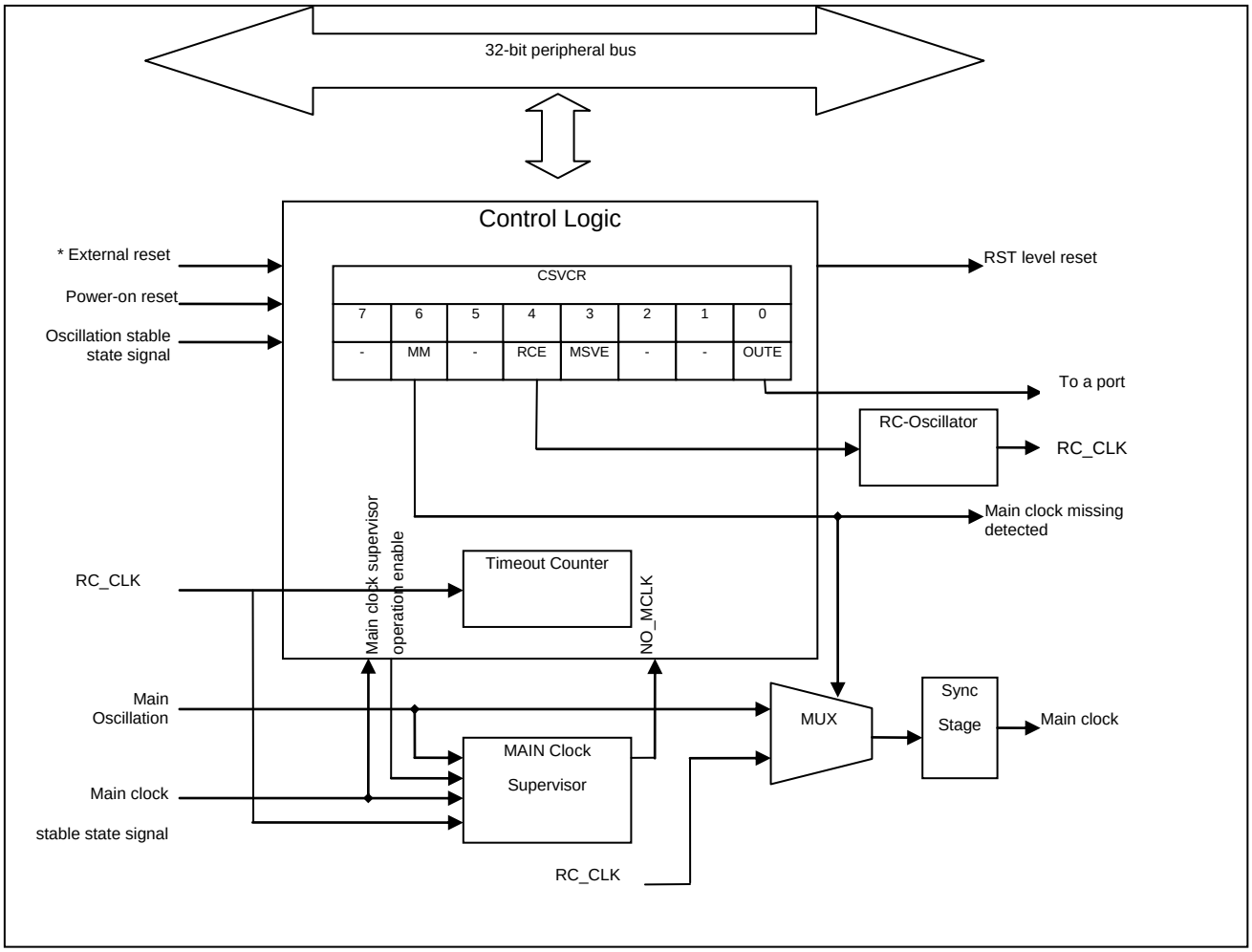
27.2 Configuration

This section explains the configuration of the clock supervisor.

The blocks that configure the clock supervisor are shown below.

- Clock supervisor
- Timeout counter
- Control logic
- CR oscillator

Figure 27-1. Block Diagram (Detailed)



* External reset: On assert of RSTX pin (including simultaneous assert with NMIX)

27.3 Register

This section explains a register of the clock supervisor.

Table 27.3-1 Register Map

Address	Register				Register function
	+0	+1	+2	+3	
0x056C	Reserved	CSVCR	Reserved	Reserved	Clock supervisor control register

27.3.1 Clock Supervisor Control Register: CSVCR (Clock SuperVisor Control Register)

This section shows the bit configuration of the clock supervisor control register (CSVCR).

This register sets the operation mode of clock supervisor.
This register has the bit that shows the breakdown of the clock.

CSVCR: Address 056D$_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	MM	Reserved	RCE	MSVE	Reserved	Reserved	OUTE
Initial value	0	0	0	1	1	1	0	0
Attribute	R0/W0	R/W	R0/W0	R/W	R/W	R/W1	R0/W0	R/W

[bit7] Reserved

Be sure to write "0" to this bit.

[bit6] MM (Main clock Missing): Main clock stop

When this bit is "1", it indicates that some problem is found in the main oscillation clock.
When this bit is "0", there are no problems in the main clock.
When the main clock is not restored, "0" write access is ignored.
This bit will be cleared to "0" on power-on or external reset. Other types of resets do not affect this bit.

MM	Read	Write
0	Main oscillation clock stop undetected	When the main clock is restored oscillating, this bit can be cleared.
1	Main oscillation clock stop detected	No effect

Note:

Do not enable the PLL/SSCG oscillation operation when this bit is "1".

[bit5] Reserved

Be sure to write "0" to this bit.

[bit4] RCE (RC-oscillator Enable): CR oscillator enable

The oscillation of the CR oscillator is permitted in the standby mode when this bit is set to "1".
Setting this bit to "0" is prohibited while main clock supervisor has been still permitted.
First of all, it is necessary to confirm the MM bit is "0" after prohibiting the supervisor. Afterwards, set the RCE bit to "0".
Please do not set the RCE bit to "0" when MM bit is "1".
This bit will be cleared to "1" on power-on or external reset. Other types of resets do not affect this bit.

RCE	Description
0	CR oscillation disabled at STBY mode
1	CR oscillation enabled at STBY mode (initial value)

[bit3] MSVE (Main clock SuperVisor Enable): Main clock supervisor enable

When this bit is set to "1", the main clock supervisor is enabled. This bit is initialized to "1" only when the power is turned on. Other types of resets do not affect this bit.

MSVE	Description
0	Main clock supervisor disabled
1	Main clock supervisor enabled (initial value)

[bit2] Reserved

Be sure to write "1" to this bit.

[bit1] Reserved

Be sure to write "0" to this bit.

[bit0] OUTE (Output Enable): Output enable

This bit is the enable bit for outputting the MM bit from a port. When "1" is set to this bit, outputting the MM bit is enabled. This bit is cleared to "0" by turning on the power supply or by external reset. Other types of resets do not affect this bit.

OUTE	Description
0	Do not enable the MM output from a port (initial value).
1	Enable the MM output from a port.

27.4 Operation

This section explains the operation of the clock supervisor.

After the clock replaces the CR oscillator, it is reset at once when the main clock stops while CPU is operating with the main clock. When the clock is not input for the period of 20μ s to 80μs, it is judged that it stops. Because the bit indicating that the main clock has stopped remains in the register, it is possible to judge that a problem has occurred with the software.

When the main clock is stopped intentionally, the main clock supervisor will stop automatically. The CR oscillator stops automatically by moving to the standby mode when the CR oscillation in the standby mode is prohibited. The CR oscillator reactivates automatically when returning from the standby mode.

Note:

Please do not permit the PLL/SSCG oscillation operation if the main clock operates as a replacement for the CR oscillator after detecting the main clock stop.

The following explains the operational mode of the clock supervisor.

27.4.1 Initial State

This section explains the initial state.

At initial setting, the oscillation of the CR oscillator and main clock supervisor function are enabled.

CR Oscillator

The oscillation is enabled when the power is turned on.

Only when changing to the standby mode with "0" written in oscillation enable bit (CSVCR.RCE) in the standby mode, it stops. When the standby mode is released, the oscillation is automatically restarted.

Main Clock Supervisor

The main clock supervisor is enabled after the main oscillation stabilization wait time has elapsed.

When the main clock supervisor is enabled, if the main clock stops, the main clock is replaced by the CR oscillation clock.

Moreover, the MM bit of the CSVCR register is set to "1" and an RST level reset is generated.

Note:

Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.

In this case, after the timeout period measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

27.4.2 Stopping the CR Oscillator and the Clock Supervisor Function

This section explains stopping the CR oscillator and the clock supervisor function.

CR Oscillator

The CR oscillator can be stopped only in the standby mode. Please change to the standby mode after setting oscillation enable bit (CSVCR.RCE) in the standby mode to "0".

When there is a problem with the main clock, the stop of the CR oscillator is prohibited. It can be confirmed whether or not the problem exists in the clock by the MM bit of the CSVCR register.

Note:

The operation clock stops, too, when the CR oscillation is stopped because the operation clock has already been replaced by the CR oscillation clock when there is a problem in the clock.

Main Clock Supervisor

The MSVE bit of the CSVCR register is set to "0".

27.4.3 Re-enabling the Clock Supervisor

This section explains re-enabling the clock supervisor.

Main Clock Supervisor

To re-enable the main clock supervisor function, set the MSVE bit of the CSVCR register to "1".

When the CR oscillator is stopped, enabling the main clock supervisor function is prohibited.

Note:

Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.

In this case, after the timeout period measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

27.4.4 Stop Mode

This section explains the stop mode of the clock supervisor.

CR Oscillator

The oscillation stops by changing to the stop mode when the oscillation enable bit (CSVCR.RCE) in the stop mode is set to "0".
After the stop mode is released, it is re-enabled automatically.

Main Clock Supervisor

When the main clock supervisor function is enabled, it automatically stops when changing to the stop mode.
The main clock supervisor enable bit (CSVCR.MSVE) does not change to "0".
After the stop mode is released, the supervisor is automatically re-enabled after waiting for the main oscillation stabilization wait time.

Note:

Because the main oscillation stabilization wait time is measured by the main clock itself, if the main clock stops before the oscillation stabilization wait time has elapsed, the main oscillation stabilization wait time does not end and the main clock supervisor is not enabled.
In this case, after the timeout period measured by the internal CR oscillator has elapsed, the main supervisor function is enabled regardless of the oscillation stabilization wait time and the main clock stop is detected.

Note:

When the main clock supervisor function is disabled, if stop mode is entered, the supervisor remains disabled even after recovering from the stop mode.

27.4.5 Watch Mode

This section explains the watch mode.

Main Clock Supervisor

The main clock supervisor function is not influenced from the transition to the watch mode.

When the main clock supervisor function enabled, the system switches to the CR oscillation clock and a reset is issued when the main clock stops.

27.4.6 Checking the Reset Factor Using the Clock Supervisor

This section shows checking the reset factor using the clock supervisor.

The method for checking whether or not the clock supervisor detected a clock problem and generated a reset is shown below.
First, read the RSTRR register (see "4.1 Reset Source Register: RSTRR (ReSeT Result Register)" in "Chapter: Reset") to check the reset factor.

If the ERST bit of the RSTRR register is "1", this indicates that any of reset input from the RSTX external pin, illegal standby mode transition detection reset, external power supply low-voltage detection, clock supervisor reset, or simultaneous assert of RSTX and NMIX external pins was generated.
Please read the CSVCR register in this case, and confirm the MM bit.

Reset factor can be checked as follows.

Table 27-1. Reset Factor

ERST	MM	Reset Factor
1	1	Clock supervisor reset
1	0	Reset input from the RSTX external pin Illegal standby mode transition detection reset External power supply low-voltage detection Simultaneous assert of RSTX and NMIX external pins

Note

Because the MM bit is not cleared in conditions other than turning the power-on and the external reset, it is necessary to confirm other reset factors reading the RSTRR register (see "4.1 Reset Source Register: RSTRR (ReSeT Result Register)" in "Chapter: Reset").

27.4.7 Returning from the CR Clock

This section shows returning from the CR clock.

Main Clock Supervisor

The main clock stops when the CPU detects that the MM bit has been set after recovering from a reset, and it is possible to determine that the system has switched to the CR oscillation clock. At this time, it is possible to return to the main clock by writing "0" in the MM bit if it can be confirmed that the main clock is restored.

When the main clock is not restored, writing "0" in the MM bit does not have any influence. The MM bit keeps maintaining "1".

The MM bit is cleared if the main clock is under operation when "0" is written in the MM bit, and the clock returns to the main clock via a synchronous stage.

It can perform polling on the MM bit until the main clock is restored.

```
ldi #_csvcr,r1
clear_CSV_loop:
bandh #0b1001,@r1                ;; Clear MM
btsth #0b0110,@r1                ;; Check: Is one of them 1?
bne clear_CSV_loop
```

28. Regulator Control

This chapter explains the regulator control.

28.1 Overview

28.2 Features

28.3 Configuration

28.4 Register

28.1 Overview

This section explains the overview of the regulator control.

The following mode is entered for operation of the regulator that generates the internal voltage.

- Main mode (during normal operation and during stop mode and watch mode)

28.2 Features

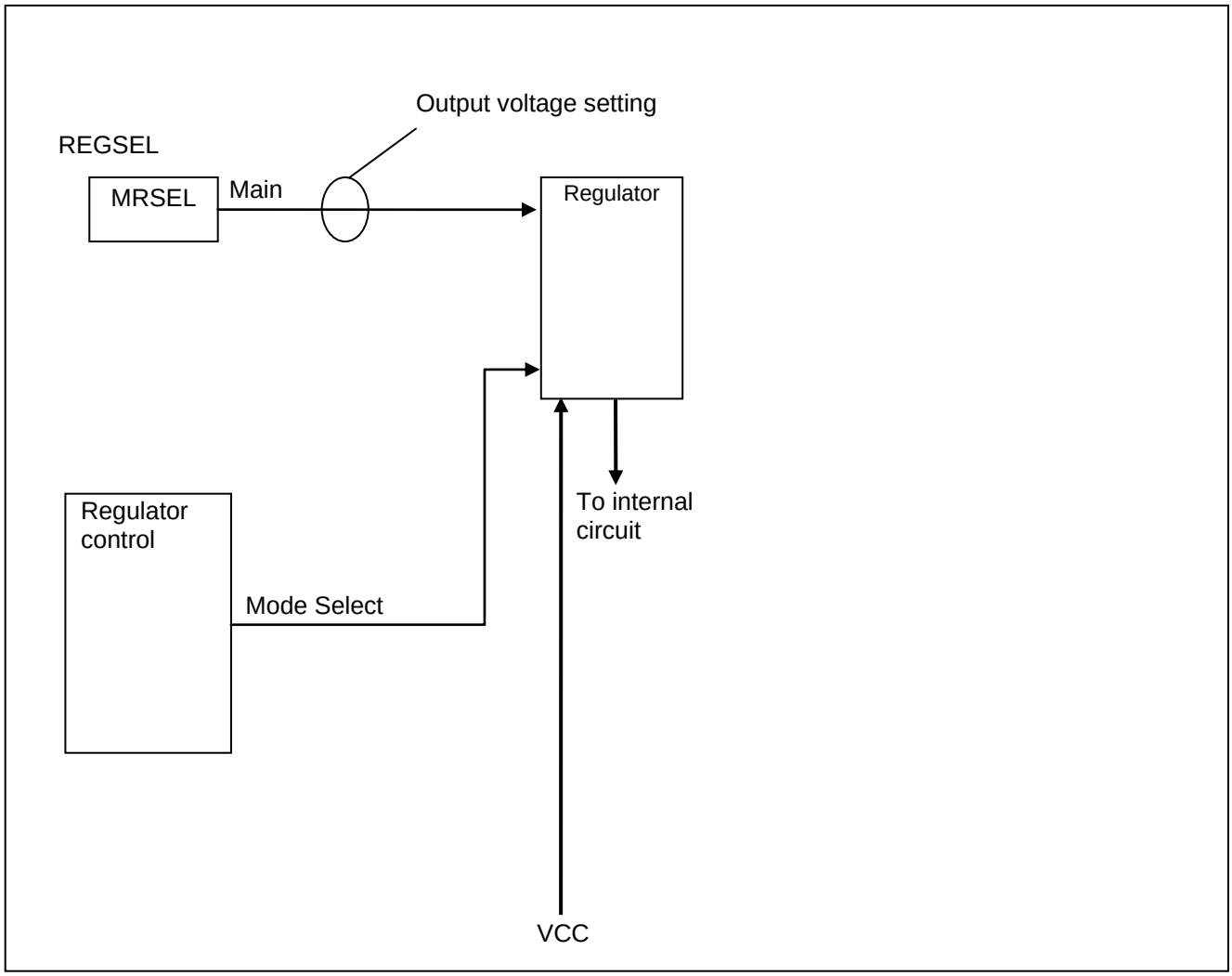
This section explains features of the regulator control.

The regulator mode is automatically changed according to the device state transition.

28.3 Configuration

This section explains the configuration of the regulator control.

Figure 28-1. Regulator Control Overview Diagram



28.4 Register

This section explains a register of the regulator control.

Table 28-1. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0580	REGSEL	Reserved	Reserved	Reserved	Regulator Output Voltage Select Register

28.4.1 Regulator Output Voltage Select Register: REGSEL (REGulator output voltage SELect register)

The bit configuration of the regulator output voltage selection register is shown below.

This register selects the output voltage level of each regulator mode (main/standby).

REGSEL: Address 0580_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MRSEL[1:0]		Reserved		Reserved			Reserved
Initial Value	0	1	1	0	0	1	1	0
Attribute	R/W0	R/W1	R1/W1	R0/W0	R0/W0	R1/W1	R1/W1	R0,WX

[bit7, bit6] MRSEL[1:0] (Main Regulator voltage SELect)

These bits set the output voltage level of main regulator (regulator mode: main mode).

MRSEL[1:0]	Main Regulator Output Voltage
00	Reserved
01	1.2 ± 0.1V
10	Reserved
11	Reserved

[bit5] Reserved

Be sure to write "1" to this bit.

[bit4] Reserved

Be sure to write "0" to this bit.

[bit3] Reserved

Be sure to write "0" to this bit.

[bit2 to bit1] Reserved

Be sure to write "1" to these bits.

[bit0] Reserved

The value "0" is always read. Writing to this bit has no influence on operation.

29. Bus Performance Counters

This chapter explains the bus performance counters.

29.1 Overview

29.2 Features

29.3 Configuration

29.4 Registers

29.5 Operation

29.1 Overview

This section explains the overview of the bus performance counters.

This series has a built-in bus performance counters (BPC) for measuring the performance of the on-chip bus. BPC measures the breakdown of traffic on the on-chip bus, and provides information for strategies to improve bus performance. Because the counters do not count while the on-chip bus is idle, use the timers in the system at the same time to measure the time.

29.2 Features

This section explains the features of the bus performance counters.

- Counter configuration

  Count clocks: Clock for the on-chip bus
  Counter bit length: 32-bit x 3 channels (BPC-A, BPC-B, BPC-C)
  Overflow detection: None
  Counter value rewrite: AllowedTrigger interrupt 0-3

- Main functions

  The following operations can be selected for counting in each channel.

- Number of read accesses in the on-chip bus
- Number of write accesses in the on-chip bus
- Number of wait cycles in the on-chip bus

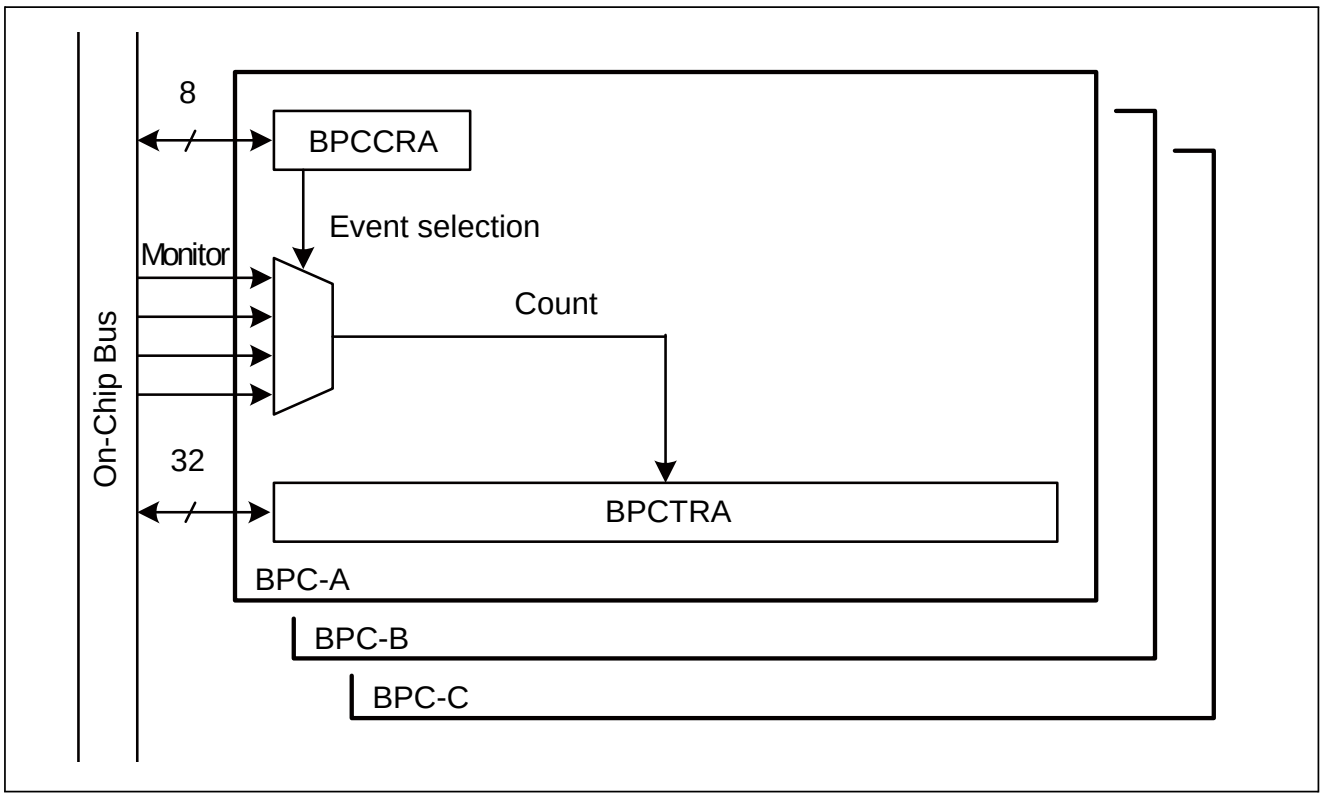
One of the following operations can be selected for counting in each channel.

- Specific bus master (CPU, DMAC, other, or all)
- Specific target (ICH, MCH, other, or all)

29.3 Configuration

This section explains the configuration of the bus performance counters.

Figure 29-1. Block Diagram



29.4 Registers

This section explains the registers of the bus performance counters.

Table 29-1. Register Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x0710	BPCCRA	BPCCRB	BPCCRC	Reserved	BPC-A control register BPC-B control register BPC-C control register
0x0714	BPCTRA				BPC-A count register
0x0718	BPCTRB				BPC-B count register
0x071C	BPCTRC				BPC-C count register

29.4.1 BPC-A Control Register: BPCCRA (Bus Performance Counter Control Register A)

The bit configuration of the BPC-A control register is shown below.

This register configures the measurement target of bus performance counter A (BPC-A).
The bus performance counters have three channels, A, B, and C, and there is a control register for each of these counters.
Each field of the control register is common to each channel.

BPCCRA: Address 0710_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FUNC[1:0]		MST[3:0]				SLV[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7, bit6] FUNC[1:0] (FUNCtion selection): Measurement event selection

These bits select the event measured by BPC.

FUNC[1:0]	Event
00	BPC-A operation stopped (initial value)
01	Number of read accesses
10	Number of write accesses
11	Number of wait cycles

[bit5 to bit2] MST[3:0] (bus MaSTer select): Bus master selection

These bits select the bus master for the events which are measured by BPC.

MST[3:0]	Bus Master
0000	All bus masters (initial value)
0001	CPU (XBS)
0010	DMAC
0011	Reserved
0100	Reserved
Except for the above	Reserved

[bit1, bit0] SLV[1:0] (SLaVe select): Slave selection

These bits select the slave for the events which are measured by BPC.

SLV	Slave
00	All slaves (initial value)
01	MCH (registers)
10	ICH (peripherals)
11	Slaves other than MCH/ICH

29.4.2 BPC-B Control Register: BPCCRB (Bus Performance Counter Control Register B)

The bit configuration of the BPC-B control register is shown below.

This register configures the measurement target of bus performance counter B (BPC-B).
The function of each bit is the same as BPCCRA.

BPCCRB: Address 0711_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FUNC[1:0]		MST[3:0]				SLV[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

29.4.3 BPC-C Control Register: BPCCRC (Bus Performance Counter Control Register C)

The bit configuration of the BPC-C control register is shown below.

This register configures the measurement target of bus performance counter C (BPC-C).
The function of each bit is the same as BPCCRA.

BPCCRC: Address 0712_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FUNC[1:0]		MST[3:0]				SLV[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

29.4.4 BPC-A Count Register: BPCTRA (Bus Performance CounTer Register A)

The bit configuration of the BPC-A count register is shown below.

This register is a 32-bit length count register that counts the events configured by BPCCRA.

BPCTRA: Address 0714_H (Access: Word)

	bit31	bit30	- - -	bit3	bit2	bit1	bit0
	BPCTRA[31:0]						
Initial value	0	0	- - -	0	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W	R/W

[bit31 to bit0] BPCTRA[31:0] (Bus Performance CounTer Register A): BPC-A count

If bit7, bit6:FUNC of the BPCCRA are set to a value other than "00", the count of the target events begins. This register is readable and writable, and can only be accessed using 32-bit access.

Because the counter is not initialized when the count is started, set the initial value when starting a new count. Furthermore, because there is no overflow control, if the counter overflows it returns to zero and continues counting.

29.4.5 BPC-B Count Register: BPCTRB (Bus Performance CounTer Register B)

The bit configuration of the BPC-B count register is shown below.

This register is a 32-bit length count register that counts the events configured by BPCCRB. The usage is the same as BPCTRA.

BPCTRB: Address 0718_H (Access: Word)

	bit31	bit30	- - -	bit3	bit2	bit1	bit0
	BPCTRB[31:0]						
Initial value	0	0	- - -	0	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W	R/W

29.4.6 BPC-C Count Register: BPCTRC (Bus Performance CounTer Register C)

The bit configuration of the BPC-C count register is shown below.

This register is a 32-bit length count register that counts the events configured by BPCCRC. The usage is the same as BPCTRA.

BPCTRC: Address 071C$_H$ (Access: Word)

	bit31	bit30	- - -	bit3	bit2	bit1	bit0
	BPCTRC[31:0]						
Initial value	0	0	- - -	0	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W	R/W

29.5 Operation

This section explains the operations.

29.5.1 Setting

29.5.2 Starting and Stopping

29.5.3 Operation

29.5.4 Measurement and Result Processing

29.5.1 Setting

This section explains the setting.

Before starting each of the BPC channels, write "0x00000000" to BPCTRA, BPCTRB, and BPCTRC, and initialize each counter. Initialize each counter in the same way when changing the measurement target. Because the counter value is undefined after reset, always write the counter value before enabling operation.

When starting each BPC channel, configure the measurement target of each counter using BPCCRA, BPCCRB, and BPCCRC.

The events monitored by the settings of the bus performance counter A (B, C) control register (BPCCRA (B, C)) are as follows. Operation is not guaranteed for any combination that does not exist in the following table. Moreover, it does not count in emulator mode.

Table 29-2. List of BPC Settings

FUNC[1:0]	MST[3:0]	SLV[1:0]	Target Event
01	0000	00	Read from XBS, DMAC
		01	MCH read from XBS, DMAC
		10	ICH read from XBS, DMAC
		11	Other than MCH/ICH read from XBS, DMAC
	0001	00	Read from XBS
		01	MCH read from XBS
		10	ICH read from XBS
		11	Other than MCH/ICH read from XBS
	0100	00	Read access from DMAC
		01	MCH read from DMAC
		10	ICH read from DMAC
		11	Other than MCH/ICH read from DMAC
10	0000	00	Write from XBS, DMAC
		01	MCH write from XBS, DMAC
		10	ICH write from XBS, DMAC
		11	Other than MCH/ICH write from XBS, DMAC
	0001	00	Write from XBS
		01	MCH write from XBS
		10	ICH write from XBS
		11	Other than MCH/ICH write from XBS
	0100	00	Write from DMAC
		01	MCH write from DMAC
		10	ICH write from DMAC
		11	Other than MCH/ICH write from DMAC

FUNC[1:0]	MST[3:0]	SLV[1:0]	Target Event
11	0000	00	Wait access of XBS, DMAC
		01	MCH wait from XBS, DMAC
		10	ICH wait from XBS, DMAC
		11	Other than MCH/ICH wait from XBS, DMAC
	0001	00	Wait access from XBS
		01	MCH wait from XBS
		10	ICH wait from XBS
		11	Other than MCH/ICH wait from XBS
	0100	00	Wait access from DMAC
		01	MCH wait from DMAC
		10	ICH wait from DMAC
		11	Other than MCH/ICH wait from DMAC

29.5.2 Starting and Stopping

This section explains the starting and stopping.

The target event count is started by setting the FUNC[1:0] bits of the bus performance counter A control register (BPCCRA) to a value other than "00". However, at this time the count starts from the current value without initializing the bus performance counter A register (BPCTRA). The operation of the bus performance counter stops when BPCCRA:FUNC[1:0] is set to "00".

29.5.3 Operation

This section explains the operations.

Once operation has been enabled by setting the control register, each of the measurement target operations continues to be counted while the on-chip bus is operating. However, the count is paused in the circumstances shown below.

- While in emulator mode

The count operation when each of the low-power consumption modes is set is as follows.

- CPU sleep mode

Each measurement target operation is counted.

- Bus sleep mode
  Only counted during DMA transfers that operate the on-chip bus. During other periods, counting is not performed because the measurement target operations do not occur.
- Standby mode (watch mode/stop mode)
  Counting is not performed because the measurement target operations do not occur.

The control register is initialized when a reset occurs. Counting is not performed immediately after a reset occurs.

29.5.4 Measurement and Result Processing

This section explains the measurement and result processing.

The use of BPC is anticipated for when ICE is connected or when using a monitor debugger. The configuring of measurements and reading of results are performed in debug mode while the user program execution is halted.

Examples of measurements are as follows.

- Measure between two points in a user program
- Measure a reference time base

These are explained below.

Measuring between two points in a user program

During this measurement, the measurement starting point and measurement ending point in the user program are configured as follows.

- Measurement starting point: Starting point of the user program execution
- Measurement ending point: Breakpoint in the user program

The measurement sequence is as follows.

1. Configure the measurement and initialize the counter in debug mode
2. Start executing the user program from the measurement starting point
3. Break on the measurement ending point and stop executing the user program
4. Switch to debug mode and read the measurement results

Measure a reference time base

During this measurement, switch to debug mode at each reference time, read out the measurement results and initialize the counters.

The following two methods are available for switching to debug mode at each reference time.

- Assert a tool break from the ICE at each reference time to switch to debug mode (when connected to ICE)
- Set the interval time of a built-in timer to the reference time, and execute the INTE instruction in the timer interrupt routine to switch to debug mode

The measurement sequence is as follows.

1. Configure the measurement and initialize the counter in debug mode
2. Begin executing the measurement target user program
3. Tool break by reference time, or execute the INTE instruction by built-in timer interrupt routine
4. Switch to debug mode and read the measurement results
5. Initialize the measurement counter
6. Repeat steps 2 to 5

Analyze the measurement results using a debugger host program, such as SOFTUNE Workbench. Visualize the analysis results by displaying them in a graph so that they can be understood intuitively (pie graph, bar graph, line graph, etc.), and provide information that is beneficial for user program tuning (bus performance analysis function). The following is an analysis example.

Analysis example:

1. Bus master access proportion
   Ex. Proportion of DMAC access vs. CPU access, specific bus master access that occupies the total access, etc.
2. Occurred event proportion
   Ex. Proportion of write access vs. read access, proportion of total cycles made up of wait cycles, etc.
3. Target accessed proportion
   Ex. Proportion of MCH vs. ICH, proportion of total accesses made up of accesses to a specific target, etc.
4. Proportion of specific accesses from a specific bus master to a specific target
   Ex. Proportion of total access made up of read accesses from CPU to MCH, etc.
5. Proportion of wait cycles occurring in specific target
   Ex. Proportion of total wait cycles made up of wait cycles during MCH access
6. Analyze operation of each bus between two specific points in a program
   Ex. Proportion of total cycles between two specific points in the program consisting of read, write, wait cycles, etc.
7. Analyze operation of each bus during progress of each specific time
   Ex. Time course of proportion of all accesses consisting of accesses to specific bus masters and specific targets, etc.

30. CRC

This chapter explains the CRC.

30.1 Overview

30.2 Features

30.3 Configuration

30.4 Registers

30.5 Operation

30.1 Overview

This section explains the overview of the CRC.

This module calculates CRC values.

CRC (Cyclic Redundancy Check) is a kind of error detection methods. CRC codes are remainders left when input data strings, regarded as high-degree polynomials, are divided by predefined generator polynomials. Normally, a CRC code is attached at the end of a data string, and received data is regarded as correct if the data leaves no remainder when divided by the same generator polynomial.

30.2 Features

This section explains features of the CRC.

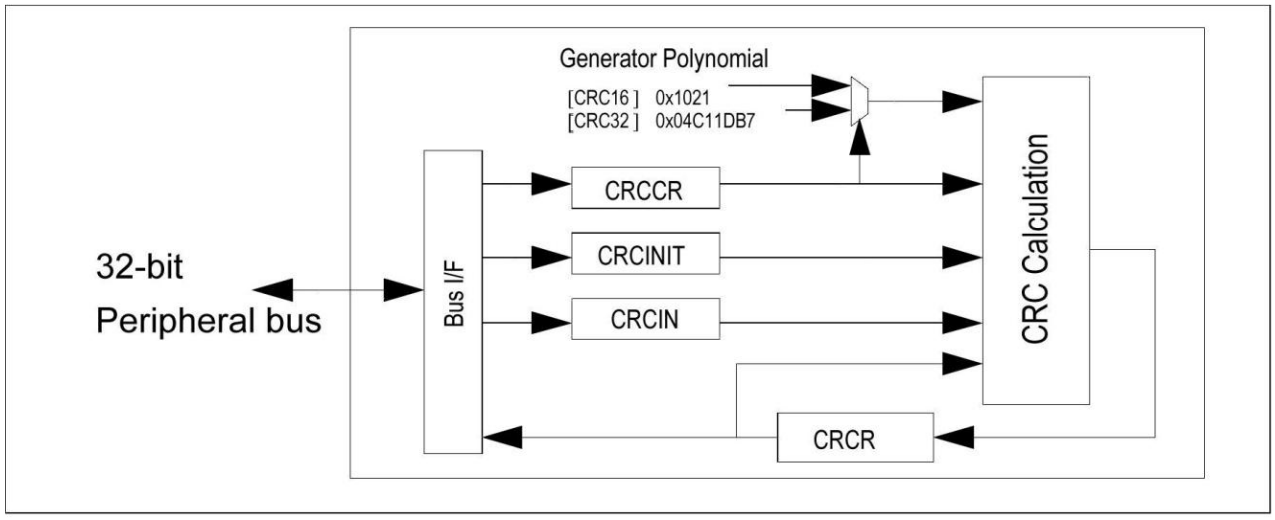
This module calculates CCITT CRC16 and IEEE-802.3 CRC32. This module cannot calculate CRC values based on other generator polynomials because the generator polynomials of this module are fixed for the values of CCITT CRC16 and IEEE-802.3 CRC32.

- CCITT CRC16 generator polynomials: 0x1021
- IEEE-802.3 CRC32 generator polynomials: 0x04C11DB7

30.3 Configuration

This section explains the configuration of the CRC.

Figure 30-1. Block Diagram



30.4 Registers

This section explains registers of the CRC.

Table 30-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x1130	Reserved			CRCCR	CRC control register
0x1134	CRCINIT				Initial value register
0x1138	CRCIN				Input Data register
0x113C	CRCR				CRC register

30.4.1 CRC Control Register: CRCCR (CRC Control Register)

The bit configuration of the CRC control register is shown below.

This register controls the CRC calculation.

CRCCR: Address 1133_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	FXOR	CRCLSF	CRCLTE	LSBFST	LTLEND	CRC32	INIT
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R0,W

[bit7] Reserved: Reserved bits

The read value of these bits is always "0".

These bits must always be written to "0".

[bit6] FXOR (Final XOR): Final XOR control bit

CRC results are output as the XOR value and XOR. The XOR values are 0xFFFFFFFF and bit strings are inverted when FXOR=1 is true. This process is made in the latter part of the CRC register, and the result is reflected in the CRC result readout value immediately after this bit setting.

[bit5] CRCLSF (CRC result LSb First): CRC result bit order setting bit

This bit sets bit orders for CRC results. Changes the bit order in a byte. When this bit is "0", MSB First is applied, and when this bit is "1", LSB First is applied. This process is made in the latter part of the CRC register, and the result is reflected in the CRC result readout value immediately after this bit setting.

[bit4] CRCLTE (CRC result LiTtle Endian): CRC result byte order setting bit

This bit sets byte orders for CRC results. Changes the byte order in a word. When this bit is "0", big endian is applied, and when this bit is "1", little endian is applied. This process is made in the latter part of the CRC register, and the result is reflected in the CRC result readout value immediately after this bit setting. When this bit is set to 1 for CRC16, the result is output in 31 to 16 bits.

[bit3] LSBFST (LSB FirST): Bit order setting bit

This bit sets bit orders. Specifies the first bit of a byte (8bit). When this bit is "0", MSB First is applied, and when this bit is "1", LSB First is applied. 4 patterns of process order can be specified by combining the LTLEND setting.

[bit2] LTLEND (LitTtLe ENDian): Byte order setting bit

This bit sets byte orders. Specifies byte orders in a writing width. When this bit is "0", big endian is applied, and when this bit is "1", little endian is applied.

[bit1] CRC32 (CRC32): CRC mode selecting bit

This bit selects a mode for CRC16 and CRC32. When CRC32=1 is true, the arithmetic operation mode of CRC32 is applied.

[bit0] INIT (INITialize): Initialization bit

Initialization bit. When "1" is written to this bit, software performs the initialization. This bit does not have a value and "0" is always returned at readout. In initialization, hardware loads the value of the initial value register to the CRC register. Initialization needs to be performed once at the beginning of the CRC calculation.

30.4.2 CRC Initial Value Register: CRCINIT (CRC Initial value register)

The bit configuration of the CRC initial value register is shown below.

This register sets the initial value for the CRC calculation.

CRCINIT: Address 1134_H (Access: Byte, Half-word, Word)

	bit 31	bit30	- - -	bit2	bit1	bit0
	D[31:0]					
Initial value	1	1	- - -	1	1	1
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

[bit31 to bit0] D (Data): Initialization value bits

These bits store the initial value for the CRC calculation. Software writes the initial value for the CRC calculation. (0xFFFF_FFFF is applied after reset.) For CRC16, D15 to D0 are used and D31 to D16 are ignored.

30.4.3 Input Data Register: CRCIN (CRC INput data register)

The bit configuration of the CRC input data register is shown below.

This register sets the input data for the CRC calculation.

CRCIN: Address 1138_H (Access: Byte, Half-word, Word)

	bit 31	bit30	- - -	bit2	bit1	bit0
	D[31:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

[bit31 to bit0] D (Data): Input Data bits

These bits set the input data for the CRC calculation. Software writes the input data for the CRC calculation. The bit width of 8, 16, 32 is used. These bits width can be mixed. Bytes or half words can be written into any position. The address position can be +0, +1, +2 or +3 for byte writing and +0 or +2 for half word writing.

30.4.4 CRC Register: CRCR (CRC Register)

The bit configuration of the CRC register is shown below.

This register outputs the result for the CRC calculation.

CRCR: Address 113C$_H$ (Access: Byte, Half-word, Word)

	bit 31	bit30	- - -	bit2	bit1	bit0
	D[31:0]					
Initial value	1	1	- - -	1	1	1
Attribute	R,WX	R,WX	- - -	R,WX	R,WX	R,WX

[bit31 to bit0] D (Data): CRC bits

These bits output the result for the CRC calculation. When software writes "1" to the initialization bit (CRCCR. INIT), the value of the initial value register (CRCINIT) is loaded to this register. When software writes the input data for the CRC calculation to the Input Data register (CRCIN), hardware immediately sets the CRC calculation result to this register. When all input data has been written, this register holds the final CRC code. When CRC16 is used, the result is output in D15 to D0 for big-endian (CRCLTE=0) byte order and in D31 to D16 for little-endian (CRCLTE=1) byte order.

30.5 Operation

This section explains the CRC.

30.5.1 CRC Definition
30.5.2 Reset Operation
30.5.3 Initialization
30.5.4 Byte and Bit Orders
30.5.5 CRC Calculation Sequence
30.5.6 Examples

30.5.1 CRC Definition

The CRC definition is shown below.

CCITT CRC16 Standard

Generator polynomials 0x1021(CRCCR:CRC32=0)

Initial value 0xFFFF

Final XOR value 0x0000(CRCCR:FXOR=0)

Bit order MSB First(CRCCR:LSBFST=0)

Output bit order MSB First(CRCCR:CRCLSF=0)

(Any byte order can be set for input and output)

IEEE-802.3 CRC32 Ethernet Standard

Generator polynomials 0x04C11DB7(CRCCR:CRC32=1)

Initial value 0xFFFF_FFFF

Final XOR value 0xFFFF_FFFF(CRCCR:FXOR=1)

Bit order LSB First (CRCCR:LSBFST=1)

Output bit order LSB First (CRCCR:CRCLSF=1)

(Any byte order can be set for input and output)

30.5.2 Reset Operation

The reset operation of the CRC is shown below.

To reset, set 0xFFFF_FFFF to the initial value register (CRCINIT) and CRC register (CRCR). Others are cleared to "0".

30.5.3 Initialization

The initialization of the CRC is shown below.

In initialization by CRCCR:INIT, the value of the initial value register is loaded to the CRC register (CRCR).

30.5.4 Byte and Bit Orders

The byte and bit orders of the CRC is shown below.

This section explains the byte and bit orders using examples. Inputs the following 1 word to the CRC calculator.

133.82.171.1 = 10000101 01010010 10101011 00000001

When the byte order is big endian (CRCCR:LTLEND=0), the transmission sequence in bytes is:

10000101	01010010	10101011	00000001
(First)	(Second)	(Third)	(Fourth)

When the bit order is LSB First (CRCCR:LSBFST=1), the transmission sequence in bits is:

10100001	01001010	11010101	10000000
(First)			(Last)

Notes:

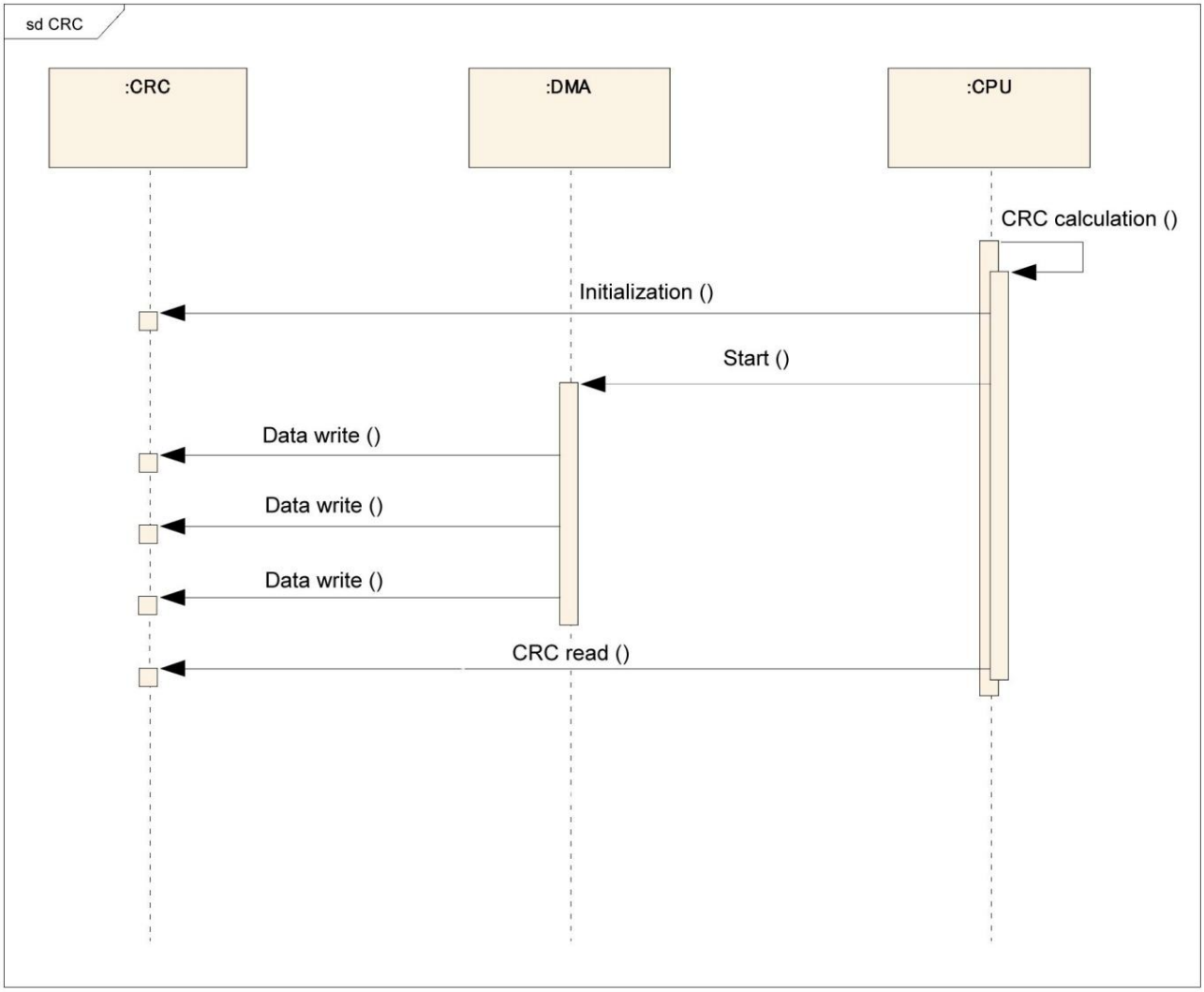
- When CRCCR:CRCLTE=1 is true, the byte order for the CRC result is changed in 32-bit width both for CRC16 and CRC32.
- Note that output position for CRC16 is bit31 to bit16.

30.5.5 CRC Calculation Sequence

The CRC calculation sequence is shown below.

The sequence for the CRC calculation is shown below. In the following explanation, the initial value register (CRCINIT) setting, CRC16/32 selection (CRCCR:CRC32), byte order and bit order settings (CRCCR:LTLEND, CRCCR:LSBFST) have been done. (When the initial value of 0xFFFFFFFF is acceptable, the setting for the initial value register (CRCINIT) can be omitted.)

Figure 30-2. CRC Calculation Sequence



- To initialize, write "1" to the initialization bit (CRCCR.INIT). The value of the initial value register will be loaded to the CRC register (CRCR).
- Input data is written to the Input Data register (CRCIN). The writing operation starts the CRC calculation. Input data can be written continuously. In addition, there can be different bit widths of writing in a sequence.
- The CRC code is obtained with the readout of the CRC register (CRCR).

30.5.6 Examples

The examples are shown below.

30.5.6.1 Example 1 CRC16, Fixed Byte Input

30.5.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths

30.5.6.3 Example 3 CRC32, Byte Order, Big-endian

30.5.6.4 Example 4 CRC32, Byte Order, Little-endian

30.5.6.1 Example 1 CRC16, Fixed Byte Input

Example 1 CRC16 and fixed byte input are shown below.

Figure 30-3. Example 1

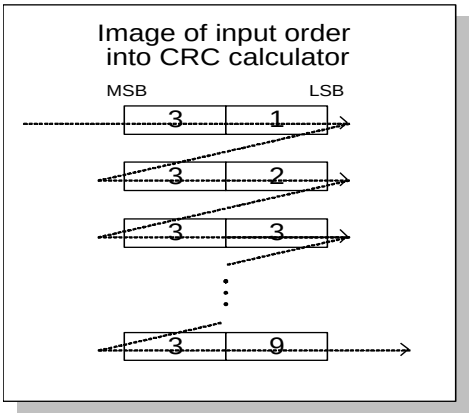
```
//****************************************
// CRC16 (CRC ITU-T)
// polynomial: 0x1021
// initial value: 0xFFFF
// CRCCR.CRC32:  0 // CRC16
// CRCCR.LTLEND: 0 // big endian
// CRCCR.LSBFST: 0 // MSB First
// CRCCR.CRCLTE: 0 // CRC big endian
// CRCCR.CRCLSF: 0 // CRC MSB First
// CRCCR.FXOR:   0 // CRC Final XOR off
//****************************************

//
// Example 1-1 (Byte-unit writing)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789"
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x29B1);
//
// Example 1-2 (CRC check)
//
// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789" + CRC
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);
B_WRITE (CRCIN, 0x29); // <-- CRC
B_WRITE (CRCIN, 0xB1); // <-- CRC

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);
```

(The following is assumed)

B_WRITE -- Byte writing
H_WRITE -- Half-word writing
W_WRITE -- Word writing

B_READ -- Byte reading
H_READ -- Half-word reading
W_READ -- Word reading

CRCCR -- Control register address
CRCINIT -- Initial value register address
CRCIN -- Input data register address
CRCR -- Current CRC register address



- Bytes and half words can be written into any position. In this example, data is written into +0 position continuously.
- When CRC16 is used, the CRC result is output in bit15 to bit0 for big-endian byte order and thus the address for H_READ8 (Half-word reading) is +2 in the example.

30.5.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths

Example 2 CRC16 and Mixture of Different Input Bit Widths are shown below.

Figure 30-4. Example 2

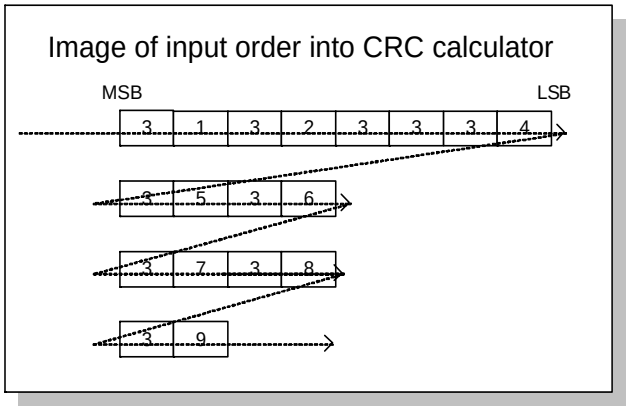
```
//************************************
// CRC16 (CRC ITU-T)
// polynomial:    0x1021
// initial value: 0xFFFF
// CRCCR.CRC32    0 // CRC16
// CRCCR.LTLEND:  0 // big endian
// CRCCR.LSBFST:  0 // MSB First
// CRCCR.CRCLTE:  0 // CRC big endian
// CRCCR.CRCLSF:  0 // CRC MSB First
// CRCCR.FXOR:    0 // CRC Final XOR off
//************************************

//
//  Example 2-1 (Mixture of writing size)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789"
W_WRITE (CRCIN, 0x31323334);
H_WRITE (CRCIN, 0x3536);
H_WRITE (CRCIN+2, 0x3738);
B_WRITE (CRCIN+3, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x29B1);

//
// Example 2-2 (CRC check)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789" + CRC
W_WRITE (CRCIN, 0x31313334);
W_WRITE (CRCIN, 0x35363738);
H_WRITE (CRCIN, 0x3929);// <-- CRC(0x29)
B_WRITE (CRCIN, 0xB1);// <-- CRC(0xB1)

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);
```

(The following is assumed)

B_WRITE -- Byte writing
H_WRITE -- Half-word writing
W_WRITE -- Word writing

B_READ -- Byte reading
H_READ -- Half-word reading
W_READ -- Word reading

CRCCR -- Control register address
CRCINIT -- Initial value register address
CRCIN -- Input data register address
CRCR -- Current CRC register address



- When the byte and bit orders are set correctly and the orders to input bits to the CRC calculator are the same, any writing width can be used.
- For example, there is a case that words are written basically and bytes or a half word is written if there is a fraction of 1, 2, or 3 bytes at the end.

30.5.6.3 Example 3 CRC32, Byte Order, Big-endian

Example 3 CRC32, the byte order and big-endian are shown below.

Figure 30-5. Example 3

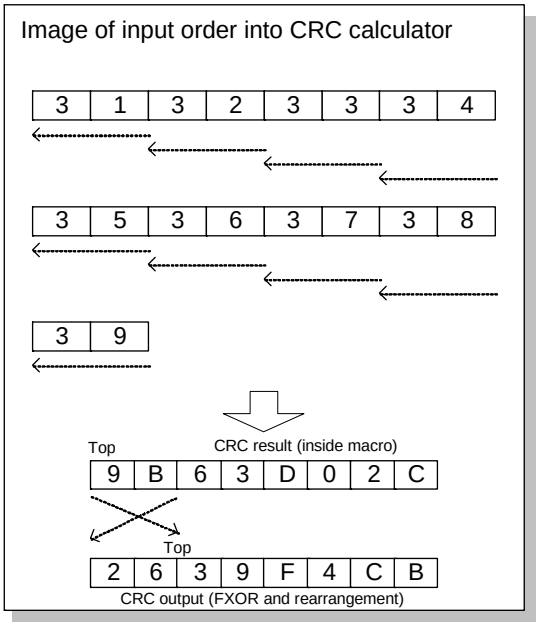
```
//**********************************
// CRC32 (IEEE-802.3)
// polynomial: 0x04C11DB7
// initial value: 0xFFFF_FFFF
// CRCCR.CRC32 1 // CRC32
// CRCCR.LTLEND: 0 // big endian
// CRCCR.LSBFST: 1 // LSB First
// CRCCR.CRCLTE: 0 // CRC big endian
// CRCCR.CRCLSF: 1 // CRC LSB First
// CRCCR.FXOR: 1 // CRC Final XOR on
//**********************************

//
// Example 3-1 (CRC32)
//

// Initialization
B_WRITE (CRCCR, 0x6B);

// data write "123456789"
W_WRITE (CRCIN, 0x31323334);
W_WRITE (CRCIN, 0x35363738);
B_WRITE (CRCIN, 0x39);

// read result
W_READ (CRCR, data);

// check CRC result
assert (data == 0x2639F4CB);
 // <- big endian & LSB First
```

(The following is assumed)

B_WRITE -- Byte writing
H_WRITE -- Half-word writing
W_WRITE -- Word writing

B_READ -- Byte reading
H_READ -- Half-word reading
W_READ -- Word reading

CRCCR -- Control register address
CRCINIT -- Initial value register address
CRCIN -- Input data register address
CRCR -- Current CRC register address



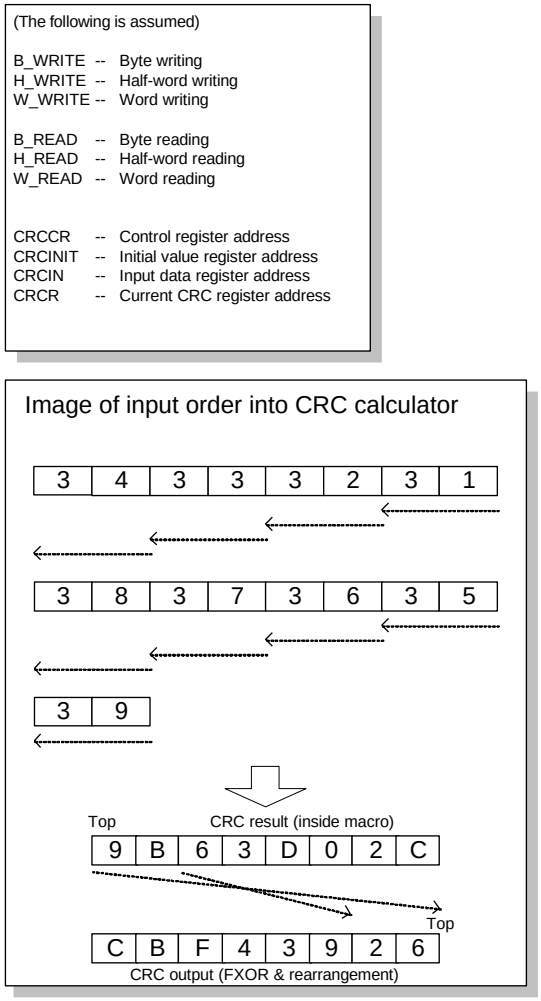
- When CRC32 (IEEE-802.3) is used, the bit order is LSB First. This CRC calculator supports both byte orders and the figure above shows the case for big endian.

30.5.6.4 Example 4 CRC32, Byte Order, Little-endian

Example 4 CRC32, the byte order and Little-endian are shown below.

Figure 30-6. Example 4

```
//*************************************
// CRC32 (IEEE-802.3)
// polynomial: 0x04C11DB7
// initial value: 0xFFFF_FFFF
// CRCCR.CRC32 1 // CRC32
// CRCCR.LTLEND: 1 // little endian
// CRCCR.LSBFST: 1 // LSB First
// CRCCR.CRCLTE: 1 // CRC little endian
// CRCCR.CRCLSF: 1 // CRC LSB First
// CRCCR.FXOR: 1 // CRC Final XOR on
//*************************************

//
// Example 4-1 (CRC32)
//

// Initialization
B_WRITE (CRCCR, 0x7F);

// data write "123456789"
W_WRITE (CRCIN, 0x34333231);
W_WRITE (CRCIN, 0x38373635);
B_WRITE (CRCIN, 0x39);

// read result
W_READ (CRCR, data);

// check result
assert (data == 0xCBF43926);
 // <- little endian & LSB First
```



- When CRC32 (IEEE-802.3) is used, the bit order is LSB First. This CRC calculator supports both byte orders and the figure above shows the case for little endian.
- When bit inversion for CRC results is not needed, the bit inversion for the current results can be canceled either by calculation through initialization using 0x3F, or setting of CRCCR.FXOR to 0 (Example: CRCCR=0x3E) after data entry.

31. RAMECC

This chapter explains the RAMECC function.

31.1 Overview

31.2 Features

31.3 Configuration

31.4 Registers

31.5 Operation

31.1 Overview

This section gives an overview of the RAMECC.

The function of the RAMECC is to correct a single-bit error and to detect a double-bit error for those data read from or written to RAM.

31.2 Features

This section explains features of the RAMECC.

Target RAM

- XBS RAM

  24 K bytes

RAMECC function

Errors up to double bits are detected for those data read from or written to RAM.
Moreover, if a single-bit error is detected it will be corrected.

Interrupt function

A double bit error is detected and RAM double bit error interrupt signal is generated.

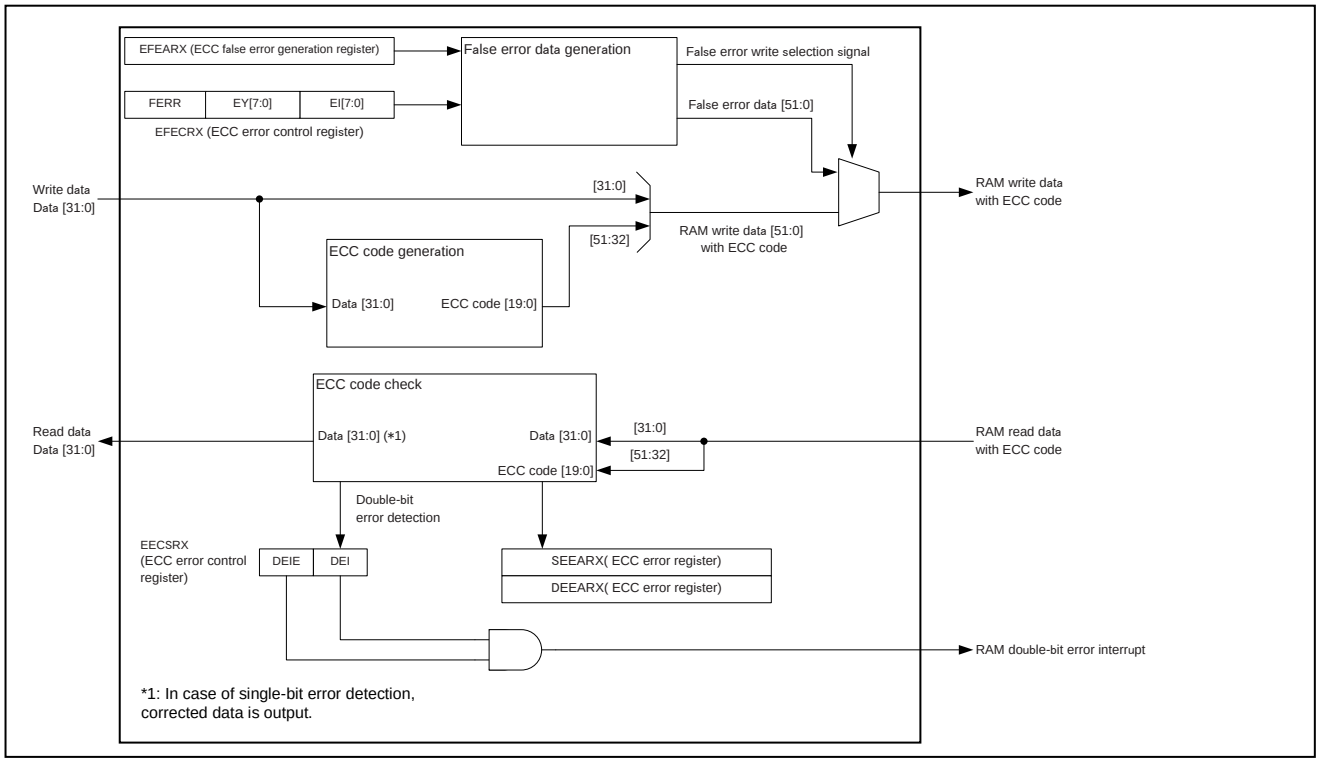
Test mode

A false error occurs for the software debugging.

31.3 Configuration

This section shows the configuration of the RAMECC.

Figure 31-1. Block Diagram of XBS RAM ECC Function (Configuration)



31.4 Registers

This section explains the registers of the RAMECC.

Table 31-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x2400	SEEARX		DEEARX		Single-bit ECC error address register XBS RAM Double-bit ECC error address register XBS RAM
0x2404	EECSRX	Reserved	EFEARX		ECC error control register XBS RAM ECC false error generation address register XBS RAM
0x2408	Reserved	EFECRX			ECC false error generation control register XBS RAM

31.4.1 Single-bit ECC Error Address Register XBS RAM: SEEARX

The bit configuration of single-bit ECC error address register XBS RAM is shown.

When the single-bit error correction is performed during the ECC check of XBS RAM, this register maintains the address at which it occurred.

SEEARX: Address 2400_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit0] D14 to D0: Single-bit error occurrence address bits

When the single-bit error correction is performed during the ECC check, these bits maintain the address at which it occurred.
If the event above is further detected when a value has already been set to these bits, the original value is maintained without overwriting these bits.

Note:

The address above is offset in words. Calculate the absolute address by adding the lower 2 bits to the offset address mentioned above, and then adding the base address of XBS RAM.
(Absolute address) = (0001_0000_H) + (Offset indicated by SEEARX + 2'b00)

31.4.2 Double-bit ECC Error Address Register XBS RAM: DEEARX

The bit configuration of double-bit ECC error address register XBS RAM is shown.

When the double-bit error detection is performed during the ECC check of XBS RAM, this register maintains the address at which it occurred.

DEEARX: Address 2402_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit0] D14 to D0: Double-bit error occurrence address bits

When the double-bit error detection is performed during the ECC check, these bits maintain the address at which it occurred.
If the event above is further detected when a value has already been set to these bits, the original value is maintained without overwriting these bits.

Note:

The address above is offset in words. Calculate the absolute address by adding the lower 2 bits to the offset address mentioned above, and then adding the base address of XBS RAM.
(Absolute address) = (0001_0000_H) + (Offset indicated by DEEARX + 2'b00)

31.4.3 ECC Error Control Register XBS RAM: EECSRX

The bit configuration of ECC error control register XBS RAM is shown.

During the ECC check of XBS RAM, this register maintains the status that indicates whether or not the single-bit error correction or the double-bit error detection has been performed and specifies whether or not to enable interrupts by the double-bit error detection.

EECSRX: Address 2404_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				DEIE	DEI	Reserved	SEI
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R(RM1),W	R/W0	R(RM1),W0

[bit7 to bit4] Reserved

Always write "0" to these bits.

[bit3] DEIE: Double-bit error factor interrupt enable bit

DEIE	Function
0	Disable interrupts.
1	Enable interrupts.

[bit2] DEI: Double-bit error occurrence bit

DEI	Read	Write
0	Double-bit error has not occurred.	Clear this bit.
1	Double-bit error has occurred.	No effect

[bit1] Reserved

Always write "0" to these bits.

[bit0] SEI: Single-bit error occurrence bit

SEI	Read	Write
0	Single-bit error has not occurred.	Clear this bit.
1	Single-bit error has occurred.	No effect

31.4.4 ECC False Error Generation Address Register XBS RAM: EFEARX

The bit configuration of the ECC false error generation address register XBS RAM is shown.

The ECC false error generation address register specifies the address where a false error of XBS RAM is generated.

EFEARX: Address 2406_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit0] D14 to D0: False error generation address setting bits

These bits set the address where false ECC error of XBS RAM is caused.
When EFECRX.FERR=1, write access to this address is generated, and an ECC error is generated by intentionally including an error in the data to be written, according to the settings of EFECRX.

Note:

The address above is offset in words. Calculate the absolute address by adding the lower 2 bits to the base address, and then adding the base address of XBS RAM.
(Absolute address) = (0001_0000_H) + (Offset set by EFEARX + 2'b00)

31.4.5 ECC False Error Generation Control Register XBS RAM: EFECRX

The bit configuration of the ECC false error generation control register XBS RAM is shown.

The ECC false error generation control register (EFECRX) specifies each false error by its byte position and its bit position where the false error is generated.

EFECRX: Address 2409_H (Access: Byte, Half-word, Word)

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							FERR
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EY7	EY6	EY5	EY4	EY3	EY2	EY1	EY0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EI7	EI6	EI5	EI4	EI3	EI2	EI1	EI0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit23 to bit17] Reserved

Always write "0" to these bits.

[bit16] FERR: False error generation enable bit

FERR	Function
0	False ECC error generation disable
1	False ECC error generation enable

This bit enables a false ECC error for XBS RAM.

"0": Prohibit a false ECC error. (Normal operation)
Also, writing "0" from software is ignored.

"1": Enable a false ECC error.

When this bit is set to "1", the following operation sequence is automatically performed.

1. Start writing data including an intentional error in the address specified by EFEARX following EY7 to EY0, EI7 to EI0.
2. Read the same address and detect ECC error.
3. Clear this bit to "0".

[bit15 to bit8] EY7 to EY0: False error generation byte setting bits

EY7 to EY0	Target Byte on RAM
EY0	RAM data[7:0]
EY1	RAM data[15:8]
EY2	RAM data[23:16]
EY3	RAM data[31:24]
EY4	RAM data[36:32]
EY5	RAM data[41:37]
EY6	RAM data[46:42]
EY7	RAM data[51:47]

These bits specify the byte position of the target that causes false ECC error for XBS RAM.
For example, when EY2 is filled with "1" and other false error generation byte setting bits are filled with "0", the target byte where a false error is generated is RAM data[23:16] only. In other bytes on the RAM, no false error is generated.
In addition to the foregoing, when both EY2 and EY3 are filled with "1" and others are filled with "0", the target byte where a false error is generated is RAM data[31:16].

[bit7 to bit0] EI7 to EI0: False error generation bit setting bits

EI7 to EI0	Target Bit on Byte
EI0	[0]
EI1	[1]
EI2	[2]
EI3	[3]
EI4	[4]
EI5	[5]
EI6	[6]
EI7	[7]

These bits specify the bit position of the target that causes false ECC error for XBS RAM.
For example, when both EY2 and EI4 are filled with "1", and others are filled with "0", the target bit where a false error is generated is RAM data[20].As a result, a single bit error can be corrected.
In addition to the foregoing, when EY2, EI4, and EI7 are filled with "1", and others are filled with "0", the target bits where a false error is generated are RAM data[23] and RAM data[20].As a result, a double bit error can be detected.
Moreover, when EY2, EY3, and EI4 are filled with "1", and others are filled with "0", the target bits where a false error is generated are RAM data[28] and RAM data[20].As a result, a single bit error can be corrected in each byte.

Note:

2408_H is reserved bit. The operations in half-word and word-access are as below.
The read value is always "0".
Always write "0".

31.5 Operation

This section explains operations.

31.5.1 RAMECC Function

31.5.2 Interrupt-related Register

31.5.3 Test Mode

31.5.4 Notes

31.5.1 RAMECC Function

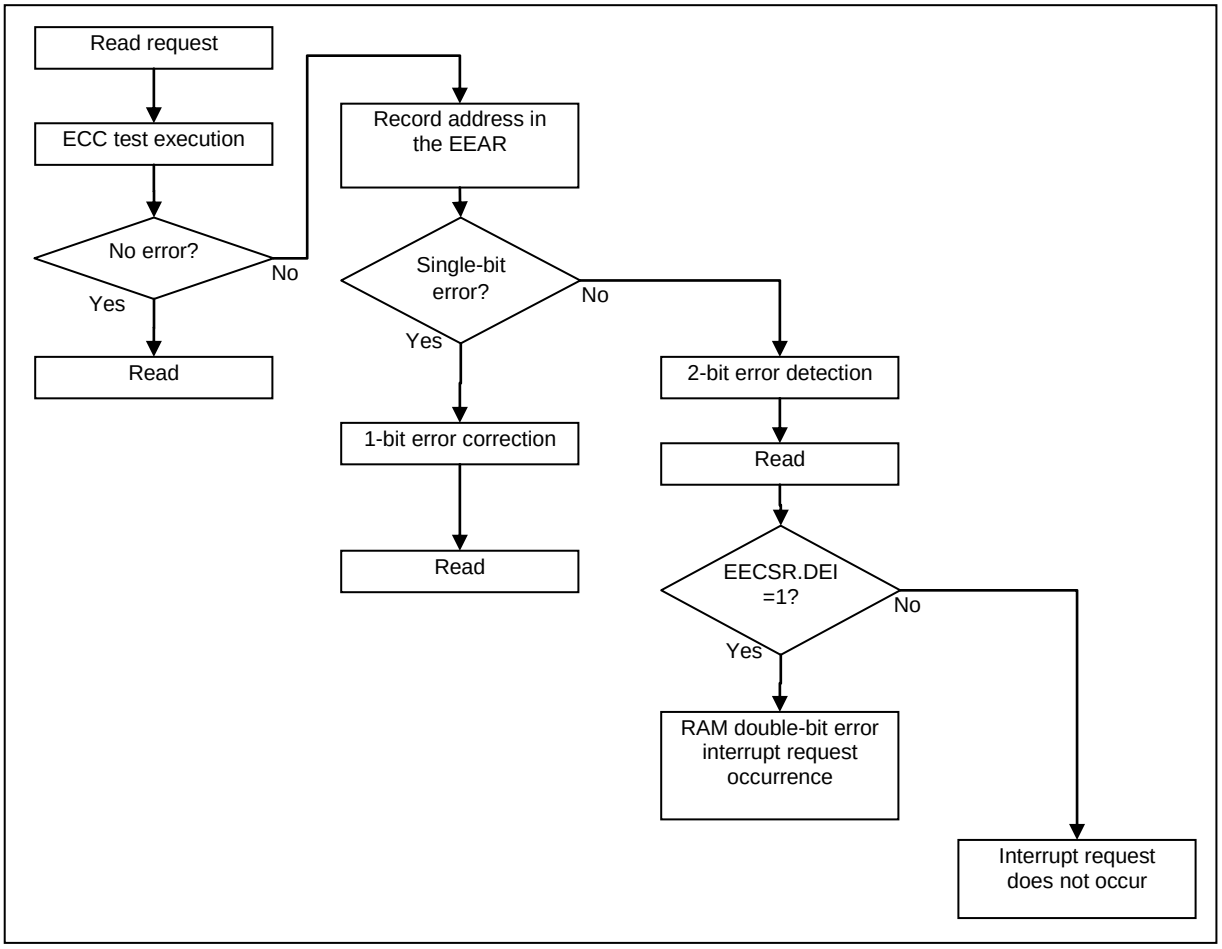
The RAMECC function is explained.

The RAMECC always functions (It, however, stops at RAM diagnosis).
When an error is detected, the address where the error occurred is held in the register EEAR. If another error is detected while the previous address is being held in the register EEAR, the EEAR register will not be overwritten. Thus the previous address is maintained.

ECC code matrix records redundant 5 bits as ECC code by byte units.

RAM data [51:0]	[7:0]	[15:8]	[23:16]	[31:24]	[36:32]	[41:37]	[46:42]	[51:47]
Data [31:0]	Data [7:0]	Data [15:8]	Data [23:16]	Data [31:24]	ECC [4:0]	ECC [9:5]	ECC [14:10]	ECC [19:15]

Flow chart of the operation is shown below.



31.5.2 Interrupt-related Register

This section explains the interrupt-related register.

Write "1" in the DEIE bit according to the usage in order to generate the interrupt, and set the RAMECC interrupt vector.

Interrupt Factor	Interrupt Vector	Interrupt Level
DEI (RAM double-bit error interrupt)	#15($000FFFC0_H$)	15(F_H) Fixed

See "Chapter: Interrupt Control (Interrupt Controller)" for details of the interrupt level and the interrupt vector.
Since the interrupt request flag (DEI) is not automatically cleared, clear the flag forcibly with software before returning from the interrupt processing. (Write "0" into the DEI bit).

31.5.3 Test Mode

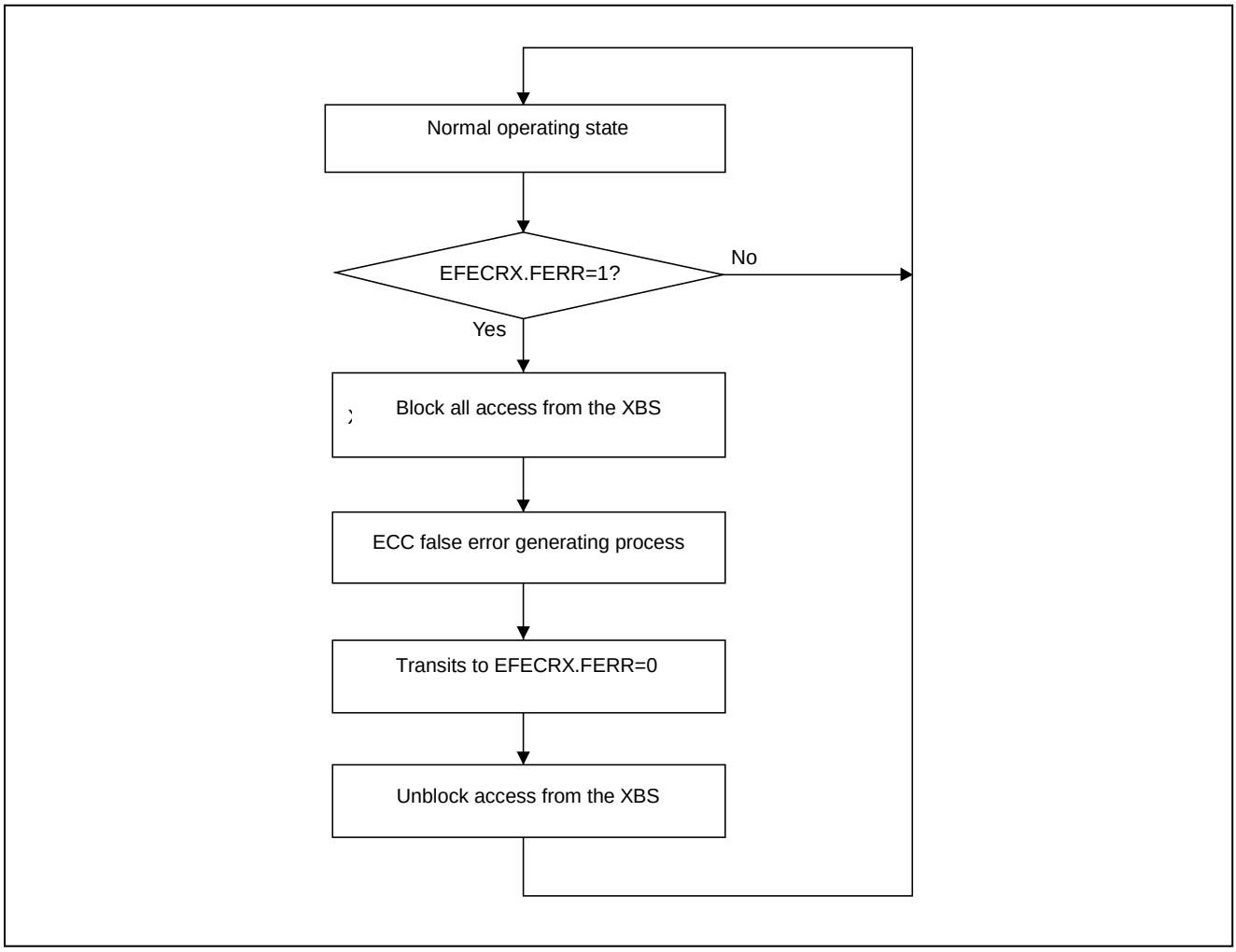
This section explains the test mode.

In this mode an ECC false error is generated in order to debug software.
The ECC false error of XBS RAM, is generated in accordance with the following procedures:

1. Specify the address where a false error is generated to the ESS false error generation address register (EFEARX).
2. Set bytes and bits in ECC false error generation control register XBS RAM (EFECRX).
   (a) Specify the byte position where a false error is generated to EFECRX.EY[7:0].
   (b) Specify the bit position where a false error is generated to EFECRX.EI[7:0].
3. Write "1" to the FERR bit of the ECC false error generation control register XBS RAM (EFECRX).

Those data including errors are intentionally written to the address specified with EFEARX, as per the contents of EY[7:0] and EI[7:0]. Then the CPU reads the data subsequently, detecting the false error.
The operation after "1" is written to the FERR bit will be performed automatically.

Flow chart of the operation is shown below.



31.5.4 Notes

This section explains notes.

A single bit error might be detected at the fault detection in 3 bits or more, and the correction operation not intended can start.

If XBS RAM is accessed while the ECC false error (a pseudo ECC error) of XBS RAM is generated, the access is blocked. Therefore, access XBS RAM after completing the process of the ECC false error (a pseudo ECC error).

32. Multi-Function Serial Interface

This chapter explains the multi-function serial interface.

32.1 Overview

32.2 Features

32.3 Configuration

32.4 Registers

32.5 Operation of UART

32.6 Operation of CSIO

32.7 Operation of LIN Interface (v2.1)

32.1 Overview

This section explains the overview of the multi-function serial interface.

This module provides, UART (Asynchronous Serial Interface), CSIO (SPI supported, Clock Synchronous Serial Interface), and LIN Interface LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)).

32.2 Features

This section explains features of the multi-function serial interface.

This product is equipped with 3-channel multi-function serial interface communication module. To use this device, you will select UART, CSIO, or LIN Interface (v2.1) using the serial mode register (SMR).

UART

UART (asynchronous serial interface) is the general-purpose serial data communication interface designed to communicate with external devices asynchronously (start-stop synchronization). It supports bidirectional communication function (normal mode), master/slave type communication function (multi-processor mode: both master and slave are supported). It is also equipped with FIFO for transmission/reception.

Item	Function
Data	■ Full-duplex double buffering (when FIFO is unused) ■ Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Serial input	Execute over-sampling to the bus clock for three times and determine the reception value by the majority of the sampling value.
Transfer format	Asynchronous
Baud rate	■ Dedicated baud rate generator (comprising 15-bit reload counter) ■ External clock input can be adjusted by the reload counter
Data length	5 to 9 bits (normal mode), 7, 8 bits (multi-processor mode)
Signaling system	NRZ (Non Return to Zero), Inverted NRZ
Start bit detection	■ Synchronize with the start bit falling edge (NRZ system) ■ Synchronize with the start bit rising edge (inverted NRZ system)
Reception error detection	■ Framing error ■ Overrun error ■ Parity error*
Timer feature	■ Employs 16-bit serial timer ■ Dividing ratio of operating clock is selectable (1/1 to 1/256)
Interrupt request	■ Reception interrupt (Reception completed, framing error, overrun error, parity error*) ■ Transmission interrupt (transmission data empty, transmission bus idle) ■ Transmission FIFO interrupt (when transmission FIFO is equal to or below interrupt trigger level or transmission FIFO is empty) ■ Both transmission and reception employ DMA function ■ Status interrupt (Serial timer interrupt)
Master/slave mode communication function (multi-processor mode)	1 (Master)-to-n (slave) communication is supported (both master and slave systems are supported)
FIFO option	■ Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) ■ Transmission FIFO and reception FIFO can be selected ■ Transmission data can be retransmitted ■ Reception FIFO interrupt timing can be modified by software ■ FIFO reset is supported independently
DMA transfer support	Transmission: Supported Reception: Supported Status: Not supported

* Parity error is for the normal mode only.

CSIO

CSIO (Clock Synchronous Serial Interface) is a general-purpose serial data communication interface for synchronous communication with external devices (SPI supported). It is also equipped with the FIFO for transmission/reception (64 bytes each).

Item	Function
Data buffer	■ Full-duplex double buffering (when FIFO is unused) ■ Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Transfer format	■ Clock synchronous (without start bit/stop bit) ■ Master/slave function ■ SPI supported (both master/slave mode supported)
Baud rate	■ Dedicated baud rate generator provided (comprising 15-bit reload counter, master mode) ■ An external clock can be entered. (Slave operation)
Data length	Can be changed to 5 to 16, 20, 24, 32 bits
Reception error detection	Overrun error
Interrupt request	■ Reception interrupt (reception completed, overrun error) ■ Transmission interrupt (transmission data empty, transmission bus idle, chip error interrupt) ■ Transmission FIFO interrupt (when transmission FIFO is equal to or below interrupt trigger level or transmission FIFO is empty) ■ Both transmission and reception employ DMA function * ■ Status interrupt (Serial timer interrupt)
Serial chip select	■ Ch.0,1,2: 2-ch control (single control, round control) ■ Variable setup/hold/deselect times can be set ■ Active level can be selected for each channel
Synchronous transmission feature	Synchronizes serial timer and is capable of automatic data transmission periodically
Timer feature	■ Employs 16-bit serial timer ■ Dividing ratio of operating clock is selectable (1/1 to 1/256)
Synchronous mode	Master or slave function
Pin access	Serial data output pin can be set to "1"
FIFO option	■ Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) ■ Transmission FIFO and reception FIFO can be selected ■ Transmission data can be retransmitted ■ Reception FIFO interrupt timing can be modified by software ■ FIFO reset is supported independently
DMA transfer support	Transmission: Supported Reception: Supported Status: Not supported

* When the access size of transmit/receive data is 16 bit and FIFO is not used, continuous transfer cannot be performed. To perform continuous transfer, set the access width of transmit/receive data to 32-bit access (SSR:AWC=1) or use FIFO.

LIN Interface (v2.1) (LIN Communication Control Interface (v2.1))

Manual Mode

LIN interface (v2.1) (LIN Communication Control Interface (v2.1)) provides functions to support LIN bus. It is also equipped with the FIFO for transmission/reception (64 bytes each).

Item	Function
Data buffer	■ Full-duplex double buffering (when FIFO is unused) ■ Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Serial input	Execute over-sampling for three times by the bus clock and determine the reception value by the majority of the sampling value.
Transfer mode	Asynchronous
Baud rate	■ Dedicated baud rate generator provided (comprising of 15-bit reload counter) ■ External clock input can be adjusted by the reload counter ■ Automatic baud rate adjustment with Sync Field reception
Data length	8 bits
Signaling system	NRZ (Non Return to Zero)
Start Bit Detection	Synchronize with the start bit falling edge
Reception error detection	■ Framing error ■ Overrun error
Interrupt request	■ Reception interrupt (Reception completed, framing error, overrun error) ■ Transmission interrupt (transmission data empty, transmission bus idle) ■ Status interrupt (LIN Break field detection, serial timer interrupt) ■ Interrupt request for ICU (LIN synch field detected: LSYN) ■ Transmission FIFO interrupt (when transmission FIFO is equal to or below interrupt trigger level or transmission FIFO is empty) ■ Both transmission and reception employ DMA function
Timer feature	■ Employs 16-bit serial timer ■ Dividing ratio of operating clock is selectable (1/1 to 1/256)
LIN bus option	■ LIN protocol revision 2.1 is supported. ■ Master device operation ■ Slave device operation ■ LIN Break field generation (can be changed to 13 to 16 bits) ■ LIN Break Delimiter generation (can be changed to 1 to 4 bits) ■ LIN Break field detection ■ Detection of start/stop edges for LIN synch field connected to input capture by input capture (See "Chapter: Input Capture.")
FIFO option	■ Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) ■ Transmission FIFO and reception FIFO can be selected ■ Transmission data can be retransmitted ■ Reception FIFO interrupt timing can be modified by software ■ FIFO reset is supported independently
DMA transfer support	■ Transmission: Supported ■ Reception: Supported ■ Status: Not supported

Assist Mode

LIN Interface (v2.1) (LIN Communication Control Interface (v2.1)) provides functions to support LIN bus. Automatic transmission/automatic detection of the header in the LIN communication is possible. It is also equipped with the FIFO for transmission/reception (64 bytes each).

Note: This supports master mode only. This cannot be used in slave mode.

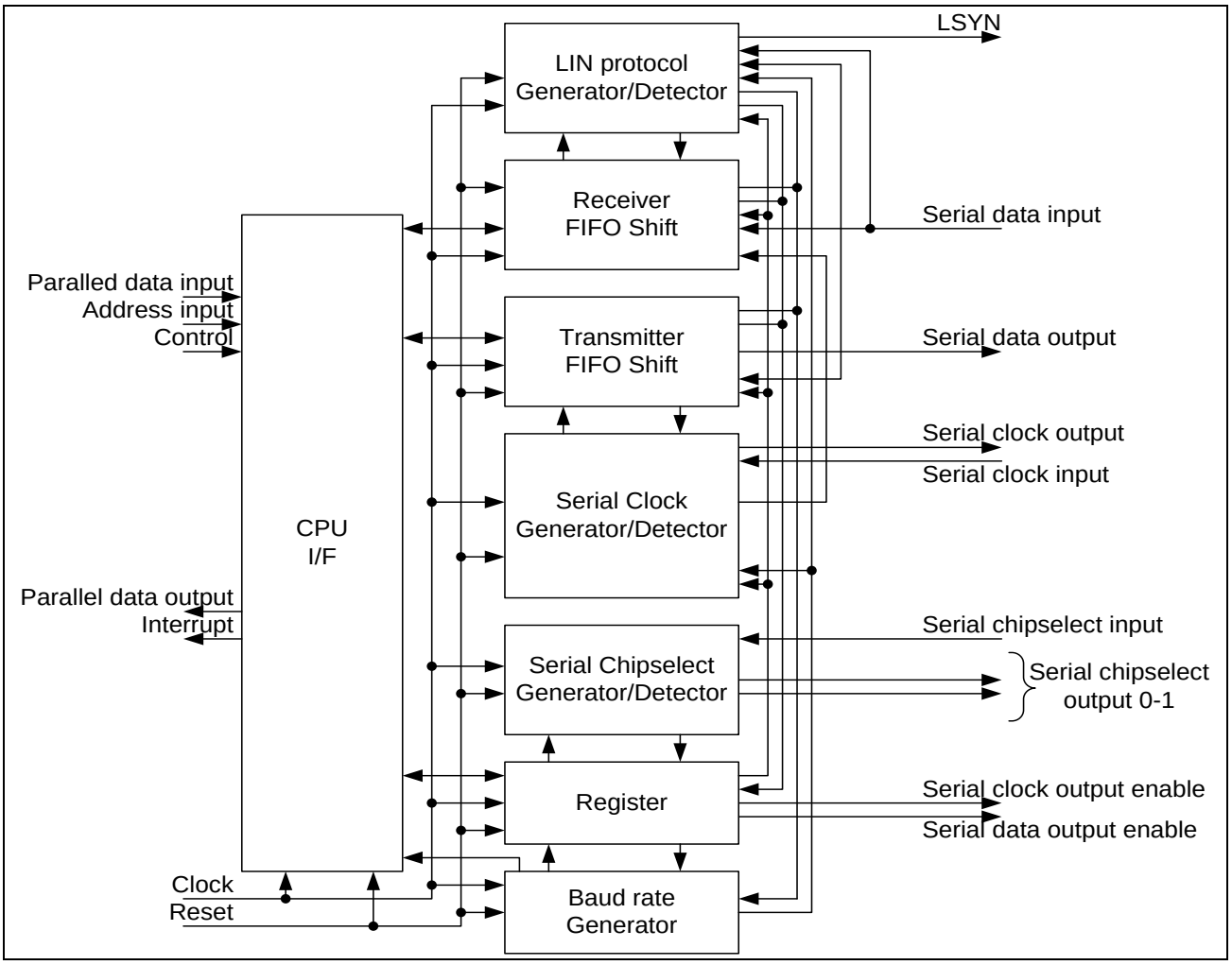
Item	Function
Data buffer	■ Full-duplex double buffering (when FIFO is unused) ■ Transmission/reception FIFO (64 bytes each) (when FIFO is used)
Serial input	Execute over-sampling for three times by the bus clock and determine the reception value by the majority of the sampling value.
Transfer mode	Asynchronous
Baud rate	■ Dedicated baud rate generator provided (comprising of 15-bit reload counter) ■ External clock input can be adjusted by the reload counter ■ Automatic baud rate adjustment with Sync Field reception
Data length	9 bits
Signaling system	NRZ (Non Return to Zero)
Start Bit Detection	Synchronize with the start bit falling edge
Reception error detection	<The error is detected by the self-check of the transmission side> ■ LIN bus error <The error is detected by the self-check of the transmission side and by the reception side > ■ Framing error ■ Overrun error ■ LIN ID parity error ■ LIN checksum error <The error is detected by the reception side of the automatic baud rate adjustment ■ prohibition> ■ LIN Sync Data error
Interrupt request	■ Transmission interrupt (1) Data transmission interrupt (Transmission data empty, transmission bus idle) (2) Transmission FIFO interrupt (When transmission FIFO is the interrupt threshold or less, or transmission FIFO is empty). ■ Reception interrupt (1) Data reception interrupt (Reception completed) (2) Reception FIFO interrupt (When reception FIFO is the interrupt threshold or more). (3) Various error interrupts (LIN bus error, LIN ID parity error, LIN Sync Data error, framing error, overrun error, and LIN checksum error) ■ Status interrupt (1) Automatic header completion interrupt (2) Sync Field detection interrupt (3) Checksum arithmetic operation completion interrupt ■ Both transmission and reception employ DMA function
Timer feature	■ Employs 16-bit serial timer ■ Dividing ratio of operating clock is selectable (1/1 to 1/256)
LIN bus option	■ LIN protocol revision 2.1 supported ■ Automatic transmission/reception of master/slave device headers (1) LIN Break Field generation (can be changed to 13 to 20-bit lengths) (2) LIN Break Delimiter generation (can be changed to 1 to 4-bit lengths) (3) Sync Field automatic generation and automatic check of data value (0x55) (4) ID Field parity value automatic generation and check (5) Check sum automatic generation/check (standard/extension supported) ■ Detection of start/stop edges for LIN sync Field connected to input capture by input capture (See "Chapter : Input Capture")
FIFO option	■ Transmission/reception FIFO equipped (transmission FIFO: 64 bytes, reception FIFO: 64 bytes) ■ Transmission FIFO and reception FIFO can be selected ■ Transmission data can be retransmitted ■ Reception FIFO interrupt timing can be modified by software ■ FIFO reset is supported independently

Item	Function
LIN communication test function	■ Serial communication test function ■ Pseudo trouble generation function (LIN bus error, LIN ID parity error, LIN checksum error, LIN Sync Data error, and framing error)

32.3 Configuration

This section explains configuration of the multi-function serial interface.

Figure 32-1. Block Diagram



32.4 Registers

This section explains registers of the multi-function serial interface.

Table of Base Addresses (Base_addr) and External Pins

Table 32-1. Table of Base Addresses (Base_addr) and External Pins

Channels	Base_addr	External pin			
		SCK	SOT	SIN	SCS
0	0x1750	SCK0	SOT0	SIN0	SCS00/SCS01
1	0x1778	SCK1	SOT1	SIN1	SCS10/SCS11
2	0x17A0	SCK2	SOT2	SIN2	SCS20/SCS21

Registers Map

Table 32-2 Registers Map

Address	Registers				Registers function
	+0	+1	+2	+3	
Base+ 00_H	[UART] SCRn [CSIO] SCRn [LIN] SCRn	[Common] SMRn	[UART] SSRn [CSIO] SSRn [LIN] SSRn	[UART] ESCRn [CSIO] ESCRn [LIN] ESCRn	[UART] Serial control register [CSIO] Serial control register [LIN] Serial control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN] Serial status register [UART] Extended communication control register [CSIO] Extended communication control register [LIN] Extended communication control register
Base+ 04_H	[UART] Reserved [CSIO] RDR1n/TDR1n [LIN] Reserved		[UART] RDR0n/TDR0n [CSIO] RDR0n/TDR0n [LIN] RDR0n/TDR0n		[CSIO] Transmission/receive data register [UART] Transmission/receive data register [CSIO] Transmission/receive data register [LIN] Transmission/receive data register
Base+ 08_H	[UART] SACSRn [CSIO] SACSRn [LIN] SACSRn		[UART] STMRn [CSIO] STMRn [LIN] STMRn		[UART] Serial aid control status register [CSIO] Serial aid control status register [LIN] Serial aid control status register [UART] Serial timer register [CSIO] Serial timer register [LIN] Serial timer register
Base+ $0C_H$	[UART] STMCRn [CSIO] STMCRn [LIN] STMCRn		[UART] Reserved [CSIO] SCSCRn [LIN] SFURn		[UART] Serial timer compare register [CSIO] Serial timer compare register [LIN] Serial timer compare register [CSIO] Serial chip select control register [LIN] Sync field upper limit register
Base+ 10_H	[UART] Reserved [CSIO] SCSTR3n [LIN] LAMSRn	[UART] Reserved [CSIO] SCSTR2n [LIN] LAMCRn	[UART] Reserved [CSIO] SCSTR1n [LIN] SFLR1n	[UART] Reserved [CSIO] SCSTR0n [LIN] SFLR0n	[CSIO] Serial chip select timing register [LIN] LIN assist mode status register [LIN] LIN assist mode control register [LIN] Sync field lower limit register
Base+ 14_H	Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Reserved [CSIO] SCSFR0n [LIN] Reserved	[CSIO] Serial chip select format register

Address	Registers				Registers function
	+0	+1	+2	+3	
Base+ 18_H	[UART] Reserved [CSIO] Reserved [LIN] LAMESRn	[UART] Reserved [CSIO] Reserved [LIN] LAMERTn	[UART] Reserved [CSIO] TBYTE1n [LIN] LAMIERn	[UART] Reserved [CSIO] TBYTE0n [LIN] LAMTIDn/ LAMRIDn	[CSIO] Transfer byte register [LIN] LIN assist mode error status register [LIN] LIN assist mode trouble examination register [LIN] LIN assist mode interrupt enable register [LIN] LIN assist mode transmission/reception ID register
Base+ $1C_H$	[UART] BGRn [CSIO] BGRn [LIN] BGRn		[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Reserved [CSIO] Reserved [LIN] Reserved	[UART] Baud rate generator register [CSIO] Baud rate generator register [LIN] Baud rate generator register
Base+ 20_H	[Common] FCR1n	[Common] FCR0n	[Common] FBYTEn		[Common] FIFO control register 1 [Common] FIFO control register 0 [Common] FIFO byte register
Base+ 24_H	[Common] FTICRn		Reserved		[Common] Transmission FIFO interrupt control register (FTICR)

32.4.1 Common Registers

Common registers are shown.

32.4.1.1 Serial Mode Register: SMR

This section explains the bit structure of the serial mode register.

This register selects the serial communication method. Bits 3 to 0 change their function according to the method selected (UART or CSIO).

SMRn (n=0 to 2): Address Base addr + 01_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
MD[2:0]			Reserved	SBL/ SCINV	BDS	SCKE/ (Reserved)	SOE	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W0	R/W	R/W (R/W0)	R/W (R/W0)	R/W (R/W0)	Attribute

[bit7 to bit5] MD[2:0] (MoDe): Operation mode

These bits set the communication method.

"000_B": Operating mode 0 (asynchronous normal mode) is set.
"001_B": Operating mode 1 (asynchronous multi-processor mode) is set.
"010_B": Operating mode 2 (CSIO mode) is set.
"011_B": Operating mode 3 (LIN communication mode) is set.

Notes:

- Settings other than those listed above are prohibited.
- Configure each register after setting the operation mode.
- [UART][CSIO][LIN] Before changing the operation mode, execute programmable clear (SCR:UPCL=1).

[bit4] Reserved bit

Always write "0" to this bit.

[bit3] SBL/SCINV (Stop Bit Length/Serial Clock INversion: Stop bit length selection bit/serial clock inversion bit.

[UART][LIN]
This bit configures the bit length of stop bit (frame end mark for transmission data):

When SBL="0" and ESCR.ESBL="0" are set: Stop bit is set to 1 bit.
When SBL="1" and ESCR.ESBL="0" are set: Stop bit is set to 2 bits.
When SBL="0" and ESCR.ESBL="1" are set: Stop bit is set to 3 bits.
When SBL="1" and ESCR.ESBL="1" are set: Stop bit is set to 4 bits.

Notes:

- When receiving, only the first bit of the stop bits will always be detected.
- This bit should be set when transmission is disabled (SCR:TXE=0).

[CSIO]
This bit inverts the serial clock format. When chip select is used in master mode (SCR: MS=0), this bit is used for serial chip select pin 0 communication.

When this bit is set to "0":

- Serial clock output mark level is set to "H".
- Transmission data is output in synchronization with a falling edge of the serial clock in the normal transfer while it is output in synchronization with a rising edge of the serial clock in the SPI transfer.
- Reception data is sampled at a rising edge of the serial clock in the normal transfer while it is sampled at a falling edge of the serial clock in the SPI transfer.

When this bit is set to "1":

- Serial clock output mark level is set to "L".
- Transmission data is output in synchronization with a rising edge of the serial clock in the normal transfer while it is output in synchronization with a falling edge of the serial clock in the SPI transfer.
- Reception data is sampled at a falling edge of the serial clock in the normal transfer while it is sampled at a rising edge of the serial clock in the SPI transfer.

Notes:

- Set this bit when transmission and reception are disabled (SCR:TXE=RXE=0).
- Set this bit when serial clock output is disabled (SCKE=0).
- After the SCINV bit is set, set reception enable (SCR: RXE=1).
- This bit is used in one of cases below.
  - ☐ When chip select pin is disabled (SCSCR:CSEN1-0="00"b)
  - ☐ While in slave mode (SCR:MS=1)
  - ☐ When data format of chip select is disabled (ESCR:CSFE=0)
  - ☐ When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active

[bit2] BDS (Bit Direction Select): Transfer direction selection bit

[LIN]
LIN must always write "0" to this bit.

[UART][CSIO]

This bit selects whether to transfer the transfer serial data from the least significant bit (LSB-first, BDS=0) or from the most significant bit (MSB-first, BDS=1).

- · When chip select is used with the master mode (SCR:MS=0), this bit is used for communication with serial chip select pin 0.

Notes:

- Set this bit when transmission and reception are disabled (SCR:TXE=RXE=0).
- [CSIO] This bit is used in one of cases below.
  - ☐ When chip select pin is disabled (SCSCR:CSEN1-0="00"b)
  - ☐ While in slave mode (SCR:MS=1)
  - ☐ When data format of chip select is disabled (ESCR:CSFE=0)
  - ☐ When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active

[bit1] SCKE (Serial ClocK Enable): Serial clock output enable bit

[CSIO]
This bit controls the I/O ports of a serial clock.
When this bit is set to "0": The SCK pin functions as a serial clock input pin.
When this bit is set to "1": It becomes a serial clock output pin allowing clock output while transmitting.

Notes:

- When you use a SCK pin as a serial clock input (SCKE=0), set a general purpose input/output port to an input port.
- After SCINV bit is set, set serial clock output enable (SCKE=1).
- When you use a SCK pin as a serial clock output, set the SCK pin as a peripheral output pin (set with EPFR). See "Chapter: I/O Ports" for how to make setups.

[UART][LIN]
This bit is a reserved bit. Always write "0" to this bit.

[bit0] SOE (Serial Output Enable): Serial data output enable bit

[UART][CSIO][LIN]
This bit enables/disables output of serial data.
When this bit is set to "0": Output of serial data stops.
When this bit is set to "1": The SOT pin functions as a serial data output pin (SOT).

Note:

Set a SOT pin as a peripheral output pin (set with EPFR). See "Chapter: I/O Ports" for how to make setups.

32.4.1.2 FIFO Control Register 1: FCR1

This section explains the bit structure of the FIFO control register 1.

FIFO control register (FCR1) is used for the test settings of FIFO, selection of transmission/reception FIFO, settings of transmission FIFO interrupt enable, and control of interrupt flag.

FCR1n (n=0 to 2): Address Base addr + 20_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved			FLSTE	FRIIE	FDRQ	FTIE	FSEL	
0	0	0	0	0	1	0	0	Initial value
R/W0	R/W0	R/W0	R/W	R/W	R(RM1),W	R/W	R/W	Attribute

[bit7 to bit5] Reserved bits

Always write "0" to these bits.

[bit4] FLSTE (Flag for data LoST detection Enable): Retransmission data lost detection enable bit

This bit enables FIFO retransmission data lost flag (FLST) detection.

When this bit is set to "0": FLST bit detection disabled
When this bit is set to "1": FLST bit detection enabled

Note:

When this bit is set to "1", set this bit to "1" after setting "1" to the FSET bit.

[bit3] FRIIE (Flag for Receive FIFO IdlE detection Enable): Reception FIFO idle detection enable bit

This bit configures whether or not to detect the reception idle state for 8-bit time or longer while the reception FIFO contains valid data. When reception interrupts are enabled (SCR:RIE=1), a reception interrupt will be generated once it detects reception idle state.

When this bit is set to "0": Reception idle state detection disabled
When this bit is set to "1": Reception idle state detection enabled

Note:

When the reception FIFO is used, set this bit to "1".

[bit2] FDRQ (transmit FIFO Data ReQuest): Transmission FIFO data request bit

It is a data request bit for transmission FIFO. When this bit is set to "1", it indicates that transmission data is being requested. When transmission FIFO interrupts are enabled (FTIE=1) at this time, a FIFO transmission interrupt request will be output.

FDRQ set condition

- When the transmission FIFO interrupt control is not used.
  - ☐ FBYTE (for transmission) = 0 (transmission FIFO is empty)
  - ☐ The transmission FIFO reset.
- When the transmission FIFO interrupt control is used.
  - ☐ FTICR setting value ≥ FTICR reading value (The storage data count of the transmission FIFO is the interrupt trigger level or less.)
  - ☐ The transmission FIFO reset.

FDRQ reset condition

- Writing "0" to this bit.
- If the transmission FIFO becomes full.

Notes:

- When transmission FIFO is enabled, writing "0" to this bit is valid.
- When FBYTE (for transmission) is "0", writing "0" to this bit is prohibited.
- When this bit is "0", the change in the FSEL bit is prohibited.
- When "1" is set to this bit, it does not affect the operation.
- If a read-modify-write instruction is executed, "1" will be read.
- If a transmit interrupt has occurred and you have written the required data in transmit FIFO, clear the interrupt request by setting the FIFO transmit data request bit (FCR1:FDRQ) to "0".
- [LIN] When it is setting value or less, writing "0" to this bit is prohibited.

[bit1] FTIE (Flag for Transmit Interrupt Enable): Transmission FIFO interrupt enable bit

It is an interrupt enable bit for transmission FIFO. If you set this bit to "1", an interrupt will be generated when the FDRQ bit is "1".

[bit0] FSEL (FIFO SELect): FIFO selection bit

This bit is used to select transmission/reception FIFO.

When this bit is set to "0", FIFO1 is assigned as the transmission FIFO, and FIFO2, the reception FIFO.
When this bit is set to "1", FIFO2 is assigned as the transmission FIFO, and FIFO1, the reception FIFO.

Notes:

- This bit will not be cleared by FIFO reset (FCL2, FCL1=1).
- When you change this bit, disable the FIFO operation (FCR0:FE2, FE1=0) first.
- If FDRQ="0", changing this bit is prohibited.

32.4.1.3 FIFO Control Register 0: FCR0

This section explains the bit structure of FIFO control register 0.

FIFO control register 0 (FCR0) is used to enable/disable FIFO operation, reset FIFO, save read pointer, and configure retransmission.

FCR0n (n=0 to 2): Address Base addr + 21_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1	
0	0	0	0	0	0	0	0	Initial value
R/W0	R,WX	R/W	R0,W	R0,W	R0,W	R/W	R/W	Attribute

[bit7] Reserved bit

This bit must always be written to "0".

[bit6] FLST (FIFO data LoST): FIFO retransmission data lost flag bit

This bit indicates that the retransmission data of transmission FIFO has been lost.

FLST set condition

- When you write (overwrite) FIFO while the FLSTE bit of FIFO control register 1 (FCR1) is "1" and the read pointers saved by the FSET bit matches the write pointer of transmission FIFO

FLST reset condition

- FIFO reset (writing "1" to FCL)
- Writing "1" to the FSET bit

If this bit is set to"1", it will overwrite the data indicated by the read pointer saved by the FSET bit. As a result, you will not be able to configure the retransmission by the FLD bit even when an error occurs. To execute a retransmission while this bit is set to "1", reset FIFO and write data to FIFO once again.

[bit5] FLD (FIFO pointer reLoaD bit) FIFO pointer reload bit

This bit reloads the data saved by the FSET bit at transmission FIFO to the read pointer. This bit is used for a retransmission in case that a communication error occurs. Once the retransmission setting has completed, this bit will be cleared to "0".

Notes:

- Do not write any other than FIFO reset while this bit is set to "1" since a reload to the read pointer is in progress.
- During the FIFO enable state or while a transmission is in progress, writing "1" to this bit is prohibited.
- [UART] [CSIO] [LIN] This bit must be set to "1" after SCR:TIE bit and SCR:TBIE bit are set to "0" and set SCR:TIE bit and SCR:TBIE bit to "1" after transmission FIFO is enabled.

[bit4] FSET (FIFO pointer SET): FIFO pointer save bit

This bit is used to save read pointer of transmission FIFO. If you save read pointer prior to communication, you will be able to retransmit while the FLST bit is "0" in case that a communication error occurs.

If this bit is set to "1": Save the current read pointer value.
If this bit is set to "0": No effect.

Note:

Set this bit to "1" when the transmission byte count (FBYTE) is 0.

[bit3] FCL2 (FIFO Clear 2): FIFO2 reset bit

This bit resets FIFO2.
When this bit is set to "1", it initializes the internal state of FIFO2.
Only the FCR0:FLST bit will be initialized while other bits of FCR1/0 register are retained.

Notes:

- Execute FIFO2 reset after disabling transmission/reception.
- Execute after clearing the transmission FIFO interrupt enable bit to "0".
- Valid data count of FBYTE2 register will be "0".

[bit2] FCL1 (FIFO Clear 1): FIFO1 reset bit

This bit resets FIFO1.
When this bit is set to "1", it initializes the internal state of FIFO1.
Only the FCR0:FLST bit will be initialized while other bits of FCR1/0 register are retained.

Notes:

- Execute FIFO1 reset after disabling transmission/reception.
- Execute after clearing the transmission FIFO interrupt enable bit to "0".
- Valid data count of FBYTE1 register will be "0".

[bit1] FE2 (FIFO Enable 2): FIFO2 operation enable bit

This bit enables/disables operation of FIFO2.

- To use FIFO2, set this bit to "1".
- When FIFO2 is selected as reception FIFO by the FSEL bit, this bit is cleared to "0" if a reception error occurs. As long as the reception error is not cleared, you will not be able to set this bit to "1".
- When FIFO2 is used as transmission FIFO, if the transmission buffer is empty (SSR:TDRE="1"), or when FIFO2 is used as reception FIFO, if the reception buffer is empty (SSR:RDRF="0"), set "1" or "0" to this bit.
- Even if you have FIFO2 disabled, the state of FIFO2 will be retained.
- [UART] [CSIO] When any data is present in FIFO2 and it is enabled for transmission (SCR:TXE=1) after FIFO2 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will immediately be started. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit.
- [CSIO] When you use FIFO2 as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "0" when reception buffer is empty (SSR:RDRF="0") and no valid data is present in the reception FIFO (FBYTE2=0).
- [CSIO] When you use FIFO2 as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "1" when reception buffer is empty (SSR:RDRF="0").
- [LIN] When FIFO2 is set for transmission FIFO, this bit is set to "1" while any data is present in FIFO2 and LIN interface (v2.1) transmission is enabled (TXE=1), transmission will immediately be started. In this case, after TIE bit and TBIE bit are set to "0", set "1" to this bit, then TIE bit and TBIE bit.

[bit0] FE1 (FIFO Enable 1) FIFO1 operation enable bit

This bit enables/disables operation of FIFO1.

- To use FIFO1, set this bit to "1".
- When FIFO2 is selected as reception FIFO by the FSEL bit, this bit is cleared to "0" if a reception error occurs. As long as the reception error is not cleared, you will not be able to set this bit to "1".
- When FIFO2 is used as transmission FIFO, if the transmission buffer is empty (SSR:TDRE="1"), or when FIFO2 is used as reception FIFO, if the reception buffer is empty (SSR:RDRF="0"), set "1" or "0" to this bit.
- Even if you have FIFO1 disabled, the state of FIFO1 will be retained.
- [UART] [CSIO] When any data is present in FIFO1 and it is enabled for transmission (SCR:TXE=1) after FIFO1 is set for transmission FIFO (FCR1:FSEL=1) and this bit is set to "1", transmission will immediately be started. In this case, after SCR:TIE bit and SCR:TBIE bit are set to "0", set "1" to this bit, then SCR:TIE bit and SCR:TBIE bit.
- [CSIO] When you use FIFO1 as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "0" when reception buffer is empty (SSR:RDRF="0") and no valid data is present in the reception FIFO (FBYTE2=0).
- [CSIO] When you use as a reception FIFO, after reception is disabled (SCR:RXE=0), set this bit to "1" when reception buffer is empty (SSR:RDRF="0").
- [LIN] When FIFO1 is set for transmission FIFO, this bit is set to "1" while any data is present in FIFO1 and LIN interface (v2.1) transmission is enabled (TXE=1), transmission will immediately be started. In this case, after TIE bit and TBIE bit are set to "0", set "1" to this bit, then TIE bit and TBIE bit.

32.4.1.4 FIFO BYTE Register: FBYTE

This section explains the bit structure of the FIFO byte register.

This register has different functions for reading and writing.
For reading, FIFO byte register (FBYTE) shows valid data count of FIFO.
For writing, you will be able to configure whether to generate a reception interrupt when the reception FIFO receives the specified data count.

FBYTEn (n=0 to 2): Address Base addr + 22_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
FBYTE2[7:0]								
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
FBYTE1[7:0]								
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

[bit15 to bit8] FBYTE2 (FIFO BYTE 2): FIFO2 data count display bits

[bit7 to bit0] FBYTE1 (FIFO BYTE 1): FIFO1 data count display bits

The FBYTE register indicates valid data count written to or received at FIFO. The table below shows the details of FCR1:FSEL bit settings.

FSEL	FIFO selection	Data count display
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 of FBYTE transfer count is 08_H.
- Set the data count at which you want to generate a reception interrupt flag with FBYTE for reception FIFO. If the specified transfer count and data count display of FBYTE register match, the interrupt flag (SSR:RDRF) will be set to "1".
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (SSR:RDRF) will be set to "1".
  - ☐ Reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1"
  - ☐ Data count contained in the reception FIFO does not reach the transfer count
- If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.
- [CSIO] To receive data in the master operation mode (master reception), clear the SCR:TIE bit and SCR:TBIE bit to "0", set the reception data count at the FBYTE register of transmission FIFO, and write "0" to the FCR1:FDRQ bit. Then, serial clocks for the volume of data configured will be output when the SCR:TXE bit is "1", which allows you to receive the data volume you have configured. To set the SCR:TIE bit and the SCR:TBIE bit to "1", set them to 1 after FCR1:FDRQ changes to "1".
- [CSIO] When transmission data is written to TDR once, transmission FIFO's FBYTE will be incremented by +1. When SSR:AWC=0 and the data length is 20, 24, 32, a transmission data writing to TDR must be separated to 2 times. The transmission FIFO's FBYTE will be incremented by +2.

- [CSIO] When reception data is read from RDR once, reception FIFO's FBYTE will be decremented by 1. When SSR:AWC=0 and the data length is 20, 24, 32, a reception data read from RDR must be separated to 2 times. The reception FIFO's FBYTE will be decremented by 2.

Notes:

- [UART] [LIN] Set FBYTE register of the transmission FIFO to "8'h00".
- [UART] [LIN] Disable reception before making any change.
- [CSIO] Other than the case of receiving data in the master operation mode, set FBYTE register of the transmission FIFO to "8'h00".
- [CSIO] When you configure the transmission data count for data reception in the master operation mode, make sure that the transmission FIFO is empty and the SCR:TIE and SSR:TBIE bits are "0".
- [CSIO] When you disable reception (SCR:RXE=0) while data is being received in the master operation mode, you will need to disable the transmission FIFO before disabling the transmission/reception FIFO.
- [CSIO] Make any change to reception FIFO's FBYTE after disabled reception.
- [LIN] After setting FIFO select bit (FCR1:FSEL), set FIFO byte register (FBYTE).
- [LIN] FIFO select bit (FCR1:FSEL) and FIFO byte register (FBYTE) cannot be set at the same time.
- [LIN] For FIFO data count display during transmission, a value with 1 subtracted from the number of writing for transmission data is displayed as a valid data count. This is because, if transmission data is written while data which has not been transmitted yet to the TDR register exists, the data is stored in transmission FIFO. After the data in the TDR register is transmitted, the data which has not been transmitted in transmission FIFO is transmitted to the TDR register.
- [LIN] For FIFO data count display during reception, the number of data which is received by reception FIFO and which has not been read yet is displayed. The data which has been received by the RDR register is not included.
- [Common] Data configured at FBYTE of the reception FIFO should be "1" or greater.
- [Common] This register cannot use the read-modify-write instruction.
- [Common] Settings that go over the FIFO capacity are prohibited.

32.4.1.5 Transmission FIFO Interrupt Control Register: FTICR

This section explains the bit structure of the transmission FIFO interrupt control register.

Transmission FIFO interrupt control register (FTICR) configures the interrupt by the transmission effective data count of the FIFO.

FTICRn (n=0 to 2): Address Base addr + 24_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
FTICR2[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
FTICR1[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit8] FTICR2: FIFO2 data count display bits

[bit7 to bit0] FTICR1: FIFO1 data count display bits

The FTICR register sets the interrupt trigger level by the effective data count of the transmission FIFO (residual quantity). The table below shows the details of FCR1:FSEL bit settings.

FSEL	Transmission FIFO selection	Transmission FIFO interrupt control register
0	FIFO1	FTICR1
1	FIFO2	FTICR2

- The initial values of the effective data count that generates the interrupt of the FTICR register are 0x00.
- FTICR register sets the data count of transmission interrupt generation to FTICR of transmission FIFO. When the display of set number of data count and effective data count of transmission FIFO (FBYTE) matches or becomes small, interrupt flag (FDRQ) is set to "1".
- The effective data count to transmission FIFO is displayed.

FTICR2, FTICR1: FIFIO2 data count display bits, FIFO1 data count display bits

Write	The effective data count that generates the interrupt is set.
Read	The effective data count is read.

Notes:

- The setting that exceeds the capacity of FIFO is prohibited.
- The set value cannot be read.
- This register cannot use the read-modify-write instruction.

32.4.2 Registers for UART

Registers for UART are shown.

32.4.2.1 Serial Control Register: SCR

This section explains the bit structure of the serial control register.

The serial control register (SCR) allows you to disable/enable transmission and reception, disable/enable transmission/reception interrupts, disable/enable transmission bus idle interrupts, and reset UART.

SCRn (n=0 to 2): Address Base addr + 00_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
UPCL	Reserved	Reserved	RIE	TIE	TBIE	RXE	TXE	
0	-	-	0	0	0	0	0	Initial value
R0,W	RX,WX	RX,WX	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	UPCL: Programmable clear bit	This bit initializes the internal state of UART. When this bit is set to "1": ■ Directly reset UART (software reset). In this case, the register settings will be maintained. Note that any active transmission or reception will be cut off immediately. ■ Baud rate generator restarts by reloading the setting value of the BGR1/0 register. ■ All the transmission and reception interrupt factors (SSR:PE,FRE,ORE,RDRF,TDRE,TBI,TINT) are initialized(0000110_B). When this bit is set to "0": No effect. A read always results in "0". Notes: ■ Execute a programmable clear after disabling interrupts. ■ When using FIFO, disable FIFO (FCR0:FE2,FE1=0) before you execute a programmable clear. ■ Even if programmable clear is executed (SSR:UPCL=1), the value of serial timer register (STMR) is not cleared.
bit6, bit5	Reserved bits	Read: The value is undefined. Write: No effect on operation.
bit4	RIE: Reception interrupt enable bit	■ This bit enables or disables the output of reception interrupt request to the CPU. ■ When the RIE bit and reception data flag bit (SSR:RDRF) are set to "1", or any of the error flag bit (SSR:PE, ORE, FRE) is set to "1", a reception interrupt request will be output.
bit3	TIE: Transmission interrupt enable bit	■ This bit enables or disables the output of transmission interrupt request to the CPU. ■ When the TIE bit and the SSR:TDRE bit are set to "1", a transmission interrupt request will be output.
bit2	TBIE: Transmission bus idle interrupt enable bit	■ This bit enables or disables the output of transmission bus idle interrupt request to the CPU. ■ When the TBIE bit and TBI bit are set to "1", a transmission bus idle interrupt request will be output.

Bit name		Function
bit1	RXE: Reception enable bit	This bit enables/disables the reception of UART. ■ If this bit is set to "0", reception is disabled. ■ If this bit is set to "1", reception is enabled. **Notes:** ■ Even when you enable reception (RXE=1), UART does not start the reception until a falling edge of the start bit (in the case of NRZ format (ESCR:INV=0)) is input. (In the case of inverted NRZ format (ESCR:INV=1), UART does not start the reception until a rising edge is input.) ■ If you disable reception (RXE=0) while a reception is in progress, it immediately stops the reception.
bit0	TXE: Transmission enable bit	This bit enables/disables the transmission of UART. ■ If this bit is set to "0", transmission is disabled. ■ If this bit is set to "1", transmission is enabled. **Note:** ■ If you disable transmission (TXE=0) while a transmission is in progress, it immediately stops the transmission.

32.4.2.2 Serial Status Register: SSR

This section explains the bit structure of the serial status register.

The serial status register (SSR) checks the status of transmission/reception and the reception error flag, and clears the reception error flag.

SSRn (n=0 to 2): Address Base addr + 02_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
REC	Reserved	PE	FRE	ORE	RDRF	TDRE	TBI	
0	-	0	0	0	0	1	1	Initial value
R0,W	RX,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

Bit name		Function
bit7	REC: Reception error flag clear bit	This bit clears the PE, FRE, ORE flags of the serial status register (SSR). ■ To clear an error flag, write "1" to this bit. ■ Writing "0" does not affect anything. A read always results in "0".
bit6	Reserved bit	Read: The value is indefinite. Write: No effect on operation.
bit5	PE: Parity error flag bit (Functions only in the operation mode 0)	"0" Read: No parity error "1" Read: There is a parity error ■ If a parity error occurs while a reception is in progress (ESCR:PEN=1), this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). ■ When the PE bit and the SCR:RIE bit are set to "1", a reception interrupt request will be output. ■ If this flag is set, data contained in the receive data register (RDR) becomes invalid. ■ When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.
bit4	FRE: Framing error flag bit	"0" Read: No framing error "1" Read: There is a framing error ■ If a framing error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). ■ When the FRE bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. ■ If this flag is set, data contained in the receive data register (RDR) becomes invalid. ■ When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.

Bit name		Function
bit3	ORE: Overrun error flag bit	"0" Read: No overrun error. "1" Read: There is an overrun error. ■ If an overrun error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). ■ When the ORE bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. ■ If this flag is set, data contained in the receive data register (RDR) becomes invalid. ■ When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.
bit2	RDRF: Reception data full flag bit	"0" Read: Receive data register (RDR) is empty "1" Read: The receive data register (RDR) contains data. ■ The flag indicates the state of the receive data register (RDR). ■ When received data is loaded in the RDR, this flag will be set to "1" and when RDR is read out, it will be cleared to "0". ■ When the RDRF bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. ■ While using reception FIFO, the RDRF will be set to "1" once the reception FIFO has received the specified number of data sets. ■ In the case where all the conditions below are met while using reception FIFO, when reception idle continues for more than 8 baud rate clocks, RDRF will be set to "1". ☐ Reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1" ☐ The reception FIFO contains data without receiving the specified number of data sets If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. ■ While using reception FIFO, the bit will be cleared to "0" once the reception FIFO becomes empty.
bit1	TDRE: Transmission data empty flag bit	"0" Read: Transmit data register (TDR) contains data. "1" Read: Transmit data register is empty. ■ The flag indicates the state of the transmit data register (TDR). ■ When a transmit data is written to TDR, this flag becomes "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmit shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. ■ When the TDRE bit and the SCR:TIE bit are set to "1", a transmission interrupt request will be output. ■ When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". ■ For details of the timing of setting/resetting the TDRE bit while using transmission FIFO, see "32.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing."

Bit name		Function
bit0	TBI: Transmission bus idle flag bit	"0" Read: Transmission is in progress "1" Read: No transmission is in progress ■ This bit indicates that UART has no transmission in progress. ■ When transmission data has been written to the transmit data register (TDR), this bit will become "0". ■ When the transmit data register is empty (TDRE=1) and no transmission is in progress, this bit will be set to "1". ■ When you set "1" to the UPCL bit of the serial control register (SCR), the TBI bit will be set to "1". ■ When this bit is "1" and transmission bus idle interrupts are enabled (SCR:TBIE=1), a transmission interrupt request will be output.

32.4.2.3 Extended Serial Control Register: ESCR

This section explains the bit structure of the extended serial control register.

The extended communication control register (ESCR) configures the data length of transmission/reception, enables/disables the parity bit, selects a parity bit, inverts the serial data format, and selects the length of stop bit.

ESCRn (n=0 to 2): Address Base addr + 03_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	ESBL	INV	PEN	P	L[2:0]			
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	Reserved bit	Always write "0" to this bit.
bit6	ESBL: Extended stop bit length select bit	This bit configures the bit length of stop bit (frame end mark for transmission data). When SBL="0" and ESCR:ESBL="0" are set: Stop bit is set to 1 bit. When SBL="1" and ESCR:ESBL="0" are set: Stop bit is set to 2 bits. When SBL="0" and ESCR:ESBL="1" are set: Stop bit is set to 3 bits. When SBL="1" and ESCR:ESBL="1" are set: Stop bit is set to 4 bits. **Notes:** ■ When receiving, only the first bit of the stop bits will always be detected. ■ This bit should be set when transmission is disabled (TXE=0).
bit5	INV: Inverted serial data format bit	This bit selects the serial data format to be either NRZ format or inverted NRZ format. ■ When this bit is set to "0": NRZ format is set. ■ When this bit is set to "1": Inverted NRZ format is set.
bit4	PEN: Parity enable bit (Functions only in the operation mode 0)	This bit configures whether to enable addition (transmission) and detection (reception) of the parity bit. ■ When this bit is set to "0", no parity bit will be added. ■ When this bit is set to "1", a parity bit will be added. **Note:** ■ In operation mode 1, this bit will be fixed to "0"internally.
bit3	P: Parity selection bit (Functions only in the operation mode 0)	When parity is enabled (ESCR:PEN=1), this bit selects odd parity"1" or even parity "0". ■ When this bit is set to "0": Selects even parity ■ When this bit is set to "1": Selects odd parity
bit2, bit1, bit0	L2, L1, L0: Data length select bits	These bits specify the data length of transmission/reception data. ■ "000_B": Data length will be set to 8 bits. ■ "001_B": Data length will be set to 5 bits. ■ "010_B": Data length will be set to 6 bits. ■ "011_B": Data length will be set to 7 bits. ■ "100_B": Data length will be set to 9 bits. **Notes:** ■ Settings other than those shown above are prohibited. ■ In operation mode 1, set the data length to 7 or 8 bits. The other settings are prohibited.

32.4.2.4 Receive Data Register/Transmit Data Register: RDR/TDR

This section explains the bit structure of the receive data register/transmit data register.

The receive data register and transmit data register are located within the same addresses. When you use the address to read, it functions as the receive data register, and when you use the address to write, it functions as the transmit data register. When FIFO is enabled, the address of RDR/TDR will be the address for reading/writing FIFO.

Read

RDR0n (n=0 to 2): Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved							D8	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

The receive data register (RDR) is a 9-bit data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN pin) are converted in the shift register and stored in the receive data register (RDR).
- Depending on the data length, "0" is inserted in the upper bit 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 is the reception data bit)

- When the received data is stored in the receive data register (RDR), the reception data full flag bit (SSR:RDRF) will be set to "1". When reception interrupts are enabled (SSR:RIE=1), a reception interrupt request will be generated.
- The receive data register (RDR) should be read out when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) will be automatically cleared to "0" when the receive data register (RDR) has been read out.
- In case a reception error occurs (SSR: PE, ORE or FRE is "1"), data in the receive data register (RDR) will become invalid.
- In operation mode 1 (multi-processor mode), the operation will be 7-bit or 8-bit long. The AD bit received will be stored at the D8 bit.
- For the 9-bit long transfer and in operation mode 1, RDR will be read in 16-bit access mode.

Notes:

- When using reception FIFO, if received data in the reception FIFO reaches specified number, "1" will be set to SSR:RDRF.
- When using reception FIFO, if the reception FIFO becomes empty, SSR:RDRF will be cleared to "0".
- If a reception error occurs (SSR: PE, ORE, or FRE is "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, reception data will not be stored in the reception FIFO.

Write

TDR0n (n=0 to 2): Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
-							D8	
-	-	-	-	-	-	-	1	Initial value
RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

The transmit data register (TDR) is the 9-bit data buffer register for sending serial data.

- When transmit operations are enabled (SCR:TXE=1), if transmission data is written to the transmit data register (TDR), the transmission data is transferred to the transmit shift register and converted to serial data, then output from the serial data output pin (SOT pin).
- Depending on the data length, data will be invalidated from the upper bit 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	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

- Transmission data empty flag (SSR:TDRE) will be cleared to "0" when the transmission data is written to the transmit data register (TDR).
- The transmission data empty flag (SSR:TDRE) will be set to "1" once a transmission starts after the transmission data has been transferred to the transmit shift register if the transmission FIFO is disabled or empty.
- You will be able to write transmission data when the transmission data empty flag (SSR:TDRE) is set to "1". If the transmission interrupt is enabled, a transmission interrupt will occur. Writing transmission data should be performed by the generation of transmission interrupt or be done when the transmission data empty flag (SSR:TDRE) is "1".
- You will not be able to write transmission data when the transmission data empty flag (SSR:TDRE) is "0" and transmission FIFO is disabled or full.
- In operation mode 1 (multi-processor mode), the operation will be 7-bit or 8-bit long. The AD bit will be transmitted by writing at the D8 bit.
- For the 9-bit long transfer and in operation mode 1, write a value to the TDR in 16-bit access mode.

Notes:

- Transmission data register is write-only register and receive data register is read-only register. The value written is different from the read value since the transmission/reception registers are located at the same address. Therefore instructions such as INC/DEC instructions which perform read-modify-write (RMW) operation cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR:TDRE) when using the transmission FIFO, see "32.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing."

32.4.2.5 Serial Aid Control Status Register: SACSR

This section explains the bit structure of the serial aid control status register.

Serial Aid Control Status Register (SACSR) configures serial test operation control, timer interrupt enable/disable, synchronous transmission enable/disable, operating clock division rate of serial timer, and serial timer enable/disable.

SACSRn (n=0 to 2): Address Base addr + 08_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
STST	Reserved						TINT	
0	0	0	0	0	0	0	0	Initial value
R/W	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TINTE	Reserved		TDIV3	TDIV2	TDIV1	TDIV0	TMRE	
0	0	0	0	0	0	0	0	Initial value
R/W	RX,W0	RX,W0	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] STST: Serial test bit

This bit selects to enable or disable serial test mode.
When serial test mode is enabled, SOT and SIN will be connected in the multi-function serial interface so that the data output from SOT can be received from SIN without any modification.
When serial test mode is enabled, SOT pin will be fixed to "H" and data input to SIN pin will be ignored.

STST	Serial test bit
0	Disables serial test mode
1	Enables serial test mode

Note:

This bit can be changed only when transmission and reception is disabled (SCR:TXE=0, SCR:RXE=0).

[bit14 to bit9] Reserved bits

Always write "0" to these bits.

Note:

These bits are invalid when external trigger enable bit (TRGE) is "0". [bit8] TINT: Timer Interrupt Flag

[bit8] TINT: Timer Interrupt Flag

When Serial Timer Register (STMR) matches Serial Timer Comparison Register (STMCR), the Serial Timer Register (STMR) becomes "0" and this bit will be set to "1".
When this bit is "1" and timer interrupt enable bit (TINTE) is "1", status interrupt request will be output.
When this bit is set to "0", it will be reset to "0".
Writing "1" to this bit is invalid.

TINT	Description
0	No timer interrupt request
1	Timer interrupt request

Notes:

- Performing software reset (SCR:UPCL="1") will reset this bit to "0".
- A read with a read-modify-write instruction will read "1".

[bit7] TINTE: Timer Interrupt Enable Bit

This bit enables/disables timer interrupt to the CPU.
When this bit is "1" and timer interrupt flag (TINT) is "1", status interrupt request will be output.

TINTE	Description
0	Disables interrupt from serial timer
1	Enables interrupt from serial timer

[bit6 to bit5] Reserved bits

Always write "0" to these bits.

[bit4 to bit1] TDIV3-0: Timer Operating Clock Division Bits

These bits set division rate of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer Operating Clock						
				Division rate	Φ= 8MHz	Φ= 10MHz	Φ= 16MHz	Φ= 20MHz	Φ= 24MHz	Φ= 32MHz
0	0	0	0	Φ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	Φ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	Φ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	Φ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	Φ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	Φ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	Φ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	Φ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	Φ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

Φ: Bus clock

Notes:

- These bits can be changed only when serial timer enable bit (TMRE) is "0".
- Any setup other than the above is prohibited.

[bit0] TMRE: Serial timer enable Bit

This bit enables/disables serial timer operations.

TMRE	Serial timer enable bit
0	Stops serial timer. While stopped, values in the Serial Timer Register (STMR) are retained.
1	When this bit is changed from "0" to "1", the value of the Serial Timer Register (STMR) will be initialized to "0" and serial timer will be started.

32.4.2.6 Serial TiMer Register: STMR

This section explains the bit structure of the serial timer register.

Serial Timer Register (STMR) indicates the timer values of the serial timer.

STMRn (n=0 to 2): Address Base addr + $0A_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit15 to bit0] TM15-0: Timer Data Bits

These bits indicate the timer values of the serial timer.
While the timer is running, the timer values of the serial timer will be increased by 1 every timer operating clock (SACSR: Set with TDIV3-0).

Note:

These bits will be initialized to "0" when the timer started running.

32.4.2.7 Serial Timer Comparison Register: STMCR

This section explains the bit structure of the serial timer comparison register.

Serial Timer Comparison Register (STMCR) sets the comparison value of a timer of the serial timer.

STMCRn (n=0 to 2): Address Base addr + $0C_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit0] TC15-0: Comparison Bits

These bits set a comparison value for serial timer.

These bits will be compared with the Serial Timer Register (STMR) and when these bits and the values of the STMR matched at the update timing of the STMR, the STMR will be set to "0". At this timing, the timer interrupt flag (SACSR:TINT) will be set to "1".

The interval of the following operations is (STMCR: TC+1) × timer operating clock (set to SACSR:TDIV3-0).

- SACSR:TINT is set to "1".

Notes:

- When $(0000)_H$ is set to this register, the Serial Timer Register still indicates "0".
- With "0000_H" set to this register while the timer interrupt flag (SACSR:TINT) will be fixed to "1" when the timer operating clock division value (SACSR:TDIV) is set to "0000_B" while the timer is running.
- This register can be changed only when serial timer is disabled (SACSR:TMRE="0").

32.4.2.8 Baud rate Generator Register: BGR

This section explains the bit structure of the baud rate generator register.

Baud rate generator register (BGR) sets the division ratio of serial clock. It can also select an external clock as the clock source of reload counter.

BGRn (n=0 to 2): Address Base addr + $1C_H$ (Access: Half-word, Word)

15	14	13	12	11	10	9	8	bit
EXT	BGR[14:8]							
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

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

[bit15] EXT (EXTernal clock): EXTernal clock select bit

This bit selects whether to use an internal clock source or an external clock source for the internal reload counter for baud rate generation. When setting EXT=0, the internal clock source will be used. When setting EXT=1, the external clock source will be used.

[bit14 to bit0] BGR14 to BGR0 (Baud rate GeneratoR): Baud rate generator bits

- These bits set division ratio of the serial clock.
- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15h00". Thus, if you wish to validate a new setting value immediately, execute programmable clear (UPCL) after you have change the setting value of BGR.
- When the reload value is an even number, As for the "H" width and "L" width of the reception serial clock, the "L" is longer by 1 cycle of the bus clock. When the reload value is an odd number, the widths of "H" and "L" become the same.
- Use a value 4 or greater to set to BGR. However, correct data may not be received depending on the baud rate error and reload setup value.
- When you change to the external clock setting (EXT=1) while baud rate generator is running, write "0" to the Baud Rate Generator (BGR) and perform programmable clear (UPCL), then set to the external clock (EXT=1).

32.4.3 Registers for CSIO

Registers for CSIO are shown.

32.4.3.1 Serial Control Register: SCR

This section explains the bit structure of the serial control register.

The serial control register (SCR) allows you to disable/enable transmission/reception interrupts, disable/enable transmission idle interrupt, disable/enable transmission and reception. Setup for connecting SPI and CSIO reset are also allowed.

SCRn (n=0 to 2): Address Base addr + 00_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	
0	0	0	0	0	0	0	0	Initial value
R0,W	R/W	R,W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	UPCL: Programmable clear bit	This bit initializes the internal state of CSIO. When this bit is set to "1": ■ Directly reset CSIO (software reset). In this case, the register settings will be retained. Note that any active transmission or reception will be cut off immediately. ■ Baud rate generator restarts by reloading the setting value of the BGR register. ■ All the transmissions/receptions and status interrupt factors (SSR:TDRE, TBI, RDRF, ORE, TINT, CSE) will be initialized. ■ All serial chip select pins become inactive. When this bit is set to "0": No effect on the operation. A read always results in "0". **Notes:** ■ Execute a programmable clear after disabling interrupts. ■ When using FIFO, disable FIFO (FCR0:FE2, FE1=0) before you execute a programmable clear. ■ Even if programmable clear is executed (SCR:UPCL=1), the value of serial timer register (STMR) is not cleared.
bit6	MS: Master/slave function select bit	This bit selects master or slave mode. When this bit is set to "0": Master mode When this bit is set to "1": Slave mode **Notes:** ■ If SMR:SCKE=0 when the slave mode is selected, an external clock will be input directly. ■ Set this bit when transmission and reception are disabled (TXE=RXE=0). ■ After MS bit is set, set reception enable (RXE=1).

Bit name		Function
bit5	SPI: SPI support bit	This bit is used to execute a SPI communication. When chip select is used in master mode (SCR: MS=0), this bit is used for serial chip select pin 0 communication. When this bit is set to "0": Normal synchronous communication When this bit is set to "1": SPI communication supported. **Notes:** ■ Set this bit when transmission and reception are disabled (TXE=RXE=0). ■ This bit is used in one of cases below. ☐ Chip select pin is disabled (SCSCR:CSEN1-0="00"$_B$) ☐ When mode is in slave (SCR:MS=1) ☐ When data format of chip select is disabled (ESCR:CSFE=0) ☐ When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active
bit4	RIE: Reception interrupt enable bit	■ When setting to "0": Reception interrupts are disabled. ■ When setting to "1": Reception interrupts are enabled. ■ This bit enables or disables the output of reception interrupt request to the CPU. ■ When the RIE bit and reception data flag bit (SSR:RDRF) are set to "1", or any of the error flag bit (ORE) is set to "1", a reception interrupt request will be output.
bit3	TIE: Transmission interrupt enable bit	■ When setting to "0": Transmission interrupts are disabled. ■ When setting to "1": Transmission interrupts are enabled. ■ This bit enables or disables the output of transmission interrupt request to the CPU. ■ When the TIE bit and the SSR:TDRE bit are set to "1", a transmission interrupt request will be output.
bit2	TBIE: Transmission bus idle interrupt enable bit	■ When setting to "0": Transmission bus idle interrupts are disabled. ■ When setting to "1": Transmission bus idle interrupts are enabled. ■ This bit enables or disables the output of transmission bus idle interrupt request to the CPU. ■ When the TBIE bit and SSR:TBI bit are set to "1", a transmission bus idle interrupt request will be output.
bit1	RXE: Reception enable bit	This bit enables/disables the reception of CSIO. ■ If this bit is set to "0", data frame reception is disabled. ■ If this bit is set to "1", data frame reception is enabled. **Notes:** ■ If you disable reception (RXE=0) while a reception is in progress, it immediately stops the reception. ■ After MS and SMR:SCINV bits are set, set reception enable (RXE=1).
bit0	TXE: Transmission enable bit	This bit enables/disables the transmission of CSIO. ■ If this bit is set to "0", data frame transmission is disabled. ■ If this bit is set to "1", data frame transmission is enabled. **Note:** If you disable transmission (TXE=0) while a transmission is in progress, it immediately stops the transmission.

32.4.3.2 Serial Status Register: SSR

This section explains the bit structure of the serial status register.

The serial status register (SSR) allows you to check the status of transmission/reception and the reception error flag as well as to clear the reception error flag.

SSRn (n=0 to 2): Address Base addr + 02_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
REC	Reserved		AWC	ORE	RDRF	TDRE	TBI	
0	0	0	0	0	0	1	1	Initial value
R0,W	R0,W0	R/W0	R/W	R,WX	R,WX	R,WX	R,WX	Attribute

Bit name		Function
bit7	REC: Reception error flag clear bit	This bit clears ORE flag of the serial status register (SSR). ■ To clear an error flag, write "1" to this bit. ■ Writing "0" does not affect the operation. A read always results in "0".
bit6, bit5	Reserved bits	Always write "0" to these bits.
bit4	AWC: Access width control bit	This bit selects 16-bit or 32-bit accesses for accessing transmission data register (TDR) or reception data register (RDR). ■ "0" is set: 16-bit access ■ "1" is set: 32-bit access **Note:** This bit can be changed only if transmission/reception is disabled (SCR:TXE=RXE=0) and TDR and RDR are empty (SSR:TDRE=1, SSR:RDRF=0). ■ To perform the DMA transfer while the transmit FIFO is not used, set "1" to this bit. ■ When the data length is 20, 24 or 32 bits, set "1" to this bit. ■ To use the SPI mode with the slave mode while the transmit FIFO is enabled (FCR0.FE2,FE1=0), set this bit to "1".
bit3	ORE: Overrun error flag bit	"0" Read: No overrun error "1" Read: There is an overrun error ■ If an overrun error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). ■ When the ORE bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. ■ If this flag is set, data contained in the receive data register (RDR) becomes invalid. ■ When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.

Bit name		Function
bit2	RDRF: Reception data full flag bit	"0" Read: Receive data register (RDR) is empty "1" Read: The received data register (RDR) contains data. ■ The flag indicates the state of the receive data register (RDR). ■ When received data is loaded in RDR, this flag will be set to "1" and when RDR is read out, it will be cleared to "0". ■ When the RDRF bit and SCR:RIE bit are set to "1", a reception interrupt request will be output. ■ While using reception FIFO, the RDRF will be set to "1" once the reception FIFO has received the specified number of data sets. ■ In the case where all the conditions below are met while using reception FIFO, when reception idle continues for more than 8 baud rate clocks, RDRF will be set to "1". ☐ Reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1" ☐ The reception FIFO contains data without receiving the specified number of data sets If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. ■ While using reception FIFO, the bit will be cleared to "0" once the reception FIFO becomes empty.
bit1	TDRE: Transmission data empty flag bit	"0" Read: Transmit data register (TDR) contains data. "1" Read: Transmit data register is empty ■ The flag indicates the state of the transmit data register (TDR). ■ When a transmit data is written to TDR, this flag becomes "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmit shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. ■ When the TDRE bit and the SCR:TIE bit are set to "1", a transmission interrupt request will be output. ■ When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". ■ For details of the timing of setting/resetting the TDRE bit while using transmission FIFO, see "32.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing."

Bit name		Function
bit0	TBI: Transmission bus idle flag bit	"0" Read: Transmission is in progress. "1" Read: No transmission operation ■ This bit indicates CSIO has no transmission in progress. ■ When transmission data has been written to the transmit data register (TDR), this bit will become "0". ■ When the transmit data register (TDR) is empty (TDRE=1) and no transmission is in progress, this bit will be set to "1". ■ When you set the UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". ■ When this bit is "1" and transmission bus idle interrupts are enabled (SCR:TBIE=1), a transmission interrupt request will be output. **Note:** This bit becomes "1" when transmission data register (TDR) is empty (TDRE=1) and serial chip select error (CSE=1) is generated.

32.4.3.3 Extended Serial Control Register: ESCR

This section explains the bit structure of the extended serial control register.

The extended communication control register (ESCR) is used to set the transmission/reception data length as well as to fix the serial output at the "H" level.

ESCRn (n=0 to 2): Address Base + 03_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
SOP	L[3]	CSFE	WT[1:0]		L[2:0]			
0	0	0	0	0	0	0	0	Initial value
R0,W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	SOP: Serial output pin set bit	■ This bit sets the serial output pin at the "H" level. When you write "1" to this bit, the SOT pin will be set to "H". However, you do not have to write "0" to this bit. ■ A read always results in "0". **Note:** Do not set this bit during serial data transmission.
bit5	CSFE: Serial chip select format enable bit	Enable or disable format setting for each serial chip select pin. ■ When set to "0": Same data format and clock format will be set to all serial chip select pins ■ When set to "1": Data format and clock format will be set to serial chip select pins separately When this bit is set to "1", following setups will be set to serial chip select pins separately. ■ Inactive level for serial chip select ■ Mark level of serial clock ■ SPI transfer/normal transfer selection ■ Direction of serial data transfer ■ Serial data length **Note:** This bit setting will be invalid when any of following is met. ■ When chip select pin is disabled (SCSCR:CSEN1-0="$00"_B$) ■ When mode is in slave (SCR:MS=1) ■ When this bit is set to "1", set the following settings: ☐ Enable reception FIFO ☐ Set the hold delay to 2 or larger (setting CSHD7-0 bit in SCSTR1 register to 2 or larger) ☐ Set the length of each serial chip select data to 9 bits or smaller, or 10 bits or larger to communicate with two or more slave devices ☐ Disable setting the serial chip select pins separately to 9 bits and smaller, and 10 bits and larger (Examples of disabled settings) serial chip select 0 = 9 bits serial chip select 1 = 10 bits (Examples of enabled settings) serial chip select 0 = 16 bits serial chip select 1 = 10 bits

Bit name		Function
bit4, bit3	WT1, WT0: Data transmission/reception wait select bits	In the master mode, these bits set the number of waits for a successive data transmission or reception. Operation in the slave mode is "00". ■ "00": SCK will be output sequentially. ■ "01": SCK will be output after waiting for 1-bit time. ■ "10": SCK will be output after waiting for 2-bit time. ■ "11": SCK will be output after waiting for 3-bit time. **Notes:** ■ If this register is used when all of the following conditions are satisfied, set WT1 and WT0 to "00". ☐ Chip select is used. ☐ The SPI mode (SCR:SPI=1) is used. ☐ "01"H is set in TBYTE register. ☐ The SCAM bit in SCSCR register is set to "1"
bit6, bit2 to bit0	L3, L2, L1, L0: Data length select bits	These bits specify the data length of transmission/reception data. When chip select is used in master mode (SCR:MS=0), these bits are used for serial chip select pin 0 communication. ■ "0000_B": Data length will be set to 8 bits. ■ "0001_B": Data length will be set to 5 bits. ■ "0010_B": Data length will be set to 6 bits. ■ "0011_B": Data length will be set to 7 bits. ■ "0100_B": Data length will be set to 9 bits. ■ "0101_B": Data length will be set to 10 bits. ■ "0110_B": Data length will be set to 11 bits. ■ "0111_B": Data length will be set to 12 bits. ■ "1000_B": Data length will be set to 13 bits. ■ "1001_B": Data length will be set to 14 bits. ■ "1010_B": Data length will be set to 15 bits. ■ "1011_B": Data length will be set to 16 bits. ■ "1100_B": Data length will be set to 20 bits. ■ "1101_B": Data length will be set to 24 bits. ■ "1110_B": Data length will be set to 32 bits. **Notes:** ■ Settings other than those listed above are prohibited. ■ This bit is used in one of cases below. ☐ When chip select pin is disabled (SCSCR:CSEN1-0="00_B") ☐ While in slave mode (SCR:MS=1) ☐ When data format of chip select is disabled (ESCR:CSFE=0) ☐ When data format of chip select is enabled (ESCR:CSFE=1) and serial chip select pin 0 is active

32.4.3.4 Receive Data Register/Transmit Data Register: RDR/TDR

This section explains the bit structure of the receive data register/transmit data register.

The receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

Read

RDR1n-0n (n=0 to 2): Address Base addr + 04_H (Access: Half-word, Word)

31	30	29	28	27	26	25	24	bit
D31	D30	D29	D28	D27	D26	D25	D24	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

23	22	21	20	19	18	17	16	bit
D23	D22	D21	D20	D19	D18	D17	D16	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

The receive data register (RDR) is a 32-bit data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN pin) are converted in the shift register and stored in the receive data register (RDR).

Depending on the data length, received data will be filled from the lower bit and other bits become "0".

Example: When the data length is 8 bits and "45_H" is received: D7-D0="45_H", D31-D8=0

- When the received data is stored in the receive data register (RDR), the reception data full flag bit (SSR:RDRF) will be set to "1". When reception interrupts are enabled (SSR:RIE=1), a reception interrupt request will be generated.
- The receive data register (RDR) should be read out when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) will be automatically cleared to "0" when the receive data register (RDR) has been read out.
- In case a reception error occurs (SSR:ORE is "1"), data in the receive data register (RDR) will become invalid.
- When you read RDR, accesses must be made with following methods.
  - ☐ SSR:AWC=0: 16-bit access to lower 16 bits of RDR
  - ☐ SSR:AWC=1: 32-bit access
- SSR:AWC=1 allows one-time read for any data length.
- SSR:AWC=0 allows one-time read for any data length from 5 to 16 bits.

Notes:

- When you use reception FIFO, if received data in the reception FIFO reaches specified number, "1" will be set to RDRF.
- When you are using reception FIFO, if the reception FIFO becomes empty, RDRF will be cleared to "0".
- If a reception error occurs (SSR:ORE is "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, data received will not be stored at the reception FIFO.
- When SSR:AWC=0, each of the read values of D31-D16 is undefined.

Write

TDR1n-0n (n=0 to 2): Address Base addr + 04_H (Access: Byte, Half-word, Word)

31	30	29	28	27	26	25	24	bit
D31	D30	D29	D28	D27	D26	D25	D24	
0	0	0	0	0	0	0	0	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

23	22	21	20	19	18	17	16	bit
D23	D22	D21	D20	D19	D18	D17	D16	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

The transmit data register (TDR) is the 32-bit data buffer register for sending serial data.

- When transmit operations are enabled (SCR:TXE=1), if transmission data is written to the transmit data register (TDR), the transmission data is transferred to the transmit shift register and converted to serial data, then output from the serial data output pin (SOT Pin).
- Depending on the data length, the transmitting data will be stored from the lower bit and other bits will become "invalid".

Example: When you transmit "45_H" with 8 bits data length, D7-D0="45_H" and D31-D8 will become invalid.

- Transmission data empty flag (SSR:TDRE) will be cleared to "0" when the transmission data is written to the transmit data register (TDR).
- The transmission data empty flag (SSR:TDRE) will be set to "1" once a transmission starts after the transmission data has been transferred to the transmit shift register if the transmission FIFO is disabled or empty.
- You will be able to write transmission data when the transmission data empty flag (SSR:TDRE) is set to "1". If the transmission interrupt is enabled, a transmission interrupt will occur. Writing next transmission data should be performed after the generation of transmission interrupt or be done when the transmission data empty flag (SSR:TDRE) is "1".
- You will not be able to write transmission data when the transmission data empty flag (SSR:TDRE) is "0" and transmission FIFO is disabled or full.
- When you write to TDR, accesses must be made with following methods.
  - ☐ SSR:AWC=0: 16-bit access to lower 16 bits of TDR
  - ☐ SSR:AWC=1: 32-bit access
- SSR:AWC=1 allows one-time write for any data length.
- SSR:AWC=0 allows one-time write for any data length from 5 to 16 bits.

Relationship between Transmit Data Register (TDR) and Transmission Data Empty Flag

In the 16-bit access mode (SSR:AWC=0), the TDR register has the 16-bit boundary, and transmission data is stored as 16-bit data for each writing. In addition, when 32-bit transmission data exists in the TDR register, the transmit data empty flag (SSR:TDRE) will be "0".

In the 32-bit access mode (SSR:AWC=1), the TDR register has the 32-bit boundary, and transmission data is stored as 32-bit data for each writing.

Table 32-3. Relationship between Transmit Data Register (TDR) and Transmission Data Empty Flag

Data access width	Data length	TDR register storage data value	TBI flag	TDRE flag	Transmission
16-bit access (SSR:AWC=0)	5-16 bits	0 bit	1	1	Disabled
		16 bits	0		Enabled
		32 bits		0	
32-bit access (SSR:AWC=1)	All data length	0 bit	1	1	Disabled
		32 bits	0	0	Enabled

Notes:

- Transmit data register is write-only register and receive data register is read-only register. The value written is different from the read value since the transmit/receive registers are located at the same address. Therefore, instructions such as INC/DEC instructions which perform read-modify-write (RMW) operations cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR: TDRE) when using the transmission FIFO, see "32.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing."

32.4.3.5 Serial Aid Control Status Register: SACSR

This section explains the bit structure of the serial aid control status register.

The serial aid control status register (SACSR) allows you to control serial test operations, enable/disable timer interrupts, enable/disable synchronous transmission, set the division value of the operating clock of the serial timer, and enable/disable the serial timer.

SACSRn (n=0 to 2): Address Base addr + 08_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
STST	Reserved	TBEEN	CSEIE	CSE	Reserved		TINT	
0	0	0	0	0	0	0	0	Initial value
R,W	R0,W0	R/W	R/W	R(RM1), W	RX,W0	RX,W0	R(RM1), W	Attribute

7	6	5	4	3	2	1	0	bit
TINTE	TSYNE	Reserved	TDIV3	TDIV2	TDIV1	TDIV0	TMRE	
0	0	0	0	0	0	0	0	Initial value
R/W	R,W	RX,W0	R,W	R,W	R,W	R,W	R/W	Attribute

[bit15] STST: Serial TeST bit

This bit is used to enable or disable the serial test mode.
When the serial test mode is enabled, SOT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOT can be received from SIN without being processed.
When the serial test mode is enabled, the SOT pin will be fixed to "H", and data input into the SIN pin will be ignored.

STST	Serial test bit
0	Serial test mode disabled
1	Serial test mode enabled

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=0, SCR:RXE=0).
- Set this bit to "0" in slave mode (SCR:MS=1).

[bit14] Reserved

Read: The read value is "0".
Write: Always write "0" to this bit.

[bit13] TBEEN: Transfer Byte Error ENable bit

If one of the following conditions applies in the master mode (SCR:MS="0") and if no valid transmission data is available (SSR:TDRE="1") for the transmission data register (TDR) when one frame has been transmitted while the number of frames that are being transmitted is smaller than the setting value of TBYTE, this bit is used to enable/disable occurrence of serial chip select errors.

- Chip select is used
- Synchronous transmission of the serial timer is used

TBEEN	Transfer Byte Error ENable bit
0	Occurrence of chip select errors in the master mode (SCR:MS=0) disabled
1	Occurrence of chip select errors in the master mode (SCR:MS=0) enabled

Note:

This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

[bit12] CSEIE: Chip select error interrupt enable bit

This bit is used to enable/disable chip select error interrupt request output.
When the CSEIE bit and the chip select error flag bit (CSE) are set to "1", a transmission interrupt request will be output.

CSEIE	Chip select error interrupt enable bit
0	Chip select error interrupt disabled
1	Chip select error interrupt enabled

[bit11] CSE: Chip select error flag bit

In the master mode (SCR:MS=0) with transfer byte error enabled (TBEEN=1), this bit is set to "1" in either of the following cases, if there is no valid transmit data (SSR:TDRE=1) in the transmit data register (TDR) when 1-frame transmit is completed while the number of transmitted frames is less than the TBYTE setting value.

- Chip select is used
- Transmission synchronized with the serial timer is used

In the slave mode (SCR:MS="1"), if a serial chip select pin becomes inactive during the transmission operation (SSR:TBI=0), this bit will be set to "1".
When this bit is set to "1" and the chip select error interrupt enable bit (CSEIE) is set to "1", a transmission interrupt request will be output.
Writing "0" to this bit will reset it to "0".
Writing "1" to this bit has no effect.

CSE	Chip select error flag bit
0	No chip select errors
1	Chip select errors

Notes:

- When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- For read-modify-write instructions, "1" will be read.
- If a chip select error occurs (CSE=1), write "0" to this bit after disabling the transmission (SCR:TXE=0). To resume transmission, write "0" to this bit, and then enable the transmission (SCR:TXE=1) and write the transmission data to the transmission data buffer (TDR).
- If the serial chip select input noise of one bus clock or more occurs during the slave mode transmission, this bit may be set to "1". In this case, resume transmission after the master transmission is completed.

[bit10, bit9] Reserved bits

Always write "0" to these bits.

[bit8] TINT: Timer interrupt flag

When the serial timer register (STMR) matches the serial timer compare register (STMCR), the serial timer register (STMR) will be set to "0", and this bit will be set to "1".
When this bit is set to "1" and the timer interrupt enable bit (TINTE) is set to "1", a status interrupt request will be output.
Writing "0" to this bit will reset it to "0".
Writing "1" to this bit has no effect.

TINT	Description
0	No timer interrupt request
1	Timer interrupt request

Notes:

- When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- For read-modify-write instructions, "1" will be read.

[bit7] TINTE: Timer interrupt enable bit

This bit is used to enable/disable timer interrupts to the CPU.
When this bit is set to "1" and the timer interrupt flag (TINT) is set to "1", a status interrupt request will be output.

TINTE	Description
0	Interrupts by the serial timer disabled
1	Interrupts by the serial timer enabled

[bit6] TSYNE: Synchronous transmission enable bit

This bit is used to enable or disable synchronous transmission.
When this bit is set to "1", transmission will be activated if the serial timer register (STMR) matches the serial timer compare register (STMCR).

TSYNE	Description
0	Synchronous transmission disabled The serial timer will be used as a timer.
1	Synchronous transmission enabled The serial timer will not be used as a timer.

Notes:

- This bit can be changed only when the serial timer enable bit (TMRE) is set to "0".
- When synchronous transmission is enabled (TSYNE="1") and transmission is disabled (SCR:TXE="0"), transmission will not be activated, even if the serial timer register (STMR) matches the serial timer compare register (STMCR).
- In the slave mode (SCR:MS="1"), this bit will be internally fixed to "0".

[bit5] Reserved bit

Always write "0" to this bit.

[bit4 to bit1] TDIV3-0: Timer operating clock division bits

These bits are used to set the division ratio of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer operating clock						
				Division ratio	Φ= 8MHz	Φ= 10MHz	Φ= 16MHz	Φ= 20MHz	Φ= 24MHz	Φ= 32MHz
0	0	0	0	Φ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	Φ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	Φ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	Φ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	Φ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	Φ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	Φ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	Φ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	Φ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

Φ: Bus clock

Notes:

- These bits can be changed only when the serial timer enable bit (TMRE) is set to "0".
- Settings other than those listed above are prohibited.

[bit0] TMRE: Serial timer enable bit

This bit is used to enable or disable the operation of the serial timer.

TMRE	Serial timer enable bit
0	The operation of the serial timer will be stopped. During stop, the value of the serial timer register (STMR) will be retained.
1	If this bit is changed from "0" to"1", the value of the serial timer register (STMR) will be initialized to "0", and the operation of the serial timer will be started.

Note:

To perform synchronous transmission by the serial timer change this bit under one of the following conditions:

- ☐ Transmission disabled (SCR:TXE="0")
- ☐ Transmission bus idle (SSR:TBI="1")

32.4.3.6 Serial Timer Register: STMR

This section explains the bit structure of the serial timer register.

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

STMRn (n=0 to 2): Address Base addr + $0A_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bt15 to bit0] TM15-0: Timer data bits

These bits are used to indicate the timer value of the serial timer.
During timer operation, 1 will be added to the timer value of the serial timer for each timer operating clock (set by SACSR:TDIV3-0).

Note:

At the start of timer operation, these bits will be initialized to "0".

32.4.3.7 Serial Timer Compare Register: STMCR

This section explains the bit structure of the serial timer compare register.

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

STMCRn (n=0 to 2): Address Base addr + $0C_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

7	6	5	4	3	2	1	0	bit
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

[bit15 to bit0] TC15-0: Compare bits

These bits are used to set compare values of the serial timer.
These bits will be compared with the serial timer register (SMTR), and when these bits match the value of the serial timer register immediately after the serial timer register (SMTR) is updated, they will set the serial timer register to "0". Then, if synchronous transmission is disabled (SACSR:TSYNE="0"), these bits will set the timer interrupt flag (SACSR:TINT) to "1", and if synchronous transmission is enabled (SACSR:TSYNE="1"), these bits will activate transmission.

The interval of the following operations is (STMCR: TC+1) × timer operating clock (set to SACSR:TDIV3-0).

- SACSR:TINT is set to "1".
- Transmission is activated synchronizing with the serial timer.

Notes:

- When "0000_H" is set to this register, the serial timer register will remain set to "0".
- When "0000_H" is set to this register with synchronous transmission disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:TINT) will be fixed to "1", if the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B" during timer operation.
- This register can be changed only when the serial timer is disabled (SACSR:TMRE="0").

32.4.3.8 Serial Chip Select Control Status Register: SCSCR

This section explains the bit structure of the serial chip select control status register.

The serial chip select control status register (SCSCR) is used to select a start pin and an end pin for serial chip select, indicate an output pin for serial chip select, retain an active level of serial chip select, invert serial chip select, and enable/disable output of serial chip select pins.

SCSCRn (n=0 to 2): Address Base addr + $0E_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved	SST0	Reserved	SED0	Reserved	SCD0	SCAM	CDIV2	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R0,W0	R/W	R0,W0	R,WX	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
CDIV1	CDIV0	CSLVL	Reserved		CSEN1	CSEN0	CSOE	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R0,W0	R0,W0	R/W	R/W	R/W	Attribute

[bit15] Reserved bit

Always write "0" to this bit.

[bit14] SST0: Serial chip select start bit

This bit is used to select a pin where serial chip select starts.
If transmission disabled (SCR:TXE="0") status is changed to transmission enabled (SCR:TXE="1") status, the serial chip select pin set by this bit will first become active.

SST0	Start pin
0	SCS0
1	SCS1

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- If the same value is set to both the serial chip select start bit (SST0) and the serial chip select end bit (SED0), only the serial chip select pin set by this bit will become active.
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- Only the serial chip select pin with serial chip select enabled (CSEN="1") will become active.

[bit13] Reserved bit

Always write "0" to this bit.

[bit12] SED0: Serial chip select end bit

This bit is used to select a pin where serial chip select ends.
When the serial chip select pin set by this bit becomes active, the serial chip select pin specified by the serial chip select start bit (SST0) will become active next time.

SED0	End pin
0	SCS0
1	SCS1

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- If the same value is set to both the serial chip select start bit (SST0) and the serial chip select end bit (SED0), only the serial chip select pin set by this bit will become active.
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- Only the serial chip select pin with serial chip select enabled (CSEN="1") will become active.

[bit11] Reserved bit

Always write "0" to this bit.

[bit10] SCD0: Serial chip select display bit

This bit is used to display a pin where serial chip select is active.

SCD0	Display pin
0	SCS0
1	SCS1

Notes:

- If a serial chip select pin is inactive, the serial chip select pin which becomes active next time will be displayed.
- This bit will be set to "0_B" when the slave mode is selected (SCR:MS="1"), software reset is triggered (SCR:UPCL="1"), or transmission is disabled (SCR:TXE="0").

[bit9] SCAM: Serial chip select active retain bit

This bit is used to select whether to retain the active state of a serial chip select pin or not. When this bit is set to "1", the serial chip select pin will not become inactive, even if transmission operation has completed (SSR:TBI="1") after a serial chip select pin becomes active. If this bit is set to "0" when a serial chip select pin is active and this bit is set to "1", the serial chip select pin will become inactive after transmission has completed.

SCAM	Serial chip select active retain bit
0	Active state of a serial chip select pin not retained
1	Active state of a serial chip select pin retained

Notes:

- If transmission is disabled (SCR:TXE="0") and software reset is triggered (SCR:UPCL="1"), a serial chip select pin will become inactive regardless of the value of this bit.
- When a serial chip error occurs (SACSR:CSE=1), a serial chip select pin will become inactive regardless of the value of this bit.
- If this register is used when all of the following conditions are satisfied, set this bit to "0".
  - ☐ The master mode (SCR:MS=0) is used.
  - ☐ The chip select is used.
  - ☐ The SPI mode (SCR:SPI=1) is used.
  - ☐ "01"H is set in TBYTE register.
  - ☐ A value other than "00" is set to WT1 and WT0 bits of ESCR register.

[bit8 to bit6] CDIV2-0: Serial chip select timing operating clock division bits

These bits are used to set the division ratio of a serial chip select timing operating clock.

CDIV2	CDIV1	CDIV0	Serial chip select timing operating clock						
			Division ratio	Φ= 8MHz	Φ= 10MHz	Φ= 16MHz	Φ= 20MHz	Φ= 24MHz	Φ= 32MHz
0	0	0	Φ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	1	Φ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	1	0	Φ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	1	1	Φ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
1	0	0	Φ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
1	0	1	Φ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
1	1	0	Φ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs

Φ: Bus clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Settings other than those listed above are prohibited.

[bit5] CSLVL: Serial chip select level setting bit

This bit is used to select "H" or "L" for the level when a serial chip select pin is inactive.
This bit is used for communication of a chip select pin 0.

CSLVL	Serial chip select level setting bit
0	Inactive level set to "L"
1	Inactive level set to "H"

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- Setting this bit is used under one of the following conditions:
  - ☐ Slave mode (SCR:MS="1")
  - ☐ Data format of chip select is disabled (ESCR:CSFE="0")
  - ☐ Data format of chip select is enabled (ESCR:CSFE="1") and a serial chip select pin 0 is active

[bit4, bit3] Reserved bits

Always write "0" to these bits.

[bit2, bit1] CSEN1-0: Serial chip select enable bits

These bits are used to enable or disable each serial chip select pin.
CSEN1 bit corresponds to SCS1 pin and CSEN0 bit to SCS0 pin.
In the slave mode (SCR:MS="1"), only CSEN0 bit is used to enable or disable a serial chip pin.

CSEN	Serial chip select enable bit
0	Operation of a serial chip select pin disabled
1	Operation of a serial chip select pin enabled

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the master mode (SCR:MS="0"), if CSEN1-0 are set to "00_B", transmission/reception operation will be performed regardless of a serial chip select pin.
- In the slave mode (SCR:MS="1"), if CSEN0 is set to "0", transmission/reception operation will be performed regardless of a serial chip select pin.

[bit0] CSOE: Serial chip select output enable bit

This bit is used to enable or disable output of serial chip select pins.

CSOE	Serial chip select output enable bit
0	Output of all serial chip select pins disabled
1	Output of all serial chip select pins enabled

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), set this bit to "0".

32.4.3.9 Serial Chip Select Timing Register: SCSTR3-0

This section explains the bit structure of the serial chip select timing register.

The serial chip select timing register (SCSTR3-0) is used to set the setup delay time for serial chip select, the hold delay time for serial chip select, and the deselect time for serial chip select.

Notes:

- In the master mode (SCR:MS=0) and the normal mode (SCR:SPI=0), set the setup delay time (CSSU7-0) or the hold delay time (CSHD7-0) so that either of the following conditions is satisfied.

  Baud rate/2[ns] < hold delay [ns] + 3 x bus clock [ns]

  Hold delay + setup delay < baud rate - 2 x bus clock [ns]

SCSTR1n-0n (n=0 to 2): Address Base addr + 12_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit8] CSSU7-0: Serial chip select setup delay bits

These bits are used to set a time interval between the timing when a serial chip select pin becomes active and the timing when the serial clock is output. If "00_H" is set to these bits, the timing when the serial clock is output will be the same as the timing when a serial chip select pin becomes active.

CSSU7	CSSU6	CSSU5	CSSU4	CSSU3	CSSU2	CSSU1	CSSU0	Setup delay time
0	0	0	0	0	0	0	0	No setup delay time
0	0	0	0	0	0	0	1	1 × Serial chip select timing operating clock
0	0	0	0	0	0	1	0	2 × Serial chip select timing operating clock
. .	. .	. .	. .	. .	. .	. .	. .	. .
1	1	1	1	1	1	1	0	254 × Serial chip select timing operating clock
1	1	1	1	1	1	1	1	255 × Serial chip select timing operating clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Regardless of the deselect time setting, once after the serial chip select pin becomes inactive, it will take at least five bus clock cycles to be active again.
- If these bits are set to "00_H", the timing when a serial chip select pin becomes active will be the same as the timing when the serial clock outputs an edge for the first time.

[bit7 to bit0] CSHD7-0: Serial chip select hold delay bits

These bits are used to set a time interval between the timing when an output of the serial time clock is terminated the timing when a serial chip select pin becomes inactive. If "00_H" is set to these bits, the timing when an output of the serial time clock is terminated will be the same as the timing when a serial chip select pin becomes inactive.

CSHD7	CSHD6	CSHD5	CSHD4	CSHD3	CSHD2	CSHD1	CSHD0	Hold delay time
0	0	0	0	0	0	0	0	No hold delay time
0	0	0	0	0	0	0	1	1 × Serial chip select timing operating clock
0	0	0	0	0	0	1	0	2 × Serial chip select timing operating clock
. .	. .	. .	. .	. .	. .	. .	. .	. .
1	1	1	1	1	1	1	0	254 × Serial chip select timing operating clock
1	1	1	1	1	1	1	1	255 × Serial chip select timing operating clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Regardless of the deselect time setting, once after the serial chip select pin becomes inactive, it will take at least five bus clock cycles to be active again.
- If these bits are set to "00_H", the timing when a serial chip select pin becomes active will be the same as the timing when the serial clock outputs an edge for the first time.

SCSTR3n-2n (n=0 to 2): Address Base addr + 10_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
CSDS15	CSDS14	CSDS13	CSDS12	CSDS11	CSDS10	CSDS9	CSDS8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit0] CSDS15-0: Serial chip deselect bits

These bits are used to set a minimum time interval between the timing when a serial chip select pin becomes inactive and the timing when the serial chip select pin becomes active next time.

CSDS15	CSDS14	CSDS13	・・・	CSDS2	CSDS1	CSDS0	Minimum deselect time
0	0	0	・・・	0	0	0	No minimum deselect time (5 bus clock time)
0	0	0	・・・	0	0	1	1 × Serial chip select timing operating clock
0	0	0	・・・	0	1	0	2 × Serial chip select timing operating clock
・ ・	・ ・	・ ・	・・・	・ ・	・ ・	・ ・	・ ・
1	1	1	・・・	1	1	0	65534 × Serial chip select timing operating clock
1	1	1	・・・	1	1	1	65535 × Serial chip select timing operating clock

Notes:

- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- Regardless of the deselect time setting, once after the serial chip select pin becomes inactive, it will take at least five bus clock cycles to be active again.

32.4.3.10 Serial Chip Select Format Register: SCSFR0

This section explains the bit structure of the serial chip select format register.

The serial chip select format register (SCSFR0) is used to select an active level of chip select for each serial chip select, invert the serial clock, configure settings for connection with SPI, and set data direction and data length of serial data output.

SCSFR0n (n=0 to 2): Address Base addr + 16_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved								
1	0	0	0	0	0	0	0	Initial value
R1,W1	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

7	6	5	4	3	2	1	0	bit
CS1 CSLVL	CS1 SCINV	CS1 SPI	CS1 BDS	CS1 L3	CS1 L2	CS1 L1	CS1 L0	
1	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Reserved bit

Always write "1" to this bit.

[bit14 to bit8] Reserved bit

Always write "0" to these bits.

[bit7] CS1CSLVL: Serial chip select level setting bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select the level when the serial chip select pin 1 is inactive.

CS1CSLVL	Serial chip select pin 1 Serial chip select setting bit
0	Inactive level set to "L"
1	Inactive level set to "H"

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit6] CS1SCINV: Serial clock invert bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to set the serial clock format when the serial chip select pin 1 is active.

When this bit is set to "0":

- Serial clock output mark level is set to "H".
- Transmission data is output in synchronization with a falling edge of the serial clock in the normal transfer while it is output in synchronization with a rising edge of the serial clock in the SPI transfer.
- Reception data is sampled at a rising edge of the serial clock in the normal transfer while it is sampled at a falling edge of the serial clock in the SPI transfer.

When this bit is set to "1":

- Serial clock output mark level is set to "L".
- Transmission data is output in synchronization with a rising edge of the serial clock in the normal transfer while it is output in synchronization with a falling edge of the serial clock in the SPI transfer.
- Reception data is sampled at a falling edge of the serial clock in the normal transfer while it is sampled at a rising edge of the serial clock in the SPI transfer.

CS1SCINV	Serial chip select pin 1 Serial clock invert bit
0	Mark level "H" format
1	Mark level "L" format

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit5] CS1SPI: SPI support bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to execute an SPI communication when the serial chip select pin 1 is active.

- When this bit is set to "0": Normal synchronous communication.
- When this bit is set to "1": SPI communication supported.

CS1SPI	Serial chip select pin 1 SPI support bit
0	Normal synchronous transfer
1	SPI supported

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit4] CS1BDS: Transfer direction select bit for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), this bit is used to select whether to transfer the transfer serial data from the least significant bit (LSB first, BDS="0") or from the most significant bit (MSB first, BDS="1") when the serial chip select pin 1 is active.

CS1BDS	Serial chip select pin 1 Transfer direction select bit
0	LSB first (transfer from the least significant bit)
1	MSB first (transfer from the most significant bit)

Notes:

- This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting this bit has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting this bit has no effect.

[bit3 to bit0] CS1 L3, L2, L1, L0: Data length select bits for chip select 1

If data format of chip select is enabled (ESCR:CSFE="1"), these bits are used to specify the data length of transmission/reception data when the serial chip select pin 1 is active.

CS1L3	CS1L2	CS1L1	CS1L0	Serial chip select pin 1 Data length select bits
0	0	0	0	8-bit length
0	0	0	1	5-bit length
0	0	1	0	6-bit length
0	0	1	1	7-bit length
0	1	0	0	9-bit length
0	1	0	1	10-bit length
0	1	1	0	11-bit length
0	1	1	1	12-bit length
1	0	0	0	13-bit length
1	0	0	1	14-bit length
1	0	1	0	15-bit length
1	0	1	1	16-bit length
1	1	0	0	20-bit length
1	1	0	1	24-bit length
1	1	1	0	32-bit length

Notes:

- Settings other than those listed above are prohibited.
- These bits can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").
- In the slave mode (SCR:MS="1"), setting these bits has no effect.
- When data format of chip select is disabled (ESCR:CSFE="0"), setting these bits has no effect.

32.4.3.11 Transfer BYTE register: TBYTE1-0

This section explains the bit structure of the transfer byte register.

The transfer byte register (TBYTE) is used to indicate the transfer data count when each serial chip select pin is active.

TBYTE1n-0n (n=0 to 2): Address Base addr + 18_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

7	6	5	4	3	2	1	0	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	Attribute

15	14	13	12	11	10	9	8	bit
TBYTE1[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TBYTE0[7:0]								
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit8, bit7 to bit0] TBYTE: Transfer data count display bits

The transfer byte register allows you to set transfer data count when each serial chip select pin is active. After a serial chip select pin has become active, the serial chip select pin will become inactive when transfer of data count set to these bits has completed.

The serial chip select pin 0 (SCS0) corresponds to TBYTE0 and the serial chip select pin 1 (SCS1) to TBYTE1.

If one of the following conditions is satisfied, the transfer byte register 0 (TBYTE0) is used for synchronous transmission. When transmission operation is started with synchronous transmission, data count set to TBYTE0 will be transferred.

- Serial chip select is disabled (SCSCR:CSEN1-0="00"b)

If the value of these bits is changed during transmission operation (SSR:TBI=0), the changed transfer data count setting will take effect after the operation of transmitting transfer data count set before the change has completed.

TBYTE	Transfer byte register
Write	Writing to TBYTE
Read	Setting value of TBYTE

Notes:

- If "00_H" is set to these bits, transfer count is 8.
- If synchronous transmission is to be performed when chip select is used during the master operation (SCR:MS="0"), transfer count varies as follows:
  - ☐ Specified number of TBYTE0 if the chip select pin 0 is active
  - ☐ Specified number of TBYTE1 if the chip select pin 1 is active
- If this register is used when all of the following conditions are satisfied, set TBYTE register to a value other than "01"H.
  - ☐ The master mode (SCR:MS=0) is used.
  - ☐ The chip select is used.
  - ☐ The SPI mode (SCR:SPI=1) is used.
  - ☐ "1" is set to SCAM bit of SCSCR register.
  - ☐ A value other than "00" is set to WT1 and WT0 bits of ESCR register.

32.4.3.12 Baud rate Generator Register: BGR

This section explains the bit structure of the baud rate generator register.

Baud rate generator register (BGR) sets the division ratio of serial clock.

BGRn (n=0 to 2): Address Base addr + $1C_H$ (Access: Half-word, Word)

15	14	13	12	11	10	9	8	bit
-	BGR[14:8]							
0	0	0	0	0	0	0	0	Initial value
RX,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

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

[bit15] Undefined

No effect for writing operations.

[bit14 to bit0] BGR (Baud rate GeneratoR): Baud rate generator bits

- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- If the reload value is an even number, the "H" and "L" widths of the serial clock depend on the SCINV bit setting as follows: If it is an odd number, the "H" and "L" widths of the serial clock are equal.
  - ☐ If SMR:SCINV="0", the "H" width of the serial clock is longer by one cycle of the bus clock.
  - ☐ If SMR:SCINV="1", the "L" width of the serial clock is longer by one cycle of the bus clock.
- Set the reload value to 3 or higher.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15h00". Thus, if you wish to validate a new setting value immediately, execute CSIO reset (SCR:UPCL) after you have changed the setting value of BGR.
- To operate in the slave mode by setting "1" to the reception FIFO idle detection enable bit (FCR1:FRIIE) when you use reception FIFO, set the baud rate at the BGR.

32.4.4 Registers for LIN

Registers for LIN is shown.

32.4.4.1 Serial Control Register: SCR

This section explains the bit structure of the serial control register.

The serial control register (SCR) allows you to disable/enable transmission/reception interrupts, disable/enable transmission idle interrupts, and disable/enable transmissions and receptions. This register also has setups for generating LIN break field and resetting LIN interface reset (v2.1).

SCRn (n=0 to 2): Address Base addr + 00_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE	
0	0	0	0	0	0	0	0	Initial value
R0,W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	UPCL: Programmable clear bit	This bit initializes the internal state of LIN interface (v2.1). When this bit is set to "1": ■ Directly reset LIN interface (v2.1) (software reset). In this case, the register settings will be maintained. Note that any active transmission or reception will be cut off immediately. ■ Baud rate generator restarts by reloading the setting value of the BGR register. ■ All transmission and reception and status interrupt sources (SSR:TDRE, TBI, RDRF, FRE, ORE, LBD, TINT, and SFD) are initialized. ■ The baud rate setting flag (SACSR:BST) is initialized. When this bit is set to "0": No effect. For reading, "0" is always read out. **Notes:** ■ Execute a programmable clear after disabling interrupts. ■ When using FIFO, disable FIFO (FCR0:FE2,FE1=0) before you execute a programmable clear ■ Transmission/reception FIFO is not cleared by a programmable clear. ■ Serial timer register (SMTR) value will not be initialized when a programmable clear is executed (SSR:UPCL=1).
bit6	MS: Master/slave select bit	This bit selects master or slave mode. "0" is set: Master mode will be set. "1" is set: Slave mode will be set.
bit5	LBR: LIN break field setting bit (Functions only in the master operation)	■ In case of LIN manual mode operation (LAMCR: LAMEN ="0"), when "1" is set to this bit, an LIN Break Field and an LIN Break Delimiter (set by the ESCR:LBL1/0 bit and ESCR:DEL1/0) are generated. ■ In case of LIN assist mode operation (LAMCR: LAMEN ="1"), when "1" is set to this bit, an LIN Break Field and an LIN Break Delimiter (set by the ESCR:LBL2/1/0 bit and ESCR:DEL1/0) are generated, and then a Sync Field and an ID Field are transmitted. Write: Writing "0": No effect. Writing "1": In case of LIN manual mode operation (LAMCR: LAMEN = "0"), generates LIN break field. In case of LIN assist mode operation (LAMCR: LAMEN = "1"), an LIN Break Field is generated, and then a Sync Field and an ID Field are transmitted. For reading, "0" will be always read out. **Notes:** ■ Functions only in the master operation (MS="0"). ■ To set an LIN Break Field during transmission of header or response, initialize the transmission/reception data. See "32.7.5.2 LIN Break Field retransmission processing in assist mode."

	Bit name	Function
bit4	RIE: Reception interrupt enable bit	■ This bit enables or disables the output of reception interrupt request to the CPU. ■ When the RIE bit and reception data flag bit (SSR:RDRF) are set to "1", or any of the error flag bits (SSR: FRE, ORE) is set to "1", a reception interrupt request will be output.
bit3	TIE: Transmission interrupt enable bit	■ This bit enables or disables the output of transmission interrupt request to the CPU. ■ When the TIE bit and the SSR:TDRE are set to "1", a transmission interrupt request will be output.
bit2	TBIE: Transmission bus idle interrupt enable bit	■ This bit enables or disables the output of transmission bus idle interrupt request to the CPU. ■ When the TBIE bit and SSR:TBI bit are set to "1", a transmission bus idle interrupt request will be output.
bit1	RXE: Reception enable bit	This bit enables/disables the reception of LIN-interface (v2.1). ■ If this bit is set to "0", data frame reception is disabled. ■ If this bit is set to "1", data frame reception is enabled. **Notes:** ■ Even when you enable reception (RXE=1), does not start the reception until a falling edge of the start bit is input. ■ In the master operation mode, data will not be received even if receptions are enabled (RXE=1) during LIN break field transmission. ■ If data reception is disabled (RXE=0) during reception in manual mode (LAMCR:LAMEN=0), the current data reception is stopped immediately. ■ In assist mode (LAMCR:LAMEN=1), disable the reception (RXE=0) during header transmission/reception and response transmission. ■ Even if the reception operation is disabled (RXE=0) during header reception in assist mode (LAMCR:LAMEN=1), header reception operation does not stop. To stop this, disable the reception (RXE=0) and set to manual mode (LAMCR:LAMEN=0). ■ If data reception is disabled (RXE=0) during response reception in assist mode (LAMCR:LAMEN=1), the current data reception is stopped immediately. ■ If data reception is enabled (RXE=1) during LIN Break Field reception, a framing error will be detected (SSR:FRE=1).
bit0	TXE: Transmission enable bit	This bit enables/disables the transmission of LIN-interface (v2.1). ■ If this bit is set to "0", data frame transmission is disabled. ■ If this bit is set to "1", data frame transmission is enabled. **Notes:** ■ If data transmission is disabled (TXE=0) during transmission in manual mode (LAMCR:LAMEN=0), the current data transmission is stopped immediately. ■ Even if the transmission operation is disabled (TXE=0) during header transmission in assist mode (LAMCR:LAMEN=1), header transmission operation does not stop. To stop this, disable the transmission (TXE=0) and set to manual mode (LAMCR:LAMEN=0). ■ If data transmission is disabled (TXE=0) during response transmission in assist mode (LAMCR:LAMEN=1), the current data transmission is stopped immediately.

32.4.4.2 Serial Status Register: SSR

This section explains the bit structure of the serial status register.

The serial status register (SSR) allows you to check the status of transmission/reception and the reception error flag and to detect the LIN break field as well as to clear the reception error flag.

SSRn (n=0 to 2): Address Base addr + 02_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
REC	Reserved	LBD	FRE	ORE	RDRF	TDRE	TBI	
0	0	0	0	0	0	1	1	Initial value
R0,W	R0,W0	R(RM1),W	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

Bit name		Function
bit7	REC: Reception error flag clear bit	This bit clears the FRE and ORE flags of the serial status register (SSR). ■ To clear an error flag, write "1" to this bit. ■ Writing "0" does not affect anything. A read always results in "0".
bit6	Reserved bit	Read: Always reads "0". Write: Always write "0".
bit5	LBD: LIN break field detection flag bit	This bit indicates that LIN break field is detected. When data with 11 or greater bits of "0" is input to serial input (SIN), LBD bit is set to "1". In this case, when "1" is set to the LIN break field interrupt enable bit (LBIE), a status interrupt will be generated. (When read) "1": LIN break field is detected. "0": LIN break field Is not detected. (When written) "0" is written: LBD bit will be cleared. "1" is written: No effect. **Note:** ■ When a read-modify-write instruction is used, "1" will be read.
bit4	FRE: Framing error flag bit	"0" Read: No framing error "1" Read: There is a framing error ■ If a framing error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). ■ When the FRE bit and RIE bit are set to "1", a reception interrupt request will be output. ■ If this flag is set, data contained in the receive data register (RDR) becomes invalid. ■ When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO. **Notes:** ■ If data reception is enabled (SCR:RXE=1) during LIN Break Field reception, a framing error will be detected before detecting LIN Break Field. However, reception of the header is not stopped. ■ In assist mode (LAMCR:LAMEN), if the master transmits another LIN Break between LIN Break Field detection and ID Field reception completion, a framing error will be detected at the "L" level of the tenth bit of the new LIN Break Field, even if data reception is disabled (SCR:RXE=0). However, reception of the header is not stopped.

Bit name		Function
bit3	ORE: Overrun error flag bit	"0" Read: No overrun error "1" Read: There is an overrun error ■ If an overrun error occurs while a reception is in progress, this bit will be set to "1". To clear this bit, write "1" to the REC bit of the serial status register (SSR). ■ When the ORE bit and RIE bit are set to "1", a reception interrupt request will be output. ■ If this flag is set, data contained in the receive data register (RDR) becomes invalid. ■ When this flag is set while using the reception FIFO, the reception FIFO enable bit will be cleared. As a result, the reception data will not be stored in the reception FIFO.
bit2	RDRF: Reception data full flag bit	"0" Read: Receive data register (RDR) is empty "1" Read: Receive data register (RDR) contains data. ■ The flag indicates the state of the receive data register (RDR). ■ When received data is loaded in RDR, this flag will be set to "1" and when RDR is read out, it will be cleared to "0". ■ When the RDRF bit and RIE bit are set to "1", a reception interrupt request will be output. ■ While using reception FIFO, the RDRF will be set to "1" once the reception FIFO has received the specified number of data sets. ■ While using reception FIFO, the bit will be cleared to "0" once the reception FIFO becomes empty. **Note:** In slave (SCR:MS=1) operation of assist mode (LAMCR:LAMEN=1), if LAMCR:LIDEN=0 is set to use the receive data register (RDR) for ID Field reception, the reception data full flag bit is set (RDRF=1) when loading the reception ID Field value to the receive data register (RDR). In addition, the LIN automatic header complete flag is also set (LAMSR:LAHC=1) at the same time.
bit1	TDRE: Transmission data empty flag bit	"0" Read: Transmit data register (TDR) contains data. "1" Read: Transmit data register (TDR) is empty ■ The flag indicates the state of the transmit data register (TDR). ■ When a transmit data is written to TDR, this flag becomes "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmit shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. ■ When the TDRE bit and the TIE bit are set to "1", a transmission interrupt request will be output. ■ When you set UPCL bit of the serial control register (SCR) to "1", the TDRE bit will be set to "1". ■ For details of the timing of setting/resetting the TDRE bit while using transmission FIFO, see "32.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing."

Bit name		Function
bit0	TBI: Transmission bus idle flag bit	"0" Read: Transmitting "1" Read: No transmission operation ■ This bit indicates that LIN Interface (v2.1) has no transmission in progress. ■ When transmission data has been written to the transmit data register (TDR), this bit will become "0". ■ When the transmit data register (TDR) is empty (TDRE=1) and no transmission is in progress, this bit will be set to "1". ■ In manual mode (LAMCR:LAMEN=0) ☐ When the LIN Break Field is set (SMR:LBR=1), this bit is set to "0". ☐ If the Transmit Data Register becomes empty after the LIN Break Field has been transmitted, this bit is set to "1". ■ In assist mode (LAMCR:LAMEN=1) ☐ During header transmission of master (SCR:MS=0), this bit is set to "0". ☐ If the Transmit Data Register becomes empty after the header (ID Field transmission) has been transmitted, this bit is set to "1". ☐ During response transmission, this bit is set to "0". ☐ If the Transmit Data Register becomes empty after the response (checksum transmission) has been transmitted, this bit is set to "1". ■ When this bit is "1" and transmission bus idle interrupts are enabled (SCR:TBIE=1), a transmission interrupt request will be output.

32.4.4.3 Extended Serial Control Register: ESCR

This section explains the bit structure of the extended serial control register.

The extended serial control register (ESCR) is used to enable/disable LIN break field interrupt, detect LIN break field, set LIN break field length and Break delimiter length, and select stop bit length.

ESCRn (n=0 to 2): Address Base addr + 03_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	ESBL	LBL2	LBIE	LBL[1:0]		DEL[1:0]		
-	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

Bit name		Function
bit7	Reserved bit	Always write "0".
bit6	ESBL: Extended stop bit length select bit	This bit configures the bit length of stop bit (frame end mark for transmission data). When SBL="0" and ESCR:ESBL="0" are set: Stop bit is set to 1 bit. When SBL="1" and ESCR:ESBL="0" are set: Stop bit is set to 2 bits. When SBL="0" and ESCR:ESBL="1" are set: Stop bit is set to 3 bits. When SBL="1" and ESCR:ESBL="1" are set: Stop bit is set to 4 bits. **Notes:** ■ When receiving, only the first bit of the stop bits will always be detected. ■ This bit should be set when transmission is disabled (TXE=0). ■ In assist mode (LAMCR:LAMEN=1), set this bit before setting the LIN Break Field (SCR:LBR=1).
bit4	LBIE: LIN break field detection interrupt enable bit	This bit enables/disables LIN break field detection interrupt. A reception interrupt occurs when LIN break field detection flag (LBD) is set to "1" and interrupts are enabled (LBIE=1). ■ "0" is set: LIN break field detection interrupt is disabled ■ "1" is set: LIN break field detection interrupt is enabled
bit5, bit3, bit2	LBL[2:0]: LIN break field length select bits (Functions only in the master operation)	"000": 13-bit length "001": 14-bit length "010": 15-bit length "011": 16-bit length "100": 17-bit length "101": 18-bit length "110": 19-bit length "111": 20-bit length ■ These bits set the length of LIN break field generation time interval (in bits). ■ Before you set LBR bit in serial control register (SCR) to "1" (LIN break field send), set this bit. ■ The timing of LIN break field detect is always the 11th bit at slave operation, regardless of the set value of this bit. **Notes:** ■ This function is enabled only in the master operation (SMR:MS="0"). ■ Set this bit before setting the LIN Break Field (SCR:LBR=1). ■ In manual mode (LAMCR:LAMEN=0), this is operated between the 13-bit length and 16-bit length. Therefore, always set LBL2 to "0".

	Bit name	Function
bit1, bit0	DEL[1:0]: LIN Break delimiter length select bits (Functions only in the master operation)	"00": 1-bit length "01": 2-bit length "10": 3-bit length "11": 4-bit length ■ These bits set the length of LIN Break delimiter (in bits). ■ Before you set LBR bit in serial control register (SCR) to "1" (LIN break field send), set this bit. **Note:** This function is enabled only in the master operation (SMR:MS="0").

32.4.4.4 Receive Data Register/Transmit Data Register: RDR/TDR

This section explains the bit structure of the receive data register/transmit data register.

The receive data register and transmit data register are located within the same addresses. When read, it functions as the receive data register and when written, it functions as the transmit data register.

Read

RDR0n (n=0 to 2): Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

The receive data register (RDR) is the data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN pin) are converted in the shift register and stored in the receive data register (RDR).
- When the received data is stored in the receive data register (RDR), the reception data full flag bit (SSR:RDRF) will be set to "1". When the reception interrupt is enabled (SSR:RIE=1), reception interrupt requests will be generated.
- The receive data register (RDR) should be read out when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) will be automatically cleared to "0" when the serial receive data register (RDR) has been read out.
- In case a reception error occurs (either SSR:ORE or SSR:FRE is "1"), data in the receive data register (RDR) will become invalid.

Notes:

- If a reception error occurs, data in the Receive Data Register (RDR) is invalid.
- Operations in assist mode (LAMCR:LAMEN=1) are shown below.
  - ☐ In slave operation, if LAMCR:LIDEN=1 is set to use the LIN assist mode reception ID register for ID Field reception, the reception ID value is not stored in the receive data register (RDR) when receiving ID Field, and the reception data full flag bit (SSR:RDRF) is not set.
  - ☐ In slave operation, if LAMCR:LIDEN=0 is set to use the receive data register (RDR) for ID Field reception, the reception ID value is stored in the receive data register (RDR) when receiving ID Field. However, the reception data full flag bit (SSR:RDRF) is not set. Check the ID value by setting the LIN automatic header completion flag (LAMSR:LAHC=1).
  - ☐ Sync Field and checksum are not stored in the receive data register (RDR), and the reception data full flag bit (SSR:RDRF) is not set.
  - ☐ The transmission data for each field is not stored in the receive data register (RDR), and the reception data full flag bit (SSR:RDRF) is not set.
  - ☐ If a reception error (SSR:FRE, ORE, LAMESR:LCSER, LSFER, LBSER, LPTER) occurs, the transmission/reception processing of assist mode is stopped. At this time, the response reception processing stops the reception data storing operation to the receive data register, regardless of data reception setting (SCR:RXE=1).
- When using reception FIFO, the operation is shown below.

- ☐ When you use reception FIFO, if received data in the reception FIFO reaches specified number, "1" will be set to SSR:RDRF.
- ☐ When you are using reception FIFO, if the reception FIFO becomes empty, SSR:RDRF will be cleared to "0".
- ☐ If a reception error occurs (either SSR:ORE or SSR:FRE is "1") while using reception FIFO, the reception FIFO enable bit will be cleared. As a result, data received will not be stored at the reception FIFO.

Write

TDR0n (n=0 to 2): Address Base addr + 06_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
Reserved								
0	0	0	0	0	0	0	0	Initial value
R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

The transmit data register (TDR) is the data buffer register for sending serial data.

- When transmit operations are enabled (SCR:TXE=1), if transmission data is written to the transmit data register (TDR), the transmission data is transferred to the transmit shift register and converted to serial data, then output from the serial data output pin (SOT pin).
- Transmission data empty flag (SSR:TDRE) will be cleared to "0" when the transmission data is written to the serial transmit data register (TDR).
- The transmission data empty flag (SSR:TDRE) will be set to "1" once a transmission starts after the transmission data has been transferred to the transmit shift register if the transmission FIFO is disabled or empty.
- If the transmission data empty flag (SSR:TDRE) is "1", the next transmission data can be written. If the transmission interrupt is enabled, a transmission interrupt will occur. Writing the next transmission data should be performed by the generation of transmission interrupt or be done when the transmission data empty flag (SSR:TDRE) is "1".
- You will not be able to write transmission data to the transmit data register (TDR) when the transmission data empty flag (SSR:TDRE) is "0" and transmission FIFO is disabled or full.

Notes:

- Transmit data register is write-only register and receive data register is read-only register. Because the two registers are located in the same address, write value and read value might be different. Therefore instructions such as INC/DEC instructions which perform read-modify-write (RMW) operation cannot be used.
- For more information about the set timing of the transmission data empty flag (SSR: TDRE) when using the transmission FIFO, see "32.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing."

32.4.4.5 Serial Aid Control Status Register: SACSR

This section explains the bit structure of the serial aid control status register.

The serial aid control status register (SACSR) allows you to control serial test operations, enable/disable timer interrupts, enable/disable synchronous transmission, set the division value of the operating clock of the serial timer, and enable/disable the serial timer.

SACSRn (n=0 to 2): Address Base addr + 08_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
STST	BST	SFD	SFDE	AUTE	Reserved		TINT	
0	0	0	0	0	0	0	0	Initial value
R,W	R,W0	R(RM1),W	R/W	R,W	RX,W0	RX,W0	R(RM1),W	Attribute

7	6	5	4	3	2	1	0	bit
TINTE	Reserved		TDIV3	TDIV2	TDIV1	TDIV0	TMRE	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W0	R,W0	R,W	R,W	R,W	R,W	R/W	Attribute

[bit15] STST: Serial test bit

This bit is used to enable or disable the serial test mode.
When the serial test mode is enabled, SOT and SIN will be connected inside the multi-function serial interface, and data to be transmitted from SOT can be received from SIN without being processed.
When the serial test mode is enabled, the SOT pin will be fixed to "H", and data input into the SIN pin will be ignored.

STST	Serial test bit
0	Serial test mode disabled
1	Serial test mode enabled

Note:

This bit can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

[bit14] BST: Baud rate setting flag

This bit indicates that automatic baud rate adjustment due to Sync Field reception was executed.

This bit is updated if Sync Field detects the fifth fall of the LIN bus.

BST	Baud rate setting flag	
	write	read
0	No influence	Automatic baud rate adjustment disabled
1		Automatic baud rate adjustment enabled

Notes:

- When automatic baud rate adjustment is disabled (AUTE=0), this bit will be fixed to "0".
- When a software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- This bit is valid only when the sync field detection flag (SACSR:SFD) is "1".
- Writing to this bit has no effect.

[bit13] SFD: Sync Field detection flag

This bit is used to indicate that Sync Field was detected.
When the fifth falling edge of LIN bus is detected in Sync Field, this bit will be set to "1".
When this bit is set to "1" and the Sync Field detection interrupt enable bit (SFDE) is set to "1", a status interrupt request will be output.
Writing "0" to this bit will reset it to "0".

SFD	Sync Field detection flag	
	Write	Read
0	Clear	No Sync Field detected
1	No influence	Sync Field detected

Notes:

- When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- Writing "1" to this bit has no effect.
- In both the master mode (SCR:MS="0") and the slave mode (SCR:MS="1"), this bit takes effect.
- A read with a read-modify-write instruction will read "1".

[bit12] SFDE: Sync Field detection interrupt enable bit

This bit is used to enable/disable Sync Field interrupts to the CPU.
When this bit is set to "1" and the Sync Field detection flag (SFD) is set to "1", a status interrupt request will be output.

SFDE	Sync Field detection interrupt enable bit
0	Interrupts by Sync Field detection disabled
1	Interrupts by Sync Field detection enabled

[bit11] AUTE: Automatic baud rate adjustment bit

This bit is used to enable/disable automatic baud rate adjustment.

AUTE	Automatic baud rate adjustment bit
0	Automatic baud rate adjustment disabled
1	Automatic baud rate adjustment enabled

Notes:

- In the master mode (SCR:MS="0"), this bit will be internally fixed to "0".
- When this bit is set to "1", the timer operating clock division bit (TDIV3-0) will be set to "3_H" (8 divisions).
- This bit can be changed from "0" to "1" only when the serial timer enable bit (TMRE) is set to "0".

[bit10, bit9] Reserved bits

Always write "0" to these bits.

[bit8] TINT: Timer interrupt flag

When the serial timer register (STMR) matches the serial timer compare register (STMCR), the serial timer register (STMR) will be set to "0", and this bit will be set to "1".
When this bit is set to "1" and the timer interrupt enable bit (TINTE) is set to "1", a status interrupt request will be output.
Writing "0" to this bit will reset it to "0".
Writing "1" to this bit has no effect.

TINT	Description	
	Write	Read
0	Clear	No timer interrupt request
1	No influence	Timer interrupt request

Notes:

- When software reset is triggered (SCR:UPCL="1"), this bit will be reset to "0".
- For read-modify-write instructions, "1" will be read.

[bit7] TINTE: Timer interrupt enable bit

This bit is used to enable/disable timer interrupts to the CPU.
When this bit is set to "1" and the timer interrupt flag (TINT) is set to "1", a status interrupt request will be output.

TINTE	Description
0	Interrupts by the serial timer disabled
1	Interrupts by the serial timer enabled

[bit6,bit5] Reserved bit

Always write "0" to these bits.

[bit4 to bit1] TDIV3-0: Timer operating clock division bits

These bits are used to set the division ratio of the serial timer.

TDIV3	TDIV2	TDIV1	TDIV0	Timer operating clock						
				Division ratio	Φ= 8MHz	Φ= 10MHz	Φ= 16MHz	Φ= 20MHz	Φ= 24MHz	Φ= 32MHz
0	0	0	0	Φ	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	Φ/2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	Φ/4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	Φ/8	1µs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	Φ/16	2µs	1.6µs	1µs	800ns	666.67ns	500ns
0	1	0	1	Φ/32	4µs	3.2µs	2µs	1.6µs	1.33µs	1µs
0	1	1	0	Φ/64	8µs	6.4µs	4µs	3.2µs	2.67µs	2µs
0	1	1	1	Φ/128	16µs	12.8µs	8µs	6.4µs	5.33µs	4µs
1	0	0	0	Φ/256	32µs	25.6µs	16µs	12.8µs	10.67µs	8µs

Φ: Bus clock

Notes:

- These bits can be changed only when the serial timer enable bit (TMRE) is set to "0".
- Settings other than those listed above are prohibited.

[bit0] TMRE: Serial timer enable bit

This bit is used to enable or disable the operation of the serial timer.

TMRE	Serial timer enable bit
0	The operation of the serial timer will be stopped During stop, the value of the serial timer register (STMR) will be retained.
1	If this bit is changed from "0" to"1", the value of the serial timer register (STMR) will be initialized to "0", and the operation of the serial timer will be started.

32.4.4.6 Serial Timer Register: STMR

This section explains the bit structure of the serial timer register.

The serial timer register (STMR) is used to indicate the timer value of the serial timer.

STMRn (n=0 to 2): Address Base addr + $0A_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TM15	TM14	TM13	TM12	TM11	TM10	TM9	TM8	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

7	6	5	4	3	2	1	0	bit
TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit15 to bit0] TM15-0: Timer data bits

These bits are used to indicate the timer value of the serial timer.
During timer operation, 1 will be added to the timer value of the serial timer for each timer operating clock (set by SACSR:TDIV3-0).

Note:

At the start of timer operation, these bits will be initialized to "0".

32.4.4.7 Serial Timer Compare Register: STMCR

This section explains the bit structure of the serial timer compare register.

The serial timer compare register (STMCR) is used to set compared values of the serial timer.

STMCRn (n=0 to 2): Address Base addr + $0C_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15 to bit0] TC15-0: Compare bits

These bits will be compared with the serial timer register (STMR), and when these bits match the value of the serial timer register immediately after the serial timer register (STMR) is updated, they will set the serial timer register to "0". Then, the timer interrupt flag (SACSR:INT) is set to "1".

Notes:

- When "0000_H" is set to this register, the serial timer register will remain set to "0".
- When "0000_H" is set to this register with synchronous transmission disabled (SACSR:TSYNE="0"), the timer interrupt flag (SACSR:TINT) will be fixed to "1", if the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B" during timer operation.
- This register can be changed only when the serial timer is disabled (SACSR:TMRE="0").
- If all the following conditions are satisfied, the serial timer register (STMR) might be reset to "0000_H" before baud rate adjustment is made. Therefore, when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "1", set a larger value to these bits than the value set by the Sync Field upper limit bit (SFUR).
  - ☐ The automatic baud rate adjustment bit (SACSR:AUTE) is set to "1"
  - ☐ These bits have a smaller value than the value set by the Sync Field upper limit bit (SFUR)

32.4.4.8 Sync Field Upper limit Register: SFUR

This section explains the bit structure of the Sync Field upper limit register.

The Sync Field upper limit register (SFUR) is used to set the upper limit of the value which can be set to the baud rate generator register for automatic baud rate adjustment.

SFURn (n=0 to 2): Address Base addr + $0E_H$ (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
-	TU14	TU13	TU12	TU11	TU0	TU9	TU8	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TU7	TU6	TU5	TU4	TU3	TU2	TU1	TU0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Undefined

The read value is "0". Writing has no effect on the operation.

[bit14 to bit0] TU14-0: Upper limit bits

These bits are used to set the upper limit of the value which can be set to the baud rate generator register (BGR) for automatic baud rate adjustment.
When the automatic baud rate adjustment bit (SACSR:AUTE) is set to "1" and the slave mode is selected (SCR:MS="1"), the value of the serial timer register (STMR) will be set to the baud rate generator register (BGR), if the value of the serial timer register (STMR) after Sync Field is received is smaller than these bits and larger than the Sync Field lower limit register (SFLR).

Note:

These bits can be changed when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "0".

32.4.4.9 Sync Field Lower limit Register: SFLR

This section explains the bit structure of the Sync Field lower limit register.

The Sync Field lower limit register (SFLR) is used to set the lower limit of the value which can be set to the baud rate generator register for automatic baud rate adjustment.

SFLR1n-0n (n=0 to 2): Address Base addr + 12_H (Access: Byte, Half-word, Word)

15	14	13	12	11	10	9	8	bit
-	TL14	TL13	TL12	TL11	TL0	TL9	TL8	
0	0	0	0	0	0	0	0	Initial value
R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
TL7	TL6	TL5	TL4	TL3	TL2	TL1	TL0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit15] Undefined

The read value is "0". Writing has no effect on the operation.

[bit14 to bit0] TL14-0: Lower limit bits

These bits are used to set the lower limit of the value which can be set to the baud rate generator register (BGR) for automatic baud rate adjustment.

When the automatic baud rate adjustment bit (SACSR:AUTE) is set to "1" and the slave mode is selected (SCR:MS="1"), the value of the serial timer register (STMR) will be set to the baud rate generator register (BGR), if the value of the serial timer register (STMR) after Sync Field is received is smaller than the Sync Field upper limit register (SFUR) and larger than these bits.

Note:

These bits can be changed when the automatic baud rate adjustment bit (SACSR:AUTE) is set to "0".

32.4.4.10 Baud Rate Generator Register: BGR

This section explains the bit structure of the baud rate generator register.

Baud rate generator register (BGR) sets the division ratio of serial clock. It can also select an external clock as the clock source of reload counter.

BGRn (n=0 to 2): Address Base addr + $1C_H$ (Access: Half-word, Word)

15	14	13	12	11	10	9	8	bit
EXT	BGR[14:8]							
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

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

[bit15] EXT (EXTernal clock): External clock select bit

This bit selects whether to use an internal clock source or an external clock source for the internal reload counter for baud rate generation. When setting EXT="0", the internal clock source will be used. When setting EXT="1", the external clock source will be used.

[bit14 to bit0] BGR (Baud rate GeneratoR): Baud rate generator bits

- These bits set division ratio of the serial clock.
- Capable of writing a reload value to be counted and reading a set value.
- Reload counter will start counting when a reload value is written.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- When you change the setting value of the baud rate generator register (BGR), a new setting value will be reloaded after the counter value becomes "15h00". Thus, if you wish to validate a new setting value immediately, execute programmable clear (UPCL) after you have changed the setting value of BGR.
- When the reload value is an even number, As for the "H" width and "L" width of the reception serial clock, the "L" width is longer by 1 cycle of the bus clock. When the reload value is an odd number, the widths of "H" and "L" for serial clock become the same.
- Use a value 3 or greater for reload values. However, correct data may not be received depending on the baud rate error and reload setup value.
- When you change to the external clock setup(EXT=1) while baud rate generator is running, write "0" to the Baud Rate Generator (BGR) and perform programmable clear (UPCL), then set to the external clock (EXT=1).

32.4.4.11 LIN Assist Mode Status Register: LAMSR

This section explains the bit structure of the LIN assist mode status register.

LIN assist mode status register (LAMSR) is used in order to check the status of automatic header transmission/reception and to check whether there is any error flag received.

LAMSRn (n=0 to 2): Address Base addr + 10_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
LER	SER	RDRF	TDRE	TBI	LCSC	Reserved	LAHC	
0	0	0	1	1	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,W	R0,W0	R,W	Attribute

[bit7] LER: LIN representative error flag bit

When the following errors occur, this bit is set to "1". As for conditions on setting and clearing the error flag bit, see the explanations for each bit of the LIN assist mode error status register (LAMESR).

- LIN bus error flag bit (LBSER)
- LIN Sync Data error flag bit (LSFER)
- LIN ID parity error flag bit (LPTER)
- LIN checksum error flag bit (LCSER)

LER	LIN representative error flag
0	No error
1	Error

Note:

In manual mode (LAMCR:LAMEN=0), the read value of this bit is always "0".

[bit6] SER: Serial interface representative error flag bit

When the following errors occur, this bit is set to "1". As for conditions on setting and clearing the error flag bit, see the explanations for each bit of the serial status register (SSR); "32.4.4.2 Serial Status Register: SSR."

- framing error flag bit (FRE)
- overrun error flag bit (ORE)

SER	Serial interface representative error flag
0	No error
1	Error

[bit5] RDRF: Reception data full flag bit

This bit is the same as the reception data full flag bit (RDRF) of the serial status register (SSR). As for the explanation for this bit, see "32.4.4.2 Serial Status Register: SSR."

[bit4] TDRE: Transmission data empty flag bit

This bit is the same as the transmission data empty flag bit (TDRE) of the serial status register (SSR). As for the explanation for this bit, see "32.4.4.2 Serial Status Register: SSR."

[bit3] TBI: Transmission bus idle flag bit

This bit is the same as the transmission bus idle flag bit (TBI) of the serial status register (SSR). As for the explanation for this bit, see "32.4.4.2 Serial Status Register: SSR."

[bit2] LCSC: LIN checksum arithmetic operation completion flag bit

This bit is a flag that shows the completion of LIN checksum arithmetic operation.
In LIN assist mode (LAMCR:LAMEN="1"), when a chunk of data with the set length (LAMCR:LDL3-0) and its checksum are received, the LIN checksum arithmetic operation is completed and then this bit is set to "1".
When the LIN checksum arithmetic operation completion flag bit (LCSC) and the checksum arithmetic operation completion interrupt enable bit (LCSCIE) are "1", the status interrupt request is output.

- When reading

"1": The checksum arithmetic operation completion has been detected.
"0": The checksum arithmetic operation completion has not been detected.

- When writing

"0": The LCSC bit is cleared.
"1": No influence.

LCSC	LIN checksum arithmetic operation completion flag	
	Write	Read
0	Clear LCSC flag	Performing checksum arithmetic operation or Waiting for the start of checksum arithmetic operation
1	No influence	Checksum arithmetic operation completed

Note:

If the read-modify-write instruction is executed, "1" will be read out.
In manual mode (LAMCR:LAMEN="0"), the read value of this bit is always "0".

[bit1] Reserved bit

Always write "0" to this bit.

[bit0] LAHC: LIN automatic header completion flag bit

- This bit is a flag that shows the state of LIN automatic header.
- In assist mode (LAMCR:LAMEN=1), when the LIN header is received this bit is set to "1".
- When the LIN automatic header completion flag bit (LAHC) and the LIN automatic header completion interrupt enable bit (LAHCIE) are "1", the status interrupt request is output.
- When reading the LIN automatic header receive ID register (LAMRID) after "1" is set to this bit, "0" is set.
  - ☐ When reading:

    "1": The LIN automatic header completion has been detected.
    "0": The LIN automatic header completion has not been detected.

  - ☐ When writing:

    "0": The LAHC bit is cleared.
    "1": No influence.

LAHC	LIN automatic header completion flag	
	Write	Read
0	Clear LAHC flag	Performing LIN automatic header reception or Waiting for reception
1	No influence	LIN automatic header reception completed

Note:

If the read-modify-write instruction is executed, "1" will be read out.
In manual mode (LAMCR:LAMEN="0"), the read value of this bit is always "0".

32.4.4.12 LIN Assist Mode Control Register: LAMCR

This section explains the bit structure of the LIN assist mode control register.

LIN assist mode control register (LAMCR) enables LIN automatic header processing, enables LIN ID register use, selects LIN checksum type, and clears TDR and sets LIN data length in LIN assist mode.

LAMCRn (n=0 to 2): Address Base addr + 11_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
LDL3	LDL2	LDL1	LDL0	LTDRCL	LCSTYP	LIDEN	LAMEN	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R0, W	R/W	R/W	R/W	Attribute

[bit7 to bit4] LDL3, LDL2, LDL1, LDL0: LIN data length setting bit

These bits set 0 to 8 bytes as the LIN response data length.
Set the value of the data length for the setting value.
In transmission operation, after the data of this data length is transmitted, a checksum is generated and transmitted.
In reception operation, the checksum of the data, received after the previously received data with this data length, is checked.

LDL3	LDL2	LDL1	LDL0	LIN data length setting bits
0	0	0	0	0 byte length
0	0	0	1	1 byte length
0	0	1	0	2 bytes length
0	0	1	1	3 bytes length
0	1	0	0	4 bytes length
0	1	0	1	5 bytes length
0	1	1	0	6 bytes length
0	1	1	1	7 bytes length
1	0	0	0	8 bytes length

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- Be sure to set these settings before transmitting/receiving the response part of the data.
- Be sure not to set "1001" to "1111" to the LDL3-0 bits.
- In a response transmission node, when the LIN data length is set to 0 byte length (LDL3-0="0000"), write the dummy value ("don't care") to the TDR register in order to transmit a checksum. At this time, the TDR setting value does not influence the checksum arithmetic operation.
- When the LIN data length is set to 0 bytes length (LDL3-0="0000"), value of the checksum is as follows.
  - ☐ When the standard checksum is set (LCSTYP=0), the value of the checksum is 0xFF.
  - ☐ When the extended checksum is set (LCSTYP=1), the value of the checksum is an inverted value of the ID Field.

[bit3] LTDRCL: Transmit data register clear bit

This bit clears the transmission data register (TDR).
"1": The transmission data register is reset.
"0": There is no influence on the operation.
For reading, "0" is always read out.

LTDRCL	Transmission data register clear bit	
	write	read
0	No influence	The read value is always "0"
1	The transmission data register (TDR) is cleared	

Notes:

- A transmission FIFO is not reset even though the transmission data register is cleared.
- When the transmission FIFO is used, be sure to clear the transmission data register after clearing the transmission FIFO (FCR0:FCL1 or FCR0:FCL2).
- When the transmit data register (TDR) with data is cleared (LTDRCL=1) while reception FIFO is not used or is empty, the transmission data empty flag bit is set (SSR:TDRE=1 and LAMSR:TDRE=1).

[bit2] LCSTYP: LIN checksum type select bit

This bit selects the checksum type of LIN.

LCSTYP	LIN checksum type select bit
0	Standard checksum
1	Extended checksum

Notes:

- This function is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- For master, set this before generating an LIN Break field (LBR="1").
- For slave, fix this bit to "0".

[bit1] LIDEN: LIN ID register use enable bit

This bit enables the use of the LIN assist mode transmission/reception ID register (LAMTID/LAMRID).

- In master mode: (SCR:MS=0)

"0": The transmission data register (TDR) is used in order to transmit the LIN ID Field.
"1": The LIN assist mode transmission ID register (LAMTID) is used in order to transmit the LIN ID Field.

- In slave mode: (SCR:MS=1)

"0": The receive data register (RDR) is used in order to receive the LIN ID Field.
"1": The LIN assist mode reception ID register (LAMRID) is used in order to receive the LIN ID Field.

LIDEN	LIN ID register use enable bit	
	Master	Slave
0	Transmission data register (TDR) is used	Reception data register (RDR) is used
1	LIN assist mode transmission ID register (LAMTID) is used	LIN assist mode reception ID register (LAMRID) is used

Notes:

- This function is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- Be sure to set this setting before transmitting/receiving the header.
- When the reception FIFO is used and this setting is set to "1" (LAMRID is used), a received ID is not stored in the reception FIFO.

[bit0] LAMEN: LIN assist mode processing enable bit

This bit enables the LIN assist mode processing.

LAMEN	LIN assist mode enable bit
0	Manual mode
1	Assist mode

Notes:

- In manual mode, be sure to change this bit when transmission/reception is prohibited in LIN (SCR:RXE="0", SCR:TXE=0).
- In assist mode, be sure not to change this bit while LIN is operating, except for the change in setting by forced stop.
- If this bit is changed, perform software reset (SCR:UPCL=1).

32.4.4.13 LIN Assist Mode Interrupt Enable Register: LAMIER

This section explains the bit structure of the LIN assist mode interrupt enable register.

LIN assist mode interrupt enable register (LAMIER) enables/disables those interrupts of LIN checksum error, LIN ID parity error, LIN Sync Data error, LIN bus error, LIN checksum operation completion, and LIN automatic header completion,

LAMIERn (n=0 to 2): Address Base addr + $1A_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	LCSERIE	LPTERIE	LSFERIE	LBSERIE	LCSCIE	Reserved	LAHCIE	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	Attribute

[bit7] Reserved bit

Always write "0" to this bit.

[bit6] LCSERIE: LIN checksum error interrupt enable bit

This bit enables/disables the LIN checksum error interrupt request output to the CPU.
When the LCSERIE bit and the LAMESR:LCSER bits are set to "1", the reception interrupt request is output.

LCSERIE	LIN checksum error interrupt enable bit
0	Disable
1	Enable

[bit5] LPTERIE: LIN ID parity error interrupt enable bit

This bit enables/disables the LIN ID parity error interrupt request output to the CPU.
When the LPTERIE bit and the LAMESR:LPTER bits are set to "1", the reception interrupt request is output.

LPTERIE	LIN parity error interrupt enable bit
0	Disable
1	Enable

[bit4] LSFERIE: LIN Sync Data error interrupt enable bit

This bit enables/disables the LIN Sync Data error interrupt request output to the CPU.
When the LSFERIE bit and the LAHESR:LSFER bits are set to "1", the reception interrupt request is output.

LSFERIE	LIN Sync Data error interrupt enable bit
0	Disable
1	Enable

[bit3] LBSERIE: LIN bus error interrupt enable bit

This bit enables/disables the LIN bus error interrupt request output to the CPU.
When the LBSERIE bit and the LAMESR:LBSER bits are set to "1", the reception interrupt request is output.

LBSERIE	LIN bus error interrupt enable bit
0	Disable
1	Enable

[bit2] LCSCIE: LIN checksum arithmetic operations completion interrupt enable bit

This bit enables/disables the LIN checksum arithmetic operations completion interrupt request output to the CPU.
When the LCSCIE bit and the LAMSR:LCSC bits are set to "1", the status interrupt request is output.

LCSCIE	LIN checksum operations completion interrupt enable bit
0	Disable
1	Enable

[bit1] Reserved bit

Always write "0" to this bit.

[bit0] LAHCIE: LIN automatic header completion interrupt enable bit

This bit enables/disables the LIN automatic header completion interrupt request output to the CPU.
When the LAHCIE bit and the LAMSR:LAHC bits are set to "1", the status interrupt request is output.

LAHCIE	LIN automatic header completion interrupt enable bit
0	Disable
1	Enable

32.4.4.14 LIN Assist Mode Transmission/Reception ID register: LAMTID / LAMRID

This section explains the bit structure of the LIN assist mode transmission/reception ID register.

LIN assist mode transmission/reception ID register (LAMTID/LAMRID) sets a value of the transmission LIN ID, indicates a parity value of the received LIN ID, and indicates a value of the received LIN ID.

LIN Assist Mode Transmission ID register (LAMTID)

LAMTIDn (n=0 to 2): Address Base addr + $1B_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	Reserved	LID5	LID4	LID3	LID2	LID1	LID0	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R0,W0	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

[bit7, bit6] Reserved bit

Always write "0" to these bits.

[bit5 to bit0] LID5 to LID0: LIN ID setting bit

(when the LIN ID are written):
When the LIN assist mode is set as master mode and LIN ID register use enable bit (LIDEN) is enabled, these bits set a value of the transmission LIN ID.

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- Be sure to set this setting (SCR:LBR="1") before LIN Break activates.

LIN Assist Mode Reception ID register (LAMRID)

LAMRIDn (n=0 to 2): Address Base addr + $1B_H$ (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
P1	P0	LID5	LID4	LID3	LID2	LID1	LID0	
0	0	0	0	0	0	0	0	Initial value
R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	Attribute

[bit7, bit6] P1, P0: LIN ID parity indication bit

(when parity is read):
In assist mode, these bits indicate a parity value of the received LIN ID.

[bit5 to bit0] LID5 to LID0: LIN ID setting bit

(when ID is read):
In assist mode, these bits indicate a value of the received LIN ID.

Note:

Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").

32.4.4.15 LIN Assist Mode Error Status Register: LAMESR

This section explains the bit structure of the LIN assist mode error status register.

LIN assist mode error status register (LAMESR) checks those flags of the LIN checksum error, the LIN ID parity error, and the LIN bus error.

LAMESRn (n=0 to 2): Address Base addr + 18_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
Reserved	LCSER	LPTER	LSFER	LBSER	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R0,W0	R,W	R,W	R,W	R,W	R0,W0	R0,W0	R0,W0	Attribute

[bit7] Reserved bit

Always write "0" to this bit.

[bit6] LCSER: LIN checksum error flag bit

When the LIN checksum error occurs, this bit is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LCSER bit and the LCSERIE bits are set to "1", the reception interrupt request is output.

LCSER	LIN checksum error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read.
- Even if a framing error is detected, the result of the checksum arithmetic operation is indicated. The result, however, is not guaranteed.

[bit5] LPTER: LIN ID parity error flag bit

When the LIN ID parity error is generated, it is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LPTER bit and the LPTERIE bit are set to "1", the reception interrupt request is output.
When this flag is read and found to be "1", the reception ID data is invalid.

LPTER	LIN ID parity error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read.
- Even if a framing error is detected, the result of the checksum arithmetic operation is indicated. The result, however, is not guaranteed.

[bit4] LSFER: LIN Sync Data error flag bit

When the LIN Sync Data error occurs, it is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LSFER bit and the LSFERIE bits are set to "1", the reception interrupt request is output.

LSFER	LIN Sync Data error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read
- This function detects an error only at automatic baud rate adjustment disabled (SACSR:AUTE="0") in slave mode (SCR:MS=1).
- Even if a framing error is detected at automatic baud rate adjustment disabled (SACSR:AUTE="0"), the result of the Sync Field value (0x55) collation is indicated. The result, however, is not guaranteed.

[bit3] LBSER: LIN bus error flag bit

When the LIN bus error occurs, this bit is set to "1".
Be sure to write "0" in order to clear this error flag bit.
When the LBSER bit and the LBSERIE bits are set to "1", the reception interrupt request is output.
When this flag is read and found to be "1" in ID Field and the data field, the data of reception data register (RDR) is invalid.

LBSER	LIN bus error flag bit	
	write	read
0	Clear error flag	No error
1	No influence	Error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- When this register is accessed in accordance with read-modify-write instructions, "1" is read.
- Even if a framing error is detected, the result of the checksum arithmetic operation is indicated. The result, however, is not guaranteed.
- When a bus error is detected in the ID Field, the LIN ID parity error is detected as well.
- When a bus error is detected by checksum, the LIN checksum error is detected as well.

[bit2 to bit0] Reserved bit

Always write "0" to these bits.

32.4.4.16 LIN Assist Mode trouble Examination Register: LAMERT

This section explains the bit structure of the LIN assist mode trouble examination register.

LIN assist mode trouble examination register (LAMERT) sets a pseudo trouble of the framing error, the LIN bus error, the LIN Sync Field error, the LIN ID parity error, and the LIN checksum error by setting the key code control bit and the pseudo trouble setting bit.

LAMERTn (n=0 to 2): Address Base addr + 19_H (Access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
KEY1	KEY0	Reserved	LCSERT	LPTERT	LSFERT	LBSERT	FRET	
0	0	0	0	0	0	0	0	Initial value
R0,W	R0,W	R0,W0	R/W	R/W	R/W	R/W	R/W	Attribute

[bit7, bit6] KEY1, KEY0: Key code control bit

These key code bits make the following pseudo trouble settings effective:

- Framing error pseudo trouble setting bit (FRET)
- LIN bus error pseudo trouble setting bit (LBSERT)
- LIN Sync Field error pseudo trouble setting bit (LSFERT)
- LIN ID parity error pseudo trouble setting bit (LPTERT)
- LIN checksum error pseudo trouble setting bit (LCSERT)

When setting a pseudo trouble, write to these bits in accordance with the following procedures:

1. KEY1-0="00"+ the pseudo trouble setting value is written.
2. KEY1-0="01"+ the pseudo trouble setting value (the same value as before) is written.
3. KEY1-0="10"+ the pseudo trouble setting value (the same value as before) is written.
4. KEY1-0="11"+ the pseudo trouble setting value (the same value as before) is written.
5. When being written in the fourth time, the pseudo trouble setting value becomes effective.

When the above-mentioned setup procedures are not followed (when another register is written or read in the middle of the writing procedures, when the value written is incorrect, or when these two bits are read in the middle of the writing procedures), writing to these two bits becomes invalid.

When releasing the pseudo trouble setting, follow the procedures similar to those to set.
As for the read value, "0" is read.

Note:

When the following error occurs in assist mode, the assist mode will stop.

- LIN bus error
- LIN framing error
- LIN Sync Data error
- LIN ID parity error
- LIN checksum error

[bit5] Reserved bit

Always write "0".

[bit4] LCSERT: LIN checksum error pseudo trouble setting bit

This bit controls occurrence of the LIN checksum error.
In assist mode, when this bit is set to "1" (errors occur) before the start bit of the checksum, the inverted checksum is outputted. When the inverted checksum is received, an LIN checksum error occurs and the flag bit (LAMESR:LCSER) is set to "1".
Until the setting of this bit is released (="0"), the pseudo trouble function is effective and causes errors.

LCSERT	LIN checksum error pseudo trouble setting bit
0	Occurrence of error
1	Non occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- The framing error is detected in the data field when the response transmission is processed after a pseudo trouble of the framing error and the LIN checksum error is set, and an automatic transmission of checksum stops.
- Therefore, the LIN checksum error is not detected.

[bit3] LPTERT: LIN ID parity error pseudo trouble setting bit

This bit controls occurrence of the LIN ID parity error.
In assist mode, when this bit is set to "1" (errors occur), before the start bit of the ID Field , all the inverted bits of the ID parity are output. When ID Field of the inverted ID parity is received, the LIN ID parity error occurs, and the flag bit (LAMESR:LPTER) is set to "1".
Until the setting of this bit is released (="0"), the pseudo trouble function is effective and causes errors.

LPTERT	LIN ID parity error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- After a pseudo trouble of the framing error and the LIN ID parity error is set, when the automatic header transmission is done, a framing error is detected in the Sync Field, and then the automatic transmission stops. As a result, the LIN ID parity error is not detected.
- When a pseudo trouble of the LIN bus error and the LIN ID parity error is set, the LIN bus error is given priority. As a result, the LIN ID parity error is not detected

[bit2] LSFERT: LIN Sync Data error pseudo trouble setting bit

This bit controls the occurrence of the LIN Sync Data error.
When the LIN assist mode is set as master mode (SCR:MS= "0") and this bit is set to "1" (errors occur) before the start bit of the Sync Field, all bits of the LIN Sync Field are output inverted.
Until the setting of this bit is released (="0"), the pseudo trouble function is effective and the output from the Sync Field is being kept inverted.

LSFERT	LIN Sync Data error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- If setting this bit (LSFERT="1"), set it together with the LIN bus error pseudo trouble setting bit (LBSERT="1"). For the master, the completion of the LIN Sync Data error pseudo trouble transmission can be identified with the detection of a LIN bus error.

[bit1] LBSERT: LIN bus error pseudo trouble setting bit

This bit controls the occurrence of the LIN bus error.
When the LIN assist mode is set as master mode, the LIN bus error will occur and the flag bit (LAMESR:LBSER) will be set to 1 in the following case:
This bit is set to "1" (errors occur) before the stop bit of each field (Sync Field, ID Field, data, and checksum).
When the LIN assist mode is set as slave mode, if this bit is set to "1" (errors occur) before the stop bit of each field (data and checksum) that does the response transmission, the LIN bus error will occur, and then the flag bit (LAMESR:LBSER) is set to "1".
Until the setting of this bit is released (="0"), the pseudo trouble function is effective and generates the error.

LBSERT	LIN bus error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- LIN bus error pseudo trouble of LIN Break Field cannot be set.
- When the LIN bus error and the LIN checksum error or the LIN ID parity error are set as a pseudo trouble at the same time, only the LIN bus error is detected. Neither the LIN checksum error nor the LIN ID parity error is detected.
- When the LIN bus error and the LIN Sync Data error or the framing error is set as a pseudo trouble before the LIN transmission start is set (SCR:LBR="1") at the same time, the LIN bus error and the LIN Sync Data error or the framing error is detected at the same time.
- When the LIN bus error is detected, transmission and reception in the assist mode will stop.

[bit0] FRET: Framing error pseudo trouble setting bit

This bit controls the occurrence of the LIN framing error.
In the assist mode, if this bit is set to "1" (errors occur) before the stop bit of each field (Sync Field, ID field, data field, and checksum field), the stop bit will be output inverted. When the inverted stop bit is received, the framing error will occur, and then the flag bit (SSR:FRE) is set to "1".
Until the setting of this bit is released (="0"), the pseudo trouble function is effective and generates the error.

FRET	Framing error pseudo trouble setting bit
0	Non occurrence of error
1	Occurrence of error

Notes:

- Function of this register is effective only in the LIN assist mode (LAMCR:LAMEN="1").
- After a pseudo trouble of the framing error and the LIN ID parity error is set, when the automatic header transmission is done, a framing error is detected in the Sync Field, and then the automatic transmission stops. As a result, the LIN ID parity error is not detected.
- After a pseudo trouble of the framing error and the LIN checksum error is set, when the response transmission is processed, the framing error is detected in the data field, and then the automatic transmission of checksum stops. As a result, the LIN checksum error is not detected.

32.5 Operation of UART

This section explains operation of UART.

32.5.1 Interrupt of UART

Interrupt of UART is shown.

There are interrupts for both transmission and reception in UART. You can generate an interrupt request for the following factors.

- Setting of reception data in the receive data register (RDR) or occurrence of a reception error
- Start of transmission after transfer of transmission data from the transmit data register (TDR) to the transmit shift register
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request

32.5.1.1 List of Interrupt of UART

This section explains the list of interrupt of UART.

Table 32-4 indicates how UART interrupt control bits relate to interrupt factors.

Table 32-4. Interrupt Control Bits and the Interrupt Factors of UART

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt factor	Interrupt factor enable bit	Interrupt request flag clear
			0	1			
Reception	RDRF	SSR	✓	✓	1-byte reception	SCR:RIE	Reading of receive data (RDR)
					Reception of as much data as specified by FBYTE		Reading of receive data (RDR) until the reception FIFO is emptied
					Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	✓	✓	Overrun error		Writing of "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	✓	✓	Framing error		
	PE	SSR	✓	-	Parity error		
Transmission	TDRE	SSR	✓	✓	Transmission register is empty	SCR:TIE	Writing to the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	TBI	SSR	✓	✓	No transmission operation	SCR:TBIE	Writing the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	FDRQ	FCR1	✓	✓	The storage data value of the transmission FIFO is FTICR setting value or less, or empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or the transmission FIFO is full
Status	TINT	SACSR	✓	✓	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR:TINTE	Writing "0" to the timer interrupt flag bit (SACSR:TINT)

* Set the TIE bit to "1" after the TDRE bit is cleared to "0".

✓: Operation mode effective

-: Operation mode non-effective

32.5.1.2 Reception Interrupts and Flag Setting Timing

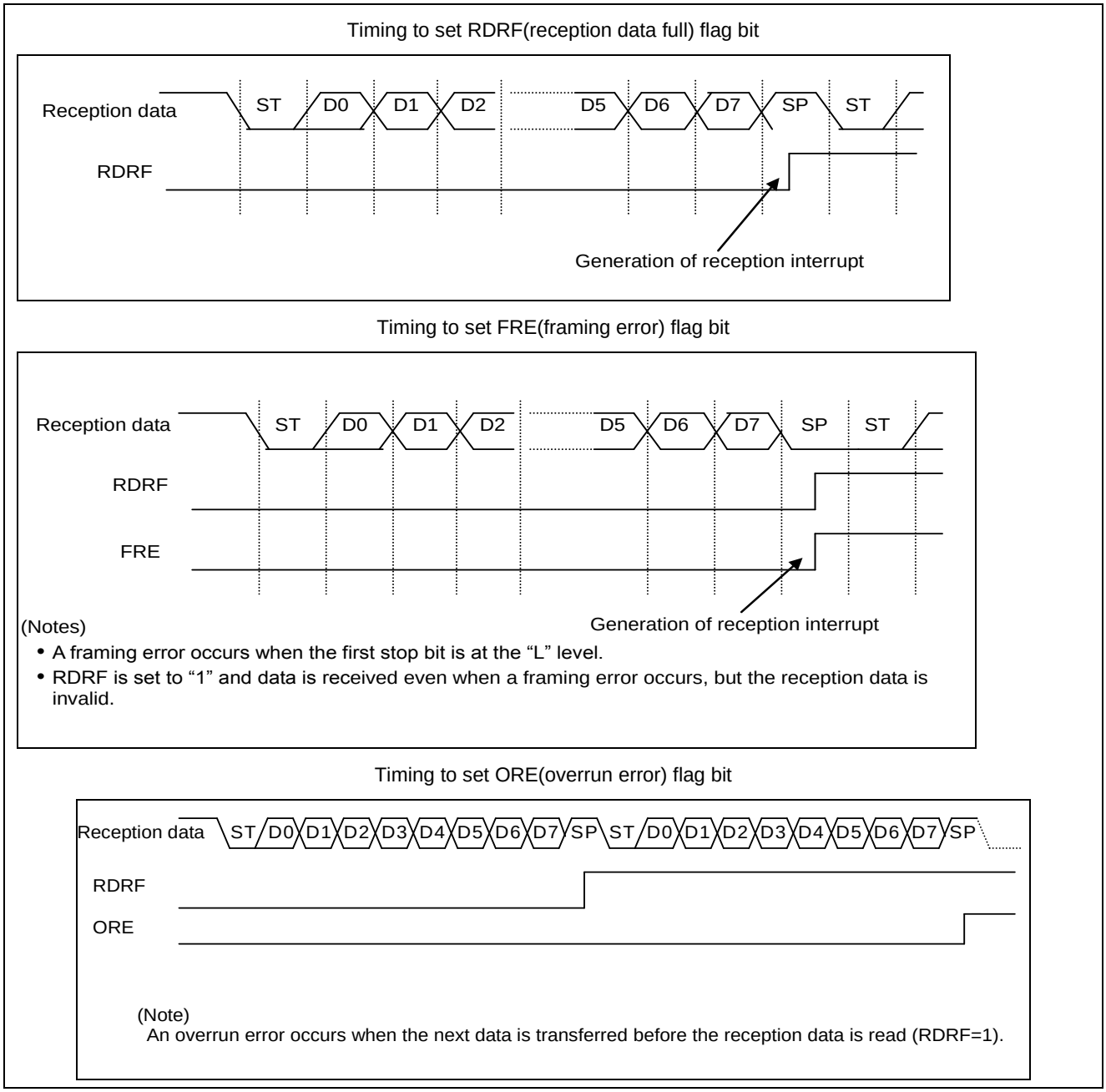
This section explains the generation of reception interrupts and flag setting timing.

Reception interrupts occur either when the reception is completed (SSR:RDRF) or when a reception error occurs (SSR:PE, ORE, FRE).
When the first stop bit is detected, reception data is stored in the receive data register (RDR). When reception is completed (SSR:RDRF=1) or a reception error occurs (SSR:PE, ORE, FRE=1), a corresponding flag is set. If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt occurs.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

Figure 32-2. Timing of Flag Bit Setting



Note:

When any of following conditions is detected while receiving at the same time of or 1 to 2 bus clocks before the sampling point for stop bit, its edge will be invalid and the next data may not be received correctly. To output frames continuously, leave some space between the frames.

- Trailing edge of serial data (when ESCR:INV="0")
- Rising edge of serial data (when ESCR:INV="1")

32.5.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing

This section explains the generation of interrupts when using reception FIFO and flag setting timing.

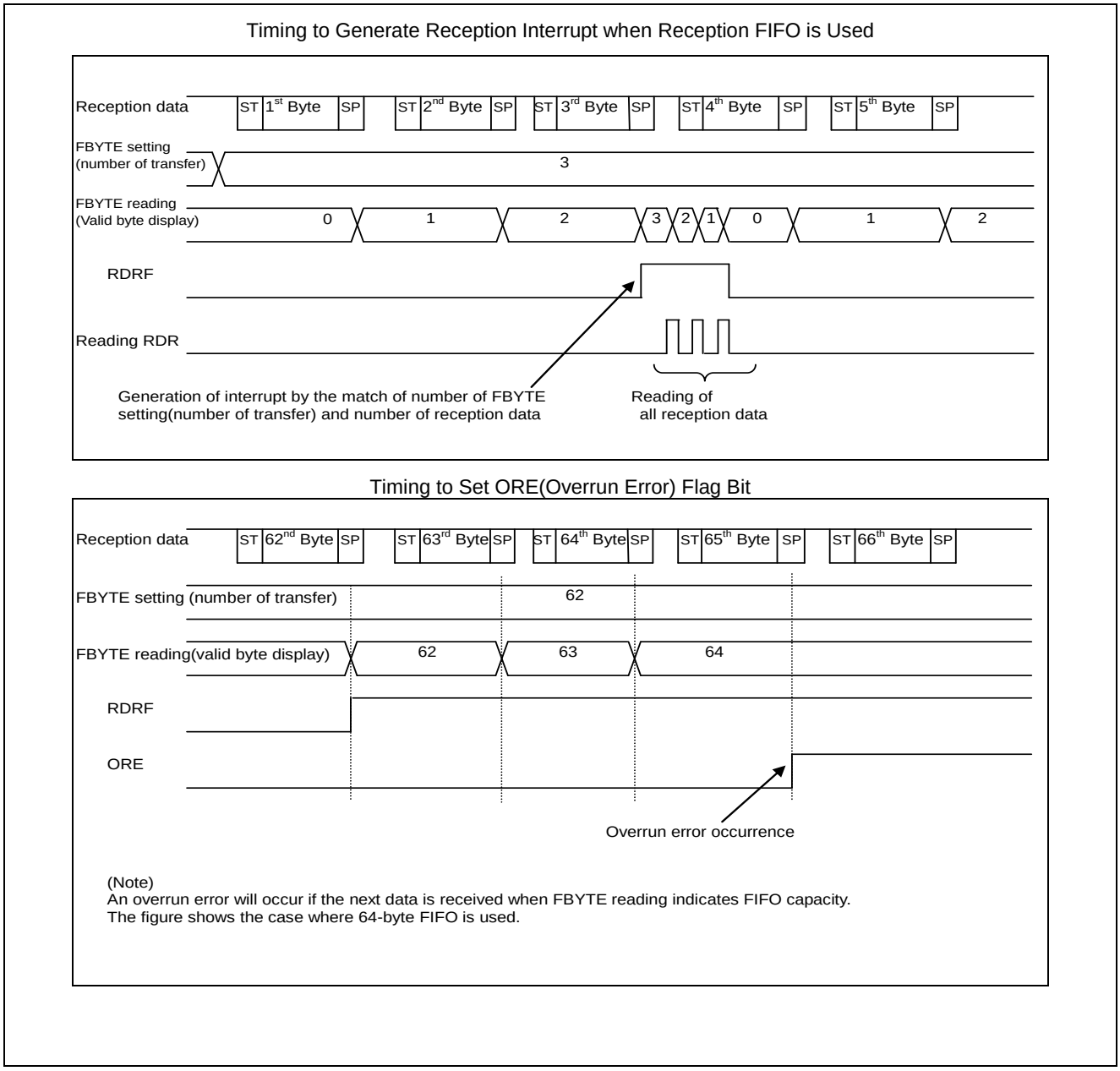
When the reception FIFO is used, an interrupt occurs after as much data as the FBYTE register (FBYTE) setting is received. The setting value of the FBYTE register determines the occurrence of an interrupt when the reception FIFO is used.

- After as much data as the transfer count setting of the FBYTE register is received, the reception data full flag of the serial status register (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SCR:RIE) at this time, a reception interrupt is generated.
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (SSR:RDRF) will be set to "1".
  - ☐ Reception FIFO idle detection enable bit (FCR:FRIIE) is "1"
  - ☐ Data count contained in the reception FIFO does not reach the transfer count

If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

- If the receive data (RDR) is read until the reception FIFO is empty, the reception data full flag (SSR:RDRF) is cleared.
- When the reception-enabled data count indication has shown the FIFO capacity, receiving the next data will generate an overrun error (SSR:ORE=1) .

Figure 32-3. Timing of Using FIFO



32.5.1.4 Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and flag setting timing.

Transmission interrupts occur either when transmission is started after transfer of transmission data from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) or when the transmission operation is idle (SSR:TBI=1).

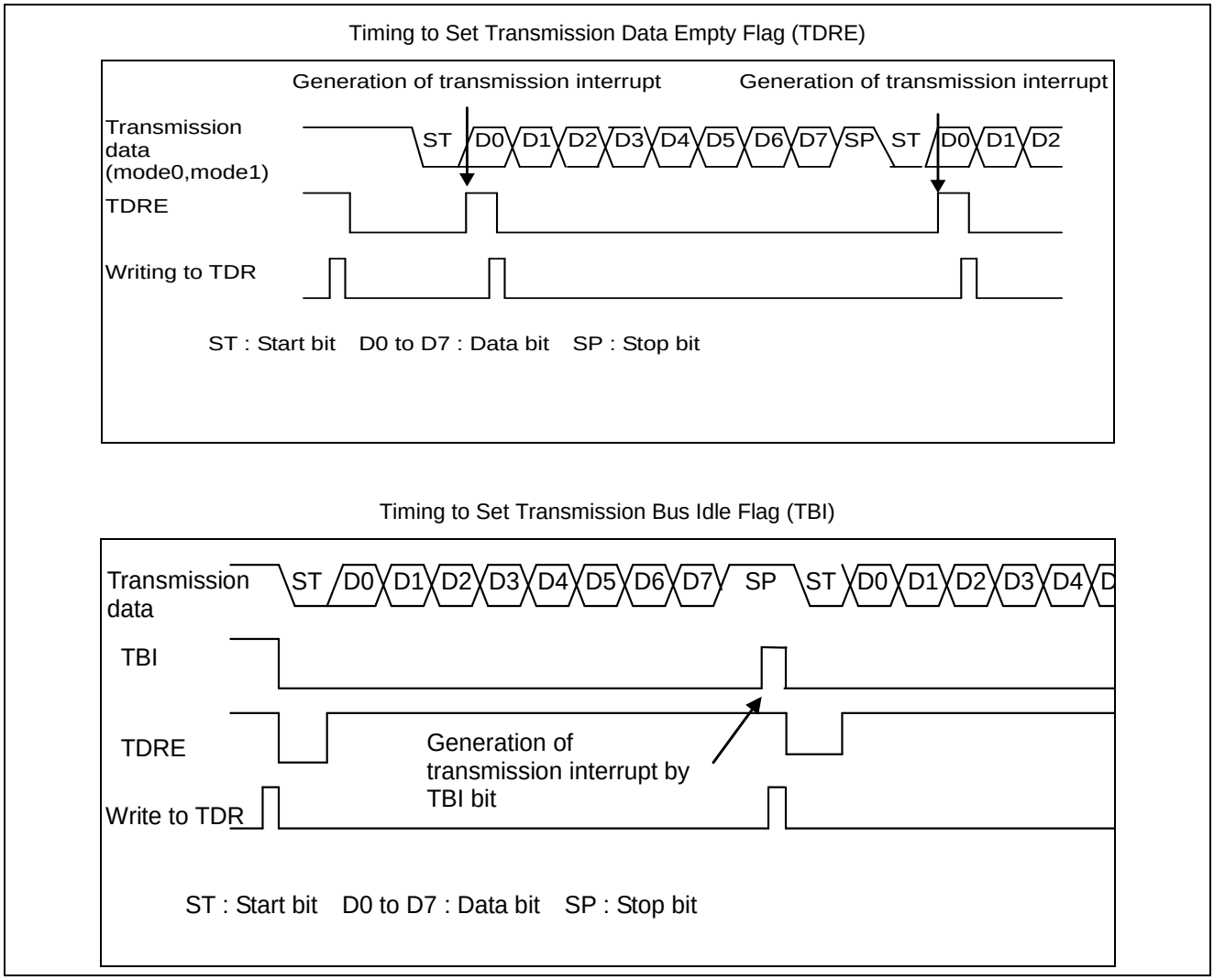
When data written to the transmit data register (TDR) is transferred to the transmit shift register, writing of next data is enabled (SSR:TDRE=1). If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs. The TDRE bit, being a read-only bit, is cleared to "0" by writing of data to the transmit data register (TDR).

When the transmit data register is empty (TDRE=1) and no transmission operation is in progress, the SSR:TBI bit is set to "

1". If transmission bus idle interrupt is enabled (SCR:TBIE=1) at this time, a transmission interrupt will occur.

When transmission data is written to the transmit data register (TDR), the SSR:TBI bit and the transmission interrupt request are cleared.

Figure 32-4. Timing of Transmission Interrupt Flag



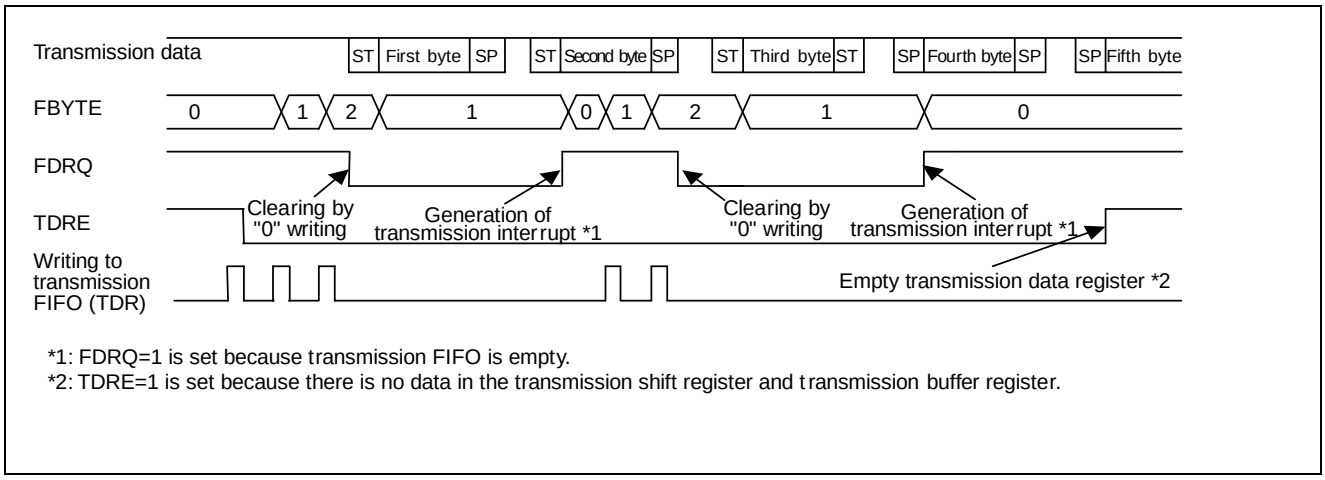
32.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

This section explains the generation of interrupts when using transmission FIFO and flag setting timing.

When the transmission FIFO is used, an interrupt is generated when the data count stored at the transmission FIFO is equal to or less than the count set for the FTICR register (FTICR).
When the transmission FIFO is used, the interrupt generation is decided depending on the FTICR register setting value.

- When the storage data value of the transmission FIFO is FTICR register (FTICR) setting value or less, the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1".
- If FIFO transmission interrupt is enabled (FCR1:FTIE="1") at this time, a transmission interrupt will occur.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
- When FBYTE=0x00 and FTICR=0x00, there is no data in the transmission FIFO.

Figure 32-5. Timing of Transmission Interrupts when Using Transmission FIFO



*1: FDRQ=1 is set because transmission FIFO is empty.
*2: TDRE=1 is set because there is no data in the transmission shift register and transmission buffer register.

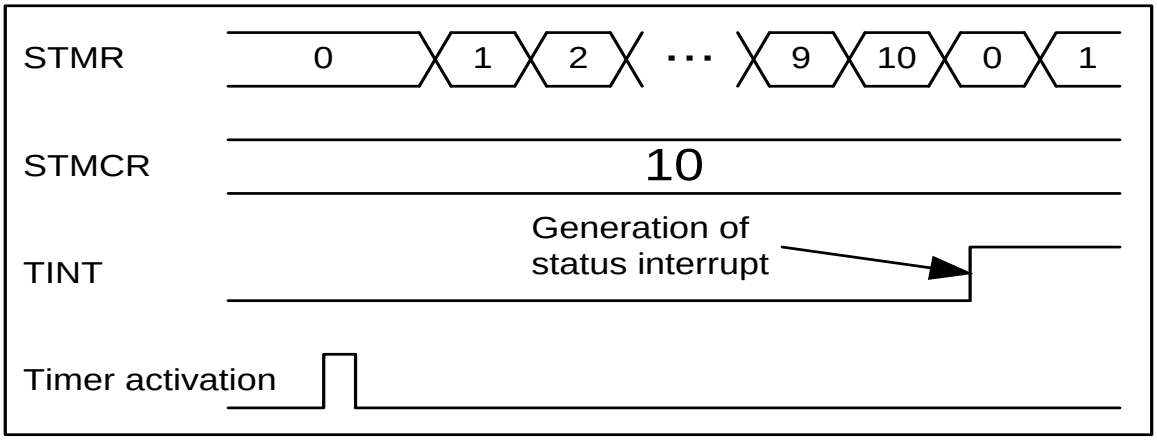
32.5.1.6 Timing of Timer Interrupt Generation and Flag Setting

This section explains the timing of timer interrupt generation and flag setting.

Timer interrupt is generated when Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR).

- When Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR), "1" will be set to timer interrupt flag (SACSR:TINT).
  At this time when the timer interrupt is enabled (SACSR:TINTE="1"), a status interrupt will be generated.

Figure 32-6. Timer Interrupt Generation Timing



32.5.2 Operation of UART

Operation of UART is shown.

UART operates with the mode 0 bidirectional serial asynchronous communication and the mode 1 master/slave multiprocessor communication.

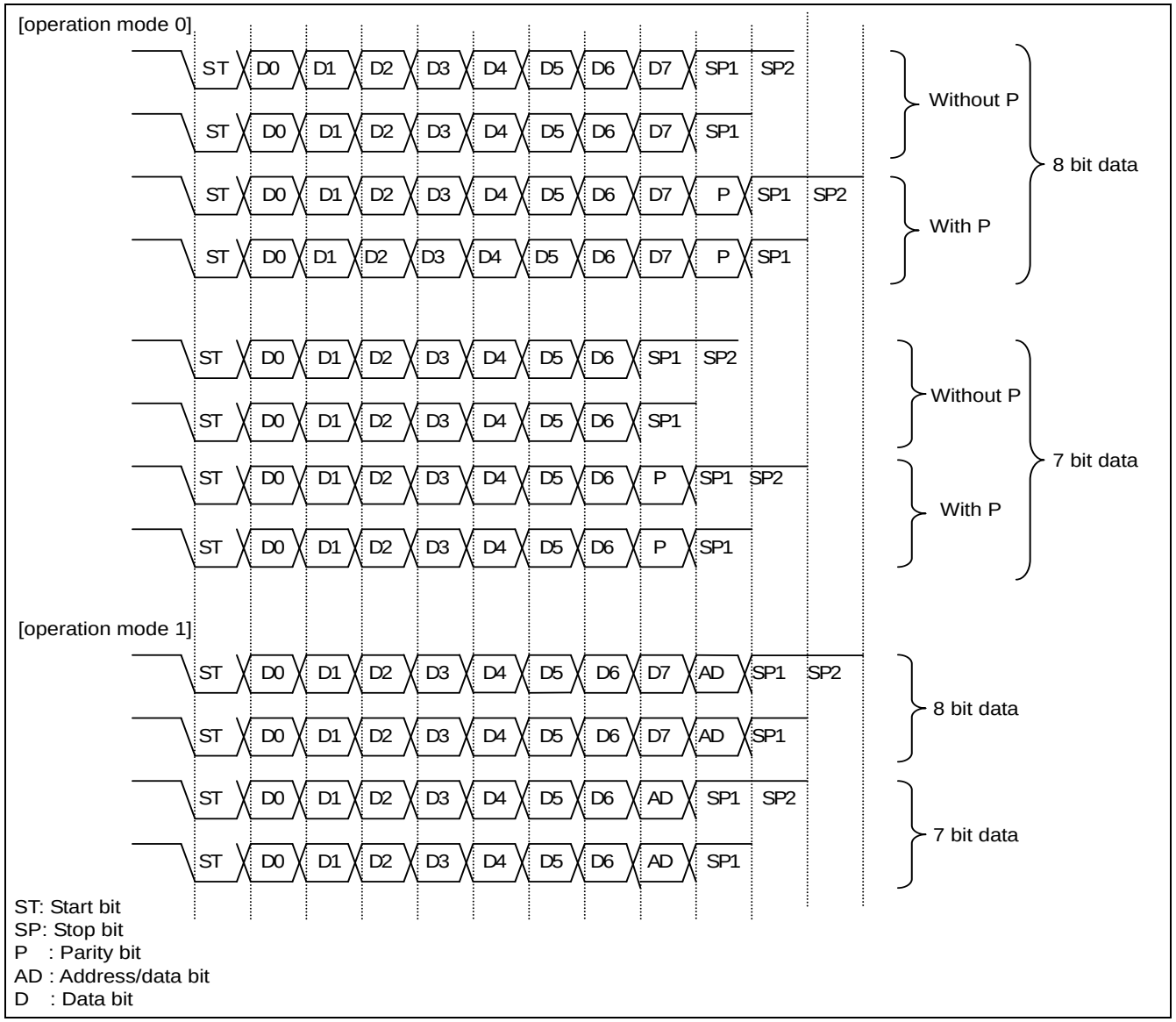
32.5.2.1 Transmission/Reception Data Format

This section explains the transmission/reception data format.

- The transmission/reception data always starts from the start bit and after the transmission/reception of data have taken place for the specified data bit length, ends at 1-bit or more length of stop bit.
- The direction of data transfer (LSB first or MSB first) is determined by the BDS bit of the serial mode register (SMR). If parity is used, the parity bit will always be placed between the last data bit and the first stop bit.
- In operation mode 0 (normal mode), you can select whether to use parity.
- In operation mode 1 (multiprocessor mode), the parity will not be added, instead AD bits will be added.

An example of transmission/reception data format (operation modes 0, 1) is shown in Figure 32-7:

Figure 32-7. Example of Transmission/Reception Data Format (Operation Modes 0, 1)



Notes:

- The Figure above shows the example of configurations with data length of 7 and 8 bits. (You can configure 5 to 9-bit data length in operation mode 0.)
- When you set "1" to the BDS bit of serial mode register (SMR) (MSB first), the bits will be processed in the order, D7, D6, D5, ..., D1, D0 (P).
- When you configure x bit of data length, the lower x bits on transmission/receive data register (RDR/TDR) will be enabled.

32.5.2.2 Transmission Operation

This section explains the transmission operation.

- If the transmission data empty flag bit (TDRE) of the serial status register (SSR) is "1", the transmission data can be written to the transmit data register (TDR). (If the transmission FIFO is enabled, transmission data can be written even if TDRE="0").
- When transmission data is written to the transmit data register (TDR), the transmission data empty flag bit (SSR:TDRE) becomes "0".
- When the transmission operation enable bit of the serial control register (SCR:TXE) is set to "1", the transmission data is loaded into the transmit shift register and the transmission starts from the start bit sequentially.
- When the transmission starts, the transmission data empty flag bit (SSR:TDRE) will be set to "1" again. If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs. In interrupt processing, the next transmission data can be written to the transmit data register.

Notes:

- As soon as the transmission interrupt is enabled (SCR:TIE), a transmission interrupt occurs, because the transmission data empty flag bit (SSR:TDRE) has the initial value "1".
- As soon as the FIFO transmission interrupt is enabled (FCR1:FTIE=1), a transmission interrupt occurs, because the FIFO transmission data request bit (FCR1:FDRQ) has the initial value "1".

32.5.2.3 Reception Operation

This section explains the reception operation.

- When reception operation is enabled (SCR:RXE=1), the reception operation will start.
- When a start bit is detected, one frame data will be received according to the data format set in the extended communication control register (ESCR:PEN, P, L2, L1, L0) and serial mode register (SMR:BDS). The start bit is detected when the falling edge (at ESCR:INV="0") or the rising edge (at ESCR:INV="1") is detected after data passes the noise filter (majority decision by sampling the serial data input with the bus clock three times), and the passed data detects "L" at the sampling point.
- When the reception of one frame data has completed, the reception data full flag bit (SSR:RDRF) will be set to "1". If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt occurs.
- Read reception data, after the one frame data reception has completed, and check for the state of error flag of the serial status register (SSR). When a reception error has detected, correct the error.
- After a read of reception data, the reception data full flag bit (SSR:RDRF) will be cleared to "0".
- When reception FIFO is enabled, if as many frames as set in the reception FBYTE have been received, the reception data full flag bit (SSR:RDRF) will be set to "1".
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (RDRF) will be set to "1".
  - ☐ Reception FIFO idle detection enable bit (FRIIE) is "1"
  - ☐ Data count contained in the reception FIFO does not reach the transfer count

  If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

  When the reception FIFO is enabled, if the error flag of the serial status register (SSR) is set to"1", the erroneous data will not be stored in the reception FIFO. Also, the reception data full flag bit (SSR:RDRF) at that time will not be set to "1". (However, when an overrun error does occur, the flag will be set to "1".) The reception FBYTE indicates the number of data items which have been successfully received before the error occurs. Unless the error flag of the serial status register (SSR) is cleared to "0", the reception FIFO will not be enabled.
- When the reception FIFO is enabled, if the reception FIFO has no more data, the reception data full flag bit (SSR:RDRF) will be cleared to "0".

Notes:

- The data on the receive data register (RDR) will be enabled when the receive data register full flag bit (SSR:RDRF) is set to "1" and a reception error does not occur (SSR:PE, ORE, FRE=0).
- When the noise passes the filter, the incorrect data is received though the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in. As measures against this, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, add the checksum of data at the end, and send it again if an error occurs).
- When any of following conditions is detected while receiving at the same time of or 1 to 2 bus clocks before the sampling point for stop bit, its edge will be invalid and the next data may not be received correctly. To output frames continuously, leave some space between the frames.
  - ☐ Trailing edge of serial data (when ESCR:INV="0")
  - ☐ Rising edge of serial data (when ESCR:INV="1")

32.5.2.4 Clock Selection

This section explains the clock selection.

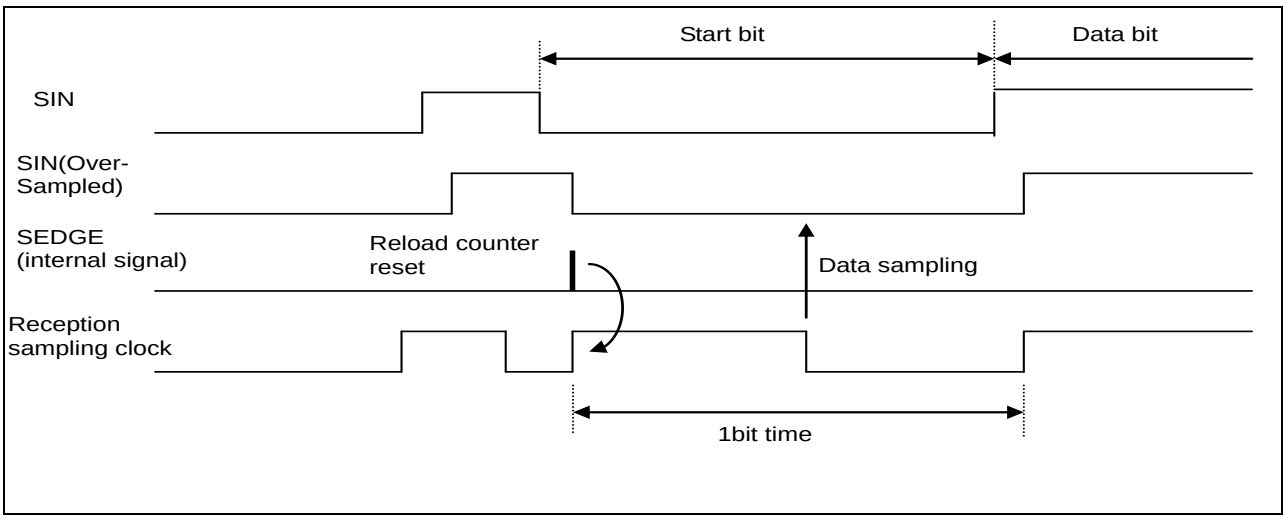
- Internal clocks or external clocks can be used.
- When you use an external clock, set BGR:EXT="1". In this case, the external clock is divided in the baud rate generator.

32.5.2.5 Start Bit Detection

This section explains the start bit detection.

- The start bit is recognized by the falling edge of the SIN signal in asynchronous mode.
  Therefore even if you enable reception operation (SCR:RXE="1"), the reception operation will not start unless the falling edge of the SIN signal is entered.
- When the falling edge of the start bit is detected, the reception reload counter of the baud rate generator will be reset, a reload will take place again, and the countdown will start. This will always launch a data sampling aimed at the center of the data.

Figure 32-8. Start Bit Detection



32.5.2.6 Stop Bit

This section explains the stop bit.

- You can select 1-4 bit length.
- The reception data full flag bit (SSR:RDRF) will be set to "1" when the first stop bit is detected.

32.5.2.7 Error Detection

This section explains the error detection.

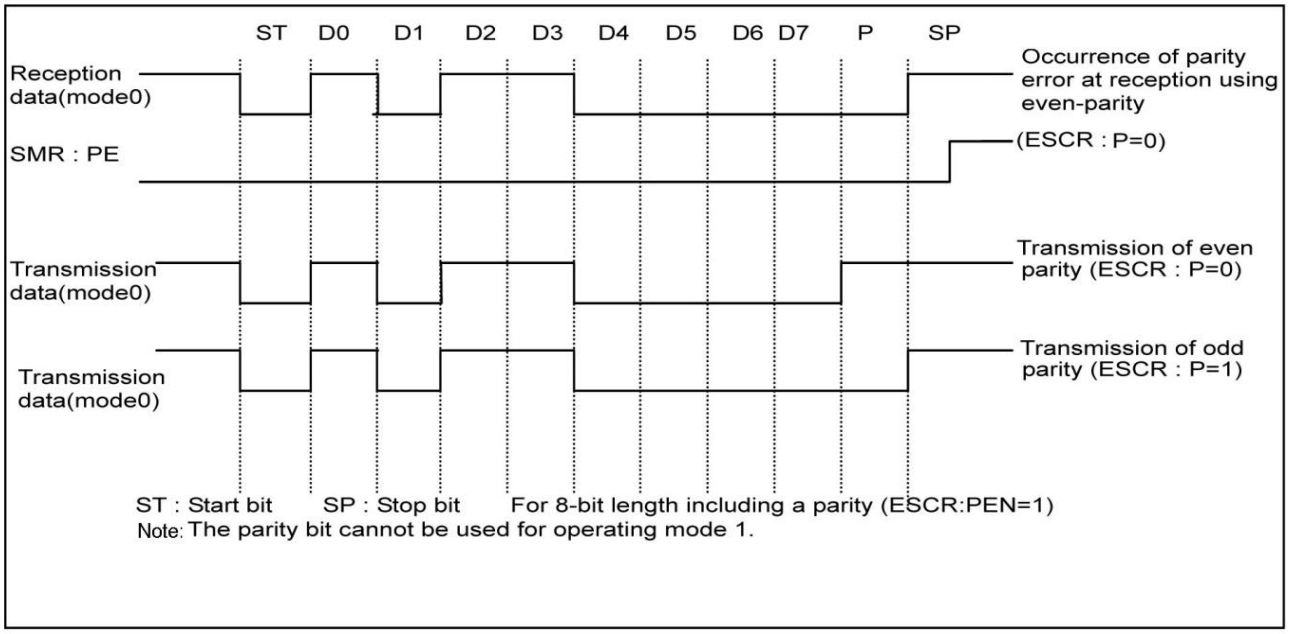
- In operation mode 0, parity errors, overrun errors, frame errors can be detected.
- In operation mode 1, overrun errors and frame errors can be detected. Parity errors cannot be detected.

32.5.2.8 Parity Bit

This section explains the parity bit.

- Parity bit can be added only in operating mode 0. The parity enable bit (ESCR:PEN) can specify whether to enable or disable the parity, and the parity selection bit (ESCR:P) can specify whether to use even parity or odd parity.
- Operation mode 1 does not use parity.

Figure 32-9. Operation with Parity Enabled

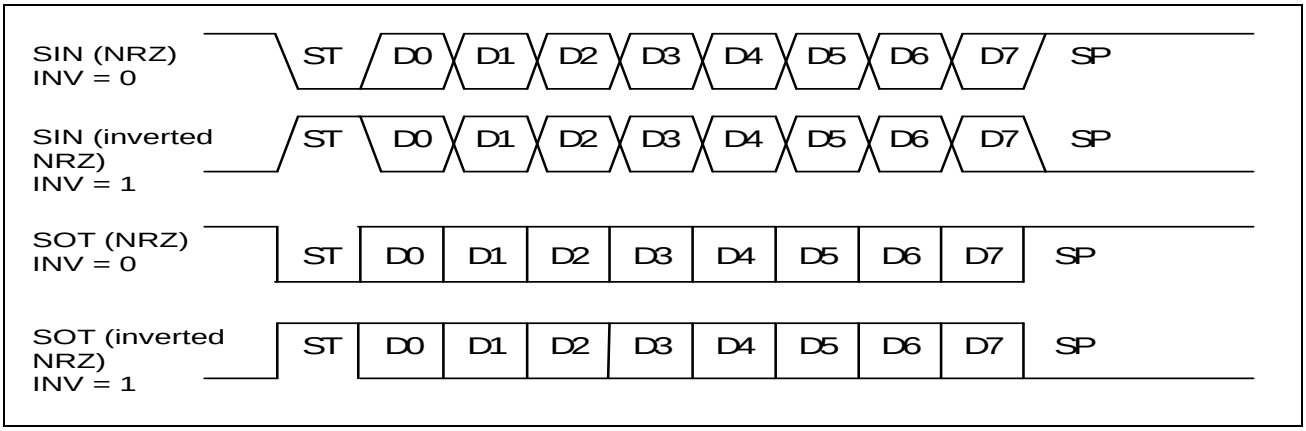


32.5.2.9 Data Signaling Method

This section explains the data signaling method.

The INV bit setting of the extended communication control register enables you to select the NRZ (Non Return to Zero) signaling method (ESCR:INV=0) or the inverted NRZ signaling method (ESCR:INV=1).

Figure 32-10. NRZ (Non Return to Zero) Signaling Method and Inverted NRZ Signaling Method



32.5.2.10 Operation of Serial Timer

This section explains the operation of the serial timer.

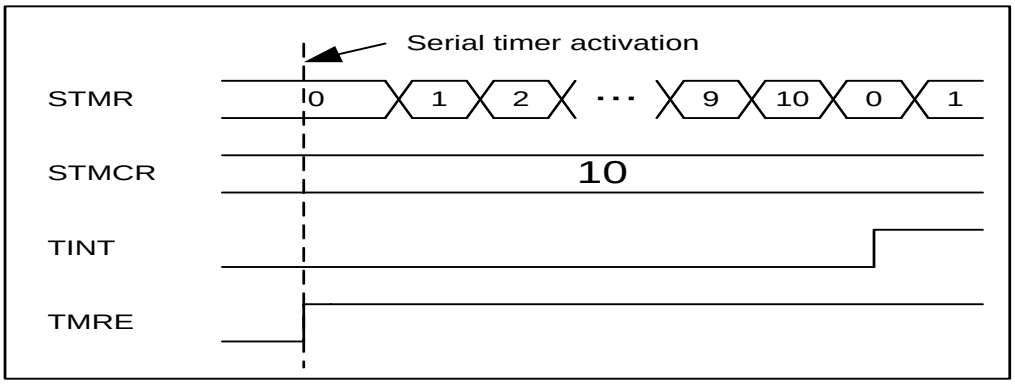
The serial timer can be used for either of the timer function or the synchronous transmission function.

How to Start Serial Timer

To start the serial timer: setting "1" to the serial timer enable bit (SACSR:TMRE).

- Start by using the serial timer enable bit (SACSR:TMRE)
  When the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer starts and the serial timer register (STMR) starts counting from 0.

Figure 32-11. Start by Using Serial Timer Enable Bit (STMCR="10",)



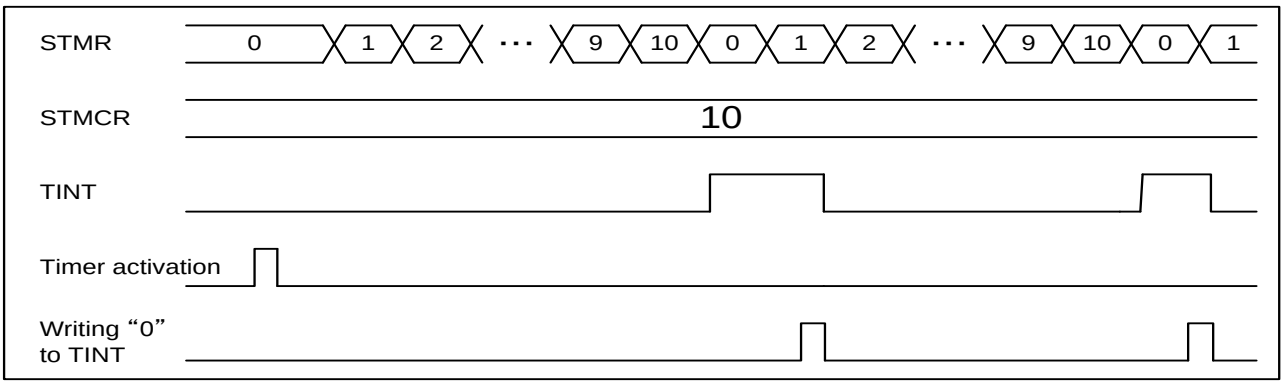
How to Stop Serial Timer

When the serial timer enable bit (SACSR:TMRE) is set to "0", the serial timer will stop. The value of the serial timer register (STMR) is retained.

Timer Operation

The serial timer operates as a timer.
If the serial timer register (STMR) matches the serial timer comparison register (STMCR), the timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to "0".

Figure 32-12. Timer Operation (STMCR="10")



Note:

When the timer comparison register (STMCR) is set to $(0000)_H$, the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B".

32.5.2.11 Test Mode

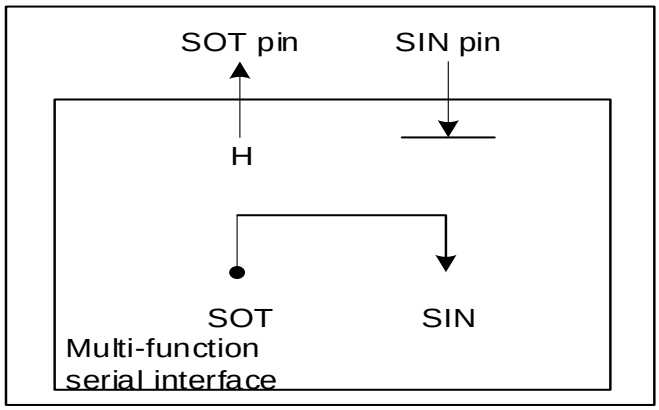
This section explains the test mode.

This section explains the operation of the test mode.

Serial Test Mode

When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and then the data sent from SOT can be received from SIN directly.
When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 32-13. Serial Test Mode



Note:

The serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

32.5.2.12 UART Baud Rate Selection/Setting

This section explains the UART baud rate selection/setting.

The UART transmission/reception baud rate generator can be configured for the settings below.

Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the internal clock

There are two internal reload counters that correspond to the transmission and reception serial clocks, respectively. The baud rate can be selected by setting a 15-bit reload value in the baud rate generator register (BGR).
The reload counter divides the internal clock with the set value.
To configure the clock source, select the internal clock (BGR:EXT=0).

Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the external clock

Use the external clock for the clock source of reload counter. The baud rate can be selected by setting a 15-bit reload value in the baud rate generator register (BGR).

The reload counter divides the external clock with the set value.

To configure the clock source, select the external clock and the baud rate generator clock (BGR:EXT=1).

This mode is designed to accommodate the case where the division of an oscillator of a special frequency is used.

Notes:

- Configure the external clock (EXT=1) after stopping the reload counter (BGR=15´h00).
- When an external clock (EXT=1) has been set, the "H" width and "L" width of the external clock should be set to 2 bus clocks or more.

Baud Rate Calculation

Set two 15-bit reload counters in the baud rate generator register (BGR).
The baud rate calculation formulas are as follows:

(1) Reload value

$V = \Phi / b - 1$

V: Reload value
b: Baud rate
Φ: Bus clock frequency, external clock frequency

(2) Example of calculation

Reload values when setting the bus clock frequency at 16 MHz, usage of internal clock, and baud rate at 19200 bps are as follows:

Reload value:

$V = (16 \times 1{,}000{,}000) / 19200 - 1 = 832$

The baud rate is:

$b = (16 \times 1{,}000{,}000) / (832+1) = 19208$ bps

(3) Baud rate error

The baud rate error can be obtained using the following formula:

Error (%) = (calculated value - desired value) / desired value × 100

(Example) Bus clock 20MHz, Target baud rate value 153600 bps

Reload value = (20 × 1,000,000) / 153600 - 1 = 129
Baud rate (calculated value) = (20 × 1,000,000) / (129 + 1) = 153846 bps
Error (%) = (153846 - 153600)/ 153600 × 100 = 0.16(%)

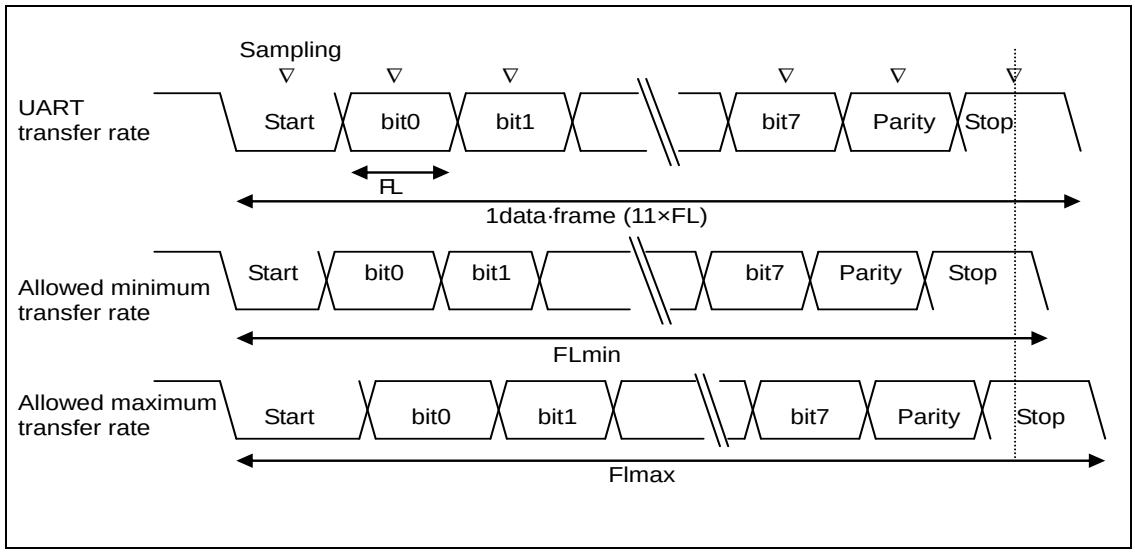
Notes:

- Set the reload value to "0" to stop the reload counter.
- If the reload value is an even number, the "L" width of the reception serial clock is 1 bus clock longer than "H" width. If it is an odd number, the "H" and "L" widths of the serial clock are equal.
- Set the reload value to 4 or higher. A normal data reception operation, however, could not be achieved due to some baud rate error and reload value settings.

Allowed Baud Rate Error Range at Reception

This section explains the amount of the destination baud rate error that can be allowed at reception.
The baud rate error at reception should be set within the allowed error range by using following formula.

Figure 32-14. Allowed Baud Rate Range at Reception



As shown in the Figure the counter set by the BGR register will determine the sampling timing of the reception data after having detected a start bit. A normal reception operation can be achieved if the last data (stop bit) have been completed within this sampling timing.

In theory, the following is expected when this is applied to 11-bit reception.

If the margin of sampling timing is 1 clock of bus clock (Φ), the allowed minimum transfer rate (FLmin) would be calculated as follows.

FLmin = (11bit × (V+1) – (V+1) / 2 +2) / Φ = (21V+25) / 2Φ (s)

V: Reload value Φ: Bus clock

Therefore, the allowed maximum baud rate (BGmax) at the destination would be calculated as follows.

BGmax = 11/FLmin = 22Φ / (21V+25) (bps)

V: Reload value Φ: Bus clock

When receiving data at the allowed maximum transfer rate (FLmax), sampling is done in the starting point of receive data in the 11th bit.

Therefore, the allowed maximum transfer rate (FLmax) is as follows.

10/11 × FLmax = (11bit × (V+1) – (V+1) / 2) / Φ

Flmax = (21/20 × 11 × (V+1)) /φ (s)

V: Reload value Φ: Bus clock

When margin (Φ) of the sampling timing is made two clocks, the allowed maximum transfer rate (FLmax) is as follows:

FLmax = (21/20 × 11 × (V+1) – 2) / Φ = (231V+191) / 20Φ (s)

V: Reload value Φ: Bus clock

Therefore, the allowed minimum baud rate (BGmin) at the destination would be calculated as follows.

BGmin = 11 / FLmax = 220Φ / (231V+191) (bps)

V: Reload value Φ: Bus clock

The allowed baud rate errors at UART and the destination can be obtained from above minimum/maximum baud rate calculation formulas, the result of which are as follows.

Table 32-5. Allowed Baud Rate Error

Reload value	Allowed maximum baud rate error	Allowed minimum baud rate error
3	0%	0%
10	2.98%	-3.24%
50	4.37%	-4.44%
100	4.56%	-4.60%
200	4.66%	-4.68%
32767	4.76%	-4.76%

Note:

The accuracy of reception depends on the number of bits in a frame, bus clock, and the reload value. The higher the bus clock and the division ratio are, the more accurate it will become.

Reload Values and Errors for Each Internal Clock (Peripheral Clock (PCLK)) and Baud Rate

Table 32-6. Reload Values and Errors for Each Internal Clock (Peripheral Clock (PCLK)) and Baud Rate

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	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	0.88	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: Setting value of the BGR register (decimal)
- ERR: Baud rate error (%)

External Clock

When the EXT bit of the baud rate generator register (BGR) is set to "1", the baud rate generator divides the external clock.

Note:

The external clock signals are synchronized with the internal clock by UART. If the external clock cannot be synchronized, therefore, the operation becomes unstable.

Reload Counter Functions

Reload counters, including transmission and reception reload counters, serve as the dedicated baud rate generators. It consists of a 15-bit register for reload values and generates a transmission/reception clock from the external or internal clock.

Count Start

When a reload value is written to the baud rate generator register (BGR), the reload counter starts counting.

Restart

The reload counter restarts under one of the following conditions:

- Common to the transmission and reception reload counters
  - Programmable reset (SCR:UPCL bit)
- Reception reload counter
  - Detection of a start bit falling edge in asynchronous mode

32.5.3 Setup Procedure and Program Flow

Setup procedure and program flow is shown.

32.5.3.1 Operation Mode 0 (One-to-One Connection)

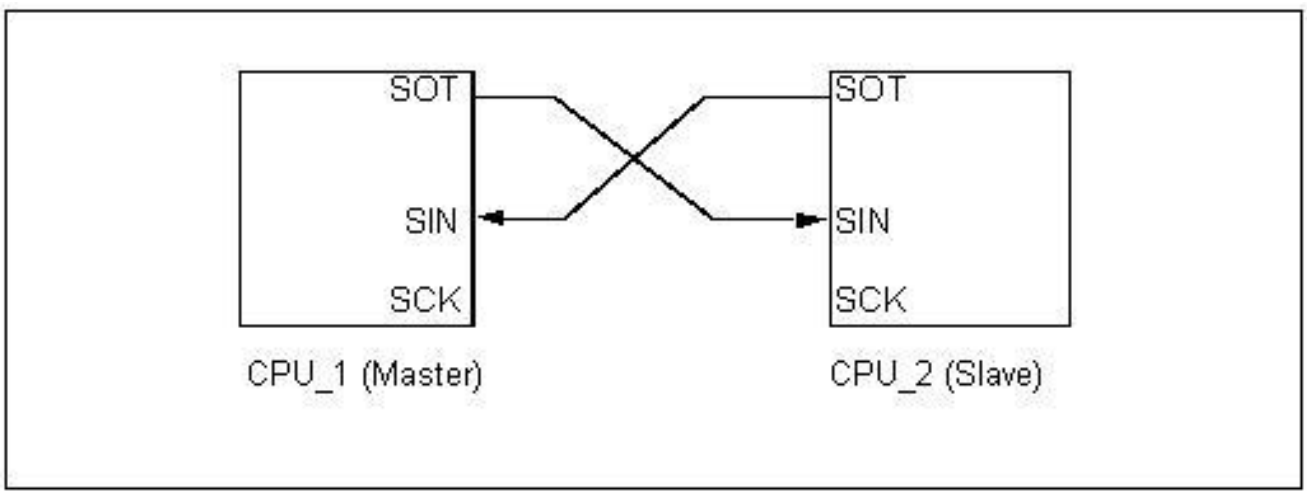
This section explains the operation mode 0 (one-to-one connection).

In operation mode 0, asynchronous serial bidirectional communications can be performed.

CPU Interconnection

In operation mode 0 (normal mode), select bidirectional communications. Two CPUs are inter-connected as shown in Figure 32-15:

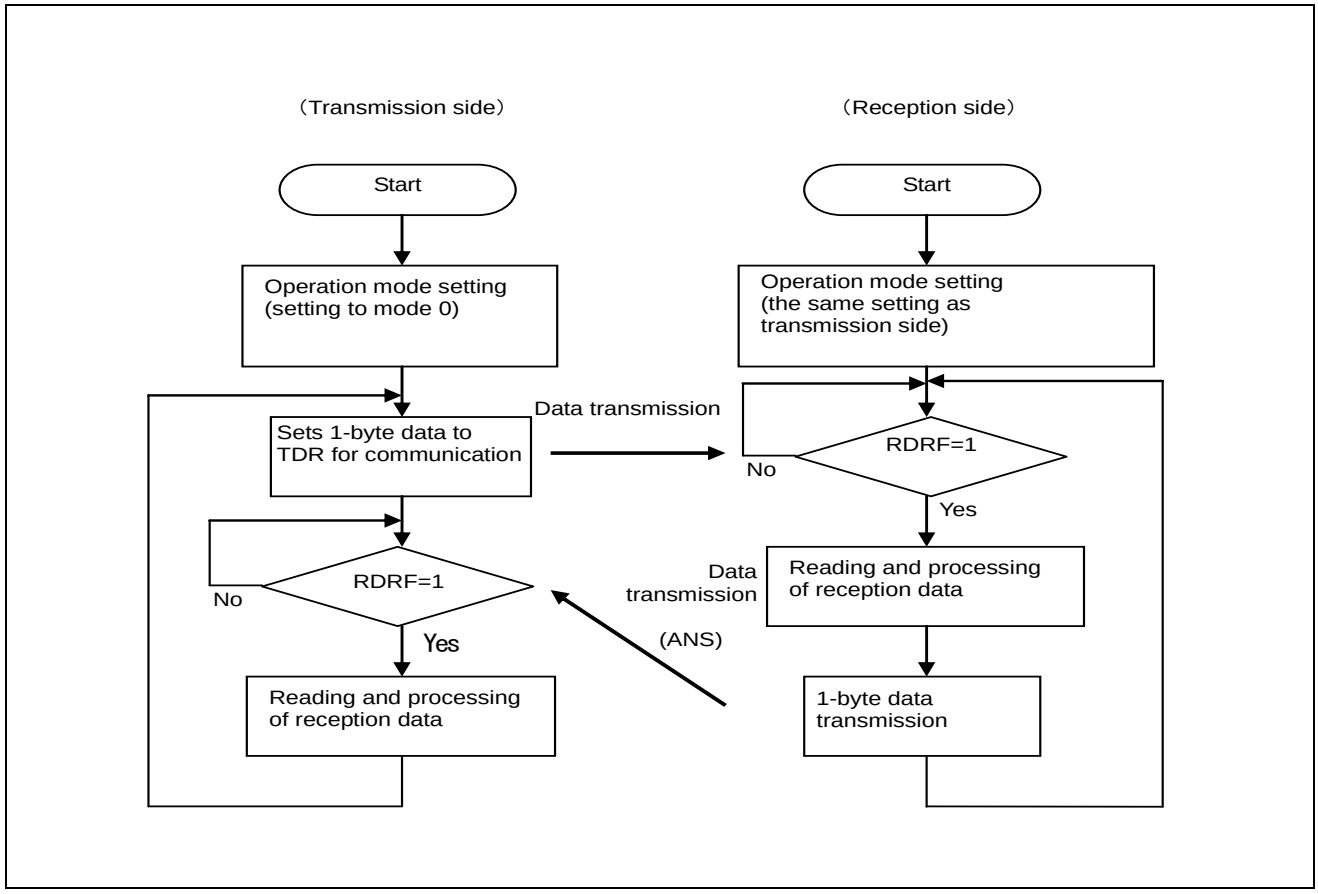
Figure 32-15. Example of Connection for Bidirectional Communications in UART Operation Mode 0



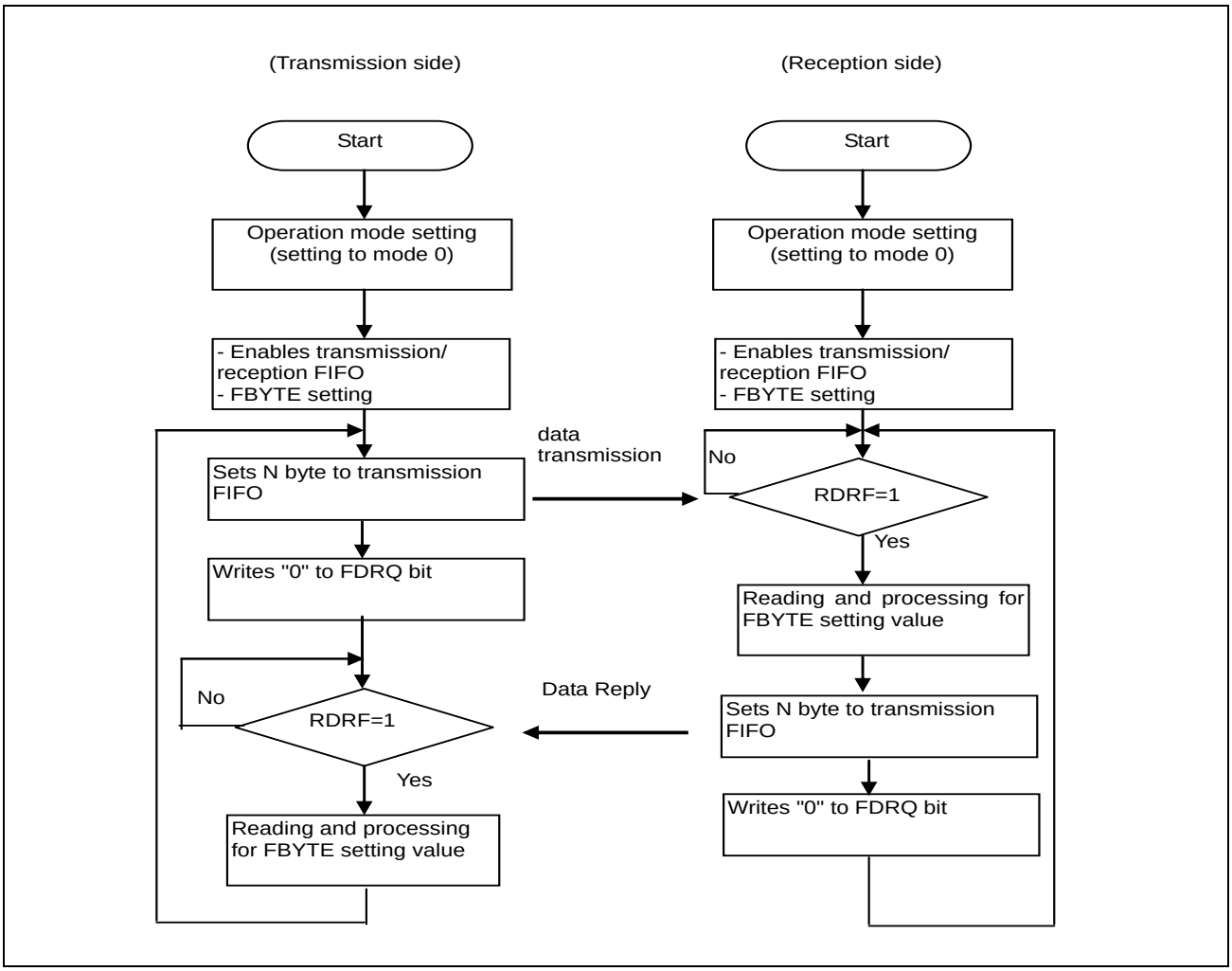
Flowchart

Figure 32-16. Example of Settings for Bidirectional Communications

FIFO Unsed



FIFO Used


(Transmission side)
(Reception side)
Start
Start
Operation mode setting
(setting to mode 0)
Operation mode setting
(setting to mode 0)
- Enables transmission/
reception FIFO
- FBYTE setting
- Enables transmission/
reception FIFO
- FBYTE setting
data
transmission
No
Sets N byte to transmission
FIFO
RDRF=1
Yes
Writes "0" to FDRQ bit
Reading and processing for
FBYTE setting value
No
Data Reply
RDRF=1
Sets N byte to transmission
FIFO
Yes
Writes "0" to FDRQ bit
Reading and processing
for FBYTE setting value

32.5.3.2 Operation Mode 1 (One-to-N Connection)

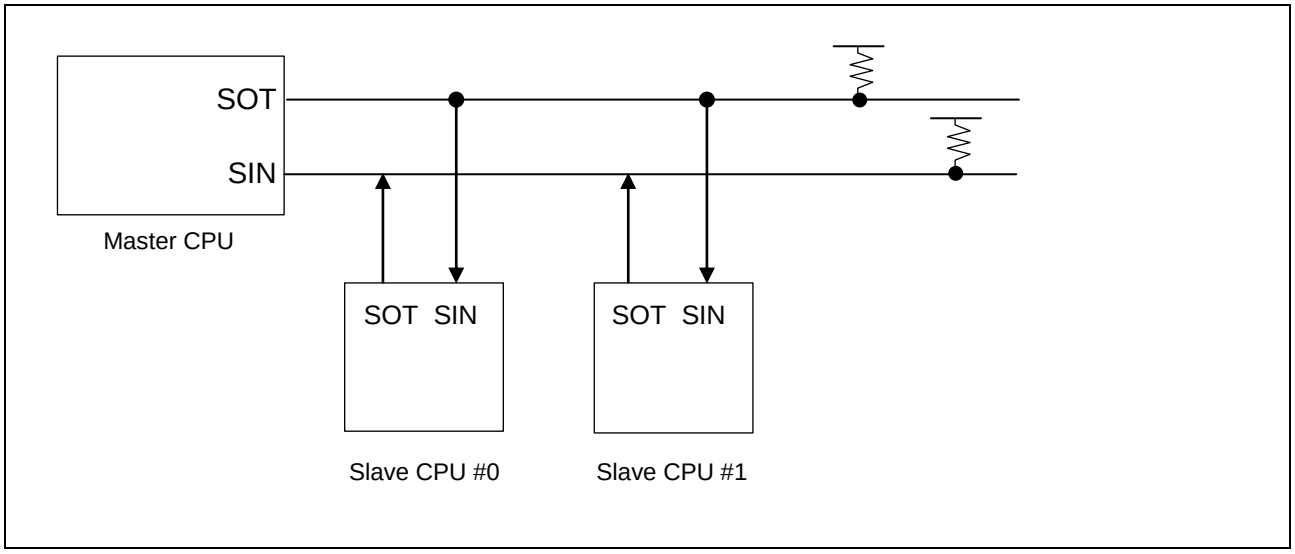
This section explains the operation mode 1 (one-to-n connection).

In operation mode 1 (multi-processor mode), communications can be performed via master-slave connection between multiple CPUs. UART can be used either as a master or slave.

CPU Interconnection

For master-slave communications, a communication system can be configured as one master CPU and multiple slave CPUs connected to two common communication lines as shown in the Figure below. UART can be used either as a master or slave.

Figure 32-17. Example of Connection for Master-Slave Communications of UART



Function Selection

For master-slave communications, select an operation mode and a data transfer method as shown in Table 32-7. Selection of Master-Slave Communication Function:

Table 32-7. Selection of Master-Slave Communication Function

	Operation mode		Data	Parity	Stop bit	Bit direction
	Master CPU	Slave CPU				
Address transmission and reception	Mode 1 (AD bit transmission)	Mode 1 (AD bit reception)	AD = "1" + 7 or 8-bit Address	None	1 bit or 2 bits	LSB or, MSB First
Data transmission and reception			AD = "0" + 7 or 8-bit Data			

Note:

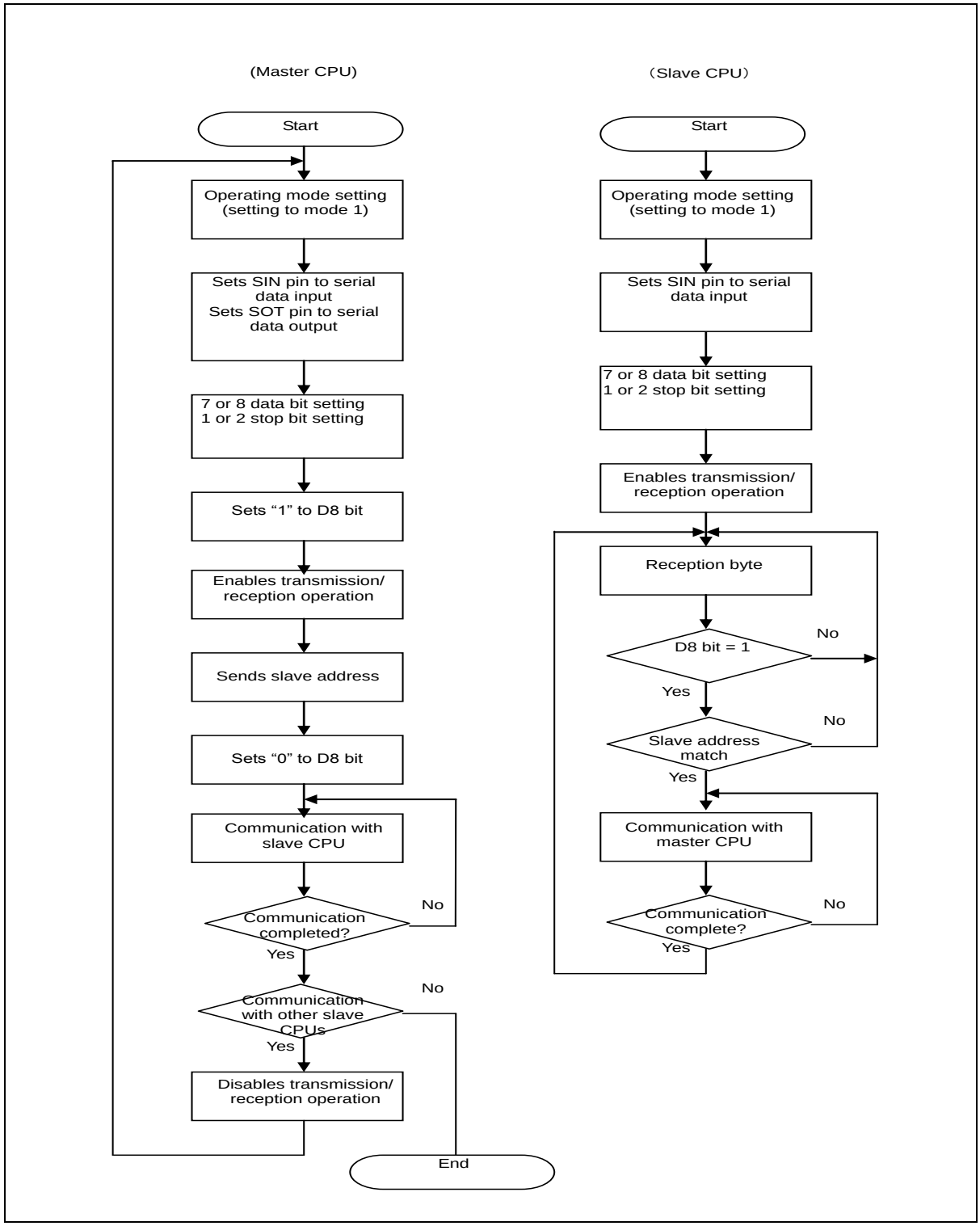
Have word access to transmitted and received data (TDR/RDR) in operation mode 1.

Communication Procedure

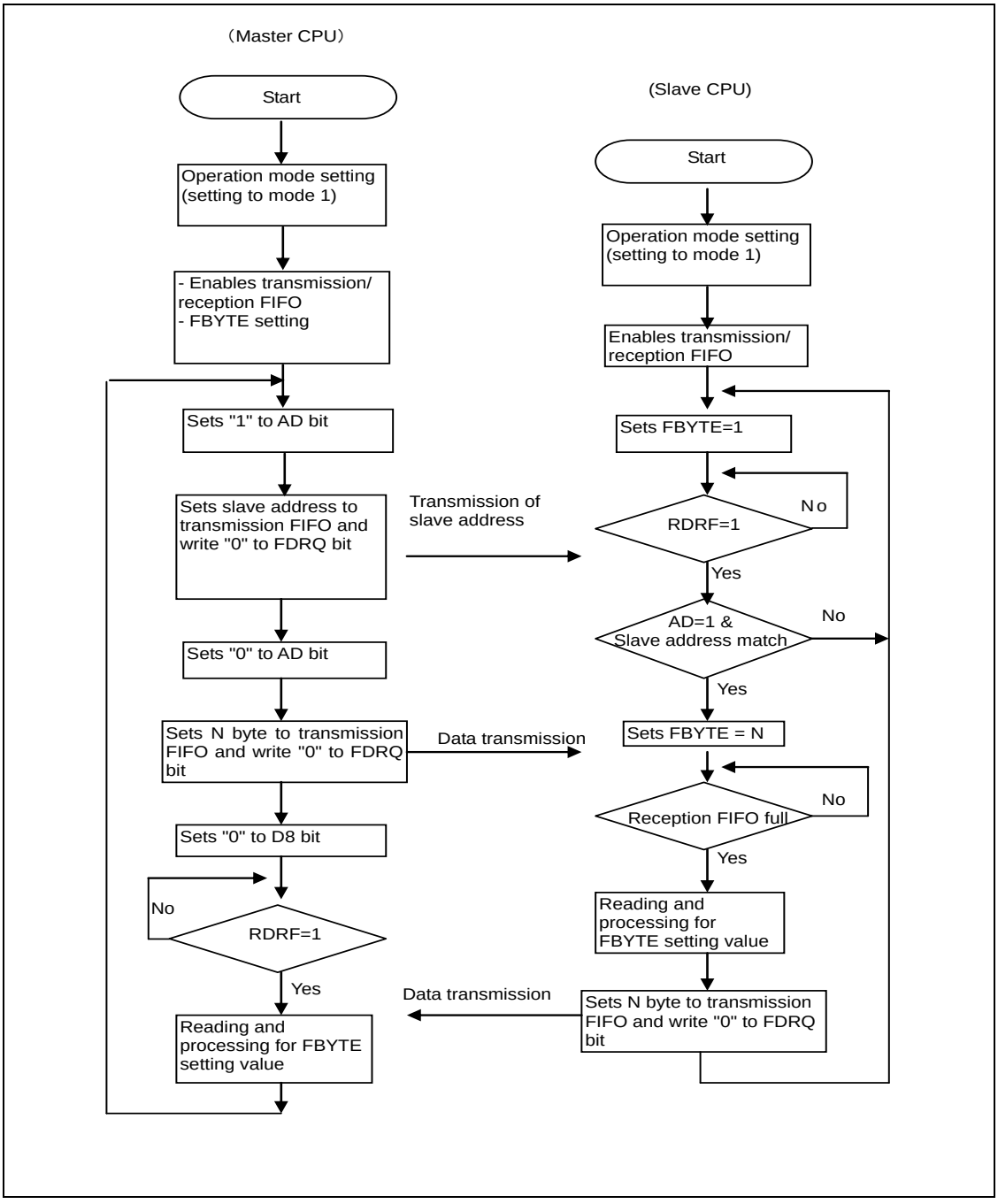
Communications start when the master CPU transmits address data. Address data refers to data with the D8 bit set to "1" and is used to select a slave CPU as the communication destination. Slave CPUs interpret address data via a program and the one with a matching address performs communications (normal data) with the master CPU. Figure 32-18 shows a flowchart of master-slave communications (multi-processor mode).

Figure 32-18 Example of Flowchart of Master-Slave Communications

FIFO Unused



FIFO Used


(Master CPU)
Start
Operation mode setting (setting to mode 1)
- Enables transmission/ reception FIFO
- FBYTE setting
Sets "1" to AD bit
Sets slave address to transmission FIFO and write "0" to FDRQ bit
Transmission of slave address
Sets "0" to AD bit
Sets N byte to transmission FIFO and write "0" to FDRQ bit
Data transmission
Sets "0" to D8 bit
No
RDRF=1
Yes
Reading and processing for FBYTE setting value
(Slave CPU)
Start
Operation mode setting (setting to mode 1)
Enables transmission/ reception FIFO
Sets FBYTE=1
No
RDRF=1
Yes
AD=1 & Slave address match
No
Yes
Sets FBYTE = N
No
Reception FIFO full
Yes
Reading and processing for FBYTE setting value
Data transmission
Sets N byte to transmission FIFO and write "0" to FDRQ bit

32.6 Operation of CSIO

This section explains operation of CSIO.

32.6.1 Interrupts of CSIO

Interrupts of CSIO are shown.

The interrupts for the CSIO (clock synchronous serial interface) include reception and transmission interrupts. An interrupt request can be generated using the following factors:

- Setting of reception data in the receive data register (RDR) or occurrence of a reception error
- Start of transmission after transfer of transmission data from the transmit data register (TDR) to the transmit shift register
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request
- Comparison value (STMCR) of the serial timer and serial timer value (STMR) match.
- Chip selection error generation

32.6.1.1 List of Interrupts of CSIO

This section explains the list of interrupts of CSIO.

Table 32-8. Interrupt Control Bits and Interrupt Factors of CSIO

Interrupt Type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request flag
Reception	RDRF	SSR	1-byte reception	SCR: RIE	Reading of receive data (RDR)
			Reception of as much data as specified by FBYTE		Reading of receive data (RDR) until the reception FIFO is emptied
			Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	Overrun error		Writing of "1" to the reception error flag clear bit (SSR:REC)
Transmission	TDRE	SSR	Transmission register is empty	SCR: TIE	Writing of transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	TBI	SSR	No transmission operation	SCR: TBIE	Writing of transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	FDRQ	FCR1	The storage data value of the transmission FIFO is FTICR setting value or less, or empty	FCR1: FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or transmission FIFO is full
Transmission	CSE	SACSR	In slave mode (SCR:MS="1"), when serial chip select pin is in inactive master mode (SCR:MS=0) during the transmission operation, the transmission count is equal to or less than the count set at TBYTE and the next transmission data is not written to TDR (SSR:TDRE=1). In master mode (SCR:MS=0), because the number of the transmission times is below the set value of TBYTE, and the next transmission data is not written to TDR (SSR:TDRE=1)	SACSR: SCEIE	Writing "0" to the serial chip select flag bit (SACSR:CSE)
Status	TINT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to the timer interrupt flag bit (SACSR:TINT)

* Set the TIE bit to "1" after the TDRE bit is cleared to "0".

32.6.1.2 Reception Interrupts and Flag Setting Timing

This section explains the generation of reception interrupts and flag setting timing.

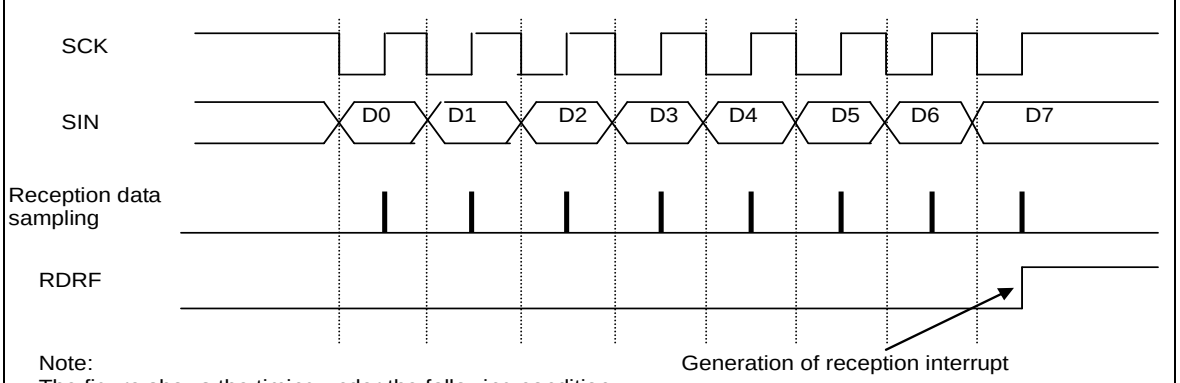
Reception interrupts occur either when the reception is completed (SSR:RDRF) or when a reception error occurs (SSR:ORE). When the last data bit is detected, reception data is stored in the receive data register (RDR). When reception is completed (SSR:RDRF=1) or a reception error occurs (SSR:ORE=1), a corresponding flag is set. If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt occurs.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

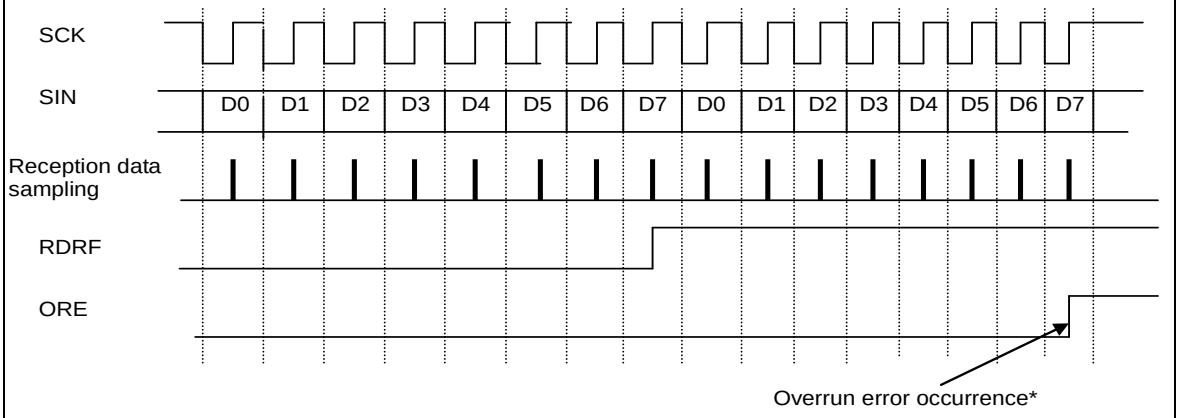
Figure 32-19. Timing of Flag Setting

Reception operation and flag setting timing



Note:
The figure shows the timing under the following condition:
SCR : MS=1, SPI=0
ESCR : L3 to L0=0000b
SMR:SCINV=0, BDS=0, SCKE=0, SOE=0

Timing to set ORE(overrun error) flag bit



* An overrun error occurs when the next data is transferred before the reception data is read(RDRF=1)

Note:
The figure shows the timing under the following condition:
SCR : MS=1, SPI=0
ESCR : L3 to L0=0000b
SMR:SCINV=0, BDS=0, SCKE=0, SOE=0

32.6.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing

This section explains the generation of interrupts when using the reception FIFO and flag setting timing.

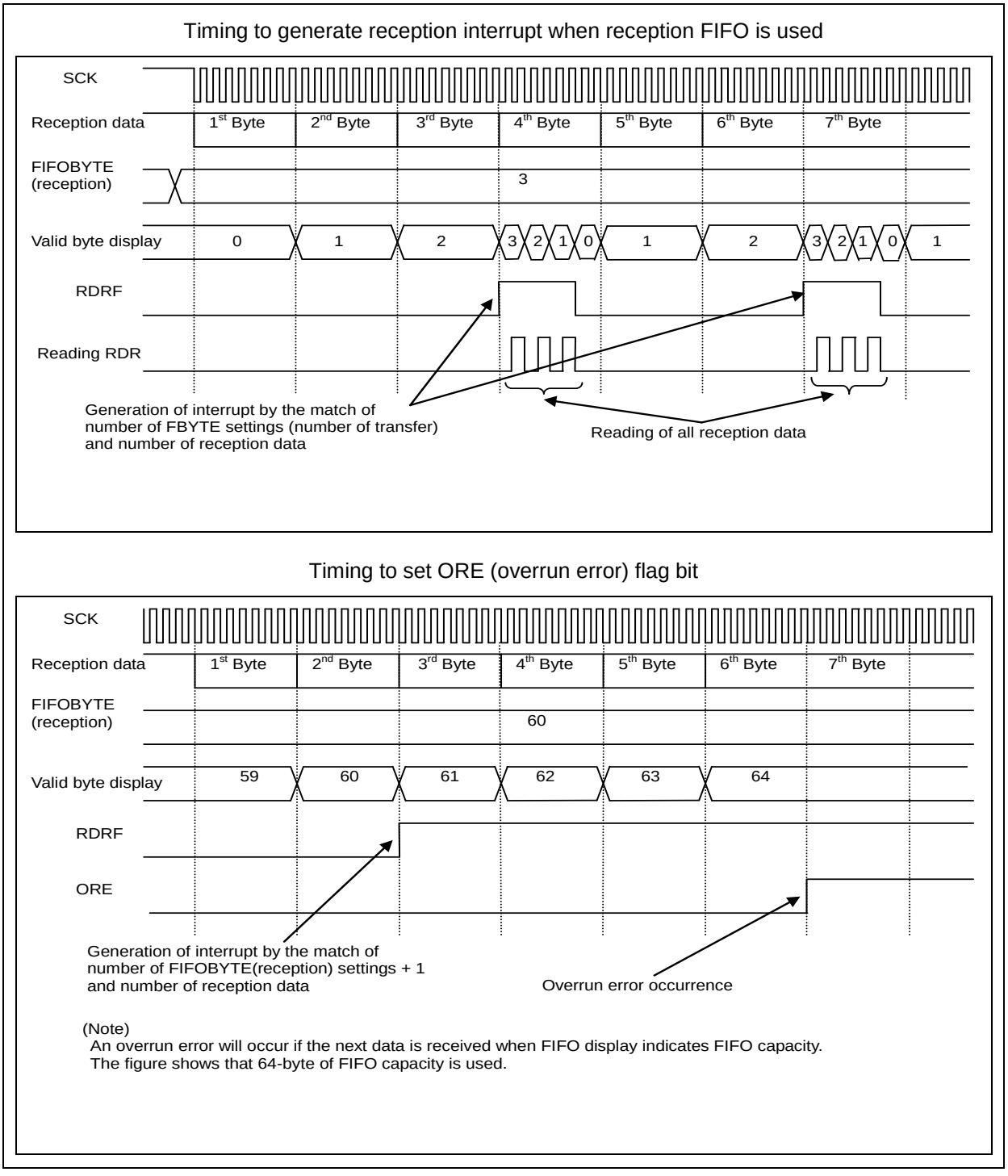
When the reception FIFO is used, an interrupt occurs after as much data as the FBYTE register (FBYTE) setting is received. The setting value of the FBYTE register determines the occurrence of an interrupt when the reception FIFO is used.

- After as much data as the transfer count setting of the FBYTE register is received, the reception data full flag of the serial status register (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SCR:RIE) at this time, a reception interrupt will be generated.
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (RDRF) will be set to "1".
  - ☐ Reception FIFO idle detection enable bit (FRIIE) is "1"
  - ☐ Data count contained in the reception FIFO does not reach the transfer count

If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

- If the receive data (RDR) is read until the reception FIFO is empty, the reception data full flag (SSR:RDRF) will be cleared.
- When the reception-enabled data count indication has shown the FIFO capacity, receiving the next data will generate an overrun error (SSR:ORE=1).

Figure 32-20. Timing of Interrupts and Flag Setting



32.6.1.4 Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and flag setting timing.

Transmission interrupts occur either when transmission is started after transfer of transmission data from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) or when the transmission operation is idle (SSR:TBI=1).

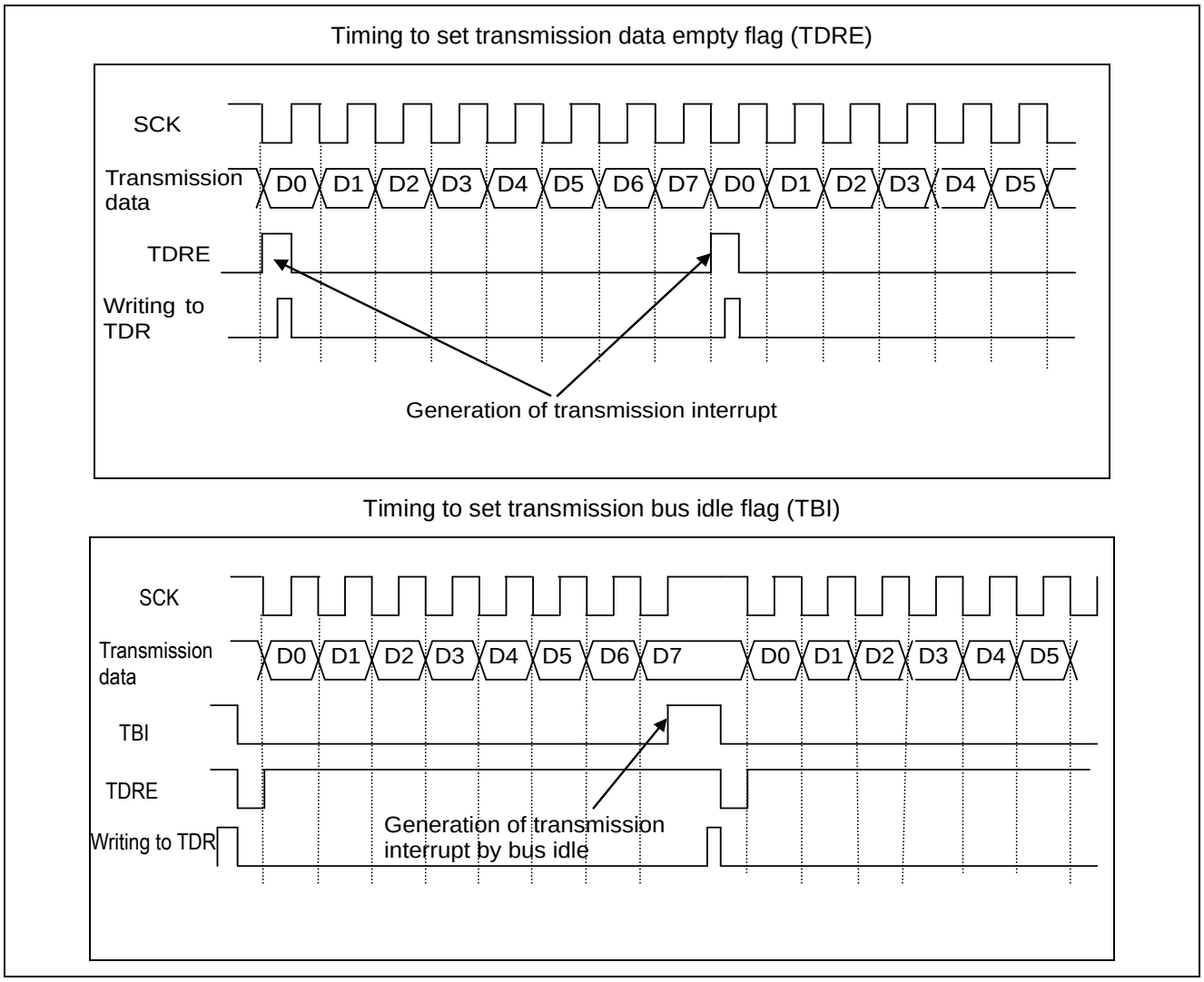
Timing of transmission data empty flag (TDRE) setting

When data written to the transmit data register (TDR) is transferred to the transmit shift register, writing of next data is enabled (SSR:TDRE=1). If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs. The SSR:TDRE bit, being a read-only bit, is cleared to "0" by writing of data to the transmit data register (TDR).

Timing of transmission bus idle flag (TBI) setting

When the transmit data register is empty (TDRE=1) and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE=1) at this time, a transmission interrupt occurs. When transmission data is written to the transmit data register (TDR), the SSR:TBI bit and the transmission interrupt request are cleared.

Figure 32-21. Timing of Flag Setting



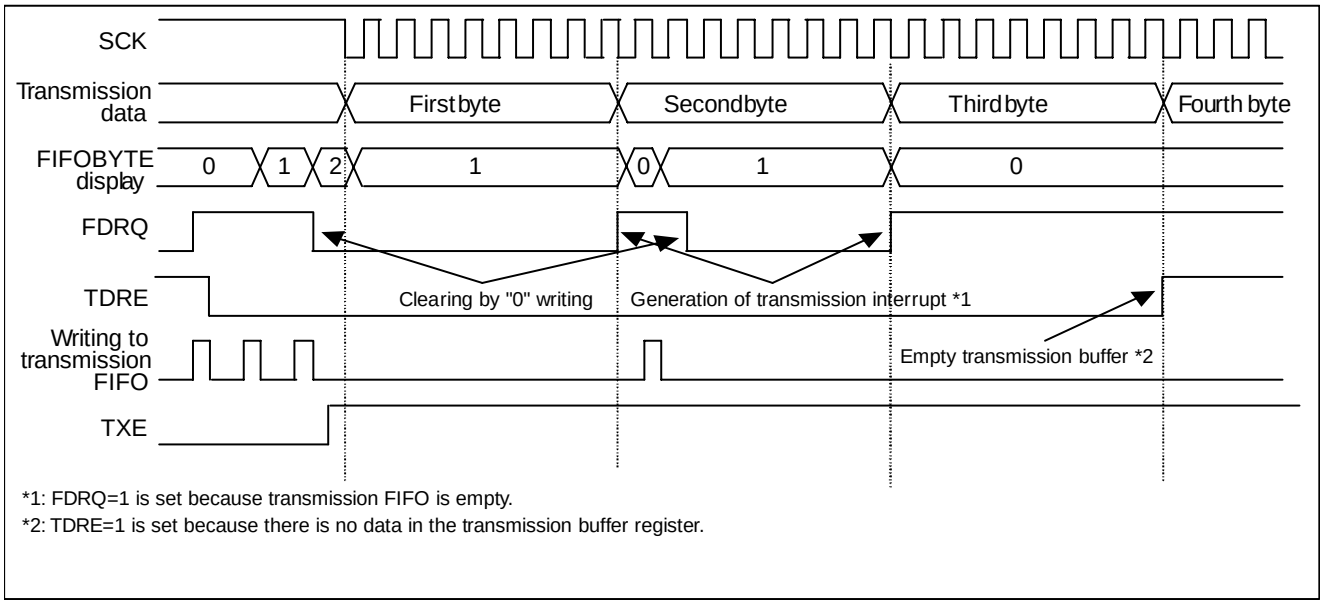
32.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

This section explains the generation of interrupts when using transmission FIFO and flag setting timing.

When the transmission FIFO is used, an interrupt generation when the storage data value of the transmission FIFO is FTICR register (FTICR) setting value or less.

- When the storage data value of the transmission FIFO is FTICR register (FTICR) setting value or less, the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1". If FIFO transmission interrupt is enabled (FCR1:FTIE="1") at this time, a transmission interrupt will occur.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
  When FBYTE=0x00 and FTICR=0x00, there is no data in the transmission FIFO.

Figure 32-22. Timing of Interrupt Generation



*1: FDRQ=1 is set because transmission FIFO is empty.
*2: TDRE=1 is set because there is no data in the transmission buffer register.

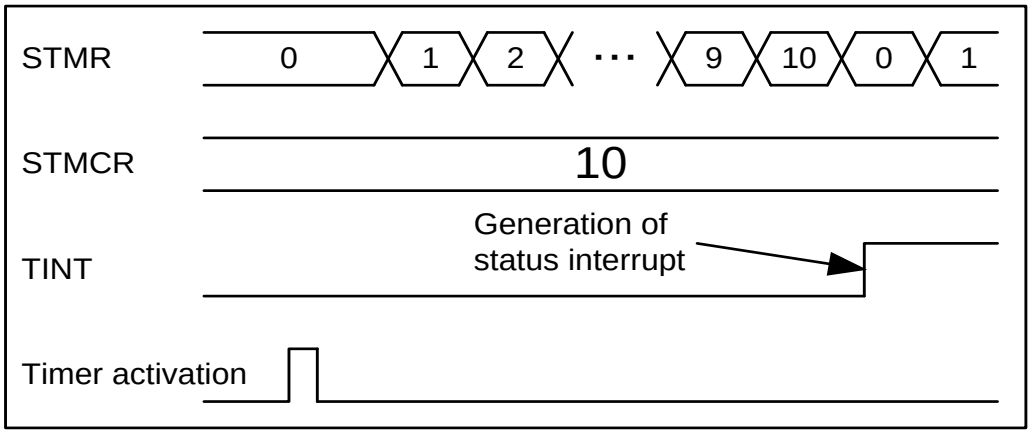
32.6.1.6 Timing of Timer Interrupt and Flag Setting

This section explains the timing of timer interrupt and flag setting.

Timer interrupt is generated when Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR).

- When Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR), "1" will be set to timer interrupt flag (SACSR:TINT).
  At this time when the timer interrupt is enabled (SACSR:TINTE="1"), a status interrupt will be generated.

Figure 32-23. Timer Interrupt Generation Timing



32.6.1.7 Timing of Chip Select Error Generation and Flag Setting

This section explains the timing of chip select error generation and flag setting.

Chip select error will be generated when the number of frames which have been transmitted is less than the setup value specified by the TBYTE and no valid data is present in the transmit data register (TDR) (SSR:TDRE="1") after one frame is transmitted while in master mode (SCR:MS="0"). This error will also be generated when chip select pin becomes inactive while transmitting in slave mode (SCR:MS=1).

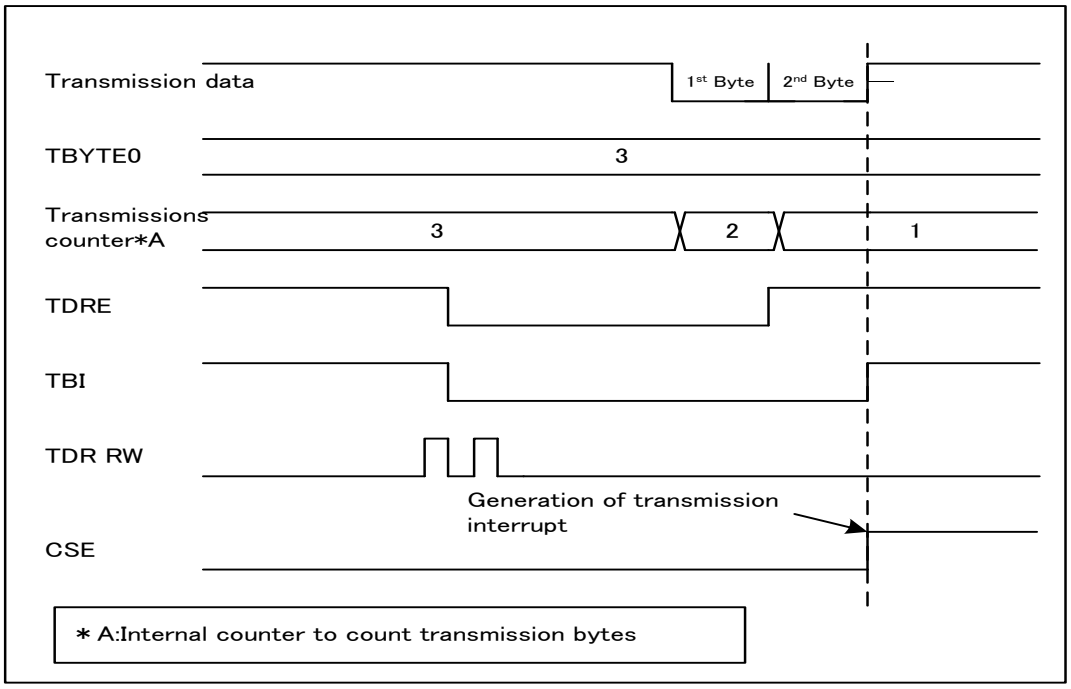
Master Mode (SCR:MS="0")

Chip select error will be generated with transfer byte error enabled (TBEEN="1") and any of following events when no valid data is present in the transmission data register (TDR) (SSR:TDRE="1") before transmitting data frame specified by the TBYTE.

- Chip select is used
- Synchronous transmission with the serial timer is used

In this case, when chip select error interrupt is enabled (SACSR:CSEIE="1"), a transmission interrupt will be generated.

Figure 32-24. Chip Select Error Generation Timing



Notes:

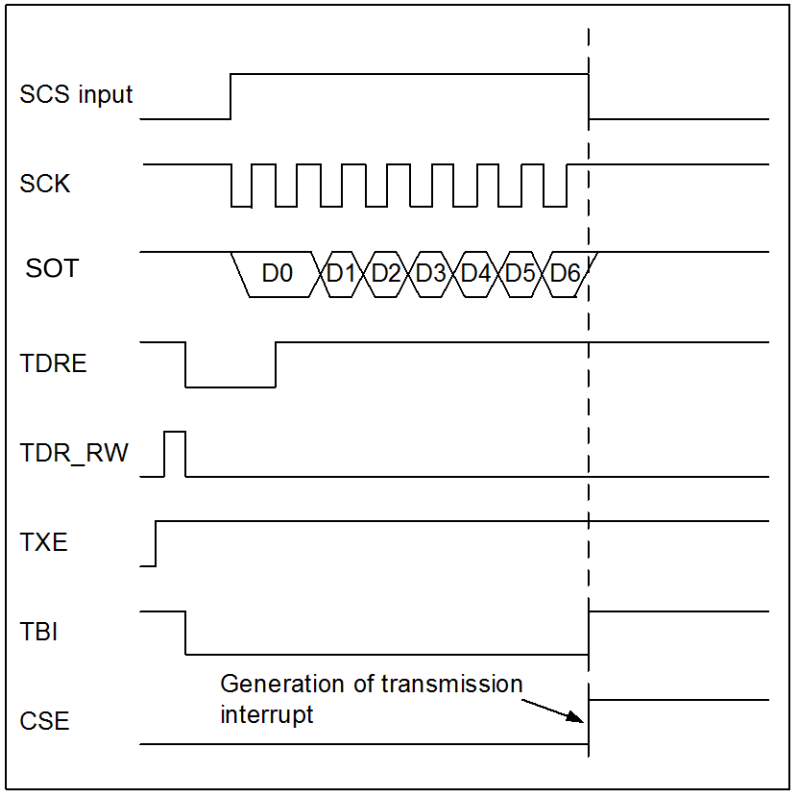
- When chip select is used, chip select error flag (SACSR:CSE) will be set to "1" after hold delay time is passed after a chip select error is generated and at the same time, the serial chip select pin will become inactive. No transmission will be started even if transmit data is written to the transmit data register (TDR) during hold delay time, and the chip select error flag (SACSR:CSE) will be set to "1" after hold delay time is passed.
- While the chip select error flag (SACSR:CSE) is set to "1", no transmission will be started even if transmission data is written to the transmit data register (TDR).
- When the chip select error flag (SACSR:CSE) is set to "1" while synchronous transmission is used with the serial timer, no transmission will be started even if serial timer register (STMR) matched the serial timer comparison register.

Slave Mode (SCR:MS="1")

Chip select error will be generated when chip select pin becomes inactive while transmitting (SSR:TBI="0").

In this case, when chip select error interrupt is enabled (SACSR:CSEIE="1"), a transmission interrupt will be generated.

Figure 32-25. Chip Select Error Generation Timing



32.6.2 Operation of CSIO

Operation of CSIO is shown.

32.6.2.1 Normal Transfer (I)

This section explains the normal transfer (I).

Features

Table 32-9. Features of Normal Transfer (I)

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Transmission data output timing	SCK falling edge
3	Reception data sampling	SCK rising edge
4	Data length	5 to 16, 20, 24, 32 bits

Register Settings

The following table lists the register settings required for normal transfer (I).

SCR:SPI=0*, SMR:MD2=0, MD1=1, MD0=0, SCINV=0*
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0

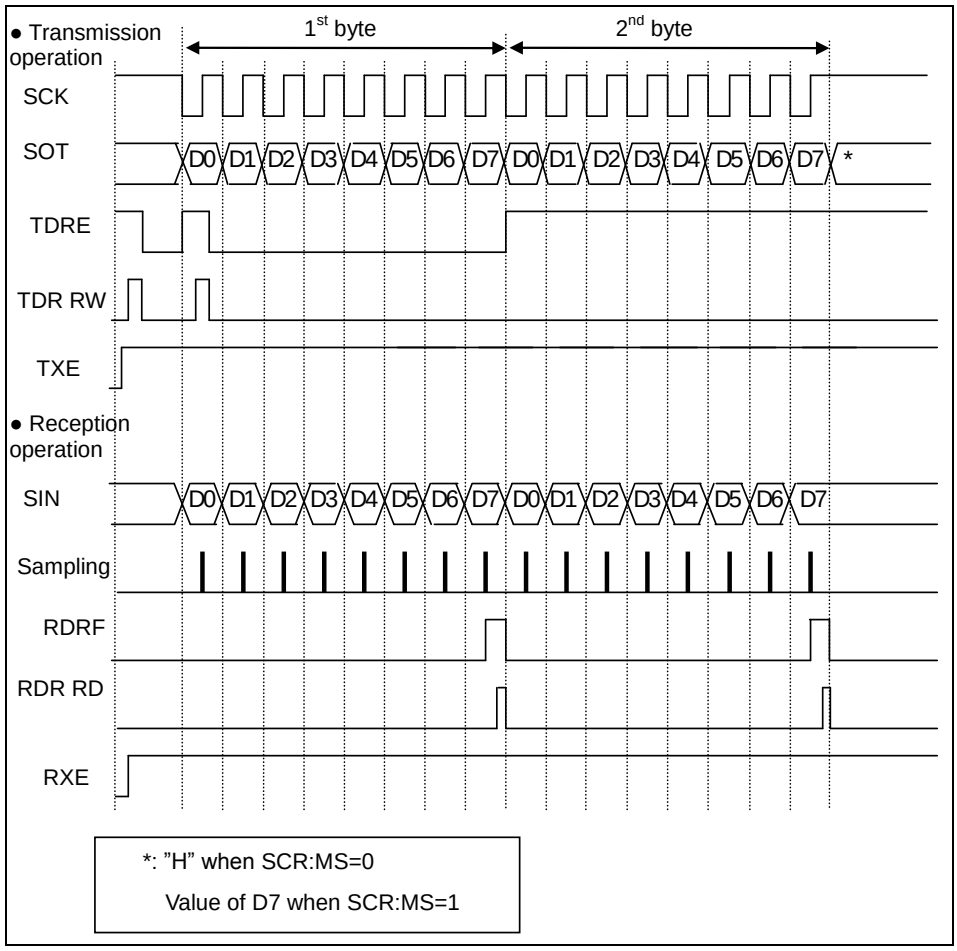
* Bit settings depend on the condition. See Table 32-9 for details.

Note:

Use proper usage for setting the registers other than the bits above.

Normal Transfer (I) Timing Chart (Serial Chip Select Pin Unused)

Figure 32-26. Normal Transfer (I) Timing Chart (Serial Chip Select Pin Unused)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSEN1-0="00"b)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting the transmission data in synchronization with a falling edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1. When the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the reception data at a rising edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1. When the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

Transmission/Reception operation

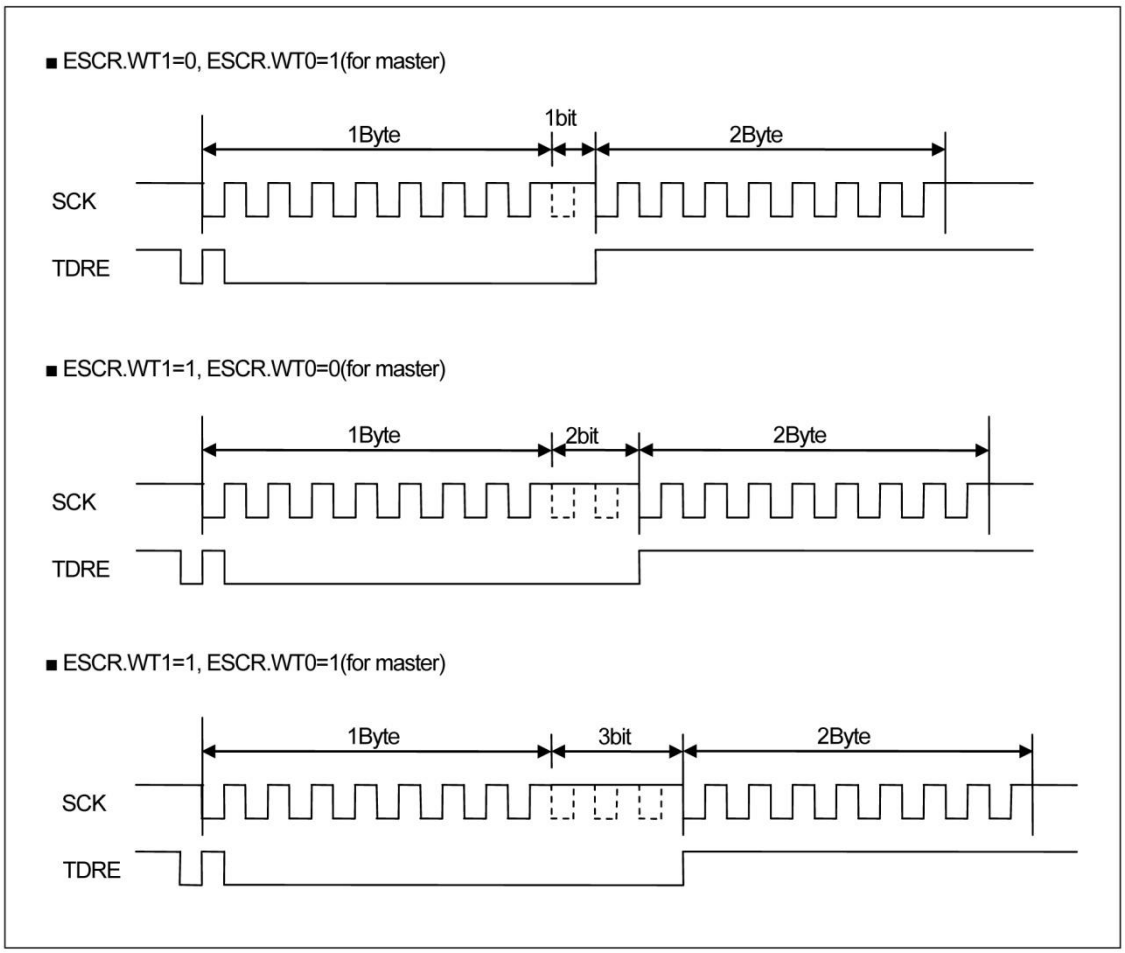
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the falling edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set. When reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=0.)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. This results in output the transmission data in synchronization with a falling edge of the serial clock (SCK) input.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1 and, if the transmission interrupt is enabled (SCR:TIE=1), outputs a transmission interrupt request. At this time, the transmission data in the second byte can be written.

Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1. When the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Transmission/Reception operation

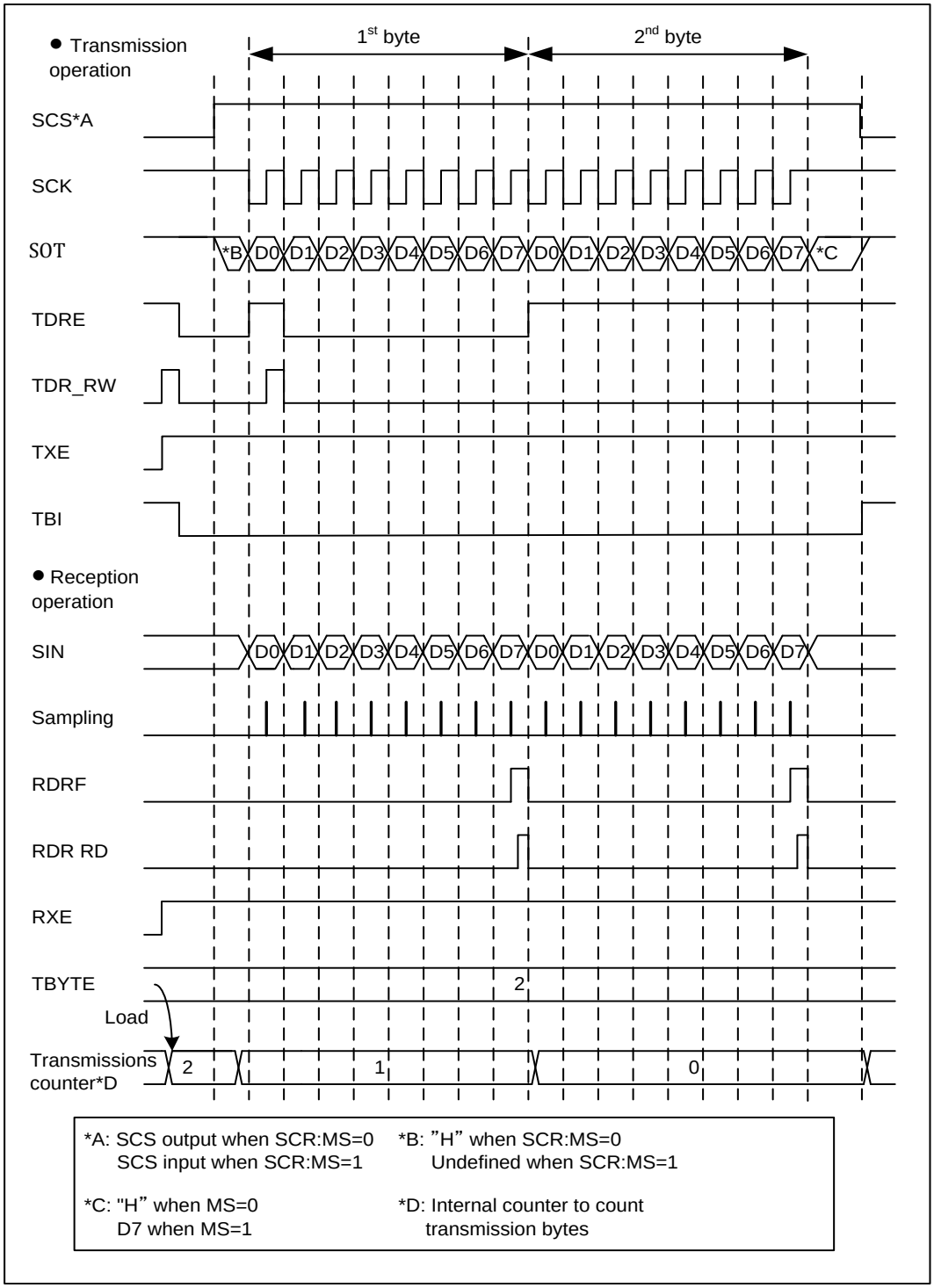
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the falling edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

Normal Transfer (I) Timing Chart (Serial Chip Select Pin Used)

Figure 32-27. Normal Transfer (I) Timing Chart (Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

* "n" shows the number of serial chip select pin used.

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. Then serial chip select pin (SCS) will become active, transmission will be started after the setup time is passed for the serial chip select pin. The start of the transmission results in outputting the transmission data in synchronization with a falling edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1, and the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(3) After the number of data set in the TBYTE is completed for transmission, transmission operation will be terminated.

(4) After hold time for the serial chip select pin is passed after the transmission operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and reception operation will be started after the setup time is passed for the serial chip select pin. The start of the reception results in sampling the reception data at a rising edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for reception, reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

Notes:

- When you make reception operation only, make sure to write a dummy data to the TDR in order to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting desired number of frames to be transferred to the FBYTE register will make serial clock (SCK) output for the setup number of frames.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and then serial chip select pin (SCS) will become active and transmission/reception will be started after the setup time has passed for the serial chip select pin. When transmission/reception is started, the transmission data is output in synchronization with the falling edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

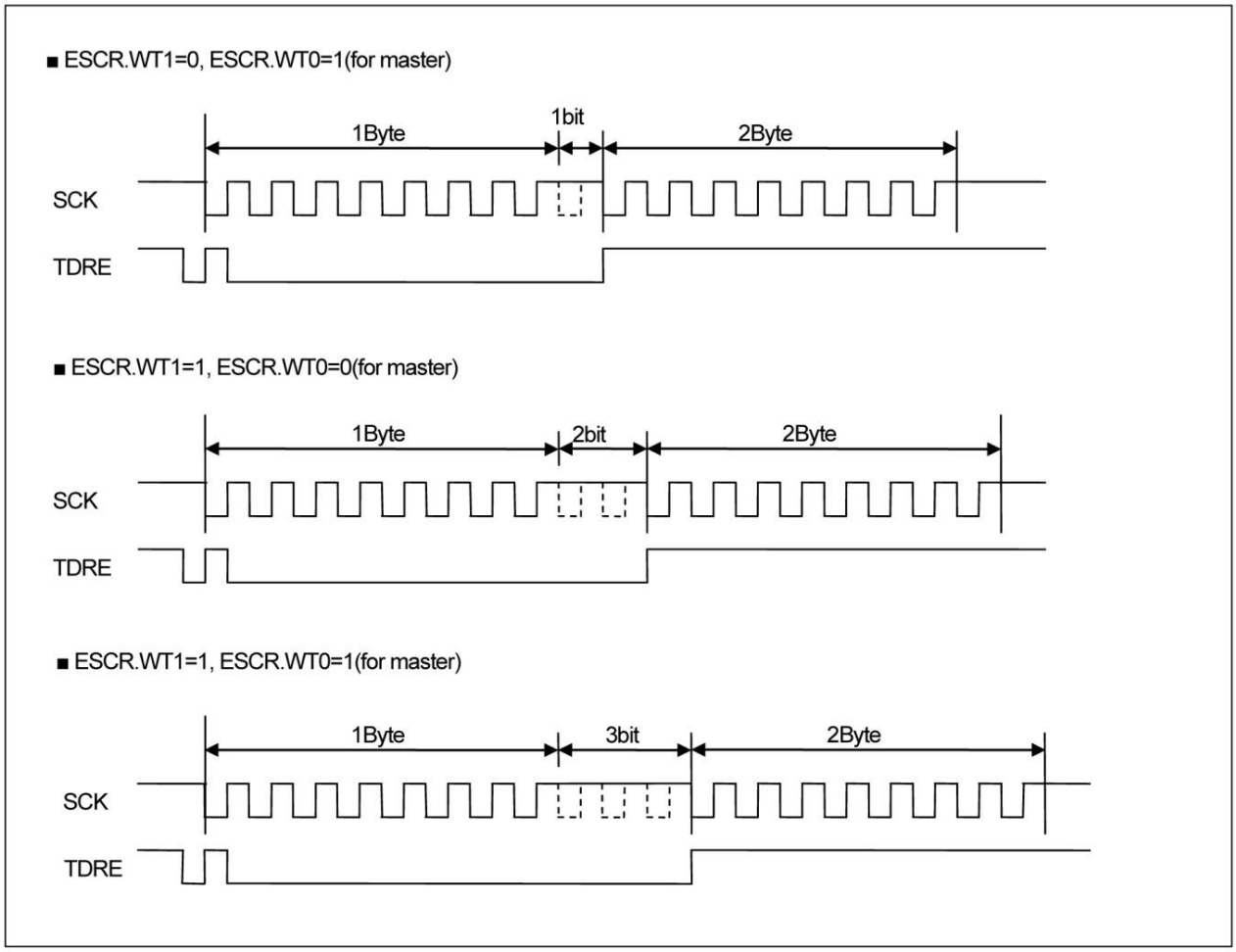
(3) While operating transmission/reception, the reception data will be sampled at a rising edge of the serial clock output (SCK). Receiving the last bit of receiving data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for transmission/reception, transmission/reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the transmission/reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

Continuous Data **Transmission or Reception Wait Operation**

(1) When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous transmission or reception, a wait will be inserted between frames.



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=1, SCSCR:CSOE=0, SCSCR:SCAM=0)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) Transmission operation will be started when serial chip select pin (SCS) becomes active, the transmission data will be output in synchronization with a falling edge of the serial clock (SCK) input.

(3) Outputting the transmission data in the first bit sets SSR:TDRE=1, and the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(4) Transmission operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), reception operation will be started when the serial chip select pin (SCS) becomes active, and reception data will be sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Reception operation will be terminated when serial chip select pin (SCS) becomes inactive.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When serial chip select pin (SCS) will become active, transmission/reception operation will be started and the transmission data is output in synchronization with the falling edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) While operating transmission/reception, the reception data will be sampled at a rising edge of the serial clock input (SCK). Receiving the last bit of receiving data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Transmission/reception operation will be terminated when the serial chip select pin (SCS) becomes inactive, and the serial output pin (SOT) becomes "H".

32.6.2.2 Normal Transfer (II)

This section explains the normal transfer (II).

Features

Table 32-10 Features of Normal Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Transmission data output timing	SCK rising edge
3	Reception data sampling	SCK falling edge
4	Data length	5 to 16, 20, 24, 32 bits

Register Settings

The following table lists the register settings required for normal transfer (II).

SCR:SPI*=0, SMR:MD2=0, MD1=1, MD0=0, SCINV*=1
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0

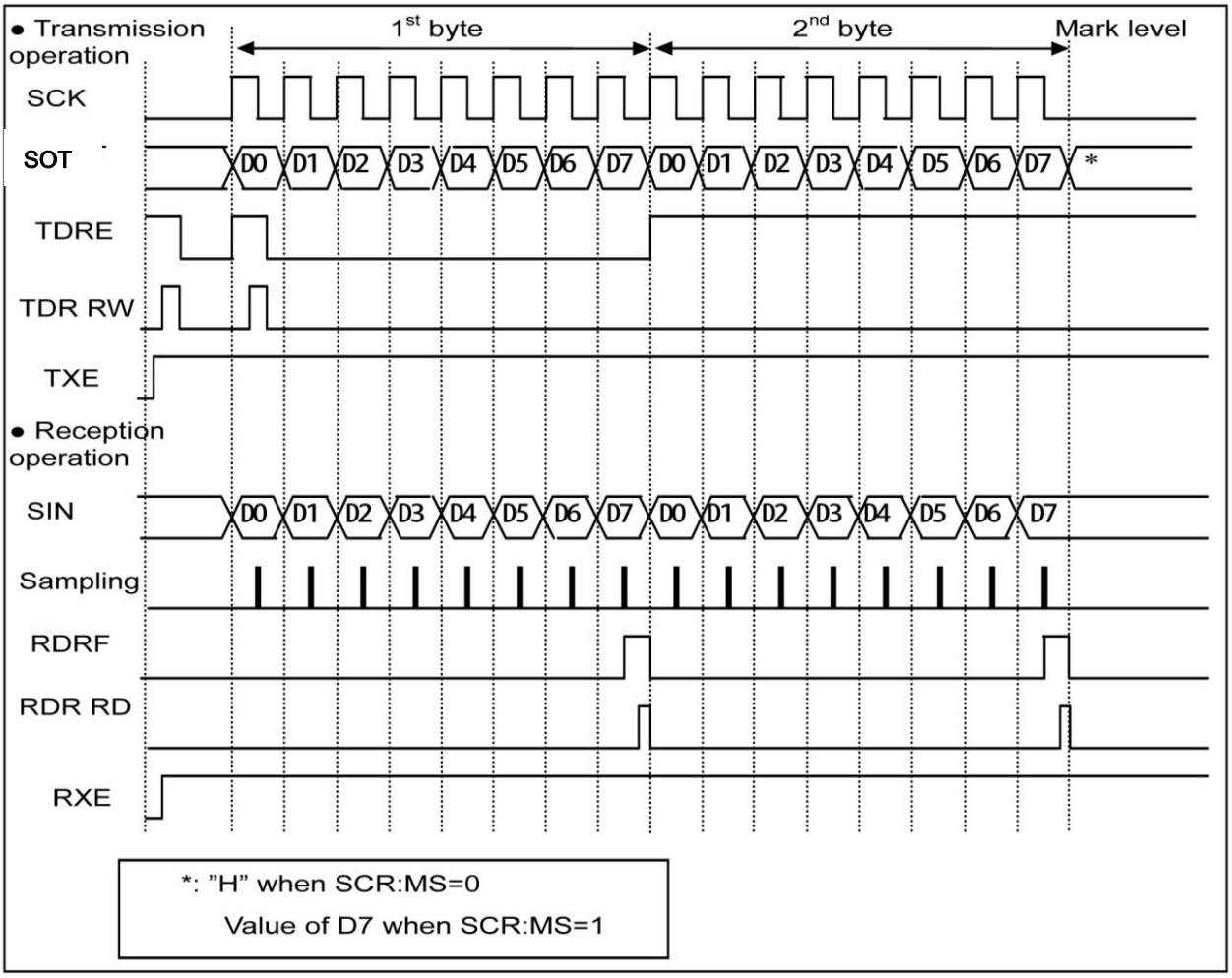
* Bit settings depend on the condition. See Table 32-13 for details.

Note:

Use proper usage for setting the registers other than the above.

Normal Transfer (II) Timing Chart (Serial Chip Select Pin Unused)

Figure 32-28. Normal Transfer (II) Timing Chart (Serial Chip Select Pin Unused)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1.)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1. When the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be output. At this time, the transmission data in the second byte can be written.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

Transmission/Reception operation

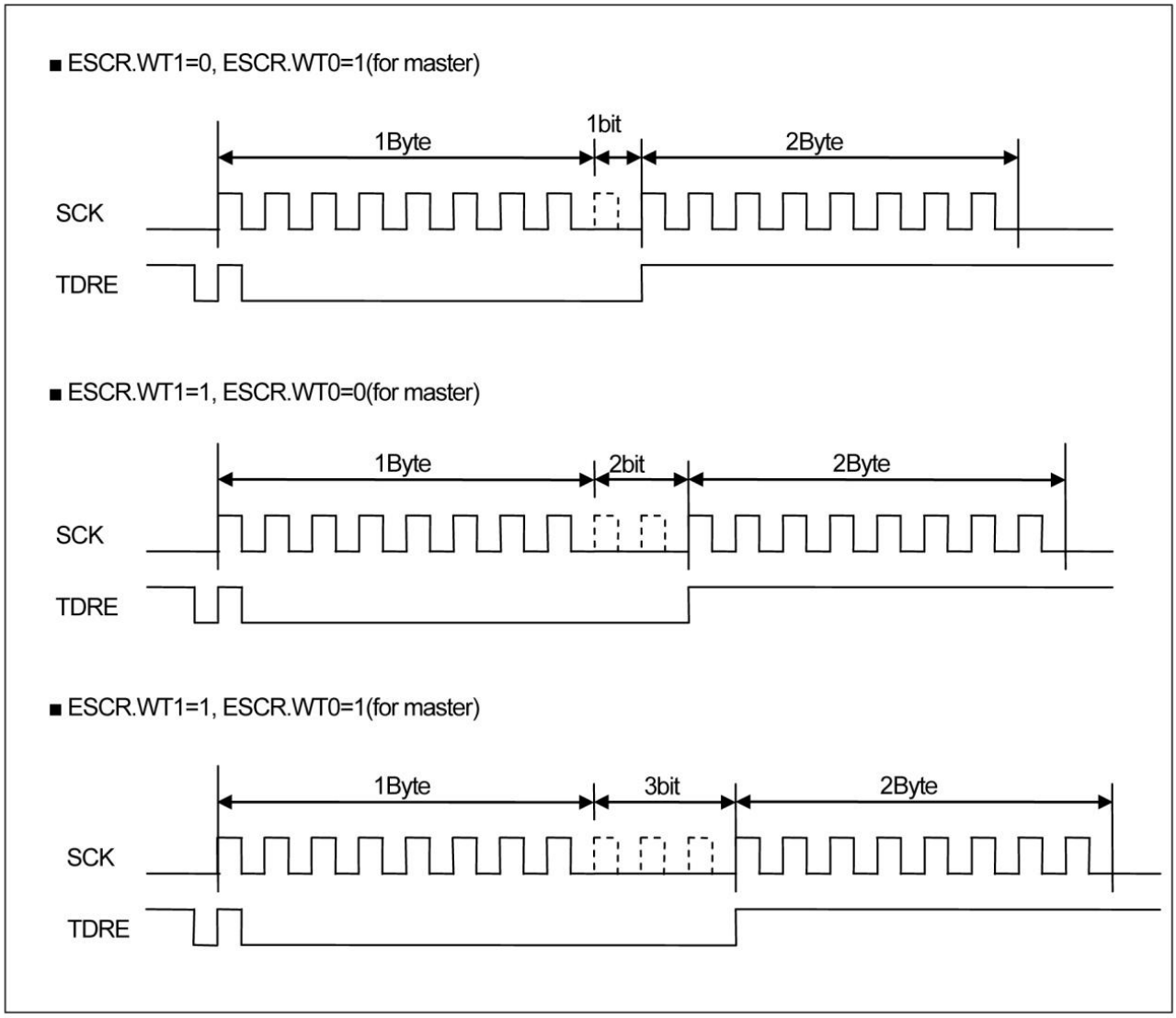
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the falling edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=0.)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) input.

(2)Outputting the transmission data in the first bit sets SSR:TDRE=1 and, if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be output. At this time, the transmission data in the second byte can be written.

Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1. When the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Transmission/Reception operation

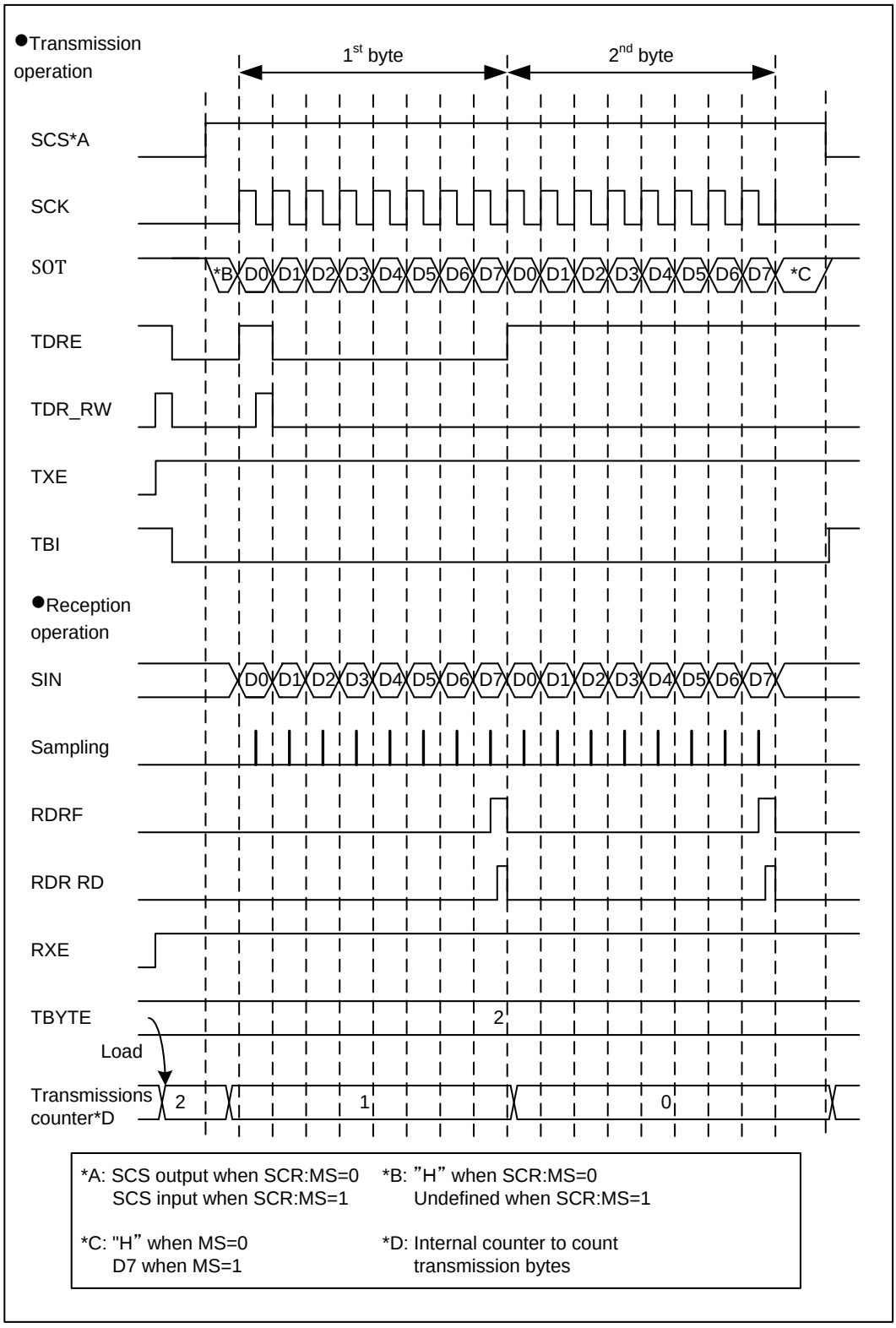
(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and the transmission data is output in synchronization with the rising edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the falling edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

Normal Transfer (II) Timing Chart (Serial Chip Select Pin Used)

Figure 32-29. Normal Transfer (II) Timing Chart (Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

* "n" shows the number of serial chip select pin used.

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. Then serial chip select pin (SCS) will become active, transmission will be started after the setup time is passed for the serial chip select pin. The start of the transmission results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) Outputting the transmission data in the first bit sets SSR:TDRE=1, and if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(3) After the number of data set in the TBYTE is completed for transmission, transmission operation will be terminated.

(4) After hold time for the serial chip select pin is passed after the transmission operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM=1) is held at this time, the serial chip select pin (SCS) will remain active.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and reception operation will be started after the setup time is passed for the serial chip select pin. The start of the reception results in sampling the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for reception, SSR:RDRF is cleared to "0".

(5) After hold time for the serial chip select pin is passed after the reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM=1) is held at this time, the serial chip select pin (SCS) will remain active.

Notes:

- When you make reception operation only, make sure to write a dummy data to the TDR in order to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting desired number of frames to be transferred to the FBYTE register will make serial clock (SCK) output for the setup number of frames.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and then serial chip select pin (SCS) will become active and transmission/reception will be started after the setup time is passed for the serial chip select pin. When transmission/reception is started, the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

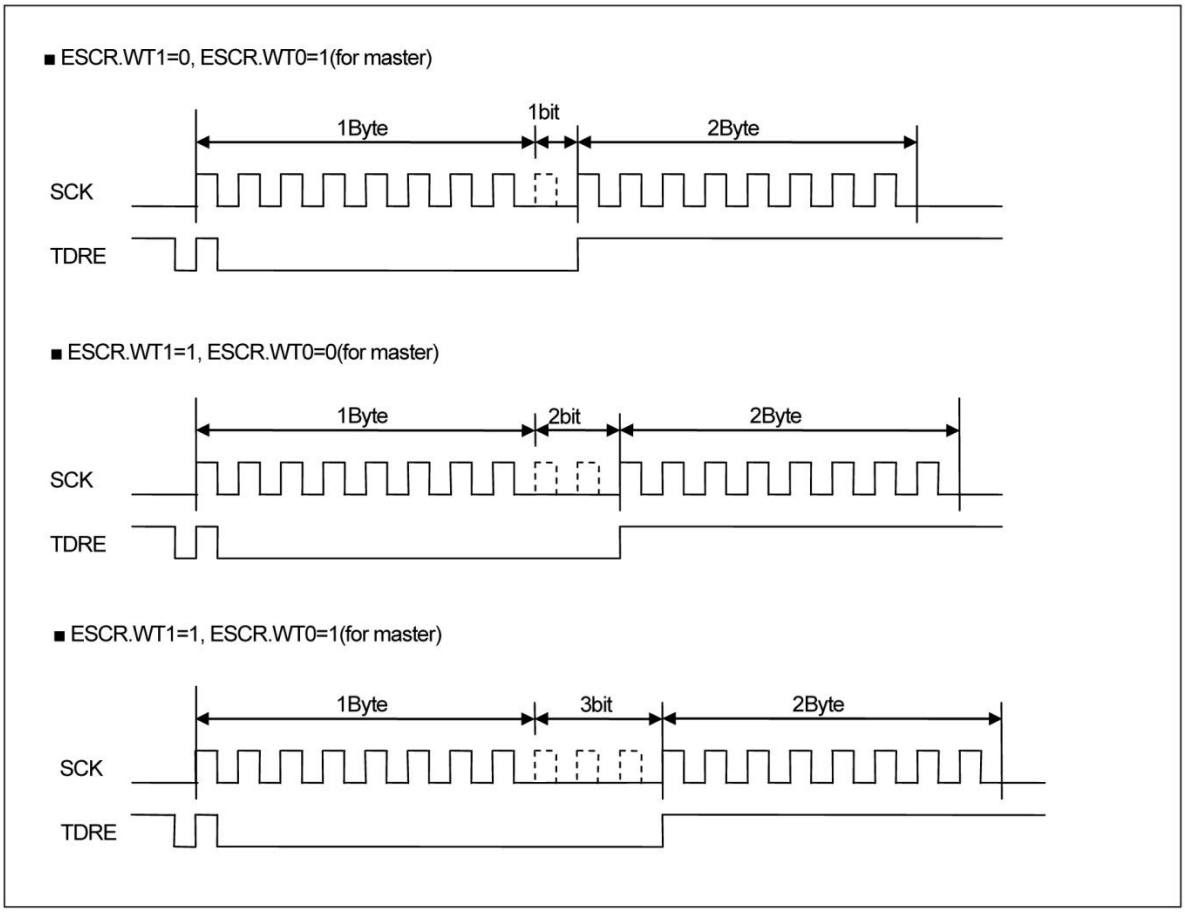
(3) While operating transmission/reception, the reception data will be sampled at a falling edge of the serial clock (SCK) output. Receiving the last bit of receiving data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for transmission/reception, transmission/reception operation will be terminated.

(5) After hold time for the serial chip select pin is has elapsed after the transmission/reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM=1) is held at this time, the serial chip select pin (SCS) will remain active.

Continuous Data **Transmission or Reception Wait Operation**

(1) When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous transmission or reception, a wait will be inserted between frames.



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=1, SCSCR:CSOE=0, SCSCR:SCAM=0)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) Transmission operation will be started when serial chip select pin (SCS) becomes active, the transmission data will be output in synchronization with a rising edge of the serial clock (SCK) input.

(3) Outputting the transmission data in the first bit sets SSR:TDRE=1, and if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(4) Transmission operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), reception operation will be started when the serial chip select pin (SCS) becomes active, and reception data will be sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Reception operation will be terminated when serial chip select pin (SCS) becomes inactive.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When serial chip select pin (SCS) will become active, transmission/reception operation will be started and the transmission data is output in synchronization with the rising edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) While operating transmission/reception, the reception data will be sampled at a falling edge of the serial clock input (SCK). Receiving the last bit of receive data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) The transmission/reception operation will be started when the serial chip select pin (SCS) becomes inactive, and the serial output pin (SOT) becomes "H".

32.6.2.3 SPI Transfer (I)

This section explains the SPI transfer (I).

Features

Table 32-11. Features of SPI Transfer (I)

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Transmission data output timing	SCK rising edge
3	Reception data sampling	SCK falling edge
4	Data length	5 to 9 bits

Register Settings

The following table lists the register settings required for SPI transfer (I).

SCR:SPI*=1, SMR:MD2=0, MD1=1, MD0=0, SCINV*=0
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0

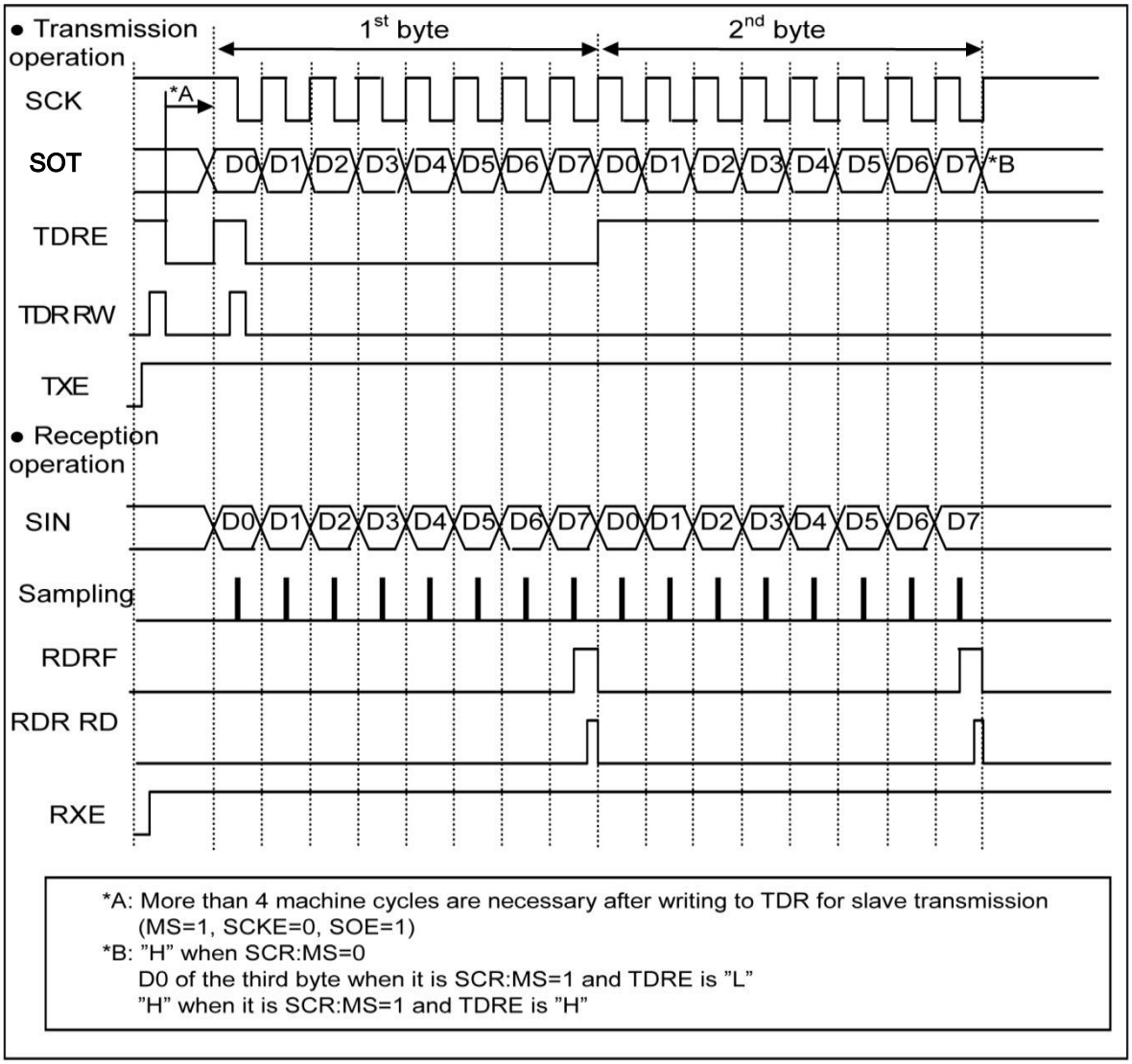
* Bit settings depend on the condition. See Table 32-13 for details.

Note:

Use proper usage for setting the registers other than the above.

SPI Transfer (I) Timing Chart (Serial Chip Select Pin Unused)

Figure 32-30. SPI Transfer (I) Timing Chart (Serial Chip Select Pin Unused)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSEN1-0="00_B")

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0 and outputs the first bit. Then, the transmission data is output in synchronization with a rising edge of the serial clock (SCK) output.

(2) Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE is set to 1. This results in generating a transmission interrupt request when the transmission interrupt is enabled (SCR:TIE=1). At this time, the transmission data in the second byte can be written.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

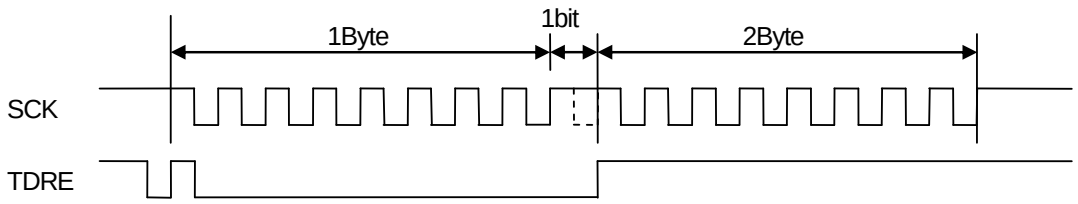
(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the falling edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set. When reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

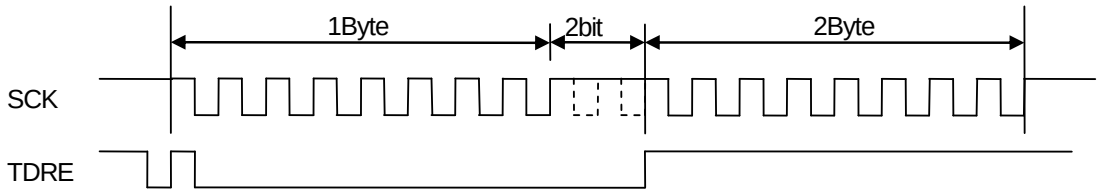
Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.

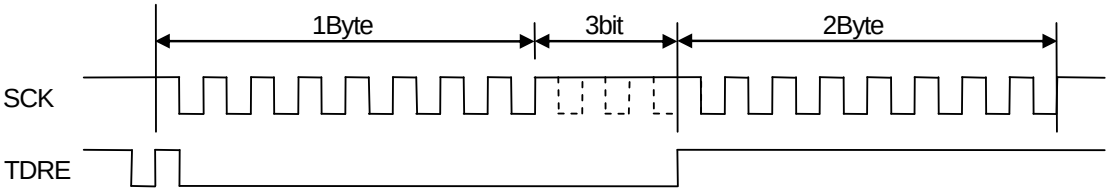
■ ESCR.WT1=0, ESCR.WT0=1(for master)



■ ESCR.WT1=1, ESCR.WT0=0(for master)



■ ESCR.WT1=1, ESCR.WT0=1(for master)



[2] Slave operation(Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN0=0.)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. Therefore, the first bit is output. This results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) When the first bit of the transmission data is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

Note:

After transmission operation is enabled (SCR:TXE=1), when you write first transmission data to the TDR while serial clock (SCK) is at a level other than mark level, the first bit of the data will not be output and correct transmission operation will not be performed. After transmission operation is enabled (SCR:TXE=1), writing the first transmitting data to the TDR must be made while serial clock (SCK) is at mark level.

Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be output.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Transmission/Reception operation

To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the rising edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the falling edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

Continuous change from reception operation to transmission operation

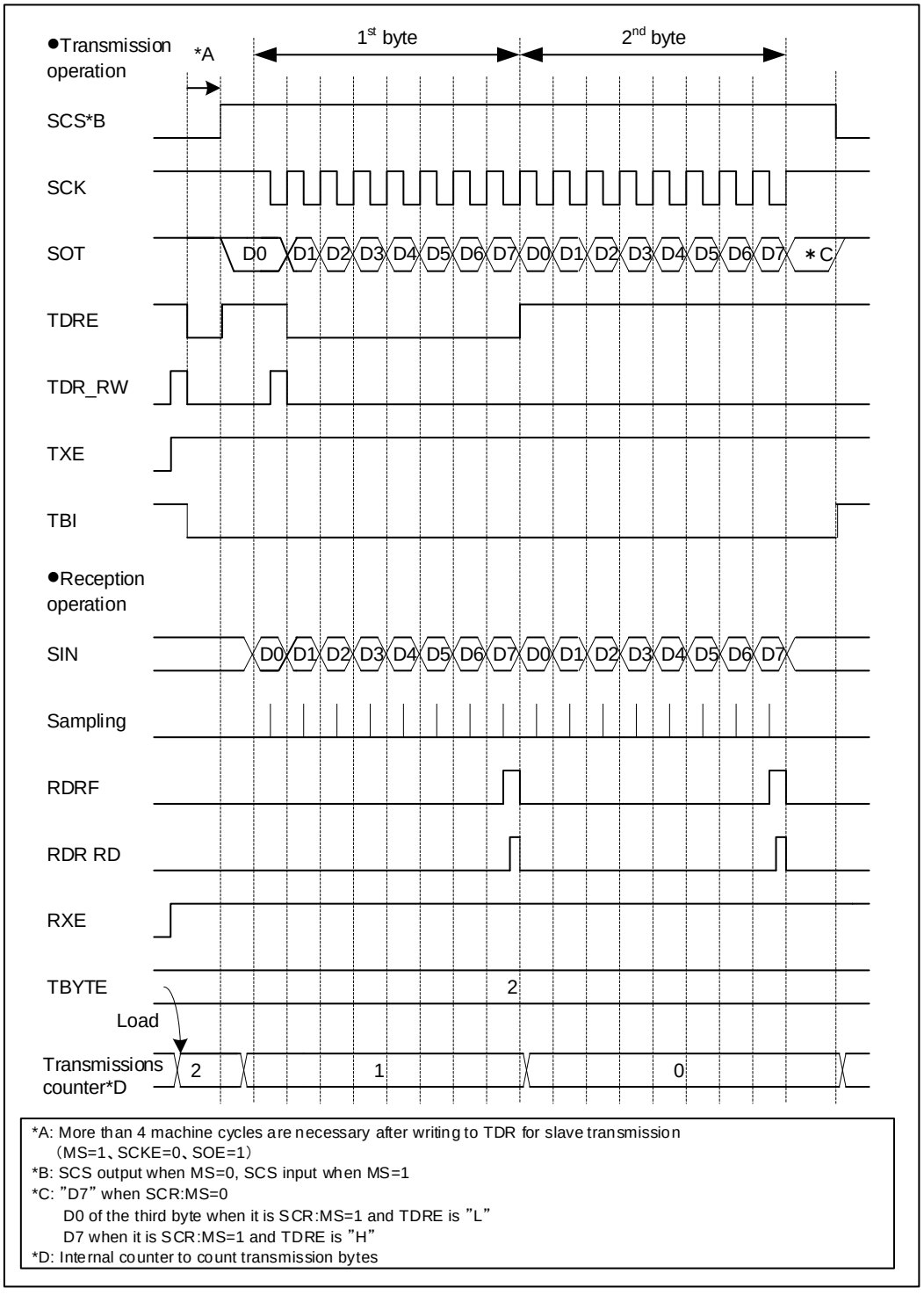
(1) Disable serial data output (SMR:SOE=0), enable reception interrupt (SCR:RIE=1), enable reception operation (SCR: RXE=1), and enable transmission operation (SCR:TXE=1). If dummy data is written to TDR when serial clock (SCK) is at the mark level, receive data is sampled at the falling edge of the serial clock input (SCK).

(2) To continue reception operation, write dummy data to TDR between the reception interrupt request and the rising edge of the next serial clock (SCK).

(3) To switch from reception operation to transmission operation, enable serial data output (SMR:SOE=1), disable reception interrupt (SCR:RIE=0), and disable reception operation (SCR:RXE=0) between the reception interrupt request and the rising edge of the next serial clock (SCK), and after transmission data is written to TDR and reception operation finishes, transmission data will be output in synchronization with the rising edge of the serial clock.

SPI Transfer (I) Timing Chart (Serial Chip Select Pin Used)

Figure 32-31. SPI Transfer (I) Timing Chart (Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

* "n" shows the number of serial chip select pin used.

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. Then serial chip select pin (SCS) will become active at the same time of outputting the first bit and transmission will be started after the setup time is passed for the serial chip select pin. The start of the transmission results in outputting the transmission data in synchronization with a rising edge of the serial clock (SCK) output.

(2) Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE is set to 1 and a transmission interrupt request will be generated when the transmission interrupt is enabled (SCR:TIE=1). At this time, the transmission data in the second byte can be written.

(3) After the number of data set in the TBYTE is completed for transmission, transmission operation will be terminated.

(4) After hold time for the serial chip select pin is passed after the transmission operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and reception operation will be started after the setup time is passed for the serial chip select pin. The start of the reception results in sampling the reception data at a falling edge of the serial clock output (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) After the number of data set in the TBYTE is completed for reception, reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

Notes:

- When you make reception operation only, make sure to write a dummy data to the TDR in order to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting desired number of frames to be transferred to the FBYTE register will make serial clock (SCK) output for the setup number of frames.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set, and then serial chip select pin (SCS) will become active at the same time of outputting the first bit and transmission/reception will be started after the setup time is passed for the serial chip select pin. When transmission/reception is started, the transmission data is output in synchronization with the rising edge of serial clock (SCK) output. Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) The reception data will be sampled at a falling edge of the serial clock output (SCK). Receiving the last bit of receive data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

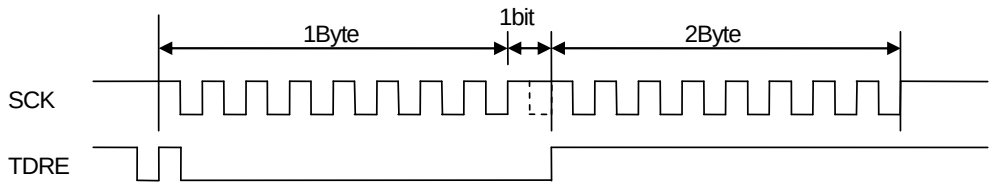
(4) After the number of data set in the TBYTE is completed for transmission/reception, transmission/reception operation will be terminated.

(5) After hold time for the serial chip select pin is passed after the transmission/reception operation, the serial chip select pin (SCS) will become inactive. However, if serial chip select active level (SCSCR:SCAM="1") is held at this time, the serial chip select pin (SCS) will remain active.

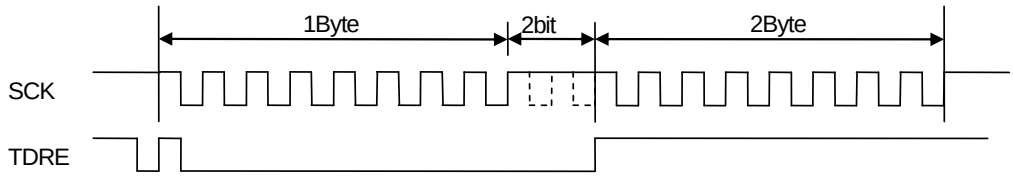
Continuous Data Transmission or Reception Wait Operation

(1) When a setup other than (ESCR:WT1, ESCR:WT0)= (0, 0) is used for continuous data transmission or reception, a wait will be inserted between frames.

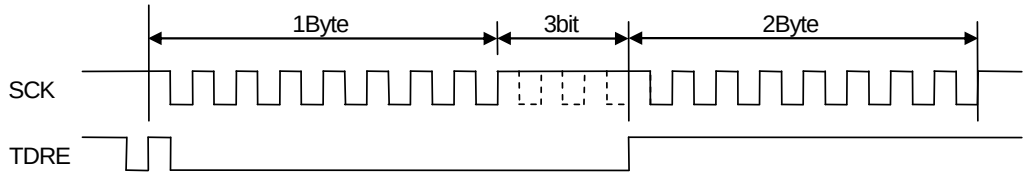
■ ESCR.WT1=0, ESCR.WT0=1(for master)



■ ESCR.WT1=1, ESCR.WT0=0(for master)



■ ESCR.WT1=1, ESCR.WT0=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN=1, SCSCR:SCAM=0)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) Transmission operation will be started and the first bit will be output when serial chip select pin (SCS) becomes active. After transmission operation is started, the transmission data will be output in synchronization with a rising edge of the serial clock (SCK) output.

(3) Outputting the transmission data in the first bit sets SSR:TDRE=1, and the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request will be generated. At this time, the transmission data in the second byte can be written.

(4) Transmission operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

Note:

After transmission operation is enabled (SCR:TXE=1), when you write first transmission data to the TDR while serial clock (SCK) is at a level other than mark level, the first bit of the data will not be output and correct transmission operation will not be performed. After transmission operation is enabled (SCR:TXE=1), writing the first transmission data to the TDR must be made while serial clock (SCK) is at mark level.

Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), reception operation will be started when the serial chip select pin (SCS) becomes active, and reception data will be sampled at a falling edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read.

(3) Reading the received data (RDR) clears SSR:RDRF to "0".

(4) Reception operation will be terminated when serial chip select pin (SCS) becomes inactive.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When serial chip select pin (SCS) will become active, transmission/reception will be started and the first bit will be output. When transmission/reception is started, the transmission data will be output in synchronization with the rising edge of serial clock (SCK) input. When the first bit of transmission data is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) The reception data will be sampled at a falling edge of the serial clock input (SCK). Receiving the last bit of receive data sets SSR:RDRF=1 and when the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be generated. At this time, the receive data (RDR) can be read. Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) Transmission/reception operation will be terminated when serial chip select pin (SCS) becomes inactive, and serial output pin (SOT) becomes "H".

32.6.2.4 SPI Transfer (II)

This section explains the SPI Transfer (II).

Features

Table 32-12. Features of SPI Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Transmission data output timing	SCK falling edge
3	Reception data sampling	SCK rising edge
4	Data length	5 to 16, 20, 24, 32 bits

Register Settings

The following table lists the register settings required for SPI Transfer (II).

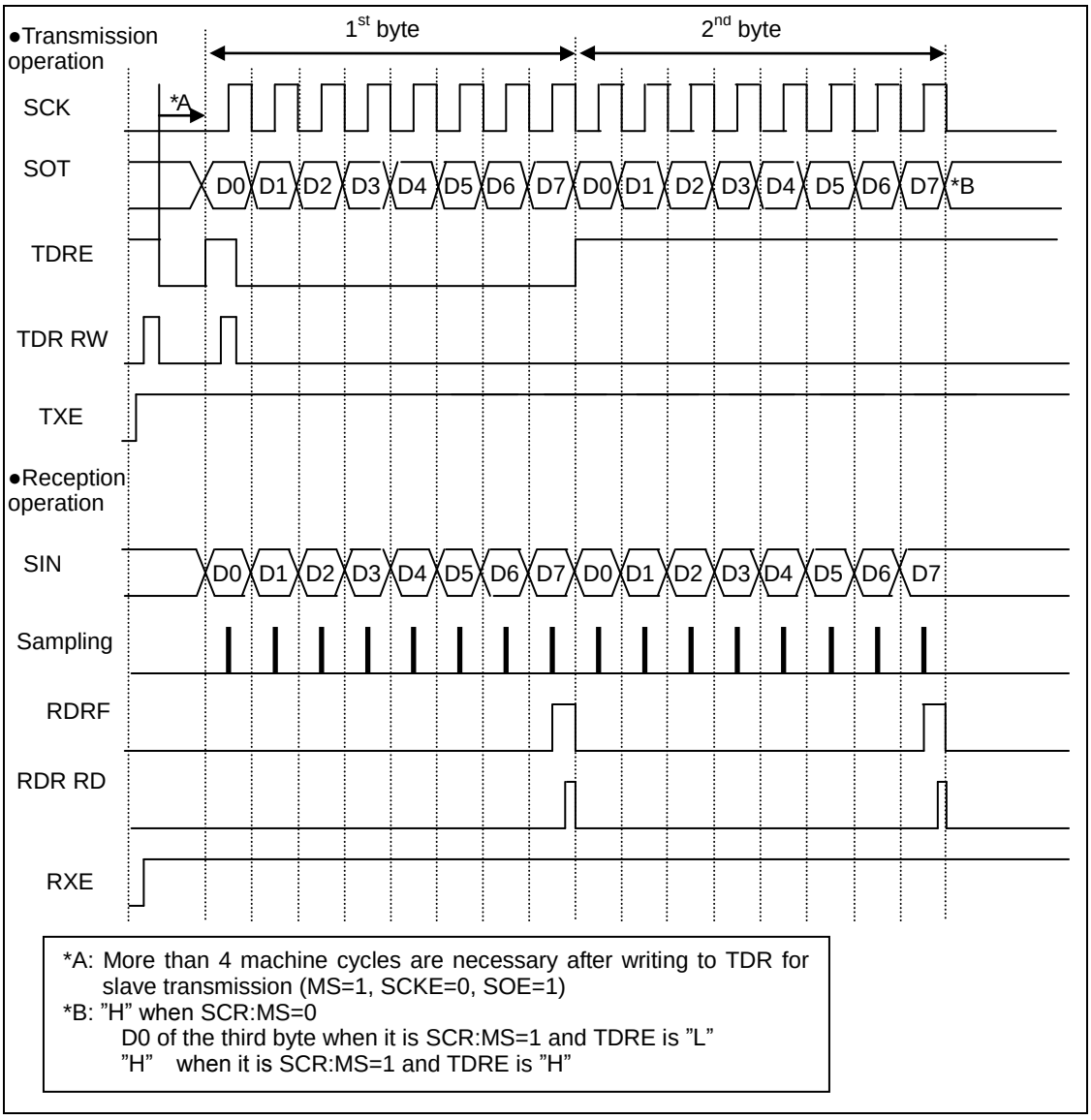
SCR:SPI*=1, SMR:MD2=0, MD1=1, MD0=0, SCINV*=1
Master operations: SCR:MS=0, SMR:SCKE=1
Slave operations: SCR:MS=1, SMR:SCKE=0
* Bit settings depend on the condition. See Table 32-13 for details.

Note:

Use proper usage for setting the registers other than the above.

SPI Transfer (II) Timing Chart (Serial Chip Select Pin Unused)

Figure 32-32. SPI Transfer (II) Timing Chart (Serial Chip Select Pin Unused)



*A: More than 4 machine cycles are necessary after writing to TDR for slave transmission (MS=1, SCKE=0, SOE=1)
*B: ”H” when SCR:MS=0
D0 of the third byte when it is SCR:MS=1 and TDRE is ”L”
”H” when it is SCR:MS=1 and TDRE is ”H”

[1] Master operation (Set SCR:MS=0, SMR:SCKE=1.)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. This results in outputting transmission data in synchronization with a falling edge of the serial clock (SCK) output.

(2) Half a cycle before a rising edge of the first serial clock (SCK), SSR:TDRE is set to 1. This results in generating a transmission interrupt request when the transmission interrupt is enabled (SCR:TIE=1). At this time, the transmission data in the second byte can be written.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR samples the receive data at a rising edge of the serial clock (SCK) output.

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request.
At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

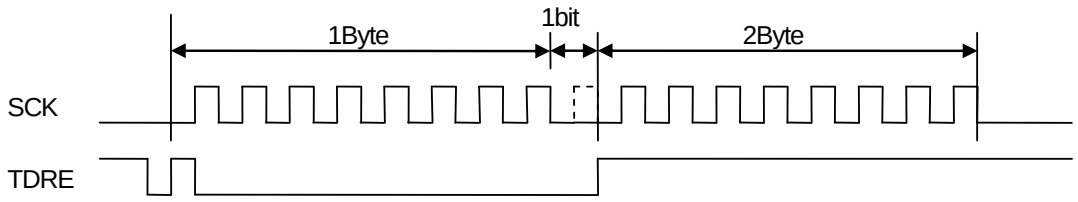
(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the falling edge of serial clock (SCK) output. Half a cycle before a rising edge of the first serial clock (SCK), SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set. When reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

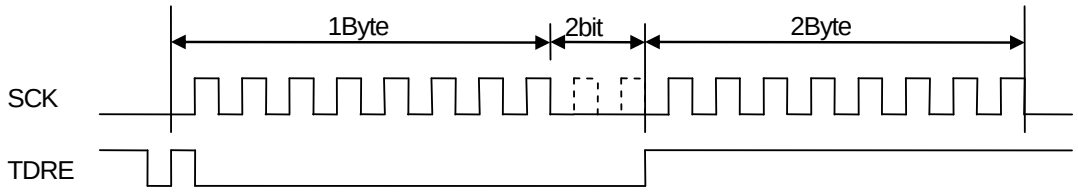
Successive data transmission or reception wait operation

(1) If setting other than (ESCR:WT1, ESCR:WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.

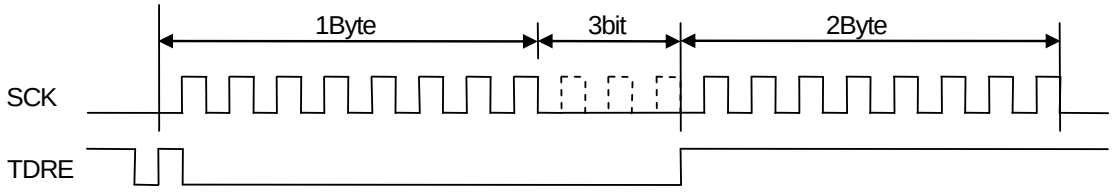
■ ESCR.WT1=0, ESCR.WT0=1(for master)



■ ESCR.WT1=1, ESCR.WT0=0(for master)



■ ESCR.WT1=1, ESCR.WT0=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0.)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0. Therefore, the first bit is output. This results in outputting the transmission data in synchronization with a falling edge of the serial clock (SCK) input.

(2) When the first bit of the transmission data is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

Note:

After the transmission operation is enabled (SCR:TXE=1), if transmission data is written to the first TDR except when serial clock (SCK) is at the mark level, the first bit of data is not output and transmission is not operated normally. After the transmission operation is enabled (SCR:TXE=1), write transmission data to the first TDR when serial clock (SCK) is at the mark level.

Reception operation

(1) With serial data output disabled (SMR:SOE=0) and reception operation enabled (SCR:RXE=1), the reception data is sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request. At this time, the receive data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set and the first bit is output. Then, the transmission data is output in synchronization with the falling edge of serial clock (SCK) input. When transmission data of the first bit is output, SSR:TDRE=1 is set, and when transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data of the second byte can be written.

(3) Receive data is sampled by the rising edge of serial clock (SCK) input. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, receive data (RDR) can be read. When receive data is read, SSR:RDRF is cleared to "0".

Continuous change from reception operation to transmission operation

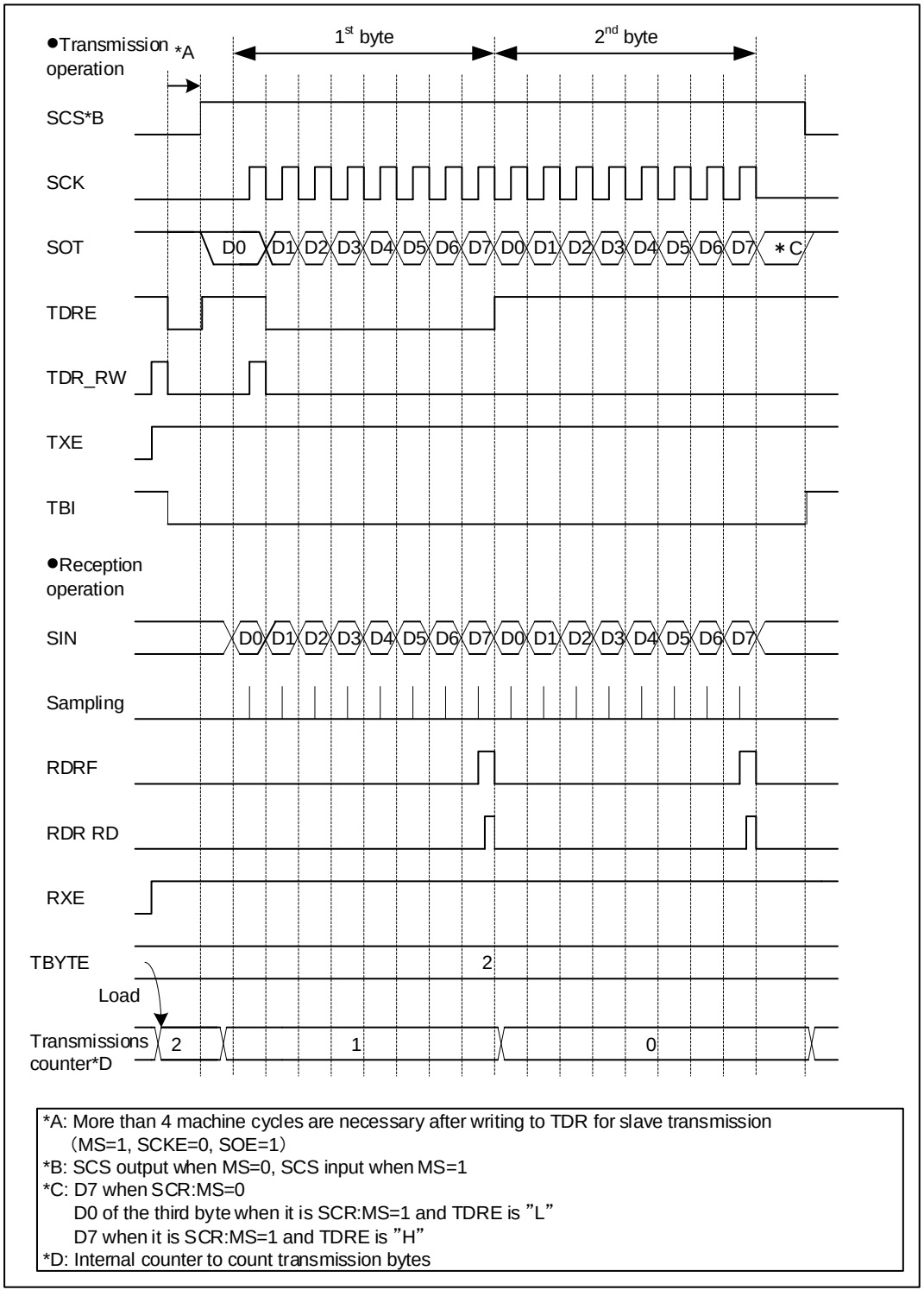
(1) Disable serial data output (SMR:SOE=0), enable reception interrupt (SCR:RIE=1), enable reception operation (SCR:RXE=1), and enable transmission operation (SCR:TXE=1). If dummy data is written to TDR when serial clock (SCK) is at the mark level, receive data is sampled at the falling edge of serial clock (SCK) input.

(2) To continue reception operation, write dummy data to TDR between the reception interrupt request and the rising edge of the next serial clock (SCK).

(3) To switch from reception operation to transmission operation, enable serial data output (SMR:SOE=1), disable reception interrupt (SCR:RIE=0), and disable reception operation (SCR:RXE=0) between the reception interrupt request and the rising edge of the next serial clock (SCK), and after transmission data is written to TDR and reception operation finishes, transmission data will be output in synchronization with the rising edge of the serial clock.

SPI Transfer (II) Timing Chart (When Serial Chip Select Pin Used)

Figure 32-33. SPI Transfer (II) Timing Chart (When Serial Chip Select Pin Used)



[1] Master operation (Set SCR:MS=0, SMR:SCKE=1, SCSCR:CSOE=1, SCSCR:CSENn*=1)

* n is the serial chip select pin number to be used.

Transmission operation

(1) With serial data output enabled (SMR:SOE=1), transmission operation enabled (SCR:TXE=1), and reception operation disabled (SCR:RXE=0), writing transmission data to TDR sets SSR:TDRE=0. After that, the serial chip select pin (SCS) becomes active at the same time the first bit is output, and the transmission operation starts after the setup time of the serial chip select pin has passed. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.

(2) Half a cycle before a falling edge of the first serial clock (SCK), SSR:TDRE is set to 1 and, if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(3) The transmission operation is terminated after the data transmission is completed as many as the number of times set with TBYTE.

(4) Then, after the hold time of serial chip select pin has passed, the serial chip select pin (SCS) becomes inactive. However, if the serial chip select active level (SCSCR:SCAM=1) is maintained at this time, the serial chip select pin (SCS) maintains its active state.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), transmission operation enabled (SCR:TXE=1), and reception operation enabled (SCR:RXE=1), writing dummy data to TDR makes the serial chip select pin (SCS) active and starts the reception operation after the setup time of that pin has passed. Starting the reception operation samples the receive data at a rising edge of the serial clock (SCK) output.

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request will be output.
At this time, the reception data (RDR) can be read.

(3) Reading the receive data (RDR) clears SSR:RDRF to "0".

(4) The reception operation is terminated after the data reception is completed as many as the number of times set with TBYTE.

(5) Then, after the hold time of serial chip select pin has passed, the serial chip select pin (SCS) becomes inactive. However, if the serial chip select active level (SCSCR:SCAM=1) is maintained at this time, the serial chip select pin (SCS) maintains its active state.

Notes:

- If only reception operation is to be performed, write dummy data to TDR to output the serial clock (SCK).
- When transmission/reception FIFO is enabled, setting the FBYTE register to the number of frames to be transferred outputs as many frames of serial clock (SCK) as the setting.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. After that, the serial chip select pin (SCS) becomes active at the same time the first bit is output, and the transmission/reception operation starts after the setup time of the serial chip select pin has passed. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) output. Half a cycle before a rising edge of the first serial clock, SSR:TDRE is set to 1 and, if the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(3) Reception data is sampled by the rising edge of the serial clock (SCK) output. When the last bit of receive data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, the reception data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".

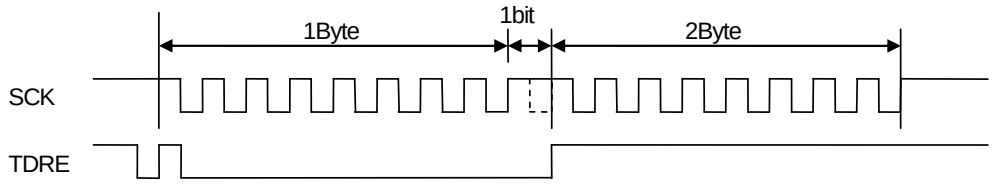
(4) The transmission/reception operation is terminated after the data transmission/reception is completed as many as the number of times set with TBYTE.

(5) Then, after the hold time of serial chip select pin has passed, the serial chip select pin (SCS) becomes inactive. However, if the serial chip select active level (SCSCR:SCAM=1) is maintained at this time, the serial chip select pin (SCS) maintains its active state.

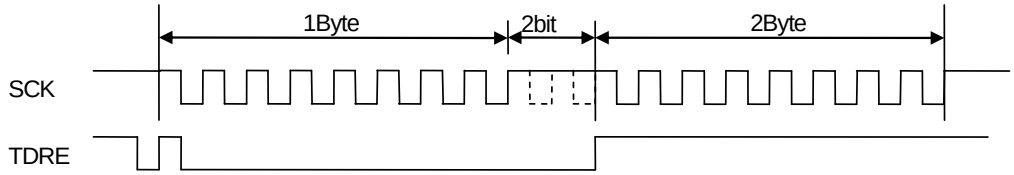
Successive data transmission or reception wait operation

(1) If setting other than (ESCR.WT1, ESCR.WT0)=(0, 0) is specified for successive data transmission or reception, a wait is inserted between frames.

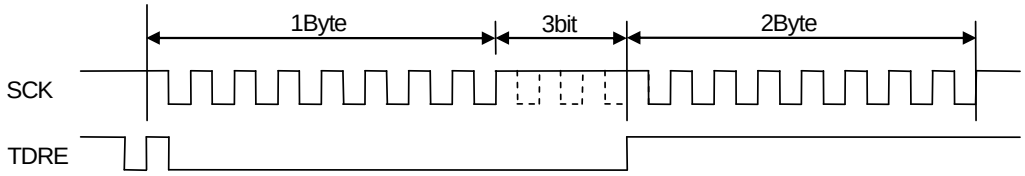
■ ESCR.WT1=0, ESCR.WT0=1(for master)



■ ESCR.WT1=1, ESCR.WT0=0(for master)



■ ESCR.WT1=1, ESCR.WT0=1(for master)



[2] Slave operation (Set SCR:MS=1, SMR:SCKE=0, SCSCR:CSEN=1, SCSCR:SCAM=0)

Transmission operation

(1) With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0.

(2) When the serial chip select pin (SCS) becomes active, the transmission operation is started and the data of the first bit is output. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) output.

(3) When the transmission data of the first bit is output, SSR:TDRE=1 is set, and when the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(4) When the serial chip select pin (SCS) becomes inactive, the transmission operation is terminated and the serial output pin (SOT) becomes "H".

Note:

After the transmission operation is enabled (SCR:TXE=1), if transmission data is written to the first TDR except when serial clock (SCK) is at the mark level, the first bit of data is not output and transmission is not operated normally. After the transmission operation is enabled (SCR:TXE=1), write transmission data to the first TDR when serial clock (SCK) is at the mark level.

Reception operation

(1) With serial data output disabled (SMR:SOE=0), and reception operation enabled (SCR:RXE=1), the reception operation starts when the serial chip select pin (SCS) becomes active, and then the reception data is sampled at a rising edge of the serial clock input (SCK).

(2) Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), outputs a reception interrupt request.
At this time, the reception data (RDR) can be read.

(3) Reading the reception data (RDR) clears SSR:RDRF to "0".

(4) When the serial chip select pin (SCS) becomes inactive, the reception operation is terminated.

Transmission/Reception operation

(1) To perform transmission and reception at the same time, enable serial data output (SMR:SOE=1) and enable transmission/reception operation (SCR:TXE, RXE=1).

(2) When transmission data is written in TDR, SSR:TDRE=0 is set. When the serial chip select pin (SCS) becomes active, the transmission/reception operation is started and the data of the first bit is output. Then, the transmission data is output in synchronization with a falling edge of the serial clock (SCK) input. When the transmission data of the first bit is output, SSR:TDRE=1 is set, and when the transmission interrupt is enabled (SCR:TIE=1), a transmission interrupt request is output. At this time, the transmission data in the second byte can be written.

(3) Reception data is sampled by the rising edge of the serial clock (SCK) input. When the last bit of reception data is received, SSR:RDRF=1 is set, and when reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is output. At this time, the reception data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".

(4) When the serial chip select pin (SCS) becomes inactive, the transmission/reception operation is terminated and the serial output pin (SOT) becomes "H".

32.6.2.5 Operation of Serial Timer

This section explains the operation of the serial timer.

The serial timer can be used for either of the timer function or the synchronous transmission function.

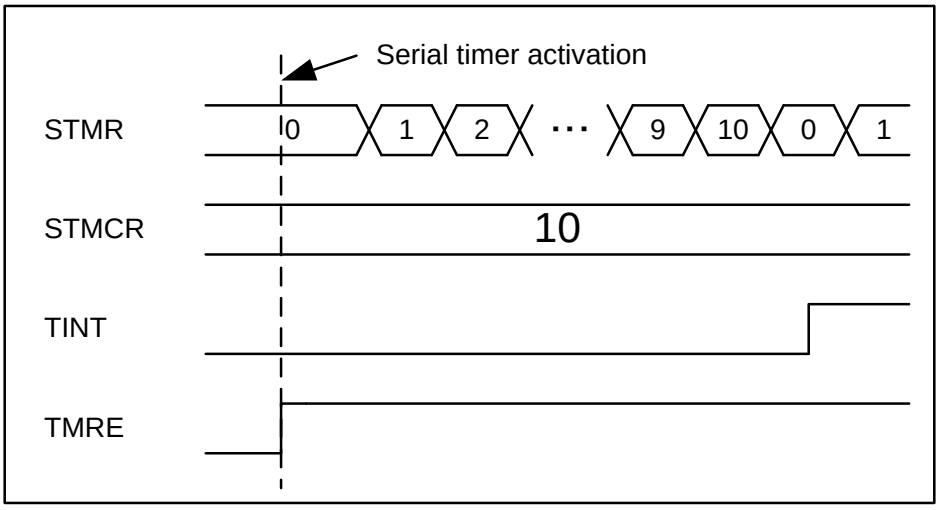
How to Start Serial Timer

To start the serial timer: setting "1" to the serial timer enable bit (SACSR:TMRE).

- Start by using the serial timer enable bit (SACSR:TMRE)

When the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer starts and the serial timer register (STMR) starts counting from 0.

Figure 32-34. Start by Using Serial Timer Enable Bit
(STMCR="10", SACSR:TSYNE="0")



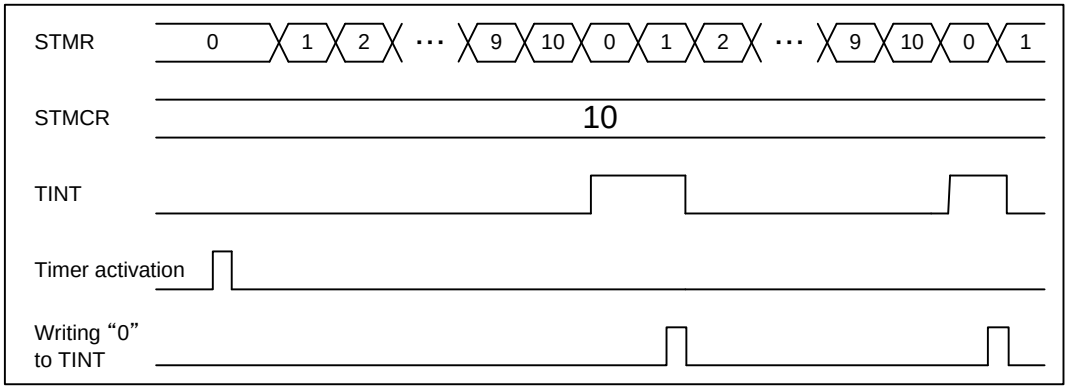
How to Stop Serial Timer

When the serial timer enable bit (SACSR:TMRE) is set to "0", the serial timer will stop. The value of the serial timer register (STMR) is retained.

Timer Operation

The serial timer operates as a timer.
If the serial timer register (STMR) matches the serial timer comparison register (STMCR), the timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to "0".

Figure 32-35. Timer Operation (STMCR="10")



Note:

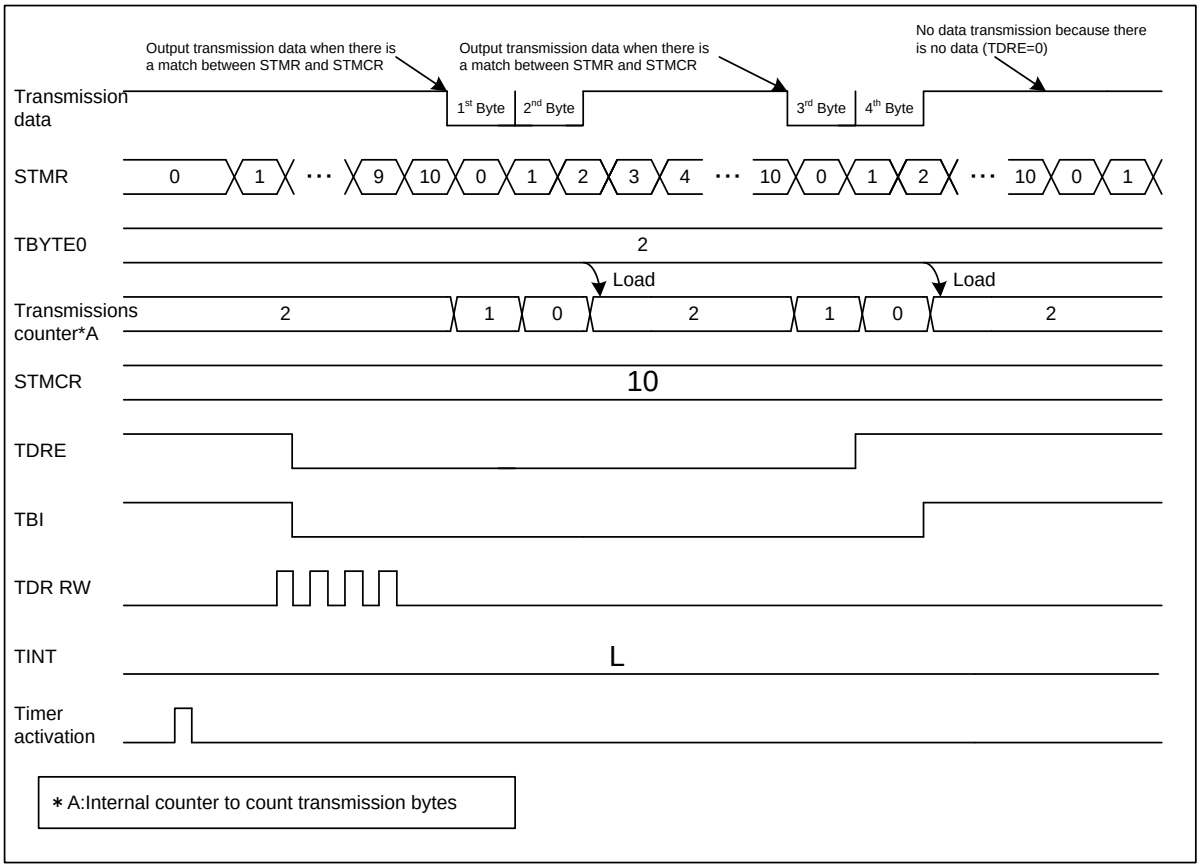
When the timer comparison register (STMCR) is set to "0000_H" with the synchronous transmission disabled (SACSR:TSYNE=0), the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B".

Synchronous Transmission Operation

When the synchronous transmission enable bit (SAGSR:TSYNE) is set to "1", the serial timer is used for a synchronous transmission.
The synchronous transmission operates as shown below:

1. If the transmission data register contains data (SSR:TDRE="0") and the serial timer register (STMR) matches the serial timer comparison register (STMCR), the transmission operation starts and the serial timer register (STMR) is reset to "0". It continues to send the same number of data as the number set to TBYTE0.
2. If the data transmission is completed as many as the number of data set to TBYTE0, then, the transmission operation stops until the serial timer register (STMR) matches the serial timer comparison register (STMCR).

Figure 32-36. Synchronous Transmission Operation (STMCR="10", TBYTE0="2")



If the synchronous transmission is enabled (SACSR:TSYNE="1") and the serial timer register (STMR) matches the serial timer comparison register (STMCR), the transmission will not start under the following conditions.

- When transmission is disabled (SCR:TXE="0")
- In slave mode (SCR:MS="1")
- When the chip select error (SACSR:CSE="1") occurred
- When there is no valid data in the transmission data register (SSR:TDRE="1")

However, if the transmission data register contains no valid data (SSR:TDRE=1), the synchronous transmission is enabled (SACSR:TSYNE="1"), and the serial timer register (STMR) matches the serial timer comparison register (STMCR), writing the transmission data to transmission data register starts a transmission immediately.
If the transmission data register (TDR) has valid transmission data (SSR:TDRE="0") after the data transmission is completed as many as the number of data set to TBYTE, then, the transmission data will not be sent until the serial timer register (STMR) matches the serial timer comparison register (STMCR).
However, if the synchronous transmission is enabled (SACSR:TSYNE="1") and the serial timer register (STMR) matches the serial timer comparison register (STMCR) under the following conditions, the transmission will not be stopped and the next transmission will be started after sending as much data as TBYTE0.

- Transmission operation in progress (SSR:TBI="0")

If synchronous reception operation is performed, disable serial data output (SMR:SOE="0"), enable transmission operation (SCR:TXE="1"), enable reception operation (SCR:RXE="1"), and write dummy data to TDR as many as the number of receptions.

Note:

- When the transmit data register (TDR) has no valid transmission data (SSR:TDRE="1") before sending the data frame of TBYTE setting, the following operation will be performed.
  - ☐ When the transfer byte error is enabled (TBEEN="1"), a chip select error (SACSR:CSE="1") occurs. When the chip select error flag (SACSR:CSE) is set to "1", writing the transmission data to the transmit data register (TDR) does not start transmission operation.
  - ☐ When the transfer byte error is disabled (TBEEN="0"), transmission operation will be stopped until transmission data is written to the transmit data register (TDR). Transmission operation will be restarted when transmission data is written to the transmit data register (TDR).

32.6.2.6 Operation of Serial Chip Select

This section explains the operation of serial chip select.

This section explains the operation of the serial chip select.

Operation of master mode (SCR:MS="0")

In master mode (SCR:MS=0), the serial chip select pin operates as shown below:

(1) With the serial chip select operation enabled (SCSCR:CSENn="1") and transmission enabled (SCR:TXE="1"), writing transmission data makes the serial chip select pin active.

(2) The transmission/reception operation starts after the setup time of the serial chip select pin has passed.

(3) The transmission/reception operation is terminated after the data transmission/reception operation is repeated as many as the number of times set with TBYTE.

(4) Then, after the hold time of serial chip select pin has passed, the serial chip select pin becomes inactive.

Figure 32-37. Operation of Serial Chip Select (Master Transmission (MS="0"), Normal Transfer (SPI="0"), SCINV="0")

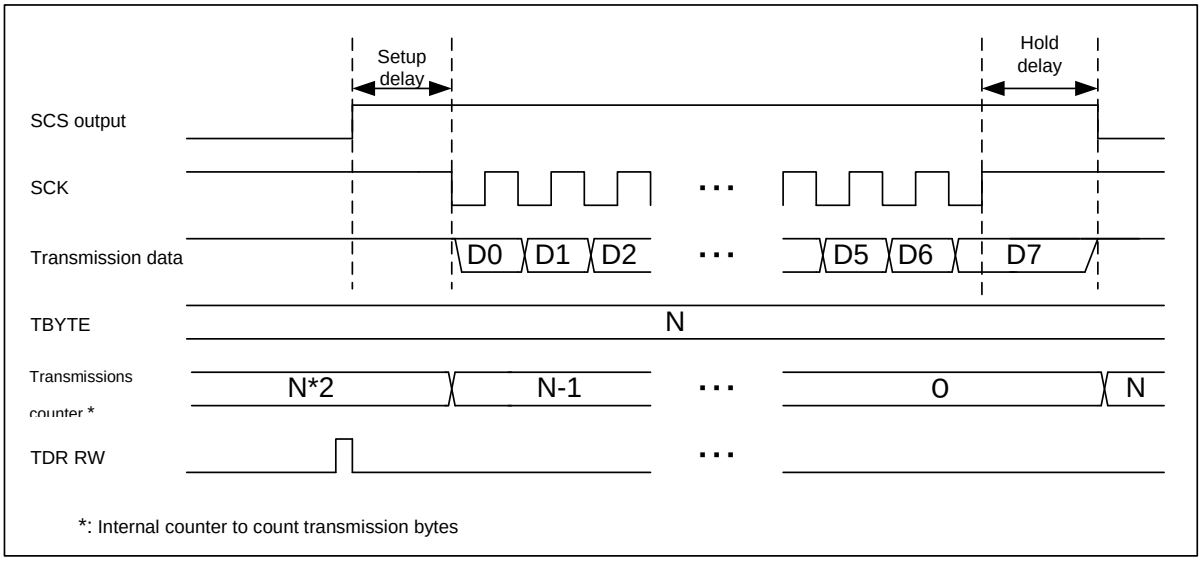
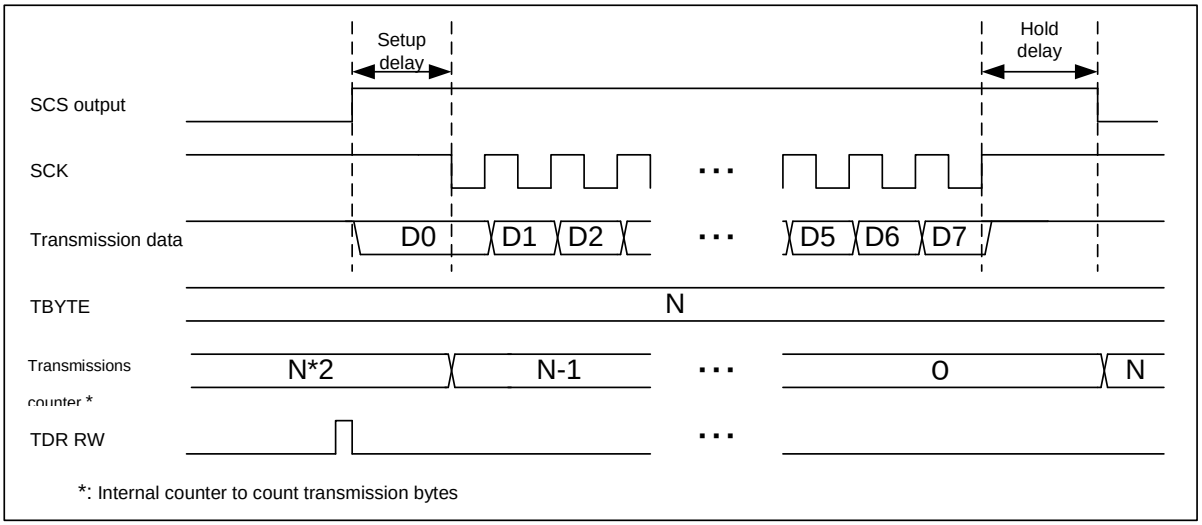


Figure 32-38. Operation of Serial Chip Select (Master Transmission (MS="0"), SPI Transfer (SPI="1"), SCINV="0")



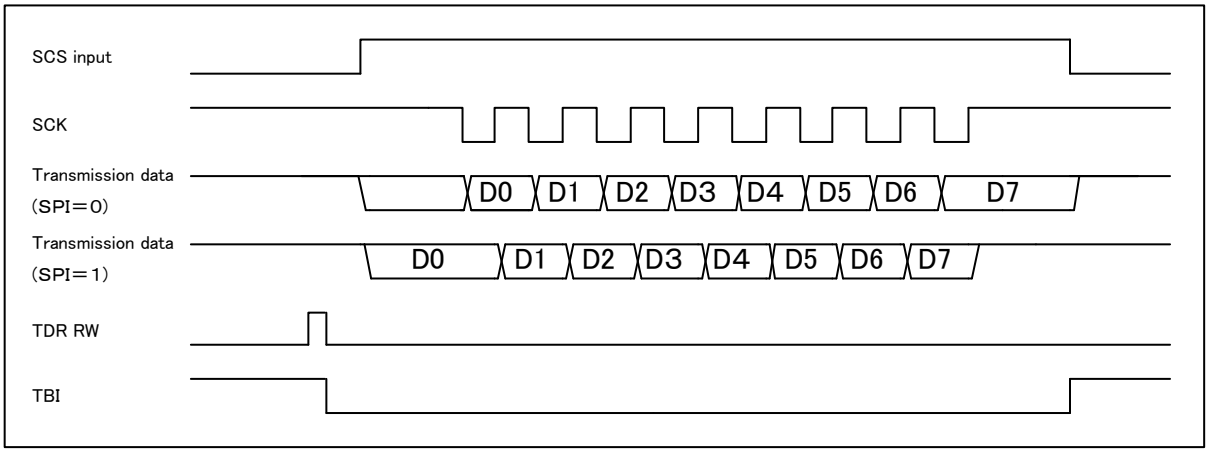
Notes:

- If the transmission is disabled (SCR:TXE="0") and the software reset is set (SCR:UPCL=1) when the serial chip select pin is active, the serial chip select pin becomes inactive.
- If the transmission data is written to the hold delay time of the serial chip select pin, the pin does not become inactive and the next transmission data is sent.
- When the serial chip select pin does not maintain its active state (SCSCR:SCAM=0) and becomes inactive, transmission bus idle (SSR:TBI=1) is set.
- If SCSCR:CSEN1 to CSEN0 is set to "00b" in master mode (SCR:MS=0), transmission/reception operation is performed regardless of the serial chip select pin.
- If the transmit data register (TDR) has no valid transmission data (SSR:TDRE=1) when 1 frame transmission is completed when the number of the frames sent is less than the number set to TBYTE, the following operation will be performed.
  - ☐ When the transfer byte error is enabled (TBEEN=1), a chip select error (SACSR:CSE=1) occurs. After the hold delay time has passed since the chip select error (SACSR:CSE=1) occurred, the serial chip select pin becomes inactive. When the chip select error flag (SACSR:CSE) is set to "1", writing the transmission data to the transmit data register (TDR) does not start transmission operation.
  - ☐ When the transfer byte error is disabled (TBEEN=0), transmission operation will be stopped until transmission data is written to the transmission data register (TDR). At this time, the serial chip select pin is active. Transmission operation will be restarted when transmission data is written to the transmit data register (TDR).

Operation of slave mode (SCR:MS="1")

When the serial chip select pin 0 (SCS0) is enabled (SCSCR:CSEN0="1") and the serial chip select pin input becomes active, the transmission or reception operation is performed in synchronization to the serial clock (SCK). Then, when the serial chip select pin input becomes inactive, the transmission or reception operation is terminated.

Figure 32-39. Operation of Serial Chip Select in Slave Mode (Slave Transmission, SCINV="0")



Notes:

- It does not operate even if the serial clock is input when the serial chip select pin input is inactive.
- If the serial chip select input becomes inactive before sampling a bit at the end during reception operation, the data being received is deleted.
- If the serial chip select input becomes inactive during transmission operation, the data being transmitted is deleted and the chip select error occurs (SACSR:CSE).
- When the serial chip select pin input becomes inactive, the transmission bus idle (SSR:TBI=1) is set.
- If SCSCR:CSEN0 is set to "0" in slave mode (SCR:MS=1), transmission/reception operation is performed regardless of the serial chip select pin.

Timing Adjustment of Serial Chip Select

When the serial chip select operation is enabled (SCSCR:CSENn="1") in master mode (SCR:MS=0), the setup delay time, hold delay time, and deselection time can be adjusted by adjusting the serial chip select timing register (SCSTR3 to SCSTR0).

- Setup delay time

  Time from the instant when the serial chip select pin becomes active to the instant when the serial clock is output. See Figure 32-40 and Figure 32-41 for the specification of the setup delay time.
  It can be adjusted by the chip select setup delay bit (SCSTR0:CSSU7 to CSSU0).

- Hold delay time

  Time from the instant when the serial clock output is completed to the instant when the serial chip select pin becomes inactive. See Figure 32-40 and Figure 32-41 for the specification of the hold delay time.
  It can be adjusted by the chip select hold delay bit (SCSTR1:CSHD7 to CSHD0).

- Deselection time

  Minimum time from the instant when the serial chip select pin becomes inactive to the next instant when the pin turns active. Even if transmission data is written to the transmit data register (TDR) during the deselection time, the serial chip select pin does not become active until the deselection time end. See Figure 32-40 and Figure 32-41 for the specification of the deselection time.

Figure 32-40. Timing Adjustment (Normal Transfer (SPI="0"), SCINV="0")

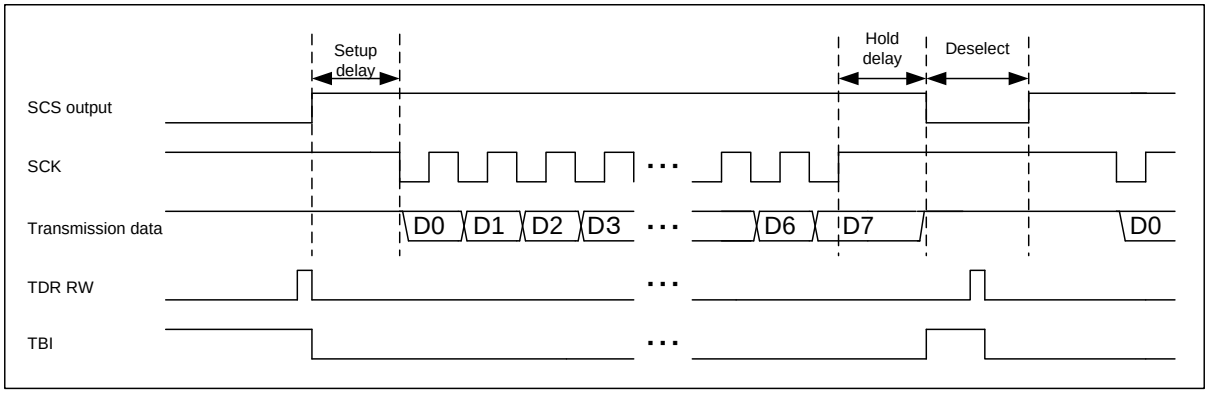
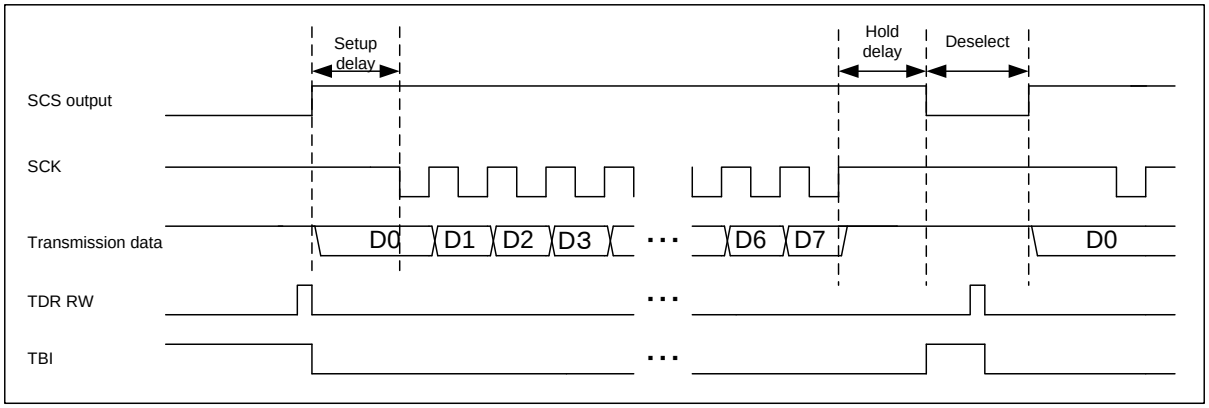


Figure 32-41. Timing Adjustment (SPI Transfer (SPI="1"), SCINV="0")



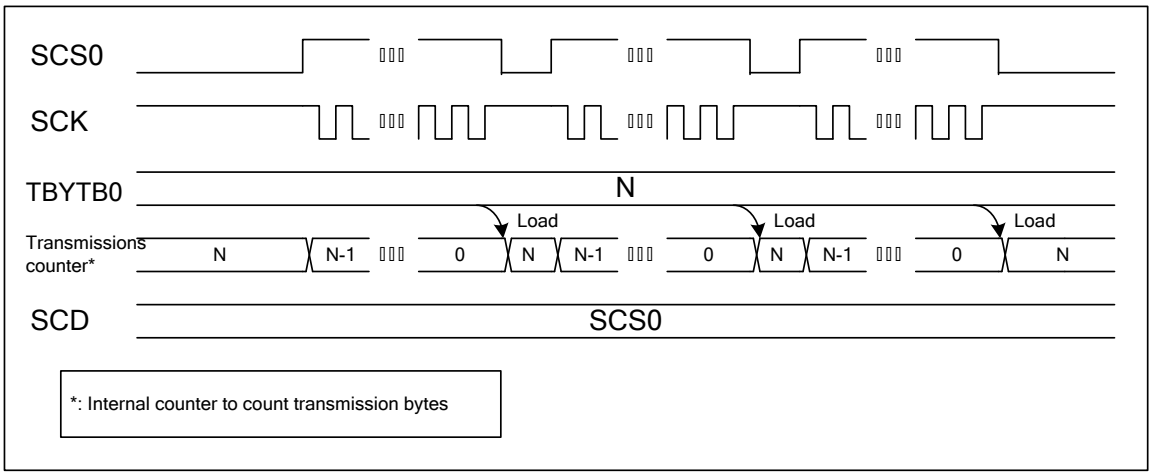
Note:

With the normal transfer (SPI=0) and no hold delay time (SCSTR1:CSHD7 to CSHD0="00_H"), the chip select pin may become inactive before the sampling of the last bit. In such cases, adjust the timing by increasing the value of SCSTR1:CSHD7 to CSHD0.

Single Operation of Chip select pin (Only Valid in Master Mode (SCR:MS="0"))

If the serial chip select start bit (SCSCR:SST0) and the serial chip select end bit (SCSCR:SED0) are equal, only the specified serial chip select pin operates.
When the serial chip select does not maintain its active state (SCSCR:SCAM=0), the serial chip select pin becomes inactive for each the data transmission/receptions for the number of times set with TBYTE.
See "Operation of Serial Chip Select to Maintain Active (SCSCR:SCAM=1) (Only Valid in Master Mode (SCR:MS=0))" for the operation of the serial chip select pin when the serial chip select maintains its active state (SCSCR:SCAM=0).

Figure 32-42. Single Operation of Chip Select (SST0="0", SED0="0", CSEN0="1", SCAM="0")



Note:

At the single operation, timing adjustment of the serial chip select pin (setup time, hold time, deselection time) is valid.

Rounding Operation of Chip Select Pin (Only Valid in Master Mode (SCR:MS=0))

(1) With the serial chip select output enabled (SCSCR:CSOE="1") and transmission enabled (SCR:TXE="1"), the serial chip select pin specified by the serial chip select start bit (SCSCR:SST0) becomes active first when transmission data is written.

(2) When the serial chip select does not maintain its active state (SCSCR:SCAM=0), the serial chip select pin becomes inactive after the data transmission/receptions are completed as many as the number of times set with TBYTE. Then, the serial chip select pin becomes active, which has the number added 1 to the number of the serial chip select pin that became active previously.
However, if the serial chip select pin to become active next is disabled (SCSCR;CSENn=0), the serial chip select pin will not become active and be skipped.

(3) If the active serial chip select pin number and the one specified by the serial chip select end bit (SCSCR:SED0) are matched, the serial chip select pin to become active next is the one specified by the serial chip select start bit (SCSCR:SST0).

See "Operation of Serial Chip Select to Maintain Active (SCSCR:SCAM=1) (Only Valid in Master Mode (SCR:MS=0))" for the operation of the serial chip select pin when the serial chip select maintains its active state (SCSCR:SCAM=0).
Figure 32-43 shows a timing chart when the start pin of serial chip select pin is SCS0 (SST0=0) and the end pin is SCS1 (SED0=1).

Figure 32-43. Round Operation of Chip Select (SST0=0, SED0=1, CSEN1=1, CSEN0=1, SCAM=0)

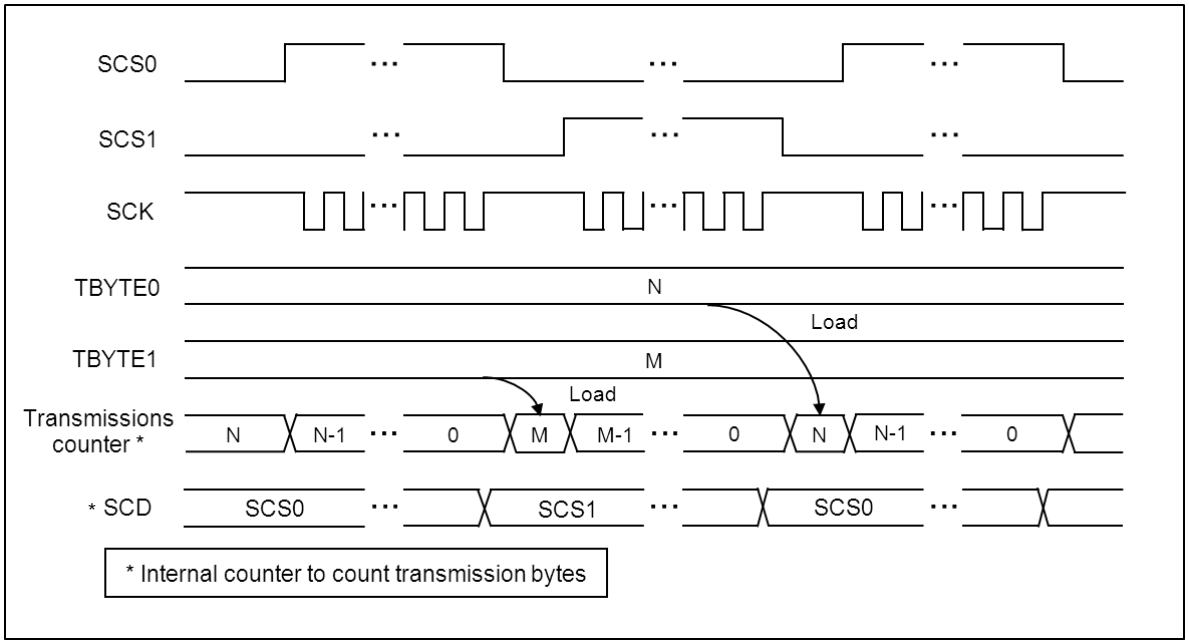
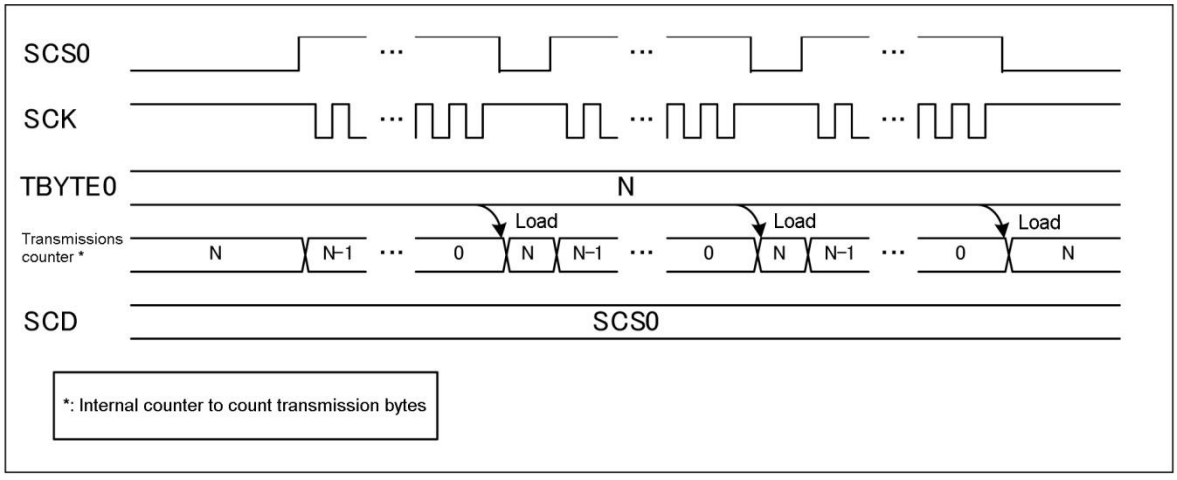


Figure 32-44 shows a timing chart when the start pin of serial chip select pin is SCS0 (SST0=0), the end pin is SCS1 (SED0=1), and the chip select pin 1 is disabled (CSEN1="0"b). After the serial chip select pin 0 becomes active, the pin 1 is skipped, and the pin 0 becomes active.

Figure 32-44. Round Operation of Chip Select (SST0=0, SED0=1, CSEN1=0, CSEN0=1, SCAM=0)



Notes:

- When transmission operation disabled (SCR:TXE=0) is changed to enabled (SCR:TXE=1), the serial chip select pin specified by the serial chip select start bit (SCSCR:SST0) becomes active first.
- When the serial chip select maintains its active level (SCSCR:SCAM=1), it does not transit to the next serial chip pin even if transmission data is written to the transmission bus idle (SCR:TBI=1).
- At the round operation, timing adjustment of the serial chip select pin (setup time, hold time, deselection time) is valid.
- After a software reset (SCR:UPCL=1), the serial chip select pin specified by the serial chip select start bit (SCSCR:SST0) becomes active first.

Operation of Serial Chip Select to Maintain Active (SCSCR:SCAM=1) (Only Valid in Master Mode (SCR:MS=0))

When the serial chip select active maintaining bit (SCSCR:SCAM) is set to "1" and transmission operation is started, the serial chip select pin is maintained to be active.
The value of the serial chip select active maintaining bit is checked for each transmission as many as the number of times set with TBYTE. After the data transmission/receptions are completed as many as the number of times set with TBYTE, the pin operates as shown below.

- If the serial chip select active maintaining bit is "0", the serial chip select pin turns inactive after the hold delay time has passed.
- If the serial chip select active maintaining bit is "1" upon serial timer synchronous transmission, the serial chip select pin is maintained to be active. Then, when the serial timer value (STMR) and the serial timer comparison value (STMCR) are matched, transmission operation is restarted. After that, the serial chip select pin is maintained to be active until the frame transmission is completed as many as the number of times set with TBYTE.

After that, the serial chip select pin is maintained to be active until the frame transmission is completed as many as the number of times set with TBYTE.

- If the serial chip select active maintaining bit is "1" and the serial timer synchronous transmission, the serial chip select pin is maintained to be active. At that time, if the transmit data register (TDR) contains the transmission data (SSR:TDRE=0), the transmission operation is continued, and the serial chip select pin is maintained to be active until the next time the frames as many as the number of times set with TBYTE are sent.

If the serial chip select active maintaining bit (SCSCR:SCAM) is written to "0", it operates as shown below.

- The serial chip select pin becomes inactive after the data transmission/receptions are completed as many as the number of times set with TBYTE and the hold delay time has passed.

Under the following conditions, the serial chip select pin becomes inactive when the serial chip select active maintaining bit (SCSCR:SCAM) is used.

- When SCSCR:SCAM=0 after as many transmissions as the TBYTE count are made
- When the chip select error occurred (SACSR:CSE=1)
- When transmission is disabled (SCR:TXE=0)
- When software reset is performed (SCR:UPCL=1)

Note:

If the transmit data register (TDR) is empty (SACSR:TDRE=1) when the transfer byte error is enabled (SACSR:TBEEN=1) and the data transmission/reception is not completed as many as the number of times set with TBYTE, the serial chip select pin is not retained and becomes inactive after the hold delay time has passed, and the chip select error (SACSR:CSE=1) occurs.

Format Setting of Serial Chip Select Pin

Active level of the chip select for each serial chip select pin, mark level of the serial clock, enabling/disabling SPI mode, and data direction/length of serial data output can be set by using bit shown in Table 32-13.

Table 32-13. Format Setting of Serial Chip Select Pin

Condition		Active level of chip select	Serial clock inversion	SPI setting	Data direction	Data length
Chip select format enabled (SCR:CSFE="1") and master mode (SCR:MS="0")	Serial chip select pin 0 output	SCSCR0: CSLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
	Serial chip select pin 1 output	SCSFR0: CS1CSLVL	SCSFR0: CS1SCINV	SCSFR0: CS1SPI	SCSFR0: CS1BDS	SCSFR0: CS1L3-0
Chip select format disabled (SCR:CSFE="0")		SCSCR0: CSLVL	SMR: SCINV	SCR:SPI	SMR:BDS	ESCR:L3-0
Slave mode (MS="1")						
Chip select unused (CSEN1 to CSEN0="00_B")						

32.6.2.7 Test Mode

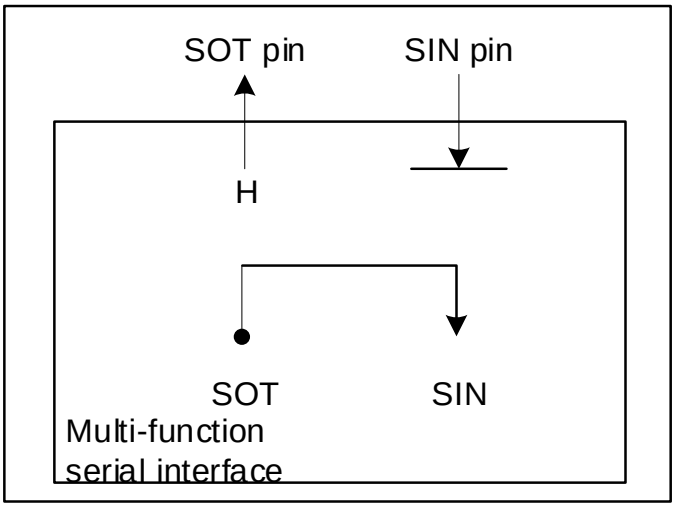
This section explains the test mode.

This section explains the operation of the test mode.

Serial Test Mode

When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and then the data sent from SOT can be received from SIN directly.
When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 32-45. Serial Test Mode



Note:

The serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

32.6.2.8 Baud Rate Generation

This section explains the baud rate generation.

The dedicated baud rate generator works only in master operation. However, if the reception FIFO is to be used, set the dedicated baud rate generator even in slave operation.

The dedicated baud rate generator settings are different between the master and slave operations.

[1] Master operation

- ☐ The dedicated baud rate generator divides the internal clock and a baud rate is selected.
- ☐ There are two internal reload counters that correspond to the transmission and reception serial clocks, respectively. The baud rate can be selected by setting a 15-bit reload value in the baud rate generator register (BGR).
- ☐ The reload counter divides the internal clock with the setting value.

[2] Slave operation

The dedicated baud rate generator does not work in slave operation (SCR:MS=1). (The external clock entered from the clock input pin SCK is used without change.)

Note:

If the reception FIFO is to be used, set the dedicated baud rate generator even in slave operation.

Baud Rate Calculation

Set two 15-bit reload counters in the baud rate generator register (BGR). The baud rate calculation formulas are as follows:

(1) Reload value

V = Φ : b – 1
V: Reload value Φ: bus clock frequency b: Baud rate

(2) Example of calculation

Reload values when setting the bus clock frequency at 16 MHz, usage of internal clock, and baud rate at 19200 bps are as follows:

Reload value:

V = (16 × 1,000,000) / 19200 – 1 = 832

Therefore, the baud rate is

b = (16 × 1,000,000) / (832+1) = 19208 bps

(3) Baud rate error

The baud rate error can be obtained using the following formula:

Error (%) = (calculated value – desired value) / desired value × 100

(Example) When you set bus clock at 20 MHz and target baud rate at 153600 bps:

Reload value = (20 × 1,000,000) / 153600 – 1 = 129

Baud rate (calculated) = (20 × 1,000,000) / (129 + 1) = 153846 bps

Error (%) = (153846 × 153600) / 153600 × 100 = 0.16(%)

Notes:

- Set the reload value to "0" to stop the reload counter.
- If the reload value is an even number, the "H" and "L" widths of the serial clock depend on the SCINV bit setting as follows: If it is an odd number, the "H" and "L" widths of the serial clock are equal.
  If SMR:SCINV="0", the "H" width of the serial clock is longer by one cycle of the bus clock.
  If SMR:SCINV="1", the "L" width of the serial clock is longer by one cycle of the bus clock.
- Set the reload value to 3 or higher.

Reload values and Baud Rate for Bus Clock Frequencies

Frequencies Example of reload values and Baud Rate are shown.

Table 32-14. Reload values and Baud Rate for Bus Clock Frequencies example

Baud Rate (bps)	8MHz		10MHz		16MHz		20MHz		24MHz		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	87	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: Setup values in the BGR register
- ERR: Baud Rate Error (%)

Reload Counter Functions

Reload counters, including transmission and reception reload counters, serve as the dedicated baud rate generators. It consists of a 15-bit register for reload values and generates a transmission/reception clock from the internal clock.

Count Start

When a reload value is written to the baud rate generator register (BGR), the reload counter starts counting.

Restart

The reload counter restarts under one of the following conditions:

Common to the transmission and reception reload counters

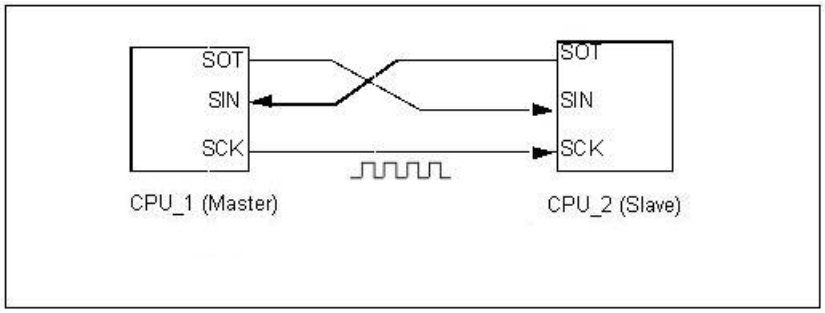
Programmable reset (SCR:UPCL bit)

32.6.3 Setup Procedure and Program Flow

Setup procedure and program flow is shown.

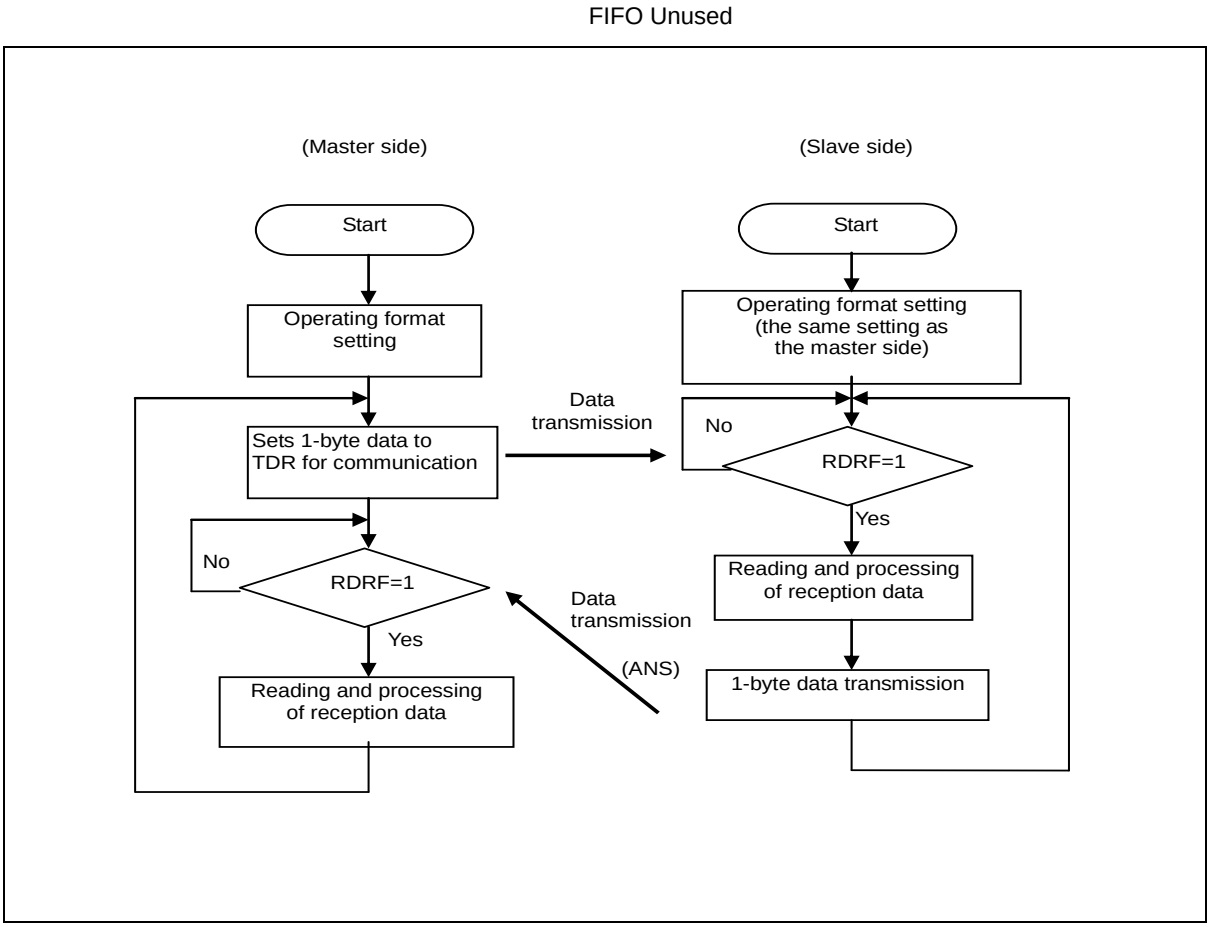
CPU Interconnection

Figure 32-46. Example of Connection between CSIO Chips

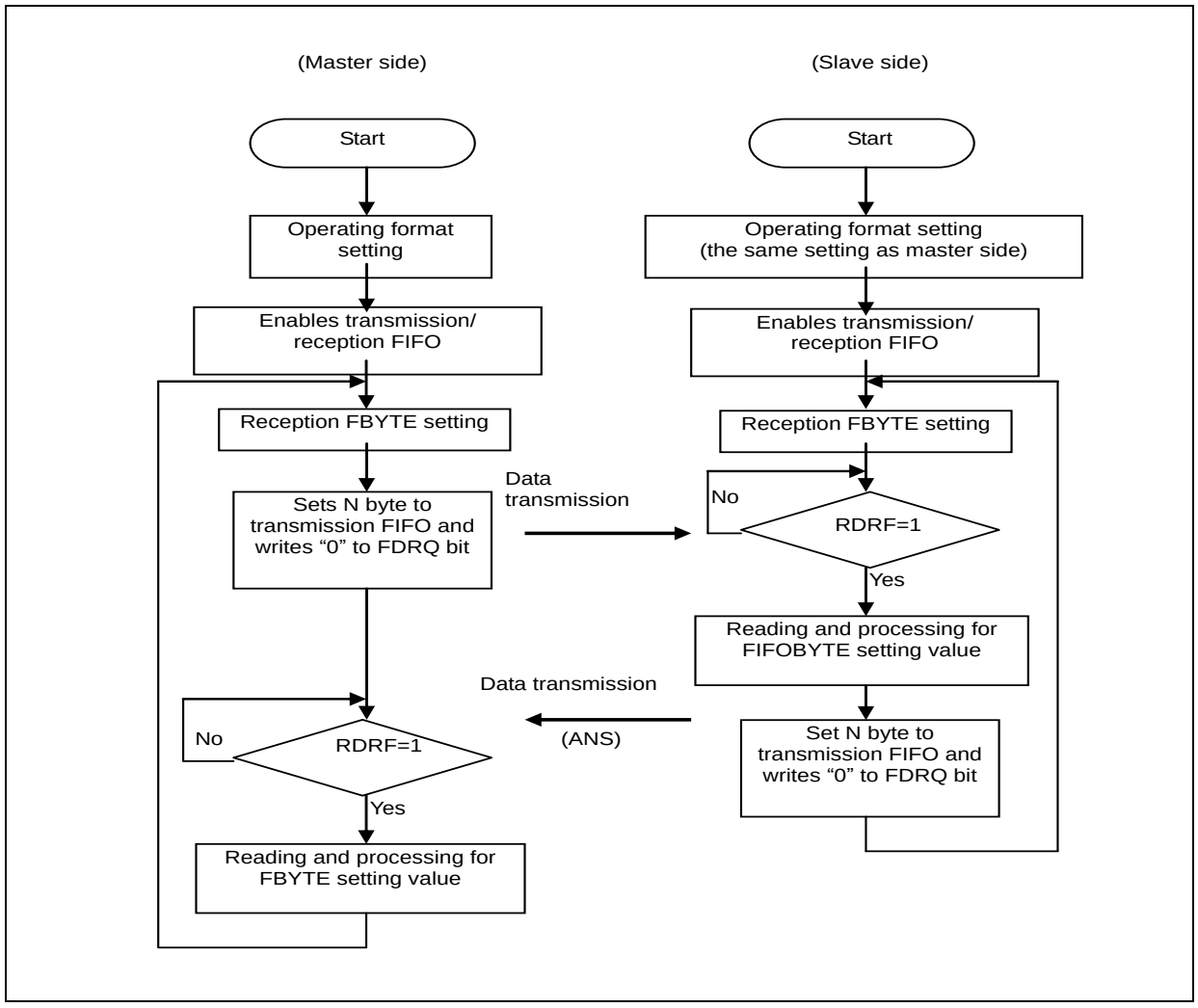


Flowchart

Figure 32-47. Flowchart Example



FIFO Used


(Master side)
(Slave side)
Start
Start
Operating format setting
Operating format setting (the same setting as master side)
Enables transmission/ reception FIFO
Enables transmission/ reception FIFO
Reception FBYTE setting
Reception FBYTE setting
Data transmission
Sets N byte to transmission FIFO and writes “0” to FDRQ bit
No
RDRF=1
Yes
Reading and processing for FIFOBYTE setting value
Data transmission
(ANS)
No
RDRF=1
Set N byte to transmission FIFO and writes “0” to FDRQ bit
Yes
Reading and processing for FBYTE setting value

32.7 Operation of LIN Interface (v2.1)

This section explains operation of the LIN Interface (v2.1).

The LIN communication function has two modes; manual mode for LIN header transmission/reception using the interrupt functions, and assist mode for automatic LIN header transmission/reception. Assist mode supports the master node of LIN protocol Revision1.X and LIN protocol Revision2.X, and the slave node of LIN protocol Revision1.X. To support the slave node of LIN protocol Revision2.X, use manual mode.

32.7.1 Interrupt of LIN Interface (v2.1) manual mode

Interrupt of LIN Interface (v2.1) manual mode is shown.

The LIN interface (v2.1) consists of reception interrupt and transmission interrupt. This interface can generate interrupt requests for the following factors:

- Setting of reception data in the receive data register (RDR) or occurrence of a reception error
- Start of transmission after transfer of transmission data from the transmit data register (TDR) to the transmit shift register
- Transmission bus idle (no transmission operation)
- Transmission FIFO data request
- LIN Break field detection

32.7.1.1 List of Interrupts of LIN Interface (v2.1) (manual mode)

This section explains the list of interrupts of the LIN interface (v2.1) (manual mode).

Table 32-15 shows the relationship between the LIN interrupt control bits and the interrupt factors in the manual mode.

Table 32-15. Interrupt Control Bits and Interrupt Factors for LIN Interface (v2.1) (Manual Mode)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	Clearing of interrupt request
Recep-tion	RDRF	SSR	1-byte reception	SCR:RIE	Reading of receive data (RDR)
			Reception of as much data as specified by FBYTE		Reading of receive data (RDR) until the reception FIFO is emptied
			Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	Overrun error		Writing of "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	Framing error		
Trans-mission	TDRE	SSR	Transmission register is empty	SCR:TIE	Writing the transmit data (TDR) or writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	TBI	SSR	No transmission operation	SCR:TBIE	Write to the transmit data (TDR), write "1" to the Lin break field setting bit(LBR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has valid data (retransmission)*
	FDRQ	FCR1	The storage data value of the transmission FIFO is FTICR setting value or less, or empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO is full
Status	LBD	SSR	Lin break field detection	ESCR:LBIE	Writing "0" to the SSR:LBD
	SFD	SACSR	Sync Field is detected	SACSR: SFDE	Writing "0" to Sync Field detection flag (SACSR:SFD)
	TINT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to timer interrupt flag bit (SACSR:TINT)
Input capture	ICP	ICS	1st falling edge of Lin Synch Field	ICS:ICE0	Disabling of ICP
	ICP	ICS	5th falling edge of Lin Synch Field		

* Wait for TDRE bit with "0" before setting "1" to TIE bit.

32.7.1.2 Reception Interrupts and Flag Setting Timing

This section explains the generation of reception interrupts and the flag setting timing.

Reception interrupts occur when the reception is completed (SSR:RDRF), when a reception error occurs (SSR:ORE, FRE), or when LIN break Field is detected.

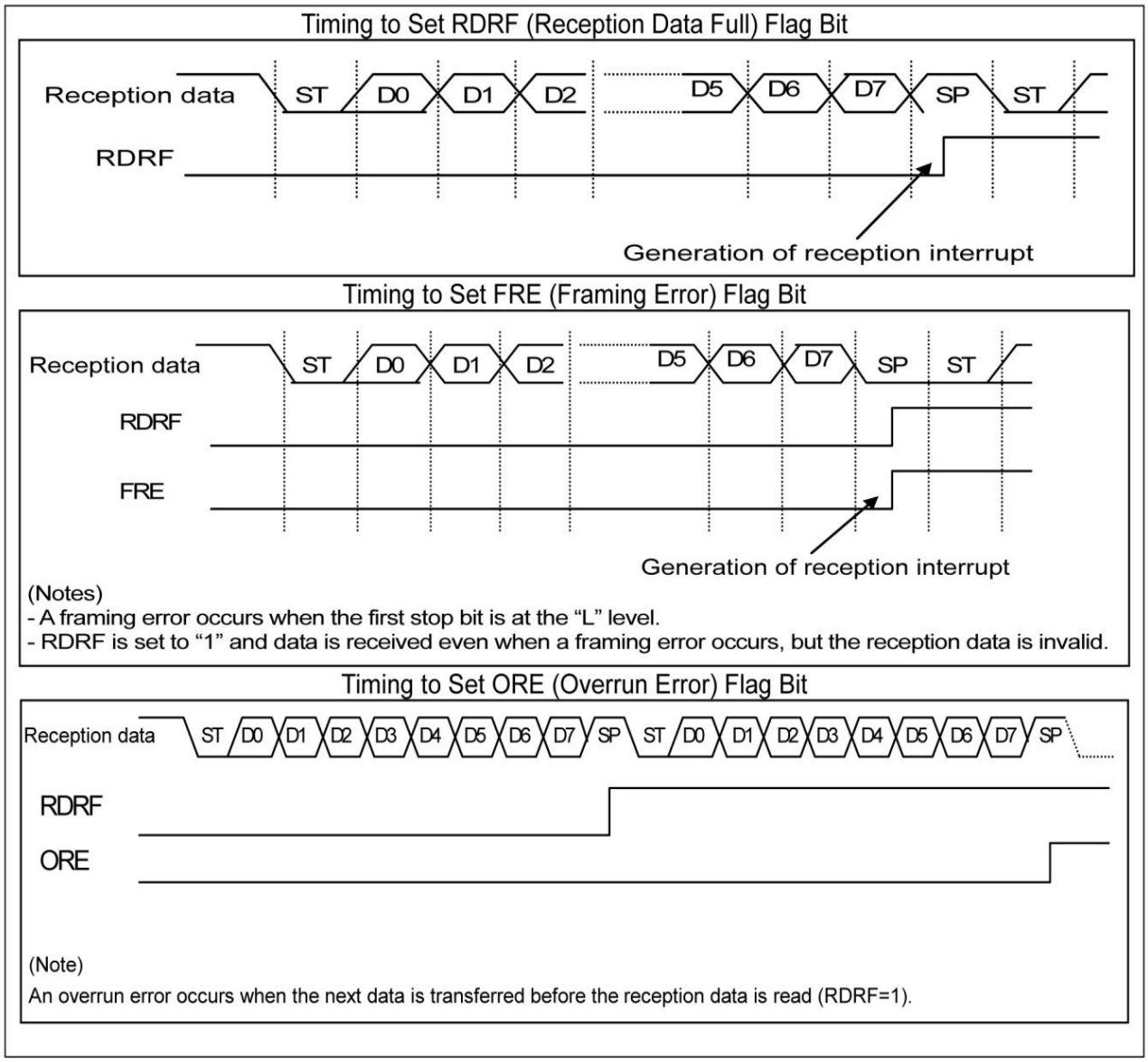
Reception Interrupts and Flag Setting Timing

When the first stop bit is detected, reception data is stored in the Receive data register (RDR). When reception is completed (SSR:RDRF=1) or a reception error occurs (SSR:ORE, FRE=1), a corresponding flag is set. If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt will occur.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

Figure 32-48. Timing of each Flag Bit Setting



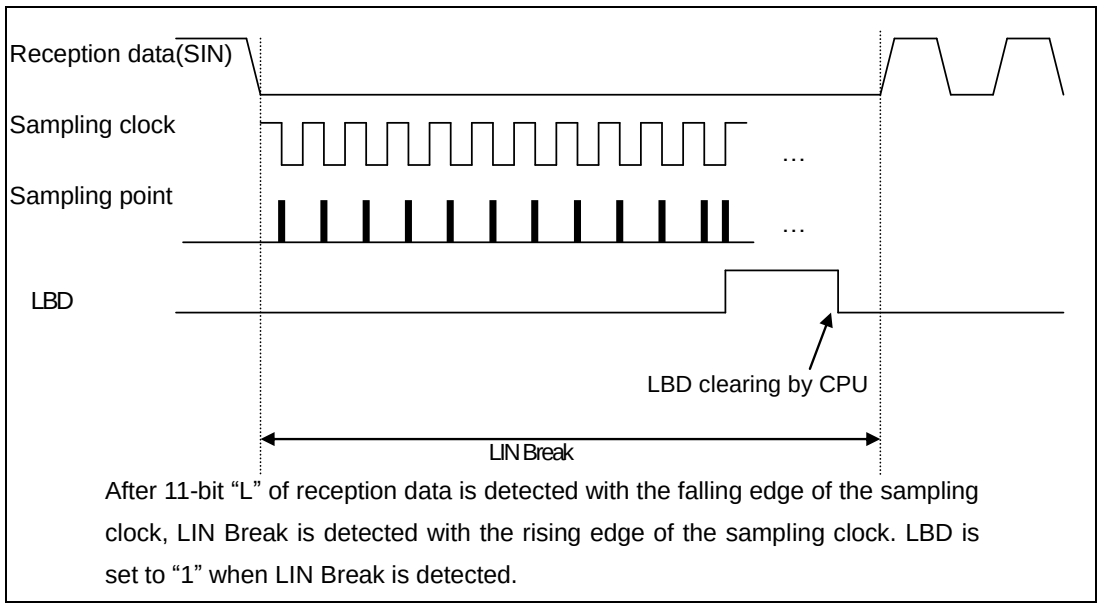
Note:

If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the data may not be received with the edge disabled. Leave a space between frames if successive frames are to be output.

Timing of LIN break Field Detection Flag (LBD) Setting

LBD bit is set to "1" when the serial input (SIN) is "0" for more than 11-bit width. In this case when LIN break Field interrupt is enabled (ESCR:LBIE=1), a reception interrupt occurs.

Figure 32-49. Timing of LIN Break Field Flag (LBD) Setting



32.7.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing

This section explains the generation of interrupts when using reception FIFO and the flag setting timing.

When the reception FIFO is used, an interrupt occurs after as much data as the FBYTE register (FBYTE) setting is received.

Reception Interrupts when Using Reception FIFO and Flag Setting Timing

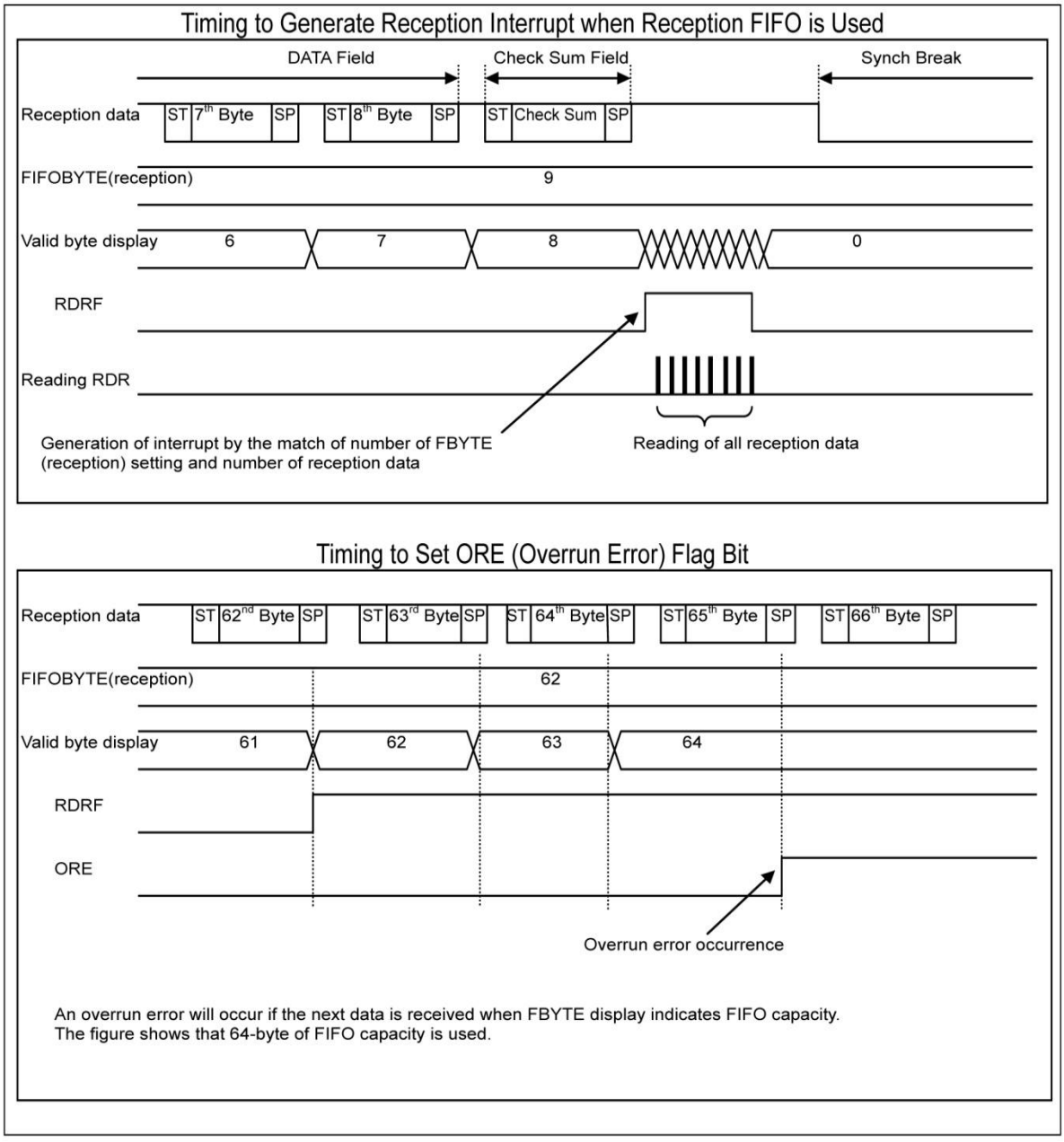
The setting value of the FBYTE register determines the occurrence of an interrupt when the reception FIFO is used.

- After as much data as the transfer count setting of the FBYTE register is received, the reception data full flag of the serial status register (SSR:RDRF) is set to "1". If the reception interrupt is enabled (SCR:RIE) at this time, a reception interrupt will be generated.
- In the case where all the conditions below are met, when reception idle continues for more than 8 baud rate clocks, interrupt flag (SSR:RDRF) will be set to "1".
  - ☐ Reception FIFO idle detection enable bit (FCR:FRIIE) is "1"
  - ☐ Data count contained in the reception FIFO does not reach the transfer count

If you read the RDR while the counter is counting 8 baud rate clocks, the counter will be reset to 0 and start counting 8 clocks again. When reception FIFO is disabled, the counter will be reset to "0". When the reception FIFO is enabled while any data is left in the reception FIFO, counting will be started once again.

- If the receive data (RDR) is read until the reception FIFO is empty, the reception data full flag (SSR:RDRF) will be cleared.
- When the reception-enabled data count indication has shown the FIFO capacity, receiving the next data will generate an overrun error (SSR:ORE=1).

Figure 32-50. Timing of Interrupt Generation



32.7.1.4 Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and the flag setting timing.

Transmission interrupts occur either when transmission is started after transfer of transmission data from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) or when the transmission operation is idle (SSR:TBI=1).

Transmission Interrupts and Flag Setting Timing

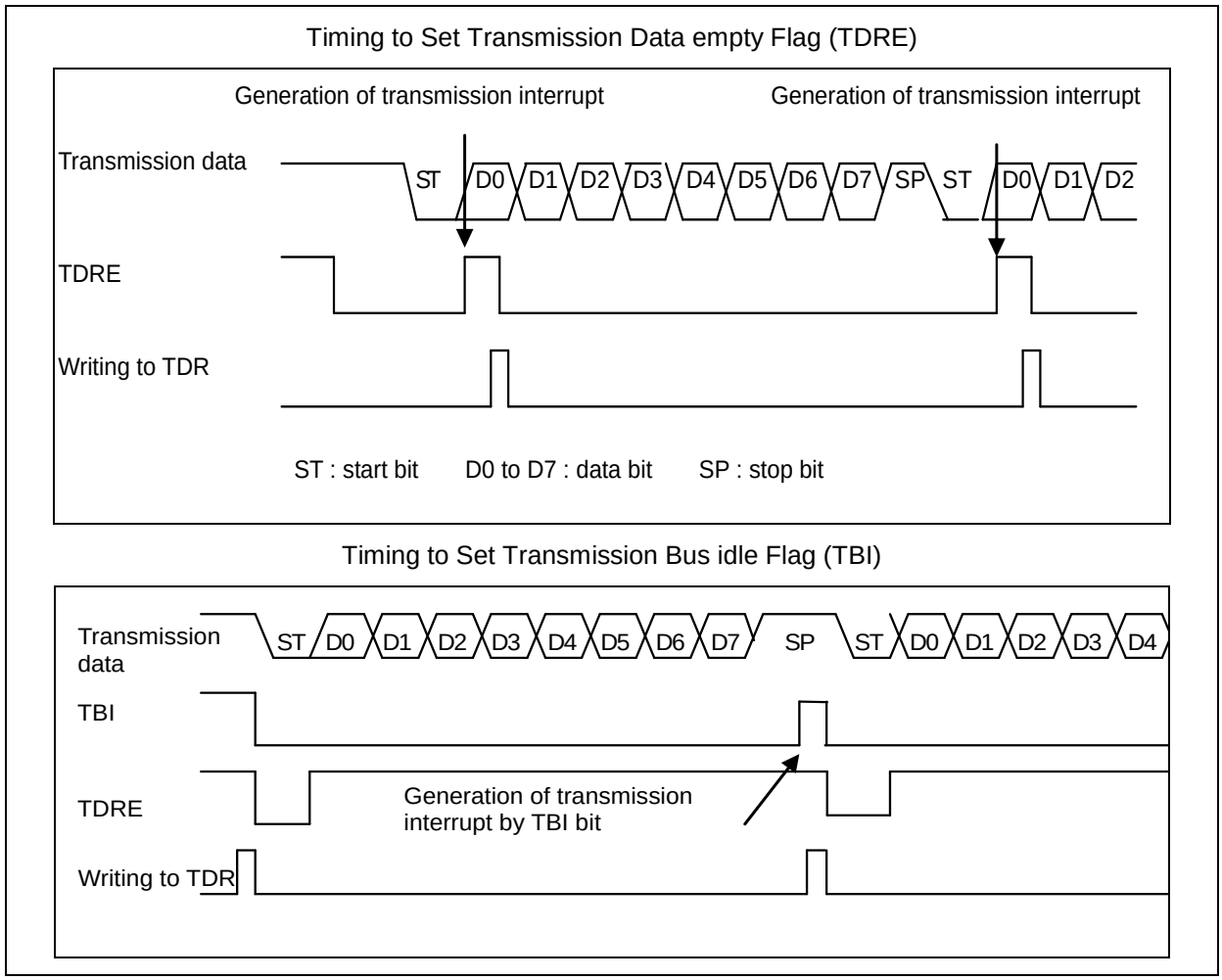
Timing of transmission data empty flag (TDRE) setting

When data written to the transmit data register (TDR) is transferred to the transmit shift register, writing of next data is enabled (SSR:TDRE=1). If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs. The TDRE bit, being a read-only bit, is cleared to "0" by writing of data to the transmit data register (TDR).

Timing of transmission bus idle flag (TBI) setting

When the transmit data register is empty (TDRE=1) and no transmission operation is in progress, the SSR:TBI bit is set to "1". If transmission bus idle interrupt is enabled (SCR:TBIE=1) at this time, a transmission interrupt occurs. When transmission data is written to the transmit data register (TDR), the TBI bit and the transmission interrupt request will be cleared.

Figure 32-51. Timing of TDRE and TBI Setting



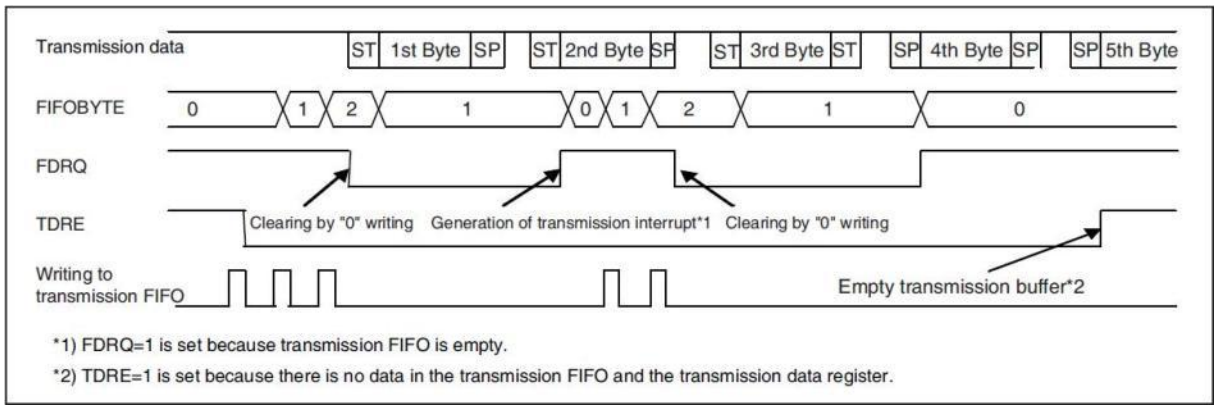
32.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

This section explains the generation of interrupts when using transmission FIFO and the flag setting timing.

When the transmission FIFO is used, an interrupt will be generated when the number of data items stored in the transmission FIFO is FTICR register (FTICR) setting value or less.

- When the number of data items stored in the transmission FIFO is FTICR register (FTICR) setting value or less. the FIFO transmission data request bit (FCR1:FDRQ) will be set to "1". If FIFO transmission interrupt is enabled (FCR1:FTIE=1) at this time, a transmission interrupt occurs.
- When required data is written to the transmission FIFO after the occurrence of a transmission interrupt, write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request.
- When the transmission FIFO is full, the FIFO transmission data request bit (FCR1:FDRQ) is set to "0".
- The presence of data in the transmission FIFO can be checked by reading the FIFO byte register (FBYTE) or the transmission FIFO interrupt control register (FTICR).
  When FBYTE=0x00, there is no data in the transmission FIFO.

Figure 32-52. Timing of Transmission Interrupts when Using Transmission FIFO



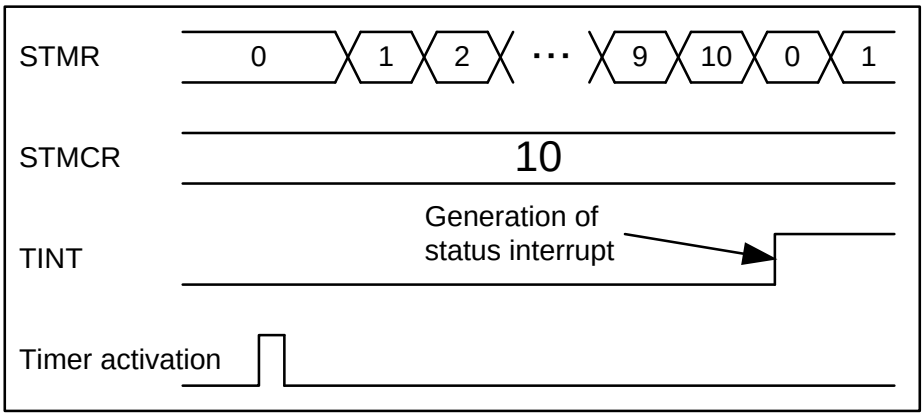
32.7.1.6 Timer Interrupt and Flag Setting Timing

This section explains the timer interrupt and the flag setting timing.

Timer interrupt occurs when the serial timer register (STMR) and the serial timer comparison register (STMCR) are matched.

- When the serial timer register (STMR) and the serial timer comparison register are matched, the timer interrupt flag (SACSR:TINT) is set to "1".
  If the timer interrupt is enabled (SACSR:TINTE="1") at this time, a status interrupt occurs.

Figure 32-53. Timer Interrupt Timing



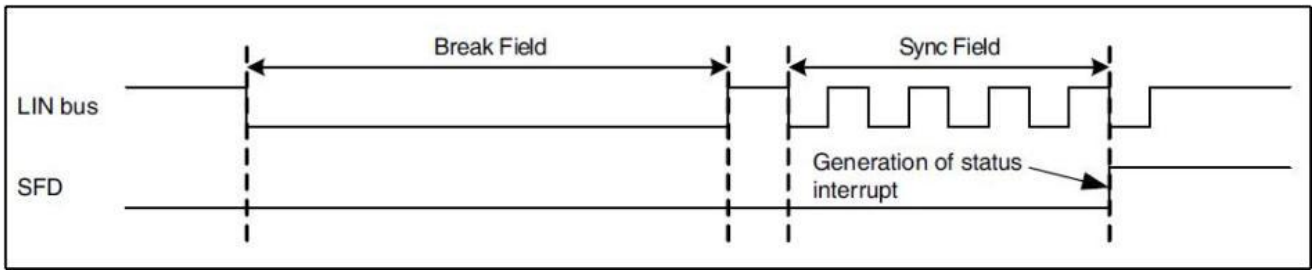
32.7.1.7 Sync Field Detection Interrupts and Flag Setting Timing

This section explains the generation of Sync Field detection interrupts and the flag setting timing.

Sync Field detection interrupt occurs when the detection of Sync Field is completed.

- When the auto baud rate adjustment is enabled (SACSR:AUTE="1") and the fifth falling of LIN bus on Sync Field is detected, the Sync Field detection flag (SACSR:SFD) is set to "1".
  If the Sync Field interrupt is enabled (SACSR:SFDE="1") at this time, a status interrupt occurs.

Figure 32-54. Sync Field Detection Interrupt Timing



32.7.2 Interrupts in LIN Interface (v2.1) Assist Mode

Interrupts in LIN Interface (v2.1) Assist Mode is shown.

The LIN interface (v2.1) consists of reception interrupt, transmission interrupt, and status interrupt.
In assist mode, this interface can generate interrupt requests for the following factors:

- When reception data is stored in the receive data register (RDR) or when a reception error occurs.
- When transmission data is transmitted from the transmit data register (TDR) to the transmit shift register, and the transmission is begun.
- Transmission bus idle (There is no transmission operation)
- Transmission FIFO data request
- LIN Break Field detection
- LIN Sync Field detection
- Both the comparison value (STMCR) of the serial timer and the value (STMR) of the serial timer are equal.
- The LIN automatic header completion or the checksum arithmetic operations completion is detected.

32.7.2.1 List of Interrupts of LIN Interface (v2.1) (assist mode)

This section explains the list of interrupts of LIN interface (v2.1) (assist mode).

Table 32-16 shows the relationship between the LIN interrupt control bits and the interrupt factors in the LIN assist mode.

Table 32-16. Interrupt Control Bits and Interrupt Factors for LIN Interface (v2.1) (Assist Mode)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt factor enable bit	How to clear the interrupt request
Recep-tion	RDRF	SSR	1-byte reception	SCR:RIE	Reading receive data (RDR)
			Reception of as much data as specified by FBYTE		Reading receive data (RDR) until the reception FIFO becomes empty
			Detection of reception idle for 8-bit time or more while there is valid data in the reception FIFO with the FRIIE bit set to "1".		
	ORE	SSR	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	Framing error		
	LBSER	LAMSER	LIN bus error detection	LAMIER: LASERIE	Writing "0" to the LAMESR:LBSER
	LPFER	LAMESR	LIN Sync Data error detection	LAMIER: LSFERIE	Writing "0" to the LAMESR:LSFER
	LPTER	LAMESR	LIN ID parity error detection	LAMIER: LPTERIE	Writing "0" to the LAMESR:LPTER
	LCSER	LAMESR	LIN checksum error detection	LAMIER: LCSERIE	Writing "0" to the LAMESR:LCSER
Trans-mission	TDRE	SSR	Transmission register is empty	SCR:TIE	Writing the transmit data (TDR), writing of "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and there is valid data in the transmission FIFO (retransmission)*
	TBI	SSR	No transmission operation	SCR:TBIE	Writing the transmit data (TDR), writing of "1" to the Lin break field set bit(LBR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has valid data (retransmission)*
	FDRQ	FCR1	The storage data value of the transmission FIFO is FTICR setting value or less, or the value is empty.	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO is full
Status(Assist Mode)	LBD	SSR	Lin break field detection	ESCR:LBIE	Writing "0" to the SSR:LBD
	SFD	SACSR	Sync Field is detected	SACSR: SFDE	Writing "0" to Sync Field detection flag (SACSR:SFD)
	TINIT	SACSR	Serial Timer Register (STMR) matched Serial Timer Comparison Register (STMCR)	SACSR: TINTE	Writing "0" to timer interrupt flag bit (SACSR:TINT)
	LAHC	LAMSR	Automatic header completion	LAMIER: LAHCIE	Writing "0" to the LAMSR:LAHC
	LCSC	LAMSR	Checksum arithmetic operation completion	LAMIER: LCSCIE	Writing "0" to the LAMSR:CRC
Input capture	ICP	ICS	1st falling edge of Lin Synch Field	ICS:ICE0	Disabling of ICP
	ICP	ICS	5th falling edge of Lin Synch Field		

* Set the TIE bit to "1" after the TDRE bit becomes "0".

32.7.2.2 Reception Interrupts and Flag Setting Timing in Assist Mode

This section explains the generation of reception interrupts and the flag setting timing in assist mode.

Reception interrupts occur when the reception is completed (SSR:RDRF) and when a reception error occurs (SSR:ORE, FRE,LAMESR:LBSER,LSFER,LPTER,LCSER).

Reception Interrupts and Flag Setting Timing

In the LIN assist mode (LAMCR:LAMEN=1), data is stored in the receive data register (RDR) when the first stop bit of each of the following fields is detected. When the reception of data is completed (SSR:RDRF=1), the flag is set. If reception interrupt is enabled (SSR:RIE=1), a reception interrupt occurs.

- ID Field when slave mode is set (SCR:MS=1) and the data reception register (RDR) is set to receive the ID Field (LAMCR:LIDEN=0)
- Data Field for response

As for the timing of setting the reception data full flag bit (SSR:RDRF), the timing is similar to those described in "32.7.1.2 Reception Interrupts and Flag Setting Timing" in manual mode. See Figure 32-48.

Notes:

- When the reception error occurs, the reception data stored in the receive data register (RDR) becomes invalid.
- When the LIN assist mode reception ID register is used in order to receive the ID Field (LAMCR:LIDEN=1), the received ID value is not stored in the receive data register (RDR) , and the reception data full flag bit (SSR:RDRF) is not set.
- Neither the Sync Field nor the checksum are stored in the receive data register (RDR), and the reception data full flag bit (SSR:RDRF) is not set.
- Those data transmitted from each field are not stored in the receive data register (RDR), and reception data full flag bit (SSR:RDRF) is not set.

Framing Error Interrupt and Flag Setting Timing

In assist mode (LAMCR:LAMEN=1), the framing error is detected and a flag of the framing error is set (SSR:FRE=1) when "L" level is detected in the stop bit of the Sync Field, the ID Field, the Data Field, and the Check Sum Field, respectively. If the reception interrupt is enabled (SSR:RIE=1), the reception interrupt will occur.
Moreover, when the framing error is detected, transmission/reception of both the header and the response is stopped in the assist mode.
While the framing error flag is being set (SSR:FRE=1), the operation enable bit of the reception FIFO is cleared (FCR0:FE1=0 or FCR0:FE2=0).
As for the timing of setting the framing error flag bit (SSR:FRE), the timing is similar to those described in "32.7.1.2 Reception Interrupts and Flag Setting Timing" in manual mode. See Figure 32-48.

Overrun Error Interrupt and Flag Set Timing

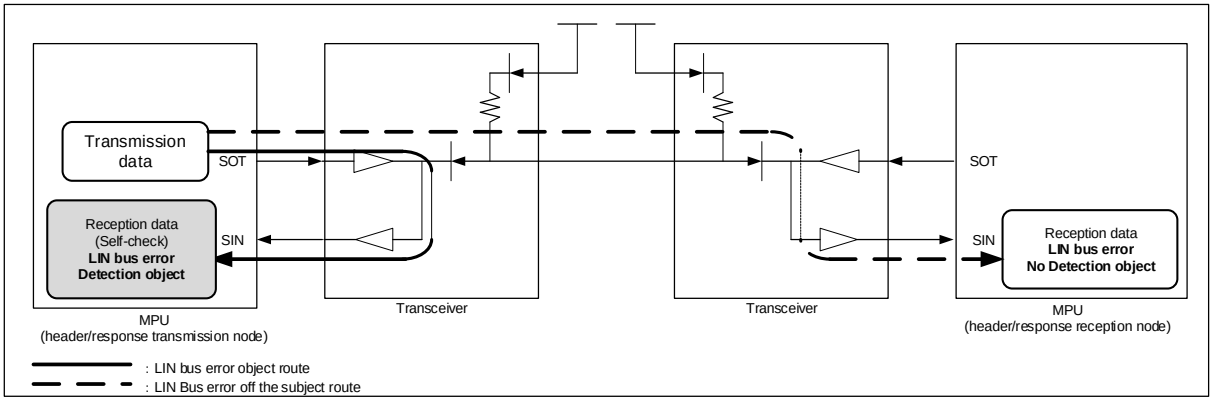
When reception of data is detected before the previously received data is read (RDRF=1), the overrun error is detected. After the reception of the next data is completed (SSR:RDRF=1), the overrun error flag is set (SSR:ORE=1). If the reception interrupt is enabled (SSR:RIE=1), the reception interrupt occurs.
Moreover, when the overrun error is detected, transmission/reception of both the header and the response is stopped in the assist mode.

While the overrun error flag is being set (SSR:ORE=1), the operation enable bit of the reception FIFO is cleared (FCR0:FE1=0 or FCR0:FE2=0).
As for the timing of setting the overrun error flag bit (SSR:ORE), the timing is similar to those described in "32.7.1.2 Reception Interrupts and Flag Setting Timing" in manual mode. See Figure 32-48.

LIN Bus Error Detection Interrupt and Flag Setting Timing

The LIN bus error is detected by the self-check done on the side where a header/response is transmitted in assist mode (LAMCR:LAMEN=1). The LIN bus error cannot be detected on the side where header/response is received.
Figure 32-55 shows the LIN bus error detection object.

Figure 32-55. LIN Bus Error Detection Object



The range of detection of the LIN bus error is a start bit and byte data of LIN Break and Sync Field/ID Field/Data Field/Check Sum Field. The stop bit is outside the detection range of the LIN bus error. When the stop bit is detected as "L" level, the framing error is detected (SSR:FRE=1).
Moreover, when the LIN bus error is detected, transmission of the header and the response is stopped in the assist mode. Even if the LIN bus error occurs when the ID Field transmission is completed, LIN automatic header completion flag (LAMSR:LAHC=1) is set.

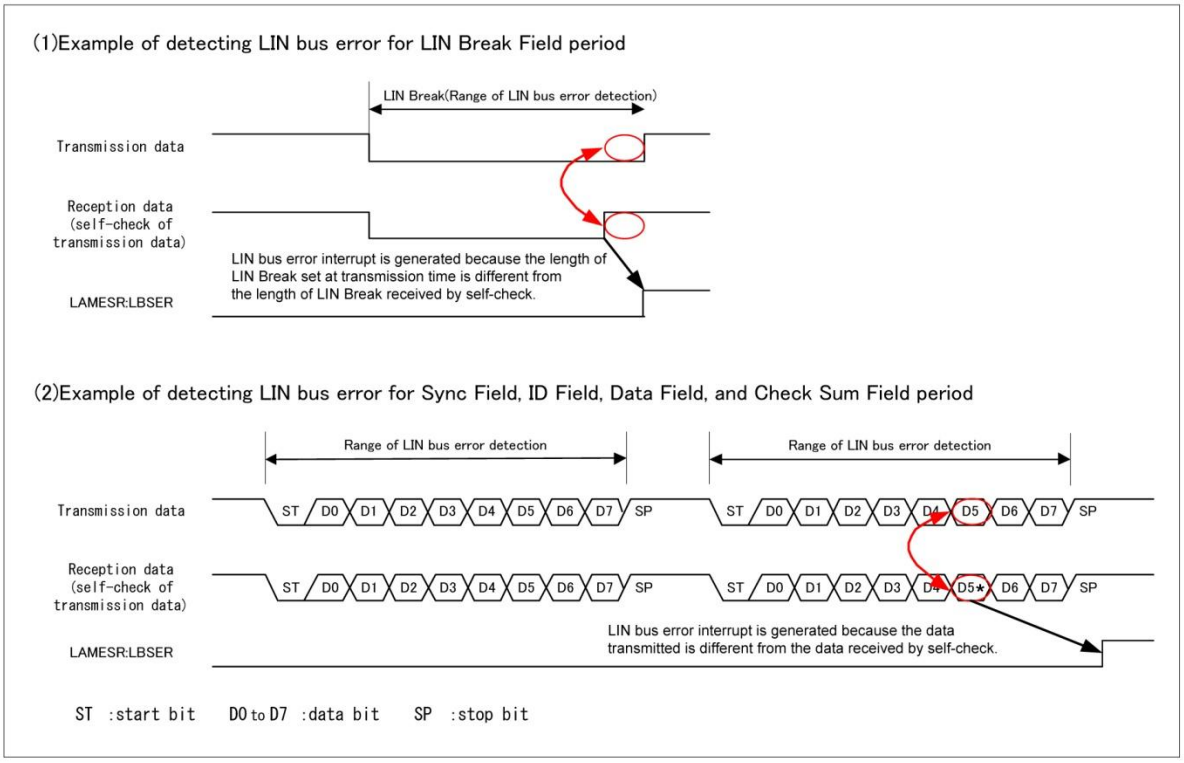
LIN bus error detection interrupt and flag setting timing on master side

On master side (SCR:MS=0), the LIN bus error is detected when a header/response is transmitted.
The LIN bus error is detected and the flag is set (LAMESR:LBSER=1) if an error is detected as a result of the comparison between transmission LIN Break length and reception LIN Break length or between the transmission data and the reception data. When this interrupt is set to be enabled (LAMIER:LBSERIE=1), the reception interrupt will occur.

LIN bus error detection interrupt and flag setting timing on slave side

On slave side (SCR:MS=1), the LIN bus error is detected when the response is transmitted.
The LIN bus error is detected and the flag is set (LAMESR:LBSER=1) if an error is detected as a result of the comparison between the transmission data and the reception data. When the interrupt is set to be enabled (LAMIER:LBSERIE=1), the reception interrupt will occur.

Figure 32-56. Setting Timing of LIN Bus Error Detection Flag (LAMESR:LBSER)



LIN Sync Data Error Detection Interrupt and Flag Setting Timing

The LIN Sync Data error is detected when automatic baud rate adjustment is disabled (SACSR:AUTE=0), in slave node (SCR:MS=1) which is set to assist mode (LAMCR:LAMEN=1).

The range of detection of the LIN Sync Data error is a start bit and byte data of the Sync Field. The stop bit is off the target of the LIN Sync Data error detection. When the stop bit is detected as "L" level, the framing error is detected (SSR:FRE=1).

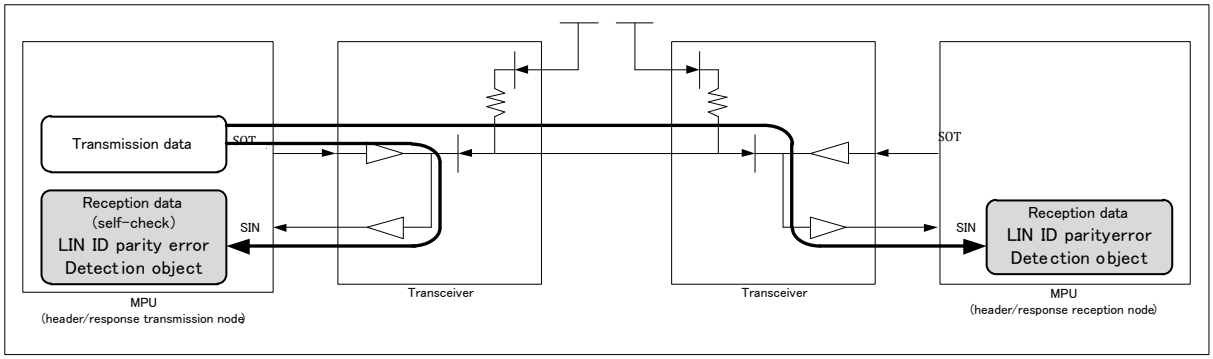
LIN Sync Data error detection interrupt and flag setting timing when automatic baud rate self adjustment is prohibited

On slave node (SCR:MS=1) which is set to automatic baud rate adjustment disabled (SACSR:AUTE=0), the LIN Sync Data error is detected and the flag is set (LAMESR:LSFER=1) if the value other than 0x55 is detected as a result of checking the data value of Sync Field. If this interrupt is set to be enabled (LAMIER:LSFERIE=1) at that time, the reception interrupt will occur.

LIN ID Parity Error Detection Flag Interrupt and Flag Setting Timing

The LIN ID parity error detection in the assist mode (LAMCR:LAMEN=1) is done by the master and slave. Here, the master transmits the ID Field and does a self-check, and the slave receives ID Field.
Figure 32-57 shows the LIN ID parity error detection object.

Figure 32-57. LIN ID Parity Error Detection Object



The range of detection of the LIN ID parity error is ID data and byte data of a parity. The start bit and the stop bit are outside the LIN ID parity error detection ranges. When "L" level is detected on the stop bit, the error is the framing error (SSR:FRE=1).

In the LIN assist mode (LAMCR:LAMEN=1), when the LIN ID parity error occurs during the transmission of an automatic header, the automatic header transmission completion flag is set (LAMSR:LAHC=1).
When the LIN ID parity error occurs while receiving an automatic header, the automatic header (reception) completion flag is set (LAMSR:LAHC=1) as well.

When the LIN ID parity error is detected, transmission/reception of the response stops in the assist mode.
While the LIN ID parity error flag is set (LAMESR:LPTER=1), the operation enable bit of the reception FIFO is cleared (FCR0:FE1 or FCR0:FE2=0).

Note:

Even though the framing error is detected in the ID Field, the result of ID parity arithmetic operation is indicated. The result at this time, however, is not guaranteed.

LIN ID parity error detection interrupt and flag setting timing on master side

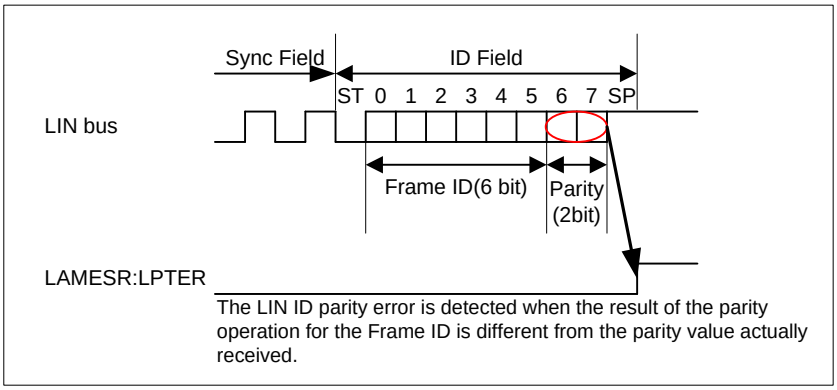
On master side (SCR:MS=0) set in the assist mode (LAMCR:LAMEN=1), detection of the LIN ID parity error is performed when the ID Field is transmitted. The master does the parity arithmetic operation for the six-bit Frame ID which is set by the transmit data register (TDR) or the LIN assist mode transmission ID register (LAMTID). Then the ID Field automatically generated is transmitted.

When the master receives the ID Field by self-check, and if there is a difference of parity between the result of the arithmetic operation for Frame ID and received value, the LIN ID parity error is detected and the flag is set (LAMESR:LPTER=1).
At that time, the interrupt occurs if the interrupt is set to be enabled (LAMIER:LPTERIE=1).

LIN ID parity error detection interrupt and flag setting timing on slave side

On slave side (SCR:MS=1) set in the assist mode (LAMCR:LAMEN=1), detection of the LIN ID parity error is performed when the ID Field is received. If the result of parity arithmetic operation for the Frame ID value in the received ID Field is different from the received parity value, the LIN ID parity error is detected and the flag is set (LAMESR:LPTER=1).
At that time, the reception interrupt occurs if the interrupt is set to be enabled (LAMIER:LPTERIE=1).

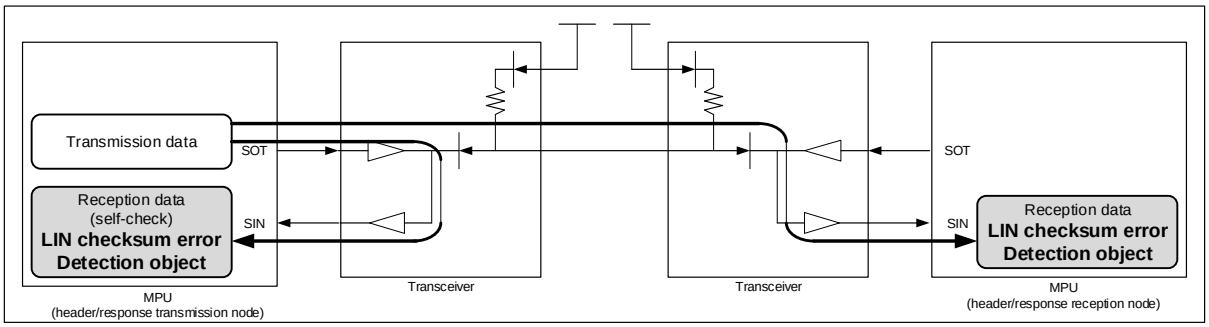
Figure 32-58. Setting Timing of LIN ID Parity Error Detection Flag (LAMESR:LPTER)



LIN Checksum Error Detection Flag Interrupt and Flag setting Timing

The LIN checksum error detection is done on both the side where the checksum is self-checked and transmitted in the assist mode (LAMCR:LAMEN=1) and the side where the checksum is received
Figure 32-59 shows the LIN checksum error detection object.

Figure 32-59. LIN Checksum Error Detection Object



The method of arithmetically operating the checksum transmitted automatically can be selected from standard (The object: data)/extended (The object: ID Field + data).
Even if the LIN checksum error occurs when the checksum transmission is completed, the transmission of data is not stopped.

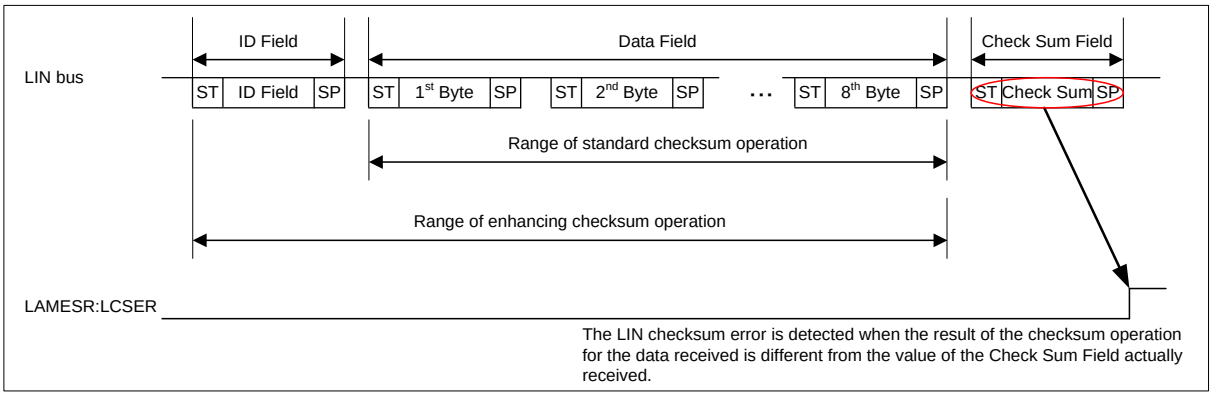
LIN checksum error detection interrupt and flag setting timing on the setting for calculating standard checksum

When the standard checksum arithmetic operation is set (LAMCR:LCSTYP=0), on the side who sends the response (data and checksum), the checksum of the transmission data for set LIN data length (LAMCR:LDL) is calculated and transmitted automatically after final data is transmitted.
On another side where the response (data and checksum) is received, the checksum of the received data for set LIN data length (LAMCR:LDL) is calculated. If the received checksum and the calculated value are different from each other, the LIN checksum error is detected and the flag is set (LAMESR;LCSER=1).
At that time, the reception interrupt occurs if the interrupt is set to be enabled (LAMIER:LCSCIE=1).

LIN checksum error detection interrupt and flag setting timing on the setting for calculating extended checksum

When the extended checksum arithmetic operation is set (LAMCR:LCSTYP=1), on the side who sends the response (data and checksum), the checksum of the ID Field and transmitted data for set LIN data length (LAMCR:LDL) is calculated and transmitted automatically after final data is transmitted.
On another side where the response (data and checksum) is received, the checksum of the received data for set LIN data length (LAMCR:LDL) is calculated. If the received checksum and the calculated value are different from each other, the LIN checksum error is detected and the flag is set (LAMESR:LCSER=1).
At that time, the reception interrupt occurs if the reception interrupt is set to be enabled (LAMIER:LCSCIE=1).

Figure 32-60. Setting Timing of LIN Checksum Error Detection Flag (LAMESR:LCSER)



Note:

When the error (LIN bus error, LIN ID error, LIN Sync Data error, and framing error) is detected in the data of the header and the response part, processing in the assist mode stops and the checksum operation is not executed, regardless of standard/extended checksum arithmetic operation.

32.7.2.3 Reception Interrupts and Flag Setting Timing when using Reception FIFO

This section explains the generation of interrupts and the flag setting timing when using reception FIFO.

The description is similar to those in "32.7.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing" in the manual mode.

32.7.2.4 Transmission Interrupts and Flag Setting Timing

This section explains the generation of transmission interrupts and flag setting timing.

A transmission interrupt occurs when the transmission data is transmitted from the transmit data register (TDR) to the transmit shift register (SSR:TDRE=1) and then the transmission is started, and when the transmission operation is not done.

Transmission Interrupts and Flag Setting Timing

Setting timing of transmission data empty flag (TDRE)

The timing is similar to those described in "32.7.1.4 Transmission Interrupts and Flag Setting Timing" in the manual mode.

Setting timing of transmission bus idle flag (TBI)

When either of the following transmission operation is not done, the transmission bus idle flag bit (SSR:TBI) is set to "1". At this time, if transmission bus idle interrupt is enabled (SCR:TBIE=1), the transmission interrupt occurs.

- The empty flag of the transmission data is set (TDRE=1) and transmission processing is not done.
- In master operation (SCR:MS=0) in assist mode (LAMCR:LAMEN=1), the header transmission processing is not done.
- In assist mode (LAMCR:LAMEN=1), the response transmission processing is not done.

Moreover, the transmission bus idle flag bit (SSR:TBI) and the transmission interrupt request are cleared by the following factors:

- Transmission data is written in the transmit data register (TDR).
- In master operation (SCR:MS=0) in assist mode (LAMCR:LAMEN=1), the header transmission is being processed (LIN Break Field, Sync Field, ID Field).
- In assist mode (LAMCR:LAMEN=1), the response transmission is being processed (data and checksum).

32.7.2.5 Interrupts and Flag Setting Timing when using Transmission FIFO

This section explains the generation of interrupts and flag setting timing when using transmission FIFO.

The description is similar to those in "32.7.1.5. Interrupts When Using Transmission FIFO and Flag Setting Timing."

32.7.2.6 Timer Interrupts and Flag Setting Timing

This section explains the generation of timer interrupts and the flag setting timing.

The description is similar to those in "32.7.1.6. Timer Interrupt and Flag Setting Timing" in the manual mode.

32.7.2.7 Status Interrupts and Flag Setting Timing in Assist Mode

This section explains the generation of status interrupts and the flag setting timing in assist mode.

The status interrupt in the assist mode occurs when LIN Break Field is detected (SSR:LBD), when the sink field is detected (SACSR:SFD), when an automatic header is completed (LAMSR:LAHC), and when the checksum arithmetic operations is completed (LAMSR:LCSC)

Setting Timing of LIN Break Field Detection Flag

When serial input (SIN) of "0" is inputted into the width of eleven bits or more, the LBD bit is set to "1". At this time, when the LIN Break Field interrupt is set enabled (ESCR:LBIE=1), the status interrupt occurs.

Notes:

- When the LIN Break Field is received, if the reception is enabled (SCR:RXE=1), a framing error is detected before the LIN Break Field is detected. However, it operates normally without stopping the header reception.
- In assist mode (LAMCR:LAMEN=1), if a new LIN Break is consecutively transmitted by the master during the time from the detection of the LIN Break Field until the completion of the ID Field reception, the framing error is detected at "L" level of the tenth bit of new LIN Break Field regardless of reception prohibition setting (SCR:RXE=0). However, it operates normally without stopping the header reception.

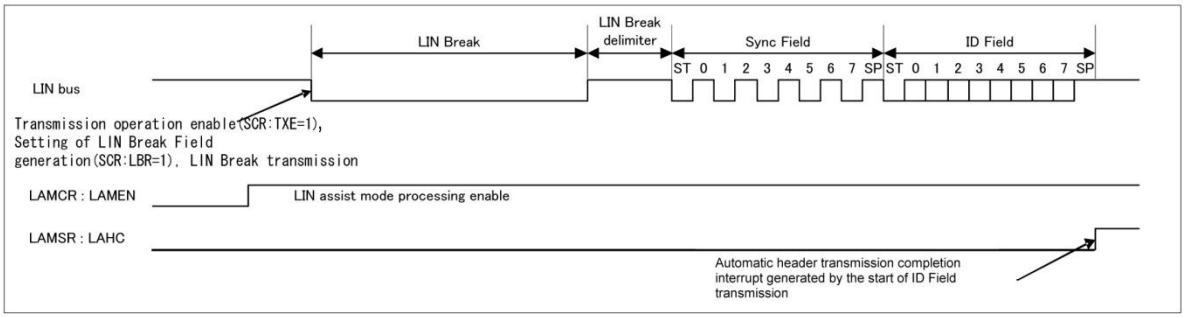
Sink Field Detection Interrupt and Flag Setting Timing

The description is similar to those in "32.7.1.7 Sync Field Detection Interrupts and Flag Setting Timing" in the manual mode.

Automatic Header Completion Interrupt and Flag Setting Timing under Transmission

On master side set in the LIN assist mode (LAMCR:LAMEN=1),when the header transmission from LIN Break to ID Field is completed, the flag is set (LAMSR:LAHC=1). If the interrupt is set enabled (LAMIER:LAHCIE =1), the status interrupt occurs.
Even when LIN bus error/LIN ID parity error/framing error occurs in the LIN assist mode for the ID Field, period the automatic header completion flag is set (LAMSR:LAHC=1). However, reception/transmission of the response stops.

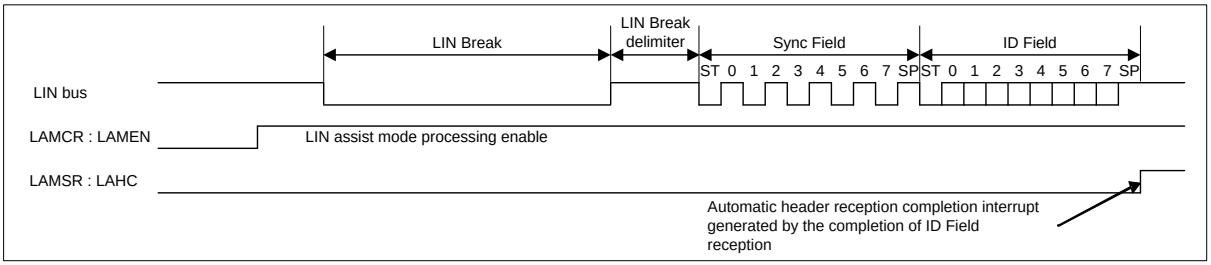
Figure 32-61. Setting Timing of Automatic Header Transmission Completion Flag (LAMSR:LAHC)



Automatic Header Completion Interrupt under Reception

On slave side set in the LIN assist mode (LAMCR:LAMEN=1),when the header reception from LIN Break to ID Field is completed, the flag is set (LAMSR:LAHC=1). If the interrupt is set enabled (LAMIER:LAHCIE =1), the status interrupt occurs.
Even when LIN bus error/LIN ID parity error/framing error occurs in the LIN assist mode for the ID Field, the automatic header completion flag is set (LAMSR:LAHC=1). However, reception/transmission processing of the response stops.

Figure 32-62. Setting Timing of Automatic Header Reception Completion Flag (LAMSR: LAHC)



LIN Checksum Detection Completion Flag Interrupt and Flag Setting Timing

In the assist mode (LAMCR:LAMEN=1), the LIN checksum detection is done on both the side where the checksum is self-checked and transmitted and the side where the checksum is received When the data of the setting length (LAMCR:LDL3-0) and the checksum are received, the checksum operation is completed and the flag is set (LAMSR:LCSC=1). When the interrupt is set enabled (LAMIER:LCSCIE=1), the status interrupt occurs.

When reception of the checksum is completed, neither the reception checksum value is stored in the RDR register nor the (SSR:RDRF) is set to "1". When FIFO is used, the received checksum value is not stored in the reception FIFO

Note:

The result of the checksum operation at this time is not guaranteed when the framing error is detected in the final data for the setting length (LAMCR:LDL3-0) or when the framing error is detected by the checksum.

32.7.3 Operation of Serial Timer

Operation of serial timer is shown.

The serial timer can be used for the timer function.

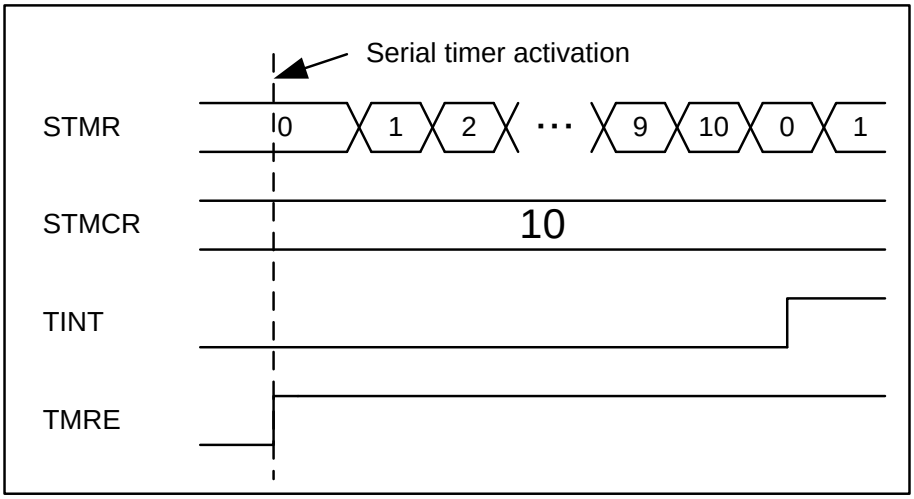
How to Start Serial Timer

There are two ways to start the serial timer: setting "1" to the serial timer enable bit (SACSR:TMRE) and starting by the Sync Field.

- Start by using the serial timer enable bit (SACSR:TMRE)

  When the serial timer enable bit (SACSR:TMRE) is set to "1", the serial timer starts and the serial timer register (STMR) starts counting from 0.

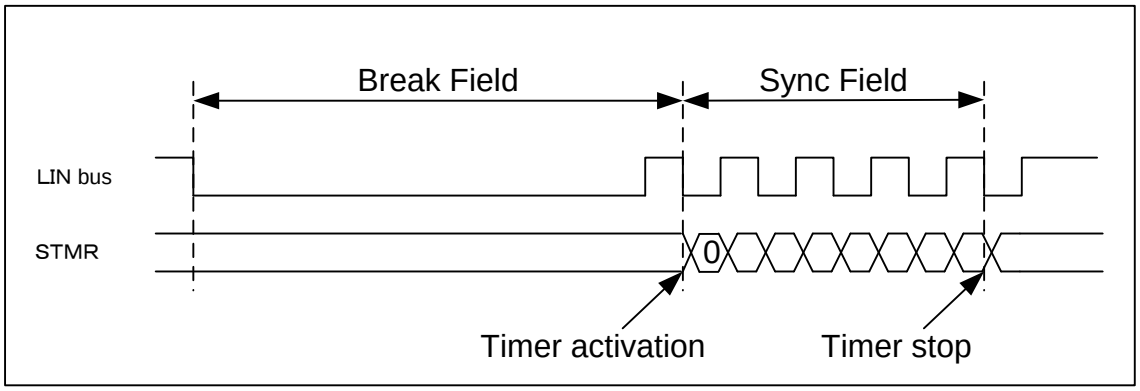
Figure 32-63. Start by Using Serial Timer Enable Bit (STMCR="10",)



- Start by Sync Field reception

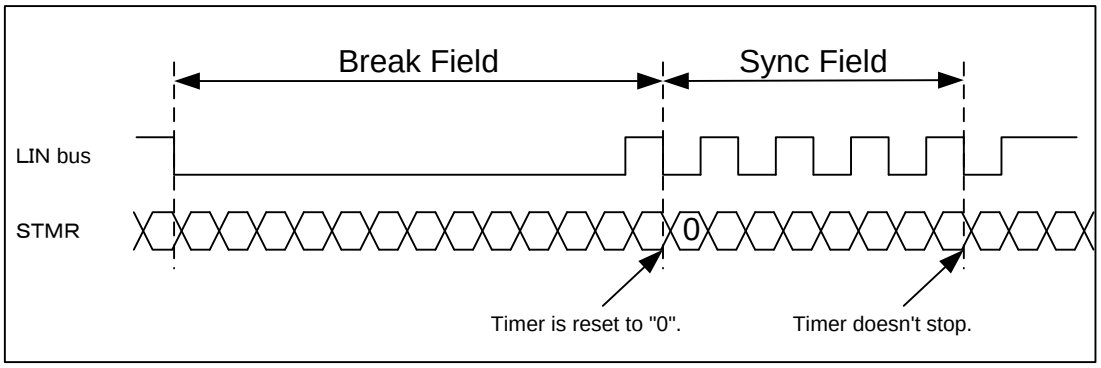
  When the serial timer is stopped and the auto baud rate adjustment bit (SACSR:AUTE) is "1", the serial timer starts and the serial timer register (STMR) starts counting from 0 if LIN interface (v2.1) detects the first falling edge of the Sync Field.

Figure 32-64. Start by Sync Field Reception During Serial Timer Stopped (SACSR:AUTE="1", TMRE="0")



When the serial timer is operating and the auto baud rate adjustment bit (SACSR:AUTE) is "1", the serial timer register (STMR) starts counting from 0 if LIN interface (v2.1) detects the first falling edge of the Sync Field.

Figure 32-65. Start by Sync Field Reception During Serial Timer Operation (SACSR:AUTE="1", TMRE="1")



How to Stop Serial Timer

The serial timer will stop under the following conditions.

- When the auto baud rate adjustment bit (AUTE) is "0", the serial timer will stop with resetting the serial timer enable bit (SACSR:TMRE) to "0". The value of the serial timer register (STMR) is retained.
- When the auto baud rate adjustment bit (AUTE) is "1" and the serial timer enable bit (SACSR:TMRE) is "1", the serial timer will stop with resetting the serial timer enable bit (SACSR:TMRE) to "0" not during Sync Field reception. The value of the serial timer register (STMR) is retained.
- When the auto baud rate adjustment bit (AUTE) is "1" and the serial timer enabled bit (SACSR:TMRE) is "0", the serial timer is stopped and the value of the serial timer register (STMR) is maintained if LIN interface (v2.1) detects the fifth falling edge of the Sync Field.

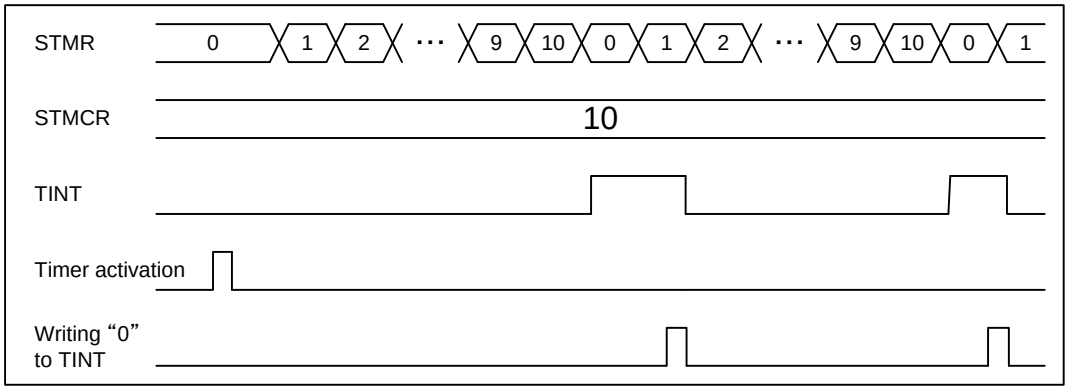
Note:

When the auto baud rate adjustment bit (AUTE) is "1" and the serial timer enabled bit (SACSR:TMRE) is "1", the serial timer is not stopped and continues the operation even if LIN interface (v2.1) detects the fifth falling edge of the Sync Field.

Timer Operation

If the serial timer register (STMR) matches the serial timer comparison register (STMCR), the timer interrupt flag (SACSR:TINT) is set to "1" and the serial timer register (STMR) is reset to "0".

Figure 32-66. Timer Operation (STMCR="10")



Notes:

- When the timer comparison register (STMCR) is set to "0000_H", the timer interrupt flag (SACSR:TINT) is fixed to "1" if the timer is operating and the division value of the timer operating clock (SACSR:TDIV) is set to "0000_B".
- If the auto baud rate adjustment bit (SACSR:AUTE) is set to "1", the serial timer register (STMR) is reset to 0 when the Sync Field is received.

32.7.4 Test Mode

Test mode is shown.

This section explains the operation of the test mode.

32.7.4.1 Manual Mode

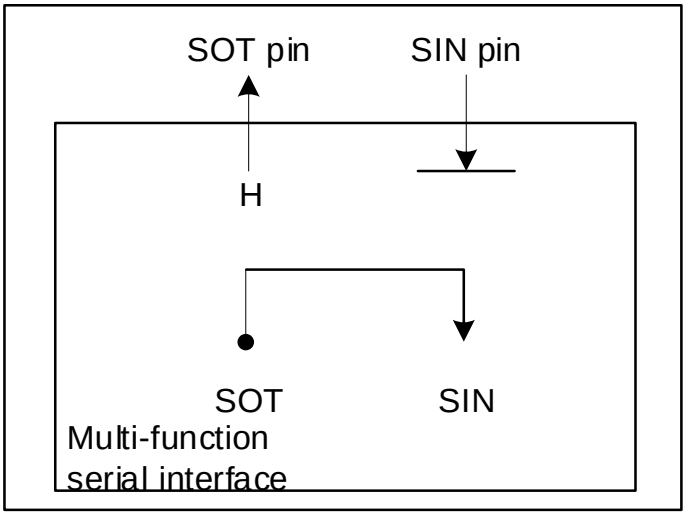
This section explains the manual mode.

Serial Test Mode

When the serial test mode is enabled (SACSR:STST="1"), SOT and SIN are connected inside the multi-function serial interface, and then the data sent from SOT can be received from SIN directly.

When the serial test mode is enabled (SACSR:STST="1"), the SOT pin is fixed to "H", and the data input to the SIN pin is ignored.

Figure 32-67. Serial Test Mode



Note:

The serial test mode enable bit (SACSR:STST) can be changed only when transmission and reception are disabled (SCR:TXE=RXE="0").

32.7.4.2 Assist Mode

This section explains the assist mode.

Serial test mode

It is similar to the serial test mode in the manual mode.
However, only master node (SCR:MS=0) can execute the serial test in LIN assist mode (LAMCR:LAMEN=1). The result of the serial test can be confirmed with the transmission/reception flag and the status flag. Please refer to Table 32-16 for the transmission/reception flag and the status flag.

Pseudo error test mode

In assist mode (LAMCR:LAMEN=1), the LIN bus error, the LIN ID parity error, the LIN checksum error, and the framing error can be artificially caused. These errors can be caused simultaneously.

Moreover, the following self-diagnoses become possible by using them together with the serial test mode.

- Pseudo LIN bus error test mode
- Pseudo LIN ID parity error test mode
- Pseudo LIN checksum error test mode
- Pseudo framing error test mode

Method of starting pseudo error test mode

It is necessary to write in key code control bit (LAMERT:KEY1,KEY0) according to the following procedure to start the pseudo error test mode, and to enable the pseudo trouble setting.

- KEY1-0="00"+ The pseudo trouble setting value is written.
- KEY1-0="01"+ The pseudo trouble setting value (the same value last time) is written.
- KEY1-0="10"+ The pseudo trouble setting value (the same value last time) is written.
- KEY1-0="11"+ The pseudo trouble setting value (the same value last time) is written.
- The pseudo trouble setting value becomes effective because of writing the fourth times.

If this procedure is not observed (When other registers are written/read during the writing procedure, the writing value is incorrect or this register is read during the writing procedure), writing becomes invalid.

The pseudo trouble setting can be released as well as setup procedure.

Note:

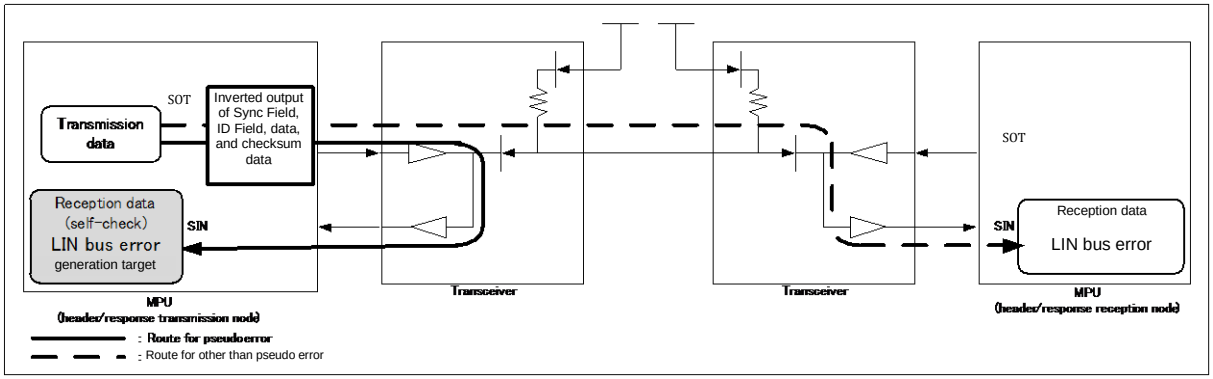
The assist mode stops when the following errors occur in the assist mode, and note the setting, please.

- LIN bus error
- LIN ID parity error
- Framing error

Overview of pseudo LIN bus error test mode

The pseudo LIN bus error test is executed by the self-check of the master/slave that transmits data. A pseudo LIN bus error cannot be detected by master/slave that receives the data.

Figure 32-68. Outline of Pseudo LIN Bus Error Test Mode



Please set the LIN bus error pseudo trouble setting bit by the method of starting the pseudo error test mode to start the pseudo LIN bus error test mode (LAMERT:LBSERT=1). The start of the pseudo LIN bus error test mode operates as follows.

- Master

  Sync Field, ID Field, data, and checksum are transmitted.
  Reception data is inverted according to the timing of the stop bit from when LIN bus error pseudo trouble was set (LAMERT:LBSERT=1), the LIN bus error is generated when self-checking it, and "1" is set to flag bit (LAMESR: LBSER).

- Slave

  Data and checksum are transmitted.
  Reception data is inverted according to the timing of the stop bit from when LIN bus error pseudo trouble was set (LAMERT:LBSERT=1), the LIN bus error is generated when self-checking it, and "1" is set to flag bit (LAMESR: LBSER).

  The LIN bus error is generated until the pseudo LIN bus error test mode setting is released (LAMESR:LBSER=0).

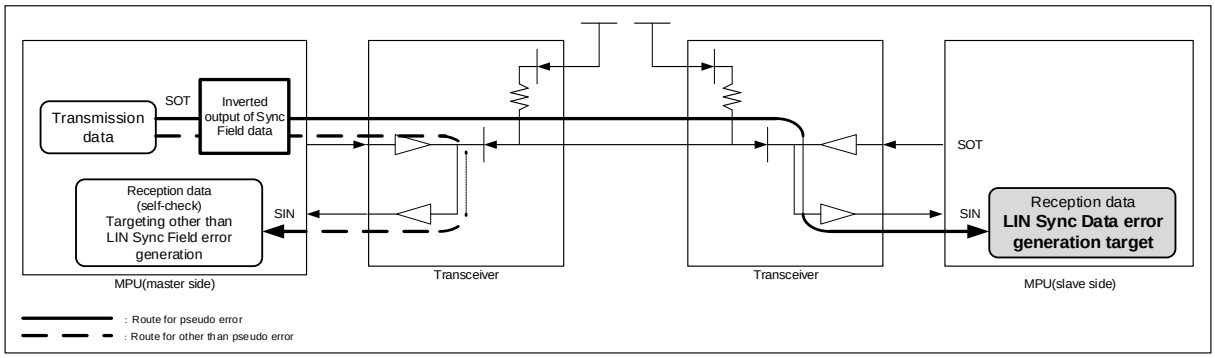
Note:

The transmission/reception processing of the header/response part of the assist mode stops by LIN bus error detection (LAMESR:LBSER=0).

Overview of pseudo LIN Sync Data error test mode

The slave that checks Sync Field value (0x55) can execute the pseudo LIN Sync Data error test.
A pseudo LIN Sync Field error cannot be detected by the master that transmits Sync Field.

Figure 32-69. Overview of Pseudo LIN Sync Field Error Test Mode



It is necessary to set the LIN Sync Data error pseudo trouble setting bit to effective (LAMERT:LSFERT=1) by using the method of starting the pseudo error test mode to start the pseudo LIN Sync Data error test mode.
The values (0x55) are all inverted by the master set to pseudo LIN Sync Data error pseudo trouble setting (LAMERT:LSFERT=1) before the start bit of Sync Field when it transmits Sync Field. It does continuing this operation until the pseudo LIN Sync Data error test mode setting is released (LAMESR:LSFERT=0).

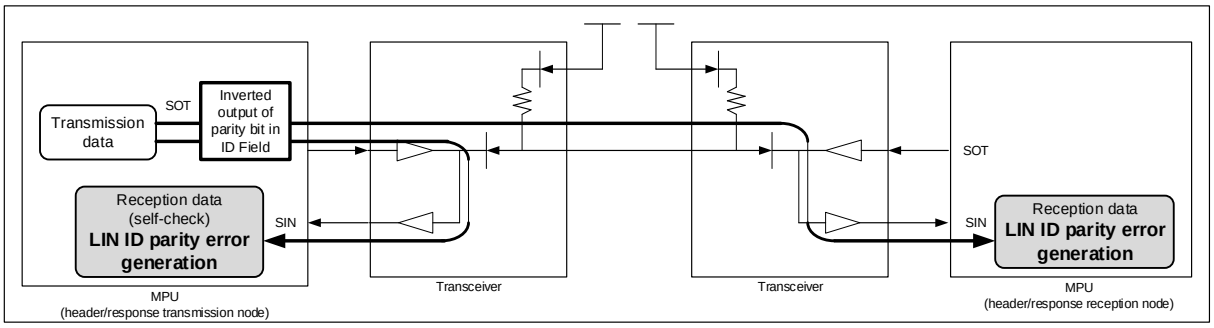
Notes:

- The detection of the LIN Sync Data error is detected in slave (SCR:MS=1) of assist mode (LAMCR:LAMEN=1).
- The header reception of the assist mode and the transmission/reception processing of the response stop by LIN Sync Data error detection (LAMESR:LBSER=1).

Overview of pseudo LIN ID parity error test mode

The pseudo LIN ID parity error test can be executed by self-check of the master that transmits ID Field and the slave that receives ID Field.

Figure 32-70. Overview of Pseudo LIN ID Parity Error Test Mode



It is necessary to set the LIN ID parity error pseudo trouble setting bit to effective (LAMERT:LPTERT=1) by the method of starting the pseudo error test mode to start the pseudo LIN ID parity error test mode.
The master to which the pseudo LIN ID parity error trouble setting (LAMERT:LPTERT=1) was enabled before the start bit of ID Field inverts and outputs all parity values (2 bit) in the ID Field when the ID Field is transmitted.
The LIN ID parity error is generated when ID Field is received, and "1" is set by flag bit (LAMESR:LPTER).
LIN ID parity error continues to be generated until the pseudo LIN ID parity error test mode is disabled (LAMESR:LPTERT=0).

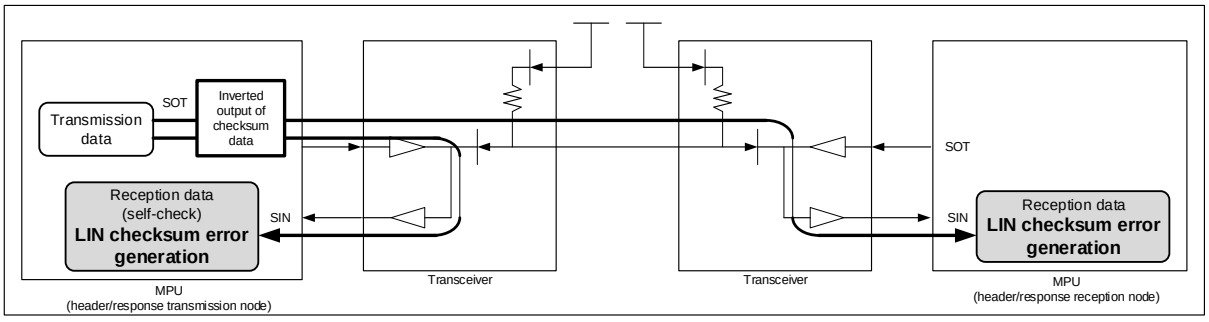
Note:

The transmission/reception processing of the response of the assist mode stops by LIN ID parity error detection (LAMESR:LPTER=1).

Overview of pseudo LIN checksum error test mode

The pseudo LIN checksum error test can be executed on the side that transmits the response for self-check and on the side that receives the response.

Figure 32-71. Outline of Pseudo LIN Checksum Error Test Mode

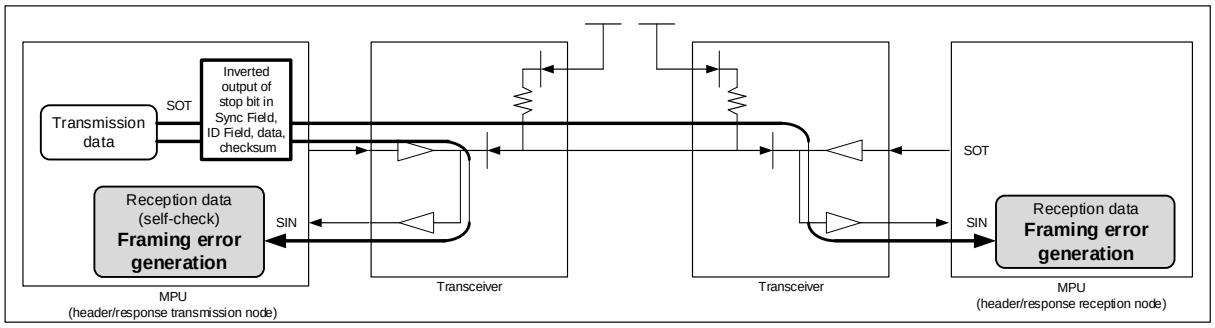


It is necessary to set the LIN checksum error pseudo trouble setting bit to effective (LAMERT:LCSERT=1) by the method of starting the pseudo error test mode to start the pseudo LIN checksum error test mode.
All the values are inverted when checksum is transmitted and the node to which the pseudo LIN checksum error pseudo trouble was set before the start bit of checksum (LAMERT:LCSERT=1) is output.
The LIN checksum error occurs when checksum is received, and "1" is set by flag bit (LAMESR:LCSER).
The LIN checksum error is generated until the pseudo LIN checksum error test mode setting is released (LAMESR:LCSERT=0).

Overview of pseudo framing error test mode

The pseudo framing error test can be executed on the side that transmits data for self-check and on the side that receives data.

Figure 32-72. Outline of Pseudo Framing Error Test Mode



It is necessary to set the framing error pseudo trouble setting bit to effective (LAMERT:FRET=1) by the method of starting the pseudo error test mode to start the pseudo framing error test mode.
The start of the pseudo framing error test mode operates as follows.

- Master

  When Sync Field, ID Field, data, and checksum are transmitted, the value of the stop bit (" H" level) is inverted to be output when the framing error pseudo trouble is set before the stop bit of each Field (LAMERT:FRET=1).
  A framing error occurs at reception, and "1" is set to flag bit (LAMESR: FRE).

- Slave

  When data and checksum are transmitted, the value of the stop bit ("H" level) is inverted to be output when the framing error pseudo trouble is set before the stop bit of each Field (LAMERT:FRET=1).
  A framing error occurs at reception, and "1" is set to flag bit (LAMESR: FRE).
  The framing error is generated until the pseudo framing error test mode setting is released (LAMESR:FRET=0).

Note:

The transmission/reception processing of the header/response part of the assist mode stops by framing error detection (LAMESR:FRE=1).

32.7.5 Operation of LIN Interface (v2.1)

Operation of LIN Interface (v2.1) is shown.

The LIN interface (v2.1) operates for the master/slave bidirectional LIN communication.

32.7.5.1 Manual mode

This section explains the manual mode.

Master Operations

Selecting Master Operation

To make the LIN interface (v2.1) work as the master device, set the SCR:MS bit to "0".

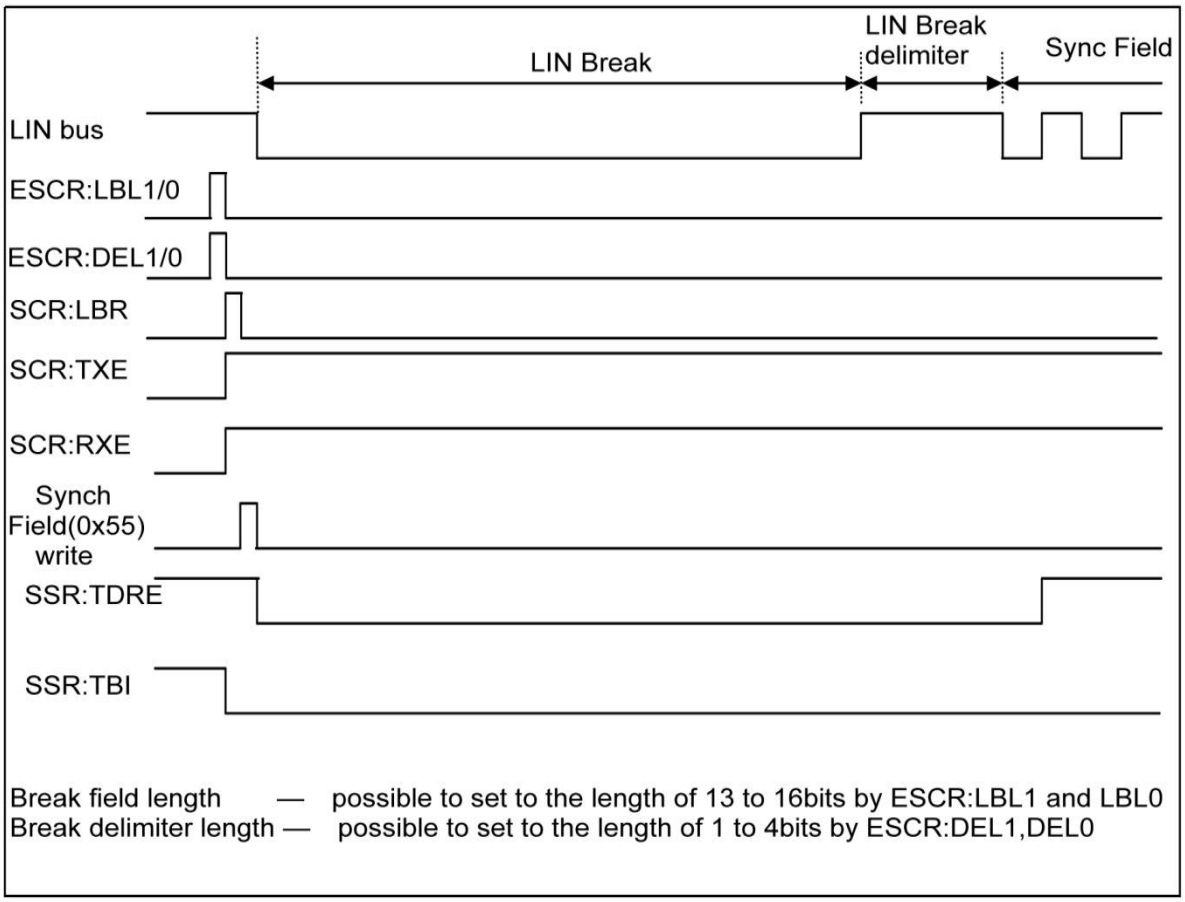
From LIN Break Field Transmission to Synch Field Transmission

- Select of the LIN Break Field length (ESCR:LBL1, LBL0) and the Break Field delimiter length (ESCR:DEL1, DEL0).
- The LIN Break Field is transmitted by enabling transmission (SCR:TXE=1) and setting the SCR:LBR bit (LIN Break field setting bit) to "1".
- The Synch Field is transmitted by writing 0x55 in the transmit data register (TDR).

Notes:

- Set 0x55 in the transmit data register (TDR) after setting the SCR:LBR bit (LIN Break field setting bit) to "1".
- Even if the SCR:RXE bit (reception enable bit) is set to "1", the LIN Break Field part is not received.

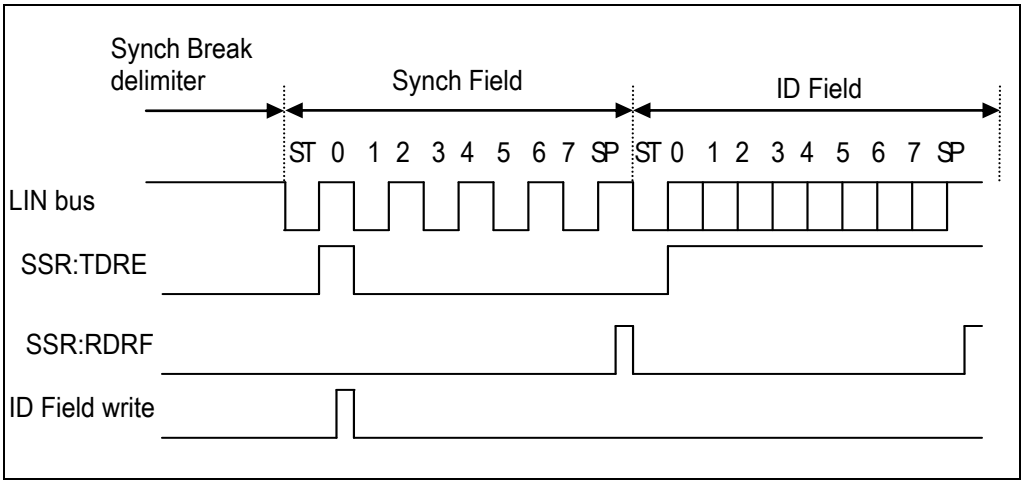
Figure 32-73. From LIN Break Field to Synch Field Transmission



From Synch Field Transmission to ID Field Transmission

- When the first bit of the Synch Field (0x55) is transmitted, the SSR:TDRE (transmission data empty) bit is set to "1". If the transmission interrupt is enabled (SCR:TIE=1) at this time, a transmission interrupt occurs.
- When this interrupt occurs, the ID Field can be written to the transmit data register (TDR).
- When a reception interrupt occurs, the reception data will be compared with the transmission data to confirm that no error has occurred.
- The ID Field is output in an LSB-first fashion with a data length of 8 bits.

Figure 32-74. From Synch Field Transmission to ID Field Transmission

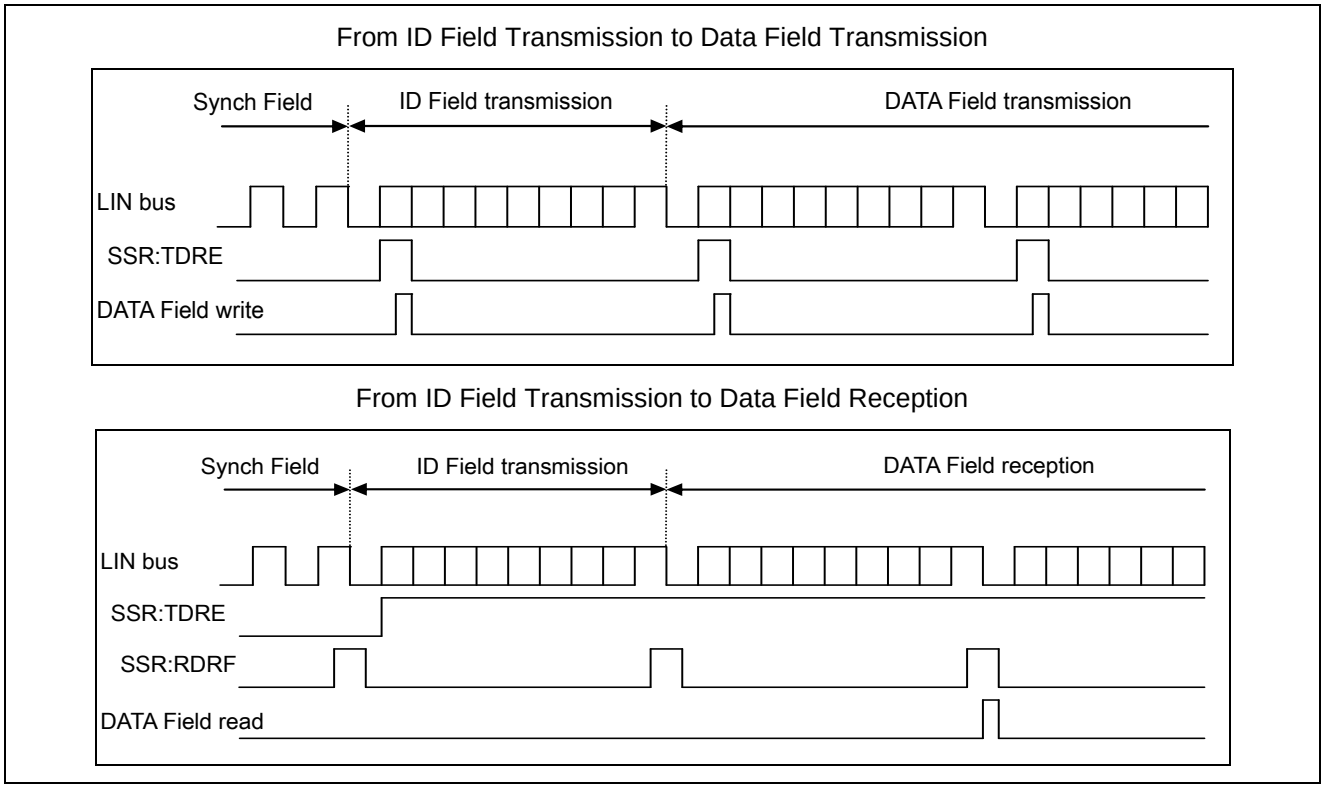


From ID Field Transmission to Data Field Transmission/Reception

Specify whether to transmit the Data Field to the slave device or receive it.

- In the case of Data Field transmission:
- When the first bit of the ID Field is transmitted, the SSR:TDRE bit is set to "1". Data can then be written in the Data Field.
- In the case of Data Field reception:
- When the first bit of the ID Field is transmitted, the SSR:TDRE bit is set to "1". However, do not write transmission data.
- Also, disable transmission interrupts (SCR:TIE=0).
- When the Data Field is received, the SSR:RDRF bit is set to "1". If reception interrupts are enabled (SCR:RIE=1) at this time, a reception interrupt will occur.
- The start bit is detected when the falling edge is detected after data passes the noise filter (majority decision by sampling the serial data input with the bus clock three times), and the passed data detects "L" at the sampling point.

Figure 32-75. From ID Field Transmission to Data Field Transmission/Reception



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. It is recommended to leave a space between frames if successive frames are to be output.

Master Operation Timing Chart (FIFO Unused)

Figure 32-76. LIN Bus Timing (at the Time of Data Field Transmission without Using FIFO)

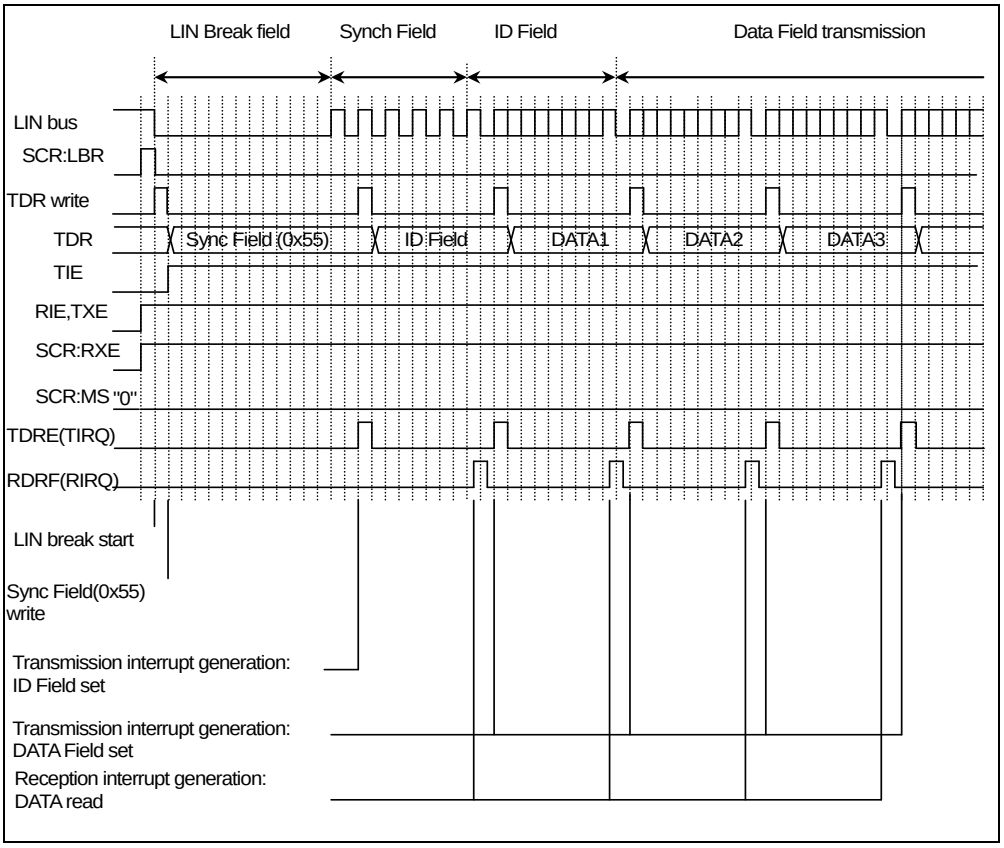
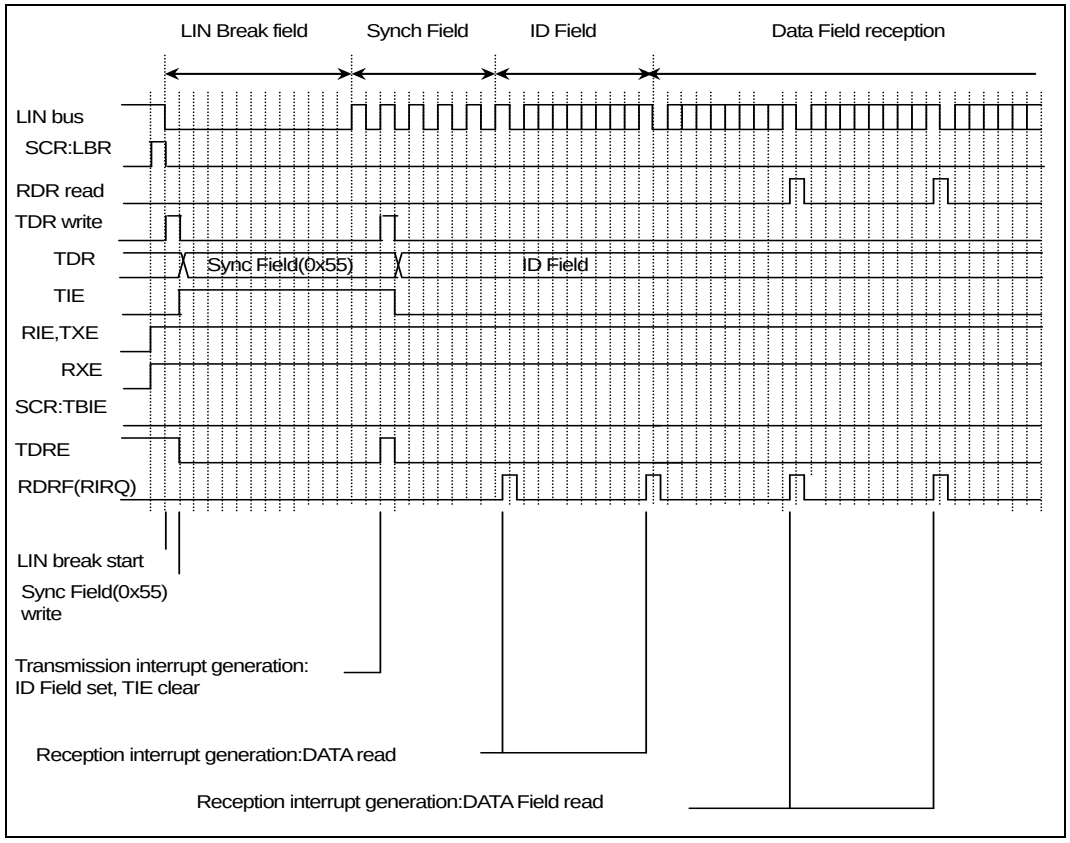


Figure 32-77. LIN Bus Timing (at the Time of Data Field Reception without Using FIFO)



Master Device Operation Timing Chart (FIFO Used)

Figure 32-78. LIN Bus Timing (at the Time of Data Field Transmission when Using FIFO)

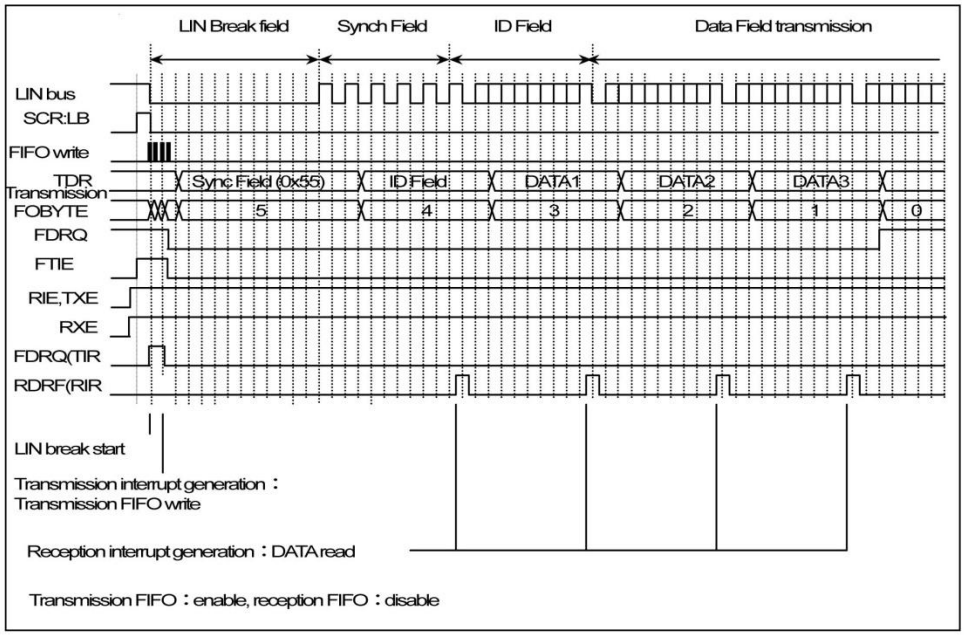
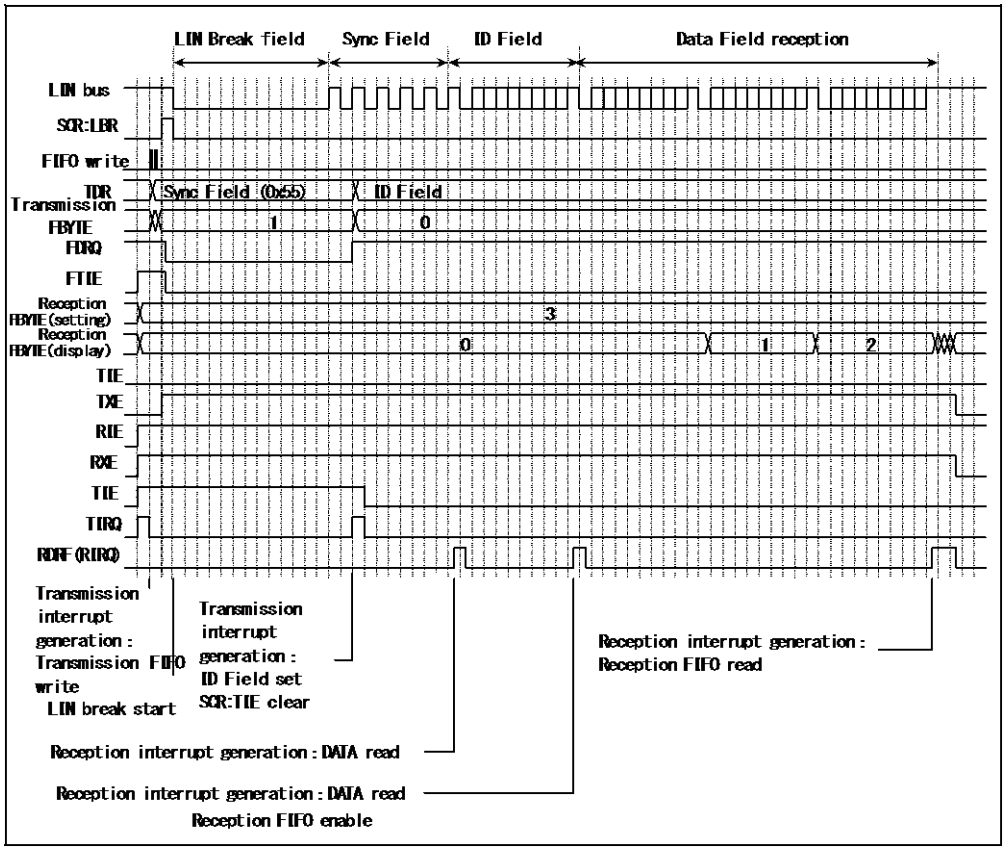


Figure 32-79. LIN Bus Timing (at the Time of Data Field Reception when Using FIFO)



Slave Operations

Selecting Slave Operation

To make the LIN interface (v2.1) work as the slave device, set the SCR:MS bit to "1".

From LIN Break Field Reception to Synch Field Reception

The method of confirming that the automatic baud rate adjustment was executed from LIN Break Field reception to Sync Field reception is as follows.

- Method of comparing BGR with STMR

Processing using this method is as follows.

Method of comparing BGR with STMR

1. The automatic baud rate adjustment is set to effective (SACSR:AUTE=1).
2. When LIN Break Field is input, LIN Break Field is detected in the 11th bit (SSR:LBD=1). At this time, if the ESCR:LBIE bit is set in "1", the status interrupt is generated. After LIN Break Field is detected (SSR:LBD=1), the serial timer is set to prohibition (SACSR:TMRE=0).
3. When LIN interface (v2.1) detects the first falling edge of Sync Field, serial timer register (STMR) is initialized to "0".
4. When the fifth falling edge of Sync Field is detected, Sync Field detection flag (SACSR:SFD) is set in "1". At this time, confirm whether the automatic baud rate adjustment was executed by checking the following.

- The reading value of baud rate generator register (BGR) becomes equal with serial timer register (STMR) at Sync Field detection (SACSR:SFD=1) when the automatic baud rate is adjusted.
- The reading value of baud rate generator register (BGR) is different from serial timer register (STMR) at Sync Field detection (SACSR:SFD=1) when the automatic baud rate adjustment is not adjusted.

Figure 32-80. From LIN Break Field Reception to Synch Field Reception
(in Case where STMR is SFUR or smaller and SFLR or larger)

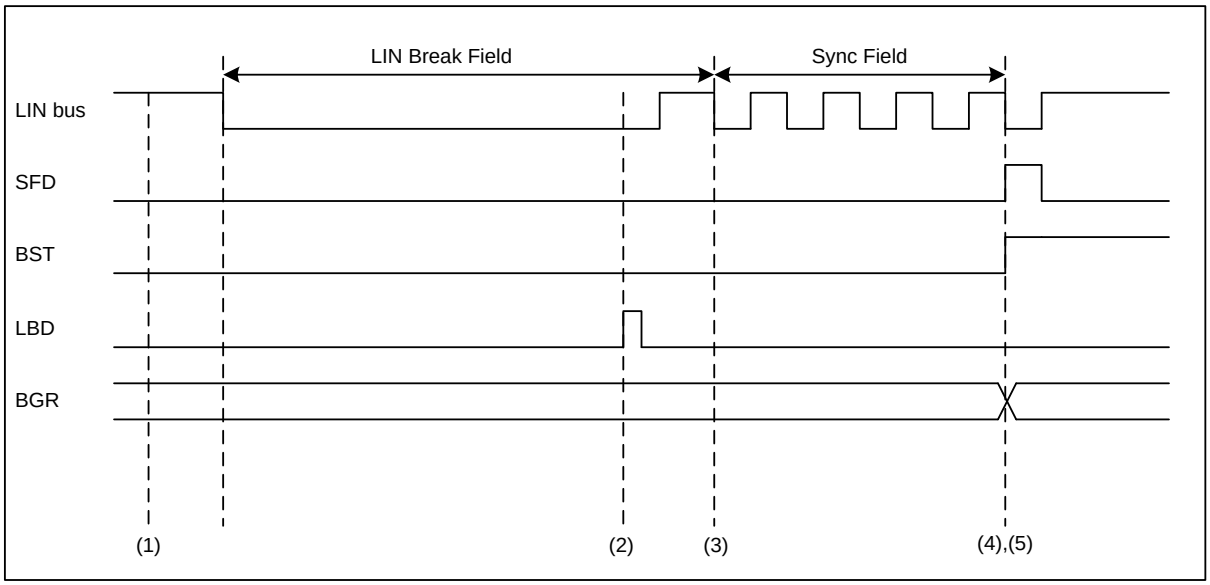
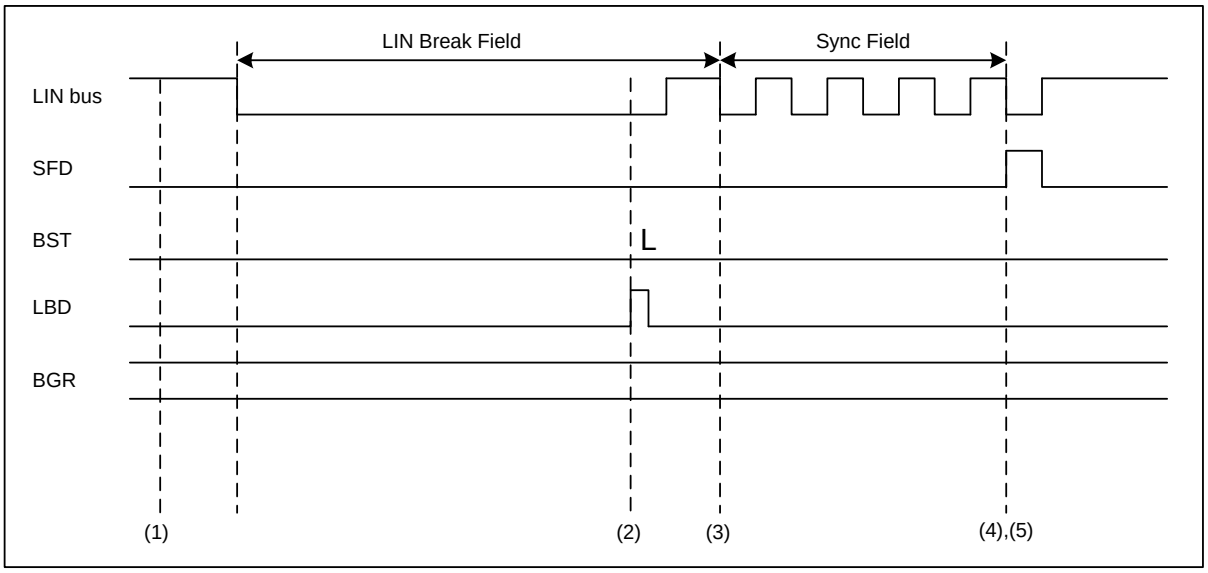


Figure 32-81. From LIN Break Field Reception to Synch Field Reception
(When STMR is not smaller than SFLR or larger than SFUR)



Note:

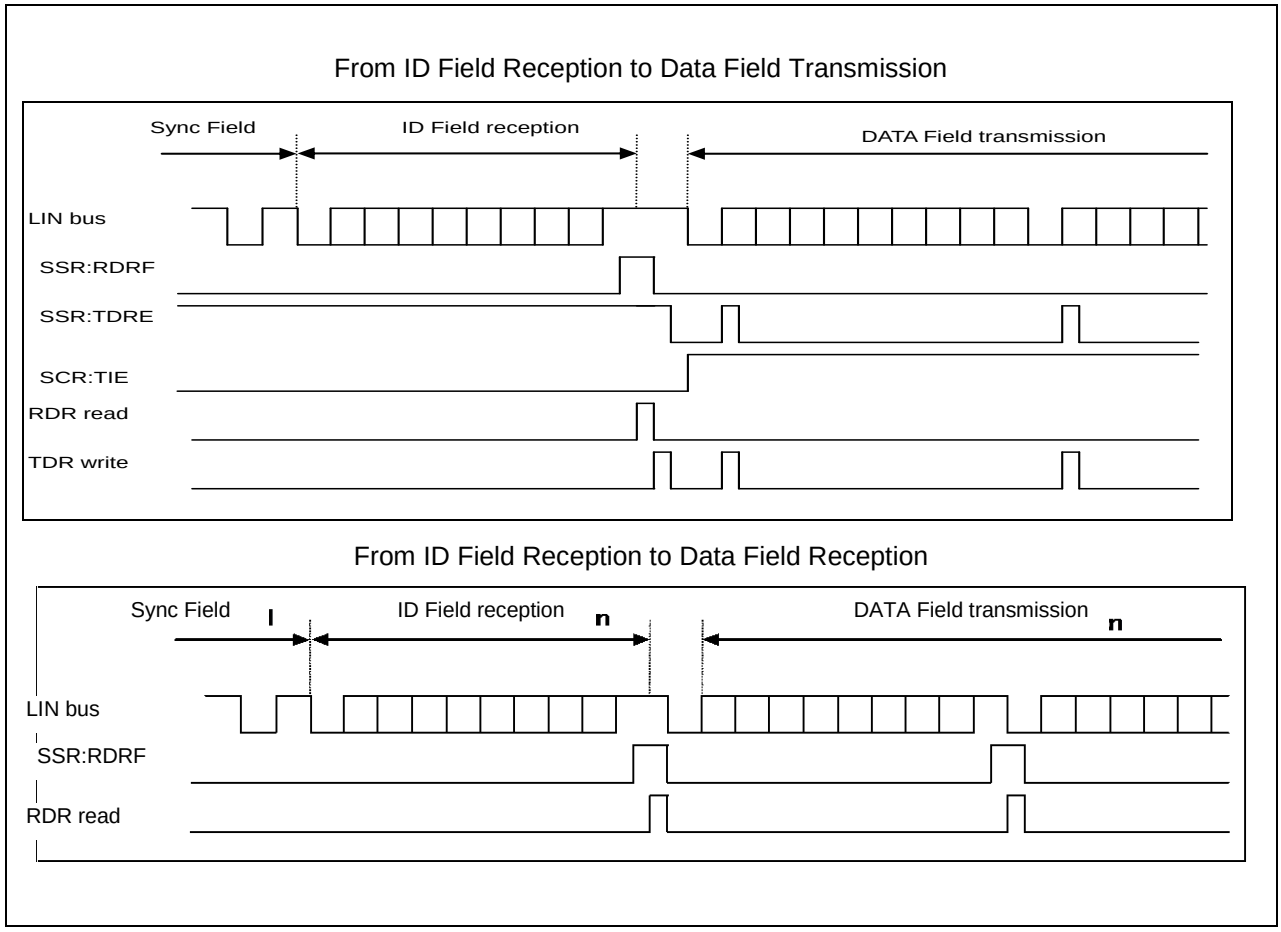
While in Break field and Sync field, set reception disabled (SCR:RXE=0).

From ID Field Reception to Data Field Transmission/Reception

After the ID Field is received, specify whether to transmit the Data Field to the master device or receive it.

- In the case of Data Field transmission:
  - ☐ After the ID Field is received, write data in the transmit data register (TDR). At this time, transmission interrupts must be enabled (SCR:TIE=1).
- In the case of Data Field reception:
  - ☐ For every Data Field reception, the SSR:RDRF bit is set to "1". If reception interrupts are enabled (SCR:RDRF=1) at this time, a reception interrupt occurs.
  - ☐ The start bit is detected when the falling edge is detected after data passes the noise filter (majority decision by sampling the serial data input with the bus clock three times), and the passed data detects "L" at the sampling point.

Figure 32-82. From ID Field Reception to Data Field Transmission/Reception



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive data normally. It is recommended to leave a space between frames if successive frames are to be output.

Slave Operation Timing Chart

Figure 32-83. LIN Bus timing (DATA Field Transmitted: FIFO Unused, AUTE=1)

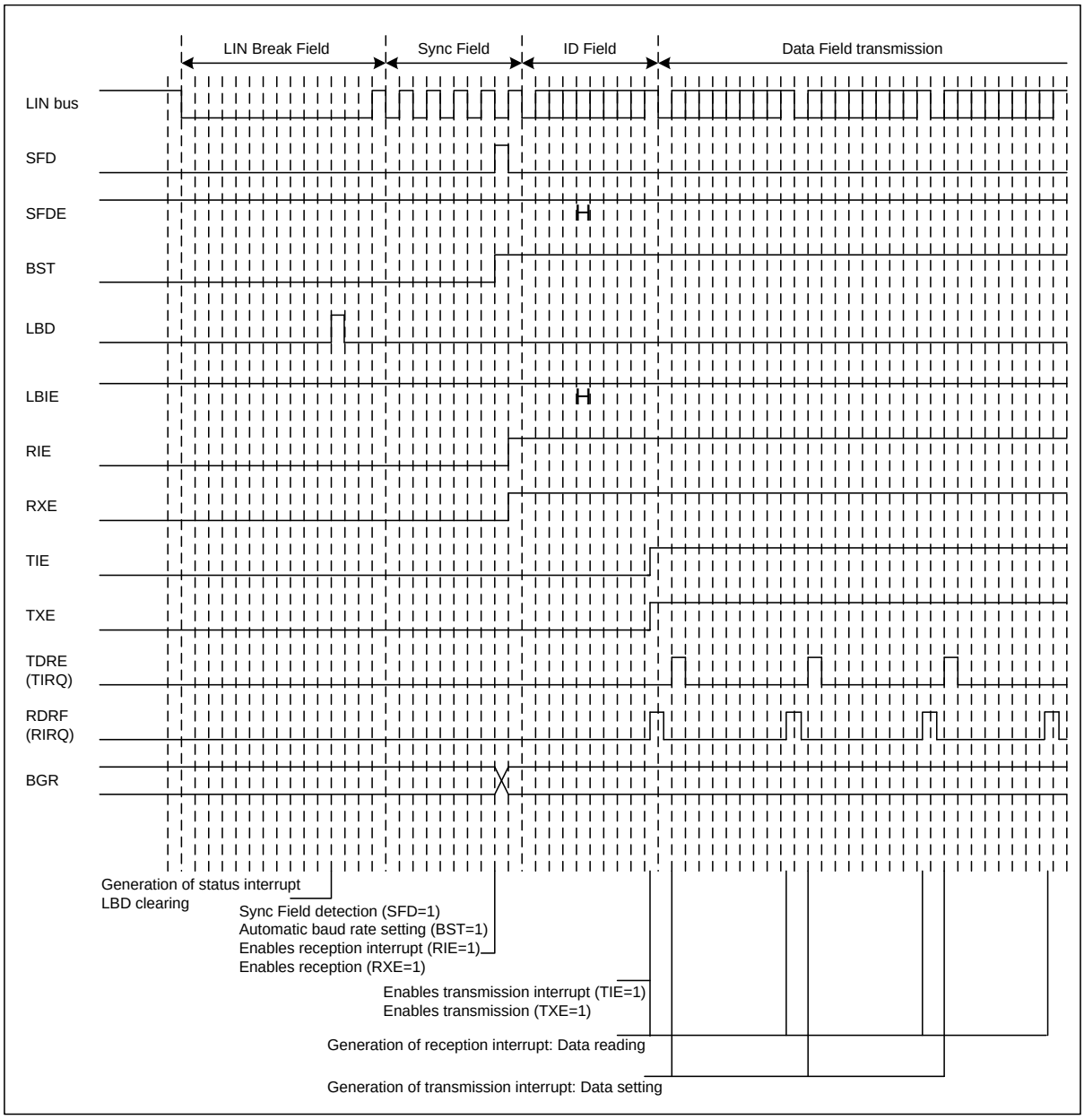
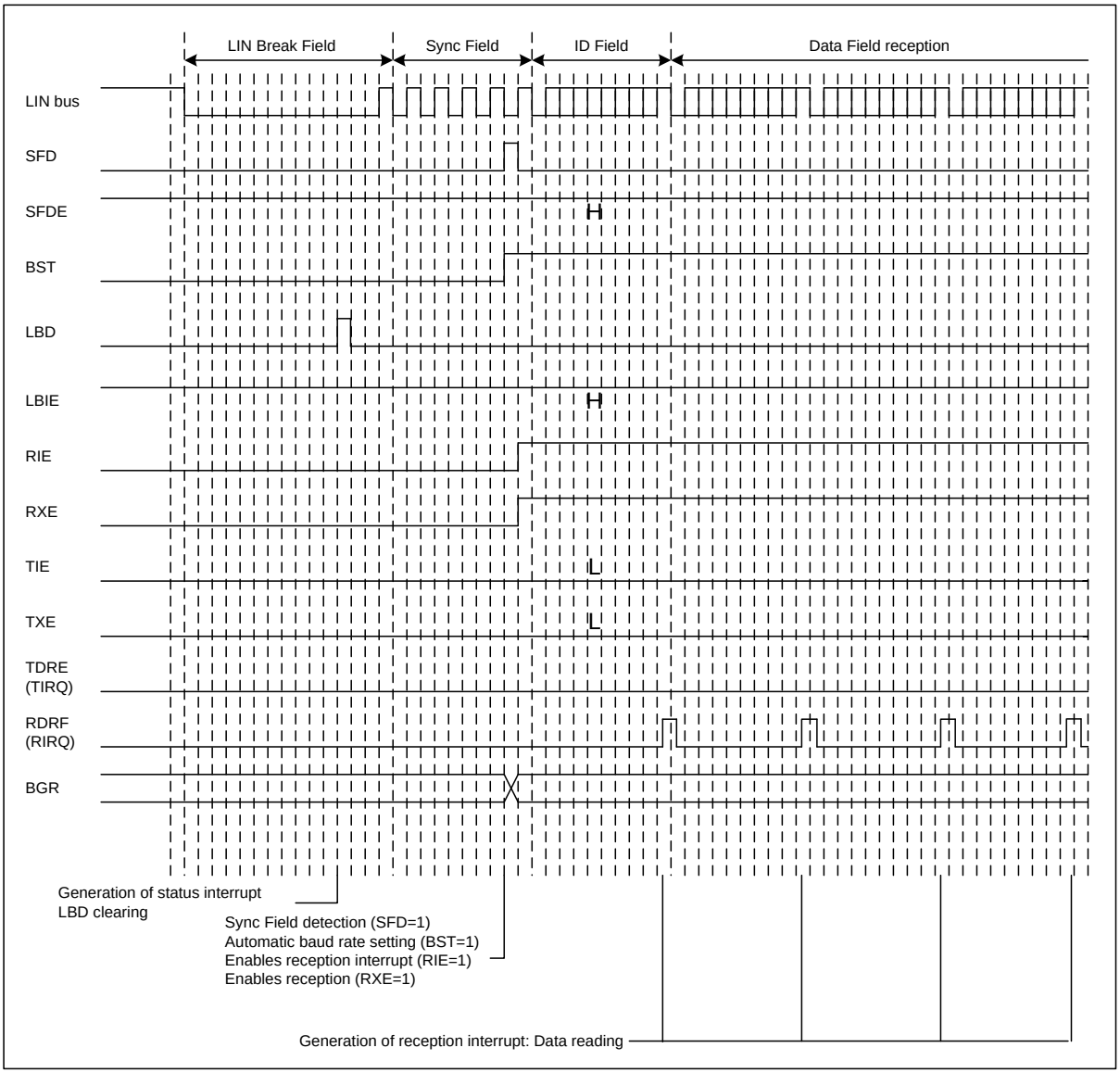


Figure 32-84. LIN Bus Timing (DATA Field Received: FIFO Unused, AUTE=1)



FIFO Used

Figure 32-85. LIN Bus Timing (DATA Field Transmitted: FIFO Used, AUTE=1)

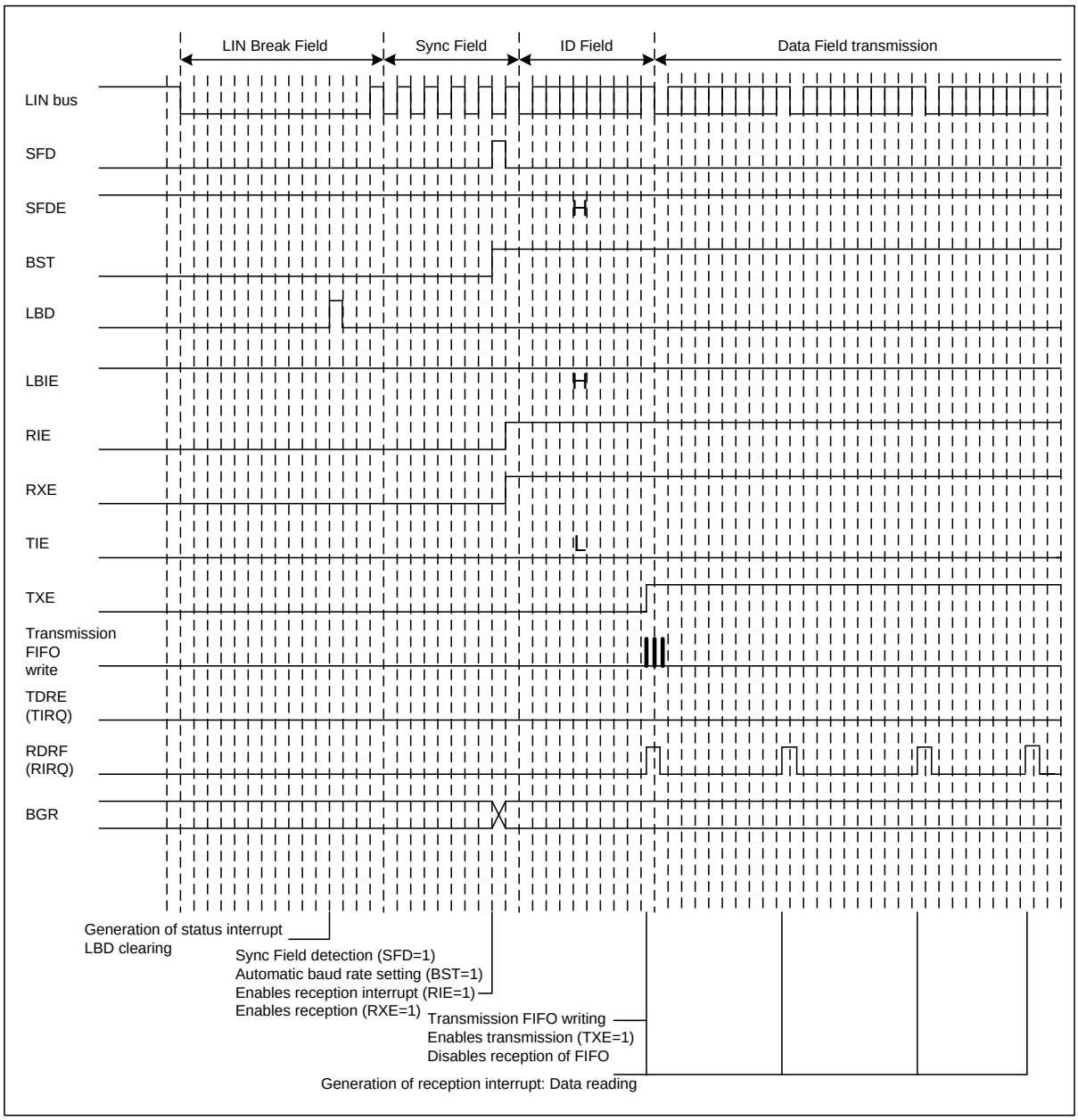
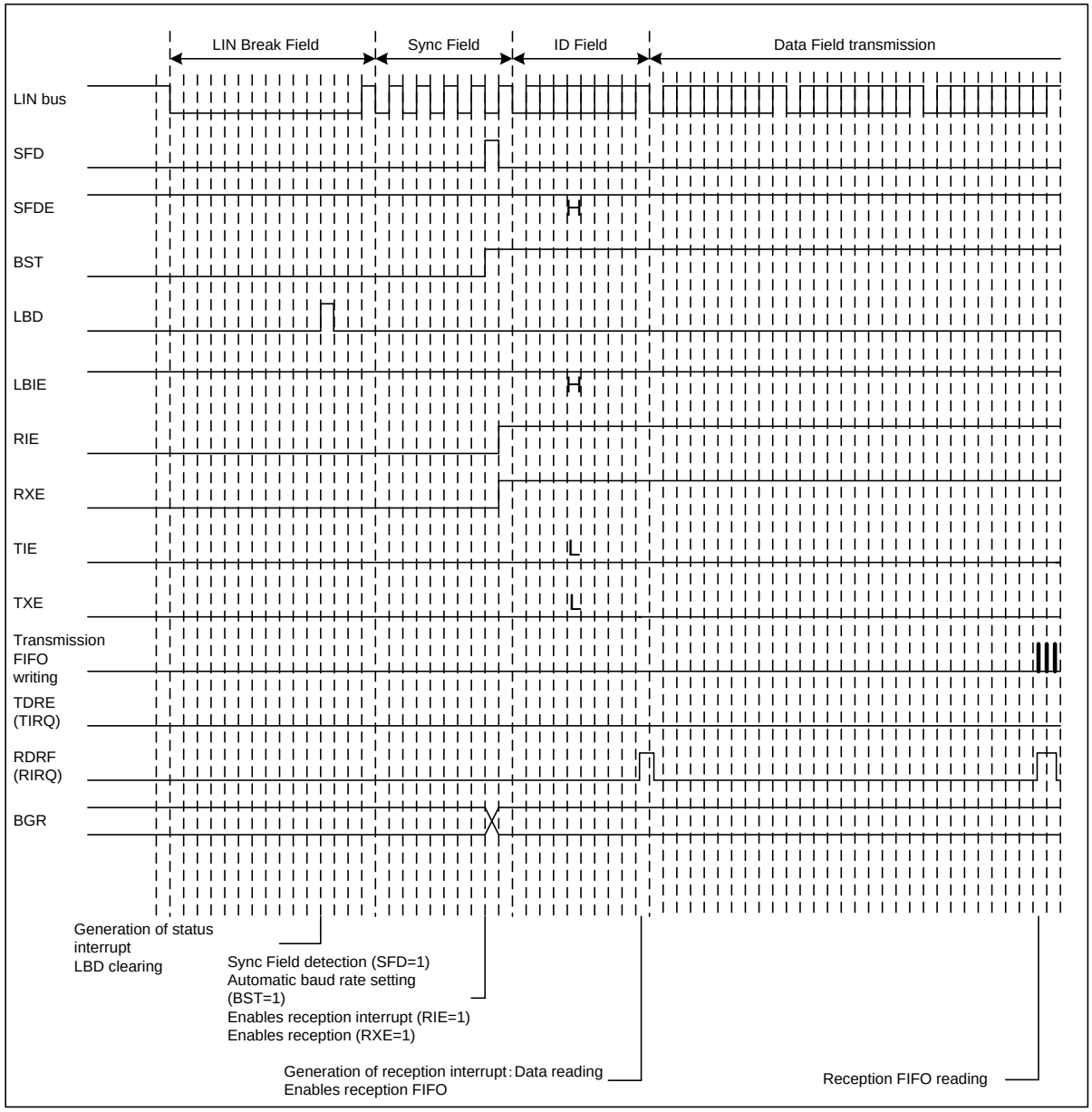


Figure 32-86. LIN Bus Timing (DATA Field Received: FIFO Used, AUTE=1)



32.7.5.2 Assist Mode

This section explains the assist mode.

The assist mode has the function to transmit/receive the LIN automatic header, and generate and check the following.

- Parity generation and check on ID Field
- Generation and check on checksum

Master operation

Automatic header transmission setting

To transmit the automatic header in the assist mode, please set the SCR:LBR bit (LIN Break Field setting bit) to "1" after initial setting. "LIN Break Field - Sync Field - ID Field" is automatically transmitted by setting the SCR:LBR bit to "1". The transmission setting is shown as follows.

- Please set the SCR:MS bit (master/slave function select bit) to "0" to operate as the master.
- Please set the LAMCR:LAMEN bit (LIN assist mode processing enable bit) to "1".
- Please set the ID Field value before the LIN assist mode begins.
- Please set the LAMCR:LIDEN bit (LIN ID register enable bit) to "1" when you use LIN assist mode transmission ID register (LAMTID).
- Please set the LAMCR:LIDEN bit to "0" when you use data transmit register (TDR).
- Please set the ID Field data to LAMTID (LIN assist mode transmission ID register) when you set the LAMCR:LIDEN bit to "1".
- Please set selection of the LIN break field length (ESCR:LBL2, LBL1, LBL0) and selection of the LIN Break delimiter length (ESCR:DEL1, DEL0).
- Please set the selection of the stop bit length (SMR:SBL and ESCR:ESBL).
- LIN Break Field transmitted on the master side is detected also on the master side. The SSR:LBD bit is set in "1" when detected. At this time, if ESCR:LBIE is set to "1", the status interrupt is generated. Please set ESCR:LBIE to "0" and change the interrupt to the prohibition setting for the LIN assist mode.
- The Sync Field value transmitted on the master side is detected also on the master side. The SACSR:SFD bit is set in "1" when detected. At this time, if the SACSR:SFDE bit is set to "1", the interrupt is generated. Please set SACSR:SFDE to "0" for the LIN assist mode and prohibit interrupting.
- Please set transmission operation enable bit (SCR:TXE) to "1" (transmission enable).

From LIN Break Field to ID Field transmission

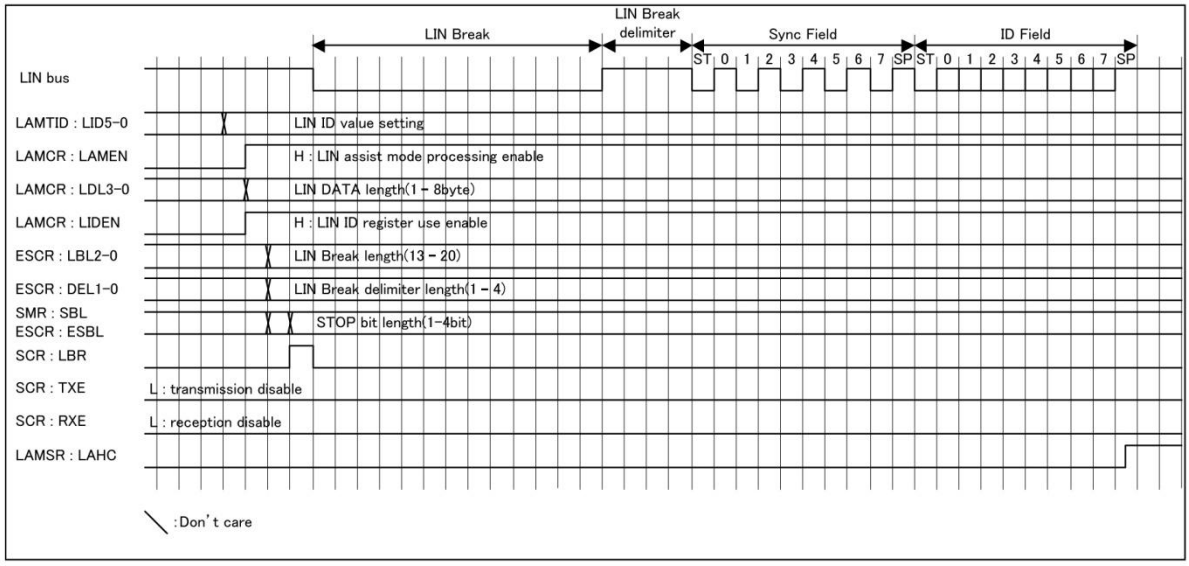
- Please set LIN Break Field setting bit (SCR:LBR) to "1" (LIN Break Field generation).
- LIN Break Field set with ESCR:LBL2 to LBL0 is transmitted.
- Please write the ID Field data in data transmission register (TDR) when you do not use LIN assist mode transmission ID register (LAMTID).
- LIN Break Field that the master transmitted is received on the master side, and the bus error is checked.
- After LIN Break Field is transmitted, the LIN Break Field delimiter set with ESCR:DEL1 and DEL0 is transmitted.
- After the LIN Break Field delimiter is transmitted, Sync Field (0x55 fixation value) is transmitted.
- Sync Field that the master transmitted is received on the master side, and the bus error is checked.
- After Sync Field is transmitted, the set ID Field value is transmitted. When the LAMCR:LIDEN bit is "0", the value set to TDR is transmitted as ID FIeld value. When the LAMCR:LIDEN bit is "1", the value set to LAMTID is transmitted as ID FIeld value.
- The LIN assist mode does the parity arithmetic operations of ID Field automatically.

- When the first bit of ID Field is transmitted, SSR:TDRE (the transmission data empty) bit is set to "1". At this time, if transmission interrupt enable (SCR:TIE=1) is done, the transmission interrupt is generated.
- When transmission interrupt (TDRE) is generated, the transmission data can be written in transmit data register (TDR).
- ID Field is eight bits in the data length, and it is output with LSB first. The LIN parity in ID Field is operated automatically.
- When the transmission of ID Field is completed, the LIN automatic header transmission completion flag is set (LAMSR:LAHC="1"). At this time, when the LIN automatic header transmission completion interrupt enable bit has been enabled (LAMIER:LAHCIE="1"), the interrupt is generated.
- The transmission stops when the following errors are generated.
  - ☐ LIN bus error
  - ☐ LIN ID parity error
  - ☐ Framing error

Notes:

- The automatic header transmission stops when SCR:TXE is set to "0" after the LIN assist mode activation.
- Please do not change the automatic header transmission setting after the LIN assist mode activation.

Figure 32-87. From LIN Break Field to ID Field Transmission



LIN Break Field retransmission processing in assist mode

Only when transmission prohibition setting (SCR:TXE=0) and state (SSR:TBI=1) of the transmission bus idle, LIN Break Field can be set (SCR:TBR=1). Therefore, when other than the transmission prohibition setting (SCR:TXE=0) or the transmission bus idle (SCR:LBR=1), the LIN Break Field setting (SCR:LBR=1) can be set after initializing the state according to the following procedure.

- First of all, execute transmission prohibition setting (SCR:TXE=0) and reception prohibition setting (SCR:RXE=0).
- Clear the transmission data before retransmission.
  - ☐ When transmission FIFO is used, reset the transmission FIFO (FCR0:FCL1=1 or FCR0:FCL2=1) after prohibiting transmission FIFO operation (FCR0:FE1=0 or FCR0:FE2=0).
  - ☐ Then, execute the transmission data register clear (LAMCR:LTDRCL=1), and set the state to transmission bus idle.

- Clear the reception data before retransmission.
  - ☐ When reception FIFO is used, reset the reception FIFO (FCR0:FCL1=1 or FCR0:FCL2=1) after prohibiting reception FIFO operation (FCR0:FE1=0 or FCR0:FE2=0).
  - ☐ Then, read the RDR register to clear the reception data register
- When LIN assist mode transmission ID register (LAMTID) is used after the above-mentioned is processed, set the ID Field data to LAMTID (LIN assist mode transmission ID register).
  - ☐ The processing that follows is equal to the preceding section "From LIN Break Field to ID Field transmission."

DATA Field transmission/reception

Whether DATA Field is transmitted or received to the slave device is selected.

(When DATA Field is transmitted)

- ☐ When LIN assist mode transmission register (LAMTID) is not used, SSR:TDRE is set to "1" if the first bit of ID Field is transmitted. At this time, DATA Field can be written.
- ☐ When LIN assist mode transmission register (LAMTID) is used, DATA Field can be written after LIN Break Field setting bit (SCR:LBR) is set to "1".
- ☐ Please set the transmission enable (SCR:TXE=1) during the period from the setting of LIN Break Field setting bit (SCR:LBR) to "1" to starting of the response transmission.
- ☐ The LIN assist mode does the checksum arithmetic operations automatically. The arithmetic operations of checksum can select the arithmetic operations method by LIN checksum type selection bit (LAMCR:LCSTYP).
- ☐ When the transmission of checksum is completed, the transmission bus idle flag (SSR:TBI) is set. At this time, when the transmission bus idle interrupt enable bit is set (SCR:TBIE="1"), the interrupt is generated.
- ☐ After the response transmission is completed (LAMSR:LCSC=1), transmission prohibition setting (SCR:TXE=0) is done.

Notes:

- The response transmission data (Data Field and checksum) when the assist mode operates cannot be stored in the RDR register.
- Please write data in FIFO after setting LIN Break Field setting bit (SCR:LBR) to "1" (LIN Break Field generation bit) when you use FIFO.
- Please write the dummy value ("don't care") to the TDR register to operate checksum automatically and to transmit when you set the LIN data length to 0 byte length (LAMCR:LDL3-0="0000") in the response transmission. The TDR setting value at this time doesn't influence checksum.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - ☐ When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - ☐ When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes inverted ID Field.

Figure 32-88. From ID Field Transmission to DATA Field Transmission (FIFO Unused when ID Register is Used)

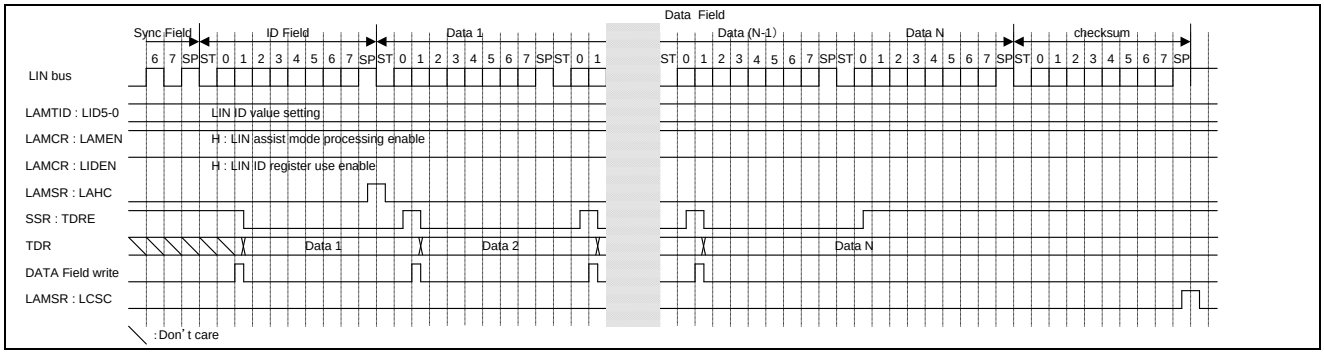
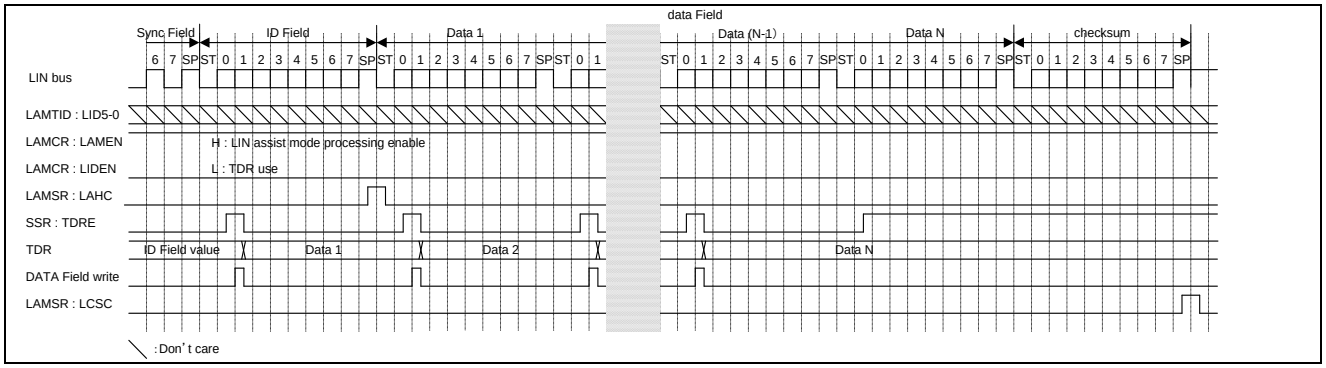


Figure 32-89. From ID Field Transmission to DATA Field Transmission (FIFO Unused when ID Register is not Used)



(When DATA Field is received)

- When LIN assist mode transmission register (LAMTID) is not used, please do not write data though SSR:TDRE is set to "1" when the first bit of ID Field is transmitted.
  Moreover, please set transmission interrupt prohibition (SCR:TIE="0").
- When LIN assist mode transmission register (LAMTID) is used, please do not write data though it is possible to write in DATA Field after LIN Break Field setting bit (SCR:LBR) is set to "1".
- Please set the reception enable (SCR:RXE=1) during the period from LIN Break Field detection (SSR:LBD=1) to the response reception starting.
- When DATA Field is received, SSR:RDRF is set to 1. At this time, if reception interrupt enable (SSR:RIE="1") is done, the reception interrupt is generated.
- When the reception of checksum is completed, the LIN checksum arithmetic operations completion flag is set (LAMSR:LCSC="1"). At this time, when the checksum arithmetic operations completion interrupt enable bit has been enabled (LAMIER:LCSCIE="1"), the interrupt is generated.
- After the checksum reception is completed (LAMSR:LCSC=1), reception prohibition setting (SCR:RXE=0) is done.
- Detection of the start bit is as follows; the falling edge is detected after passing through the noise filter (which samples serial data input in 3 bus clock and decides the value by majority), and the "L" is detected in the data after the noise filter at the sampling point.

Figure 32-90. From ID Field Transmission to DATA Field Reception (FIFO Unused when ID Register is Used)

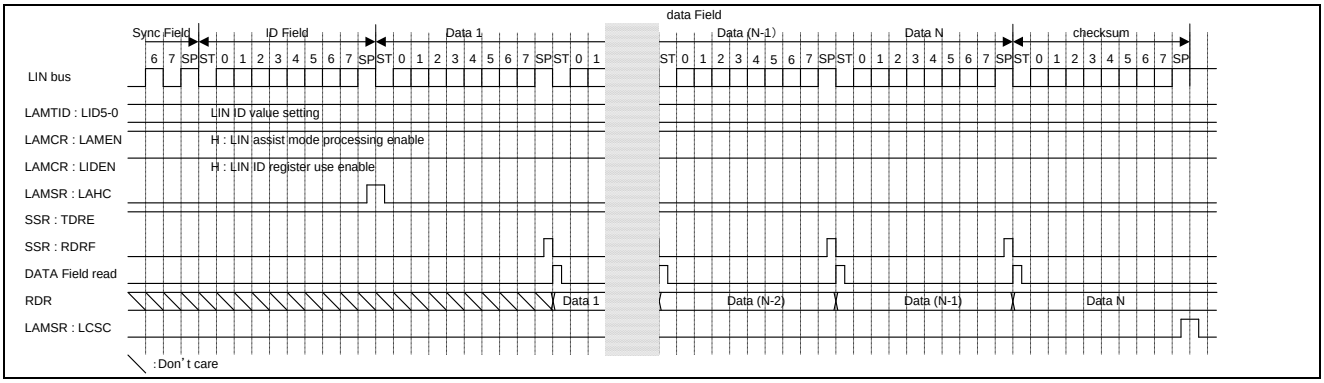
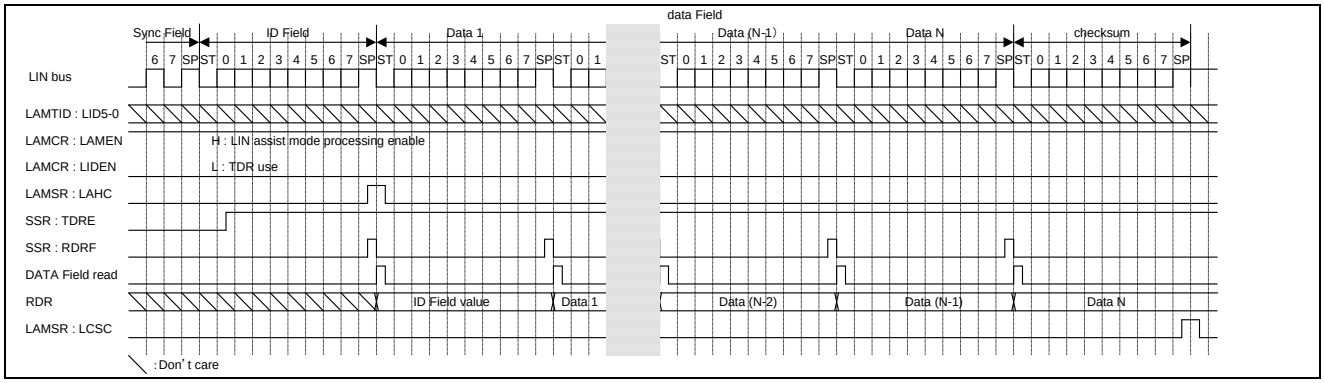


Figure 32-91. From ID Field Transmission to DATA Field Reception (FIFO Unused when ID Register is not Used)



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. If successive frames are to be output, leaving a space between frames is recommended.
- The checksum value of the response reception when the assist mode operates is not stored in the RDR register.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - ☐ When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - ☐ When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes reversed ID Field.

Master operation timing chart (When FIFO unused)

Figure 32-92. LIN Bus Timing (ID Register Use, DATA Field Transmission, and FIFO Unused)

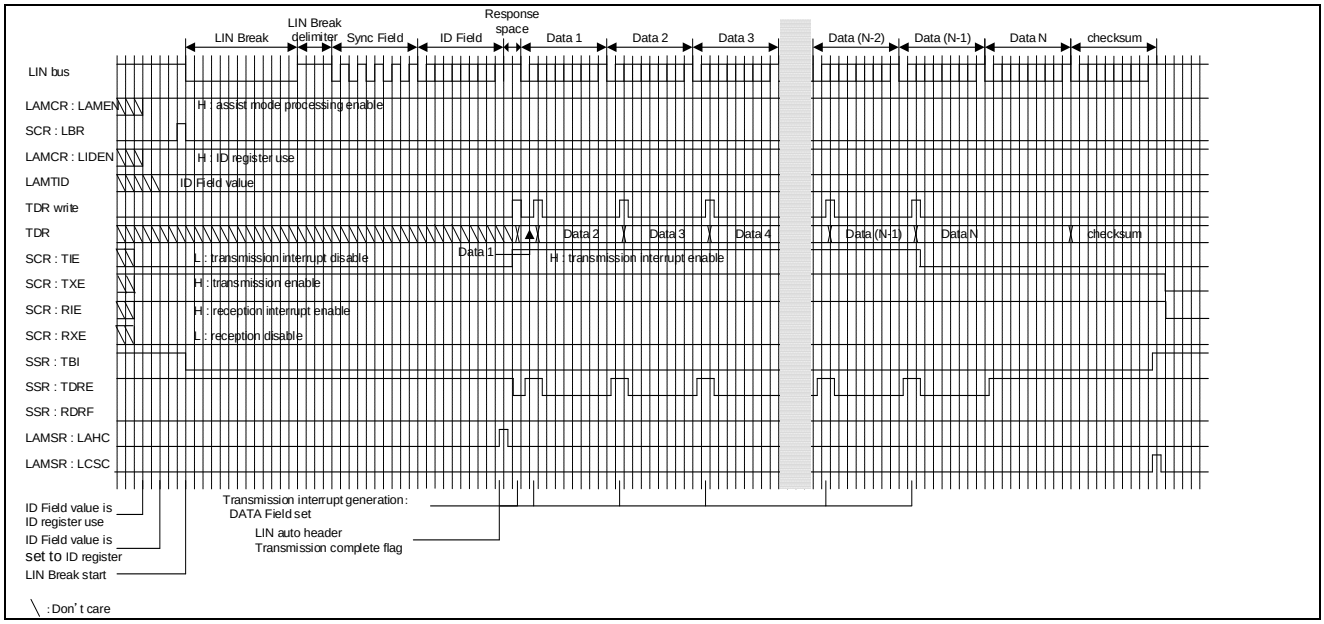


Figure 32-93. LIN Bus Timing (ID Register Unused, DATA Field Transmission, and FIFO Unused)

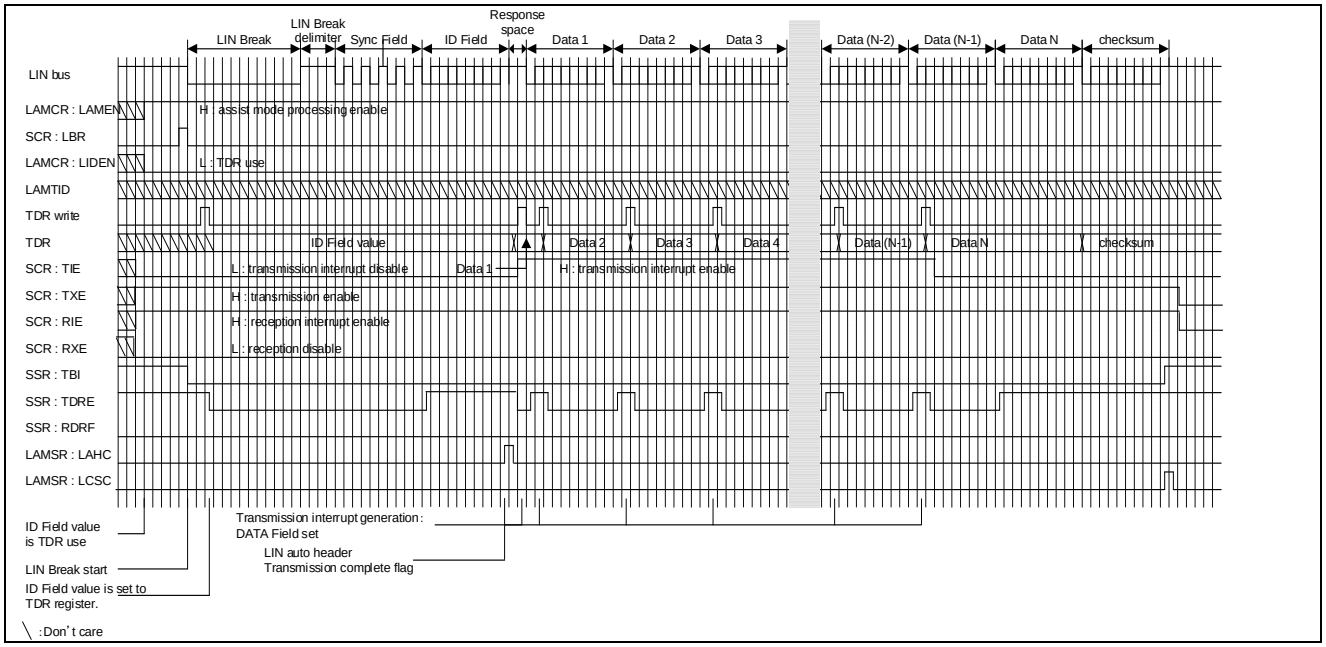


Figure 32-94. LIN Bus Timing (ID Register Use, DATA Field Reception, and FIFO Unused)

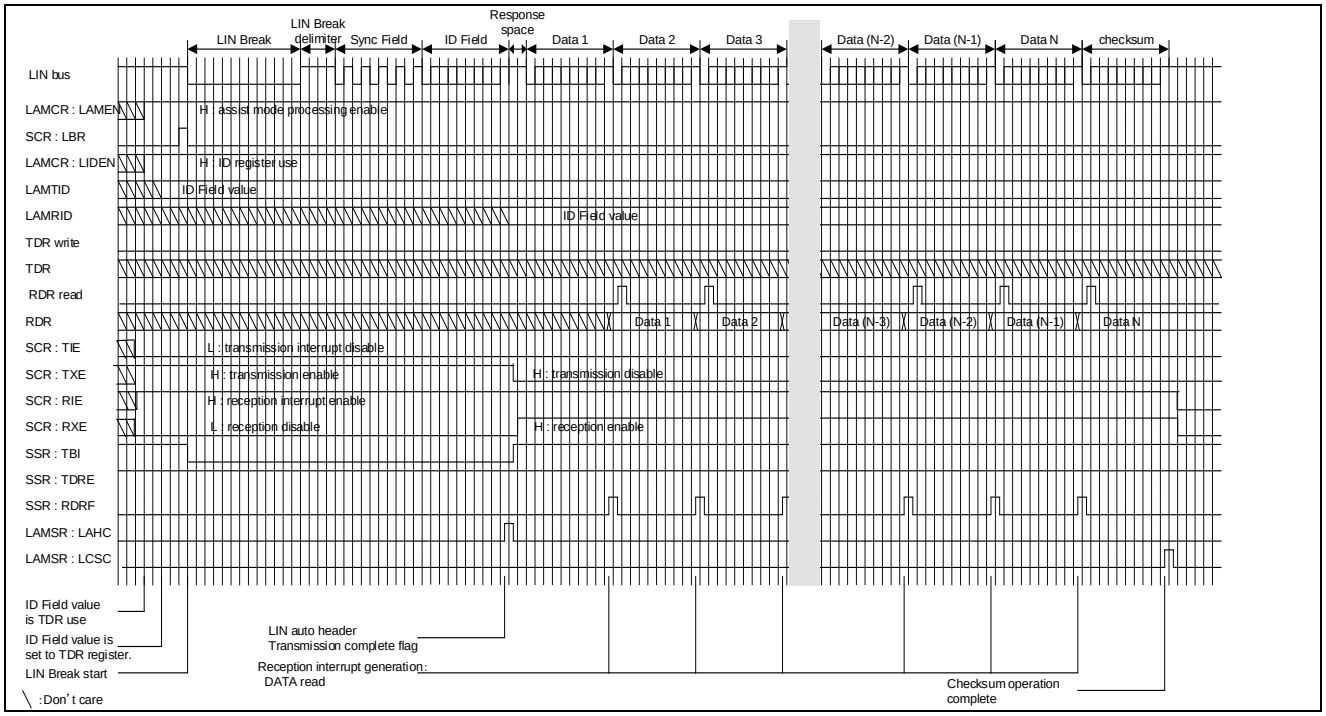


Figure 32-95. LIN Bus Timing (ID Register Unused, DATA Field Reception, and FIFO Unused)

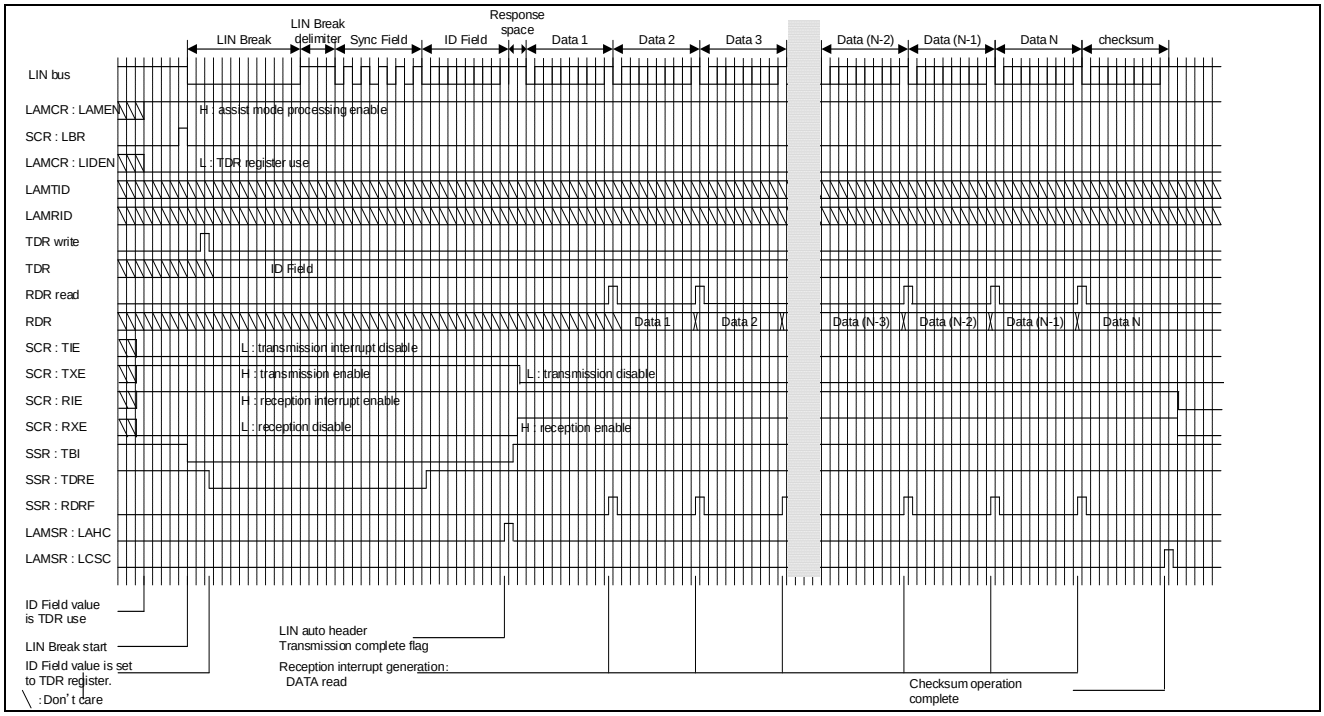


Figure 32-96. LIN Bus Timing (ID Register Use, DATA Field Transmission, and FIFO Use)

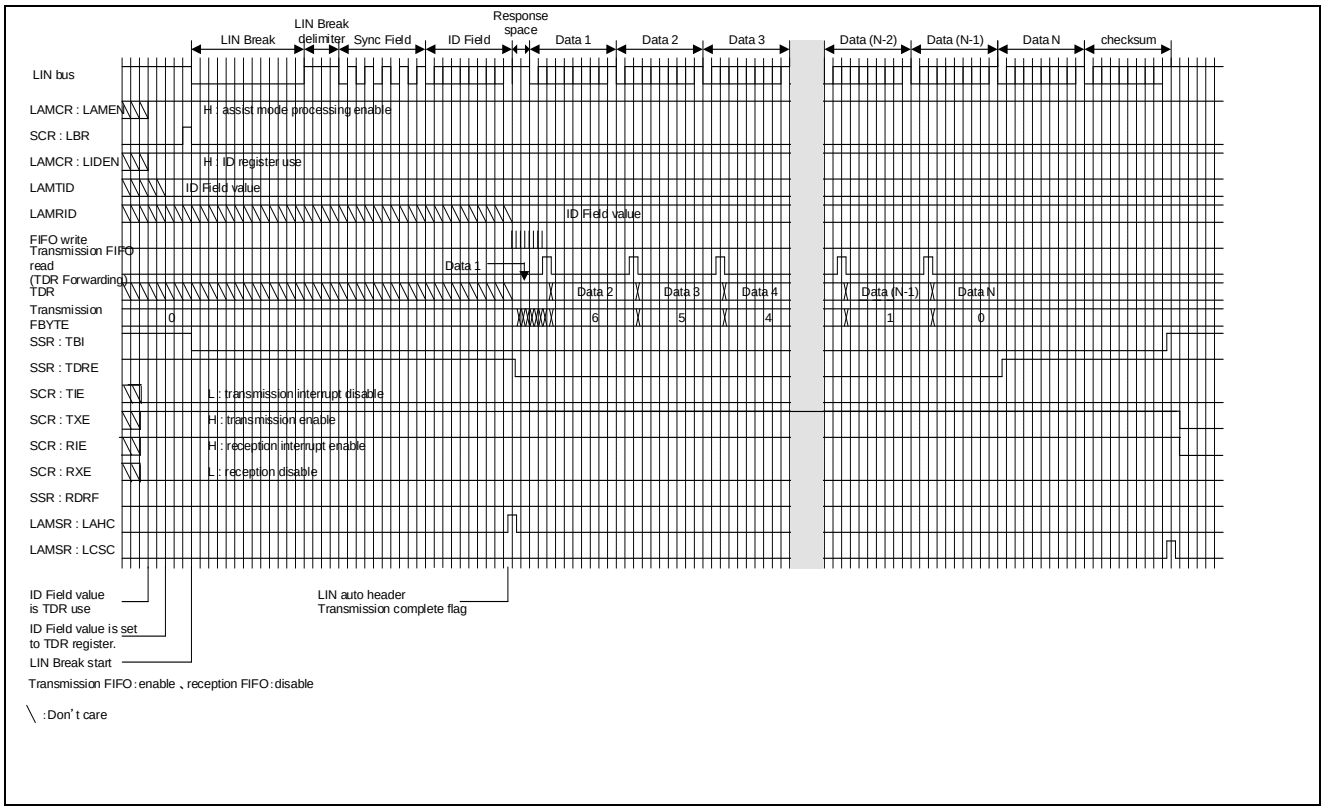


Figure 32-97. LIN Bus Timing (ID Register Unused, DATA Field Transmission, and FIFO Use)

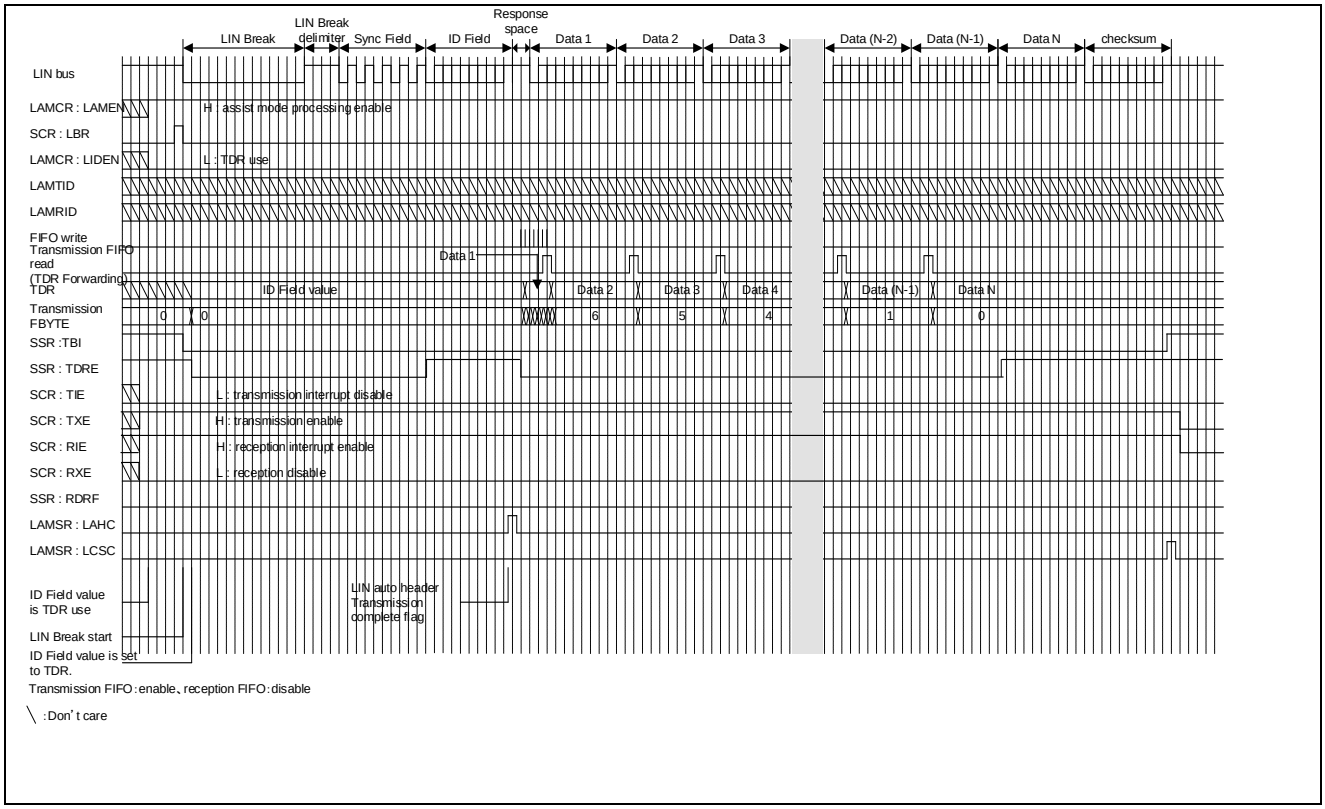


Figure 32-98. LIN Bus Timing (ID Register Use, DATA Field Reception, and FIFO Use)

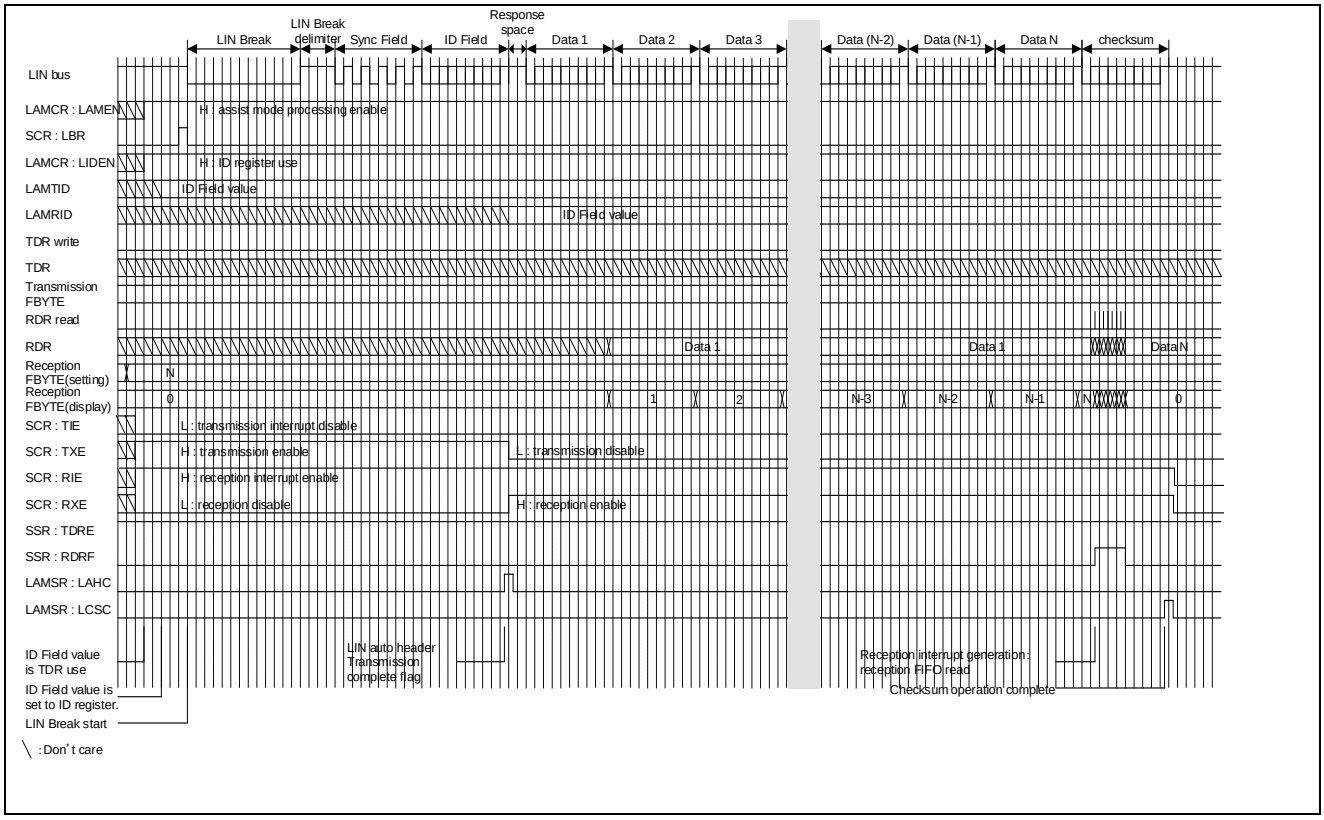
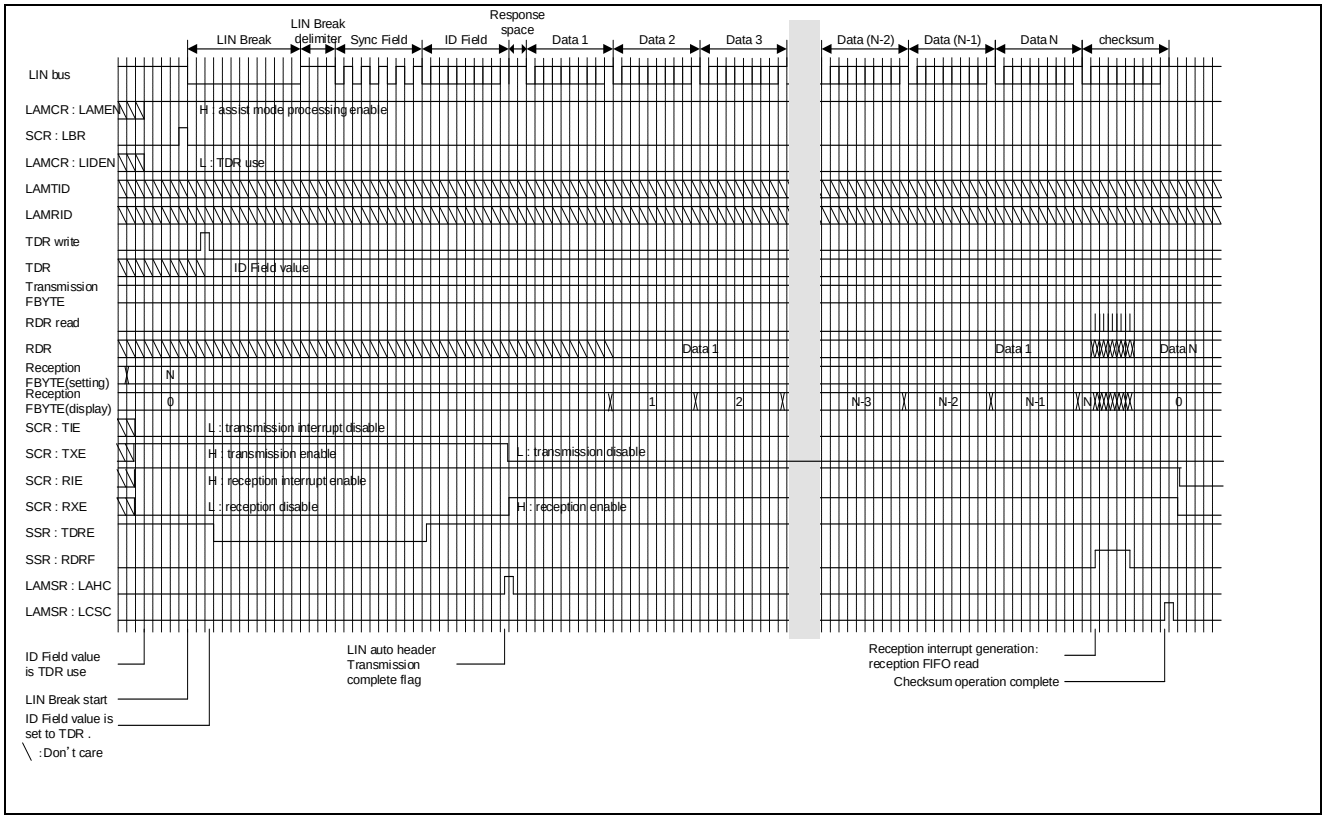


Figure 32-99. LIN Bus Timing (ID Register Unused, DATA Field Reception, and FIFO Use)



Slave operation

Automatic header reception setting

Please set the following to receive an automatic header in the assist mode.

- Please set the SCR:MS bit to "1" to operate as a slave.
- Please set the LAMCR:LAMEN bit to "1" to operate as LIN assist mode.
- The ID Field value is written in LAMRID (LIN assist mode reception ID register) or RDR (receive data register). Please set the LAMCR:LIDEN bit to "1" when you write the value of ID Field in LAMRID. Please set the LAMCR:LIDEN bit to "0" when writing it in RDR.
- Please set the SACSR:AUTE bit to "1" when you do the baud rate adjustment automatically.
- Please set reception enable bit (SCR:RXE) to "1" (reception enable).

From LIN Break Field reception to ID Field reception

1. When LIN Break Field is input, LIN Break Field is detected in the 11th bit (SSR:LBD=1). At this time, if the ESCR:LBIE bit is set to "1", the interrupt is generated. Please set the ESCR:LBIE bit to "0", and prohibit the status interrupt in the LIN assist mode.

Operation when the automatic baud rate is adjusted is shown below.

2. When LIN interface (v2.1) detects the first falling edge of Sync Field, the serial timer register (STMR) is initialized to 0.
3. When the fifth falling edge of Sync Field is detected, sync field detection flag (SACSR:SFD) is set to "1". At this time, if the SACSR:SFDE bit is set to "1", the status interrupt is generated. Please set the ESCR:SFDE bit to "0", and prohibit the status interrupt in the LIN assist mode.

4. When the fifth falling edge of Sync Field is detected, it operates according to the value of the serial timer register (STMR) as follows.

   ☐ When the counter value of the STMR is within the defined range of SFLR and SFUR, the baud rate setting flag (SACSR:BST) is set to "1" and the BGR value adopts the STMR counter value..

   ☐ When the value of the serial timer register (STMR) is smaller than the sync field lower limit register (SFLR) or larger than the sync field upper limit register (SFUR), the value of the baud rate generator register (BGR) will not be changed and the baud rate setting flag (SACSR:BST) will be reset to "0".

Figure 32-100. From LIN Break Field Reception to ID Field Reception (STMR is SFUR or smaller, and SFLR or larger)

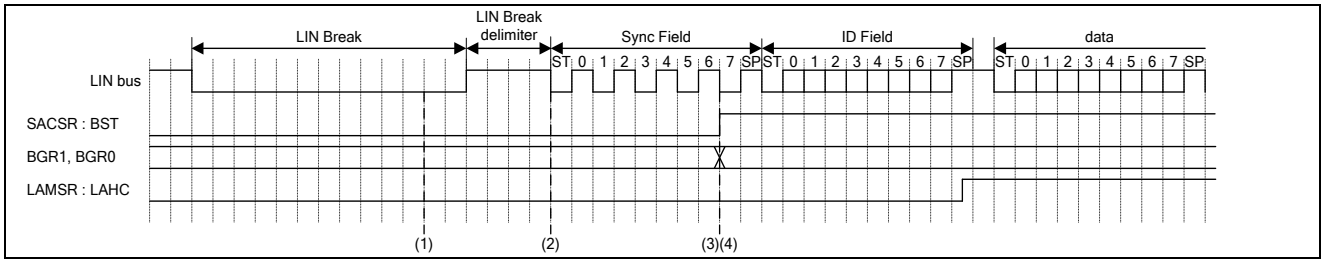
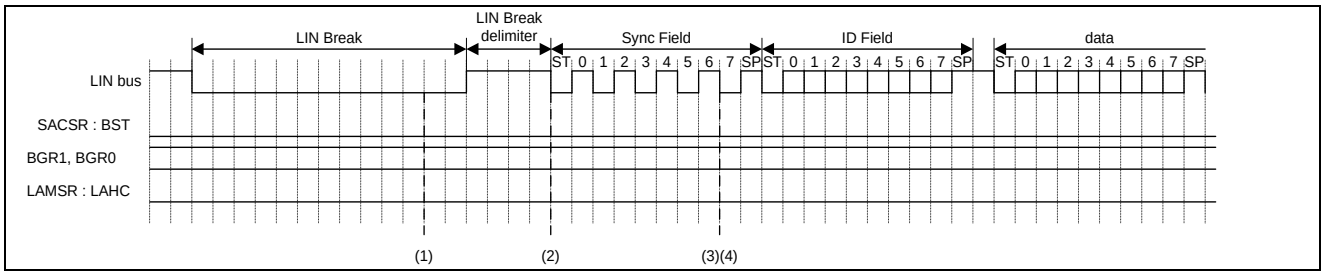


Figure 32-101. From LIN Break Field Reception to ID Field Reception (STMR is smaller than SFUR, and larger than SFLR)



5. When the automatic header reception in the LIN assist mode is completed, the LAMSR:LAHC bit is set to "1". Also when the LIN parity error is generated in ID Field, the LAMSR:LAHC bit becomes "1". Therefore, when the LAMSR:LAHC bit is set to "1", it is necessary to confirm the error has not been detected.

6. Please set LIN data length setting bit (LAMCR:LDL2 to LDL0) when ID Field is normally received.

Figure 32-102. From LIN Break Field Reception to ID Field Reception (Parity Error is Generated)

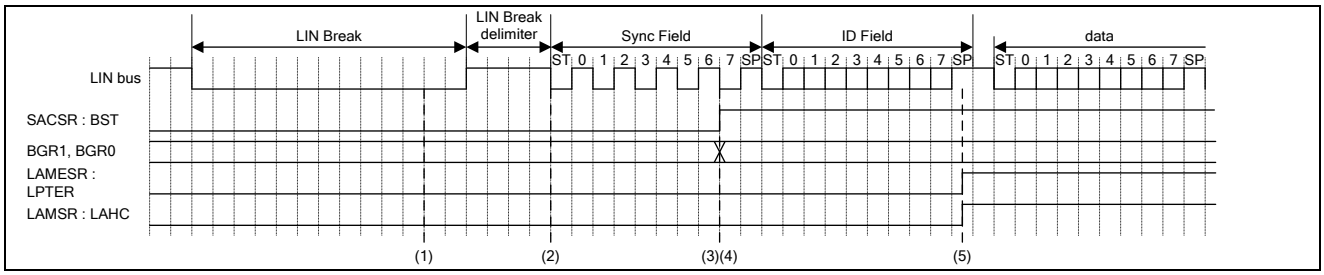
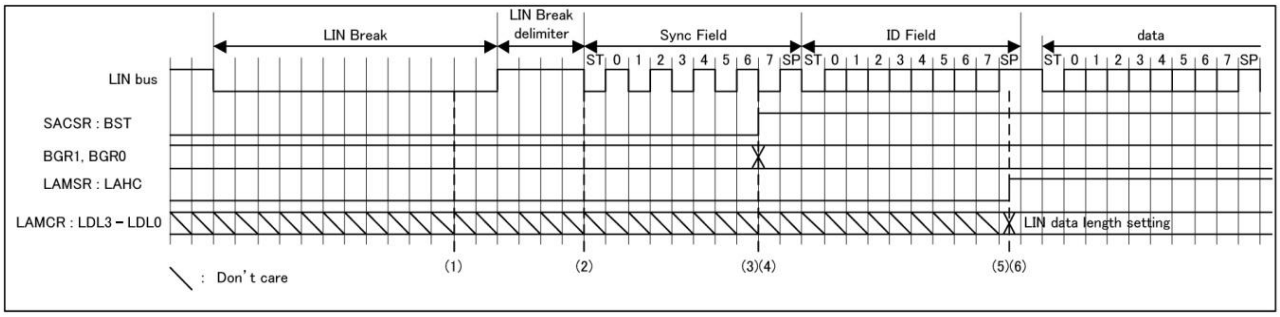


Figure 32-103. From LIN Break Field Reception to ID Field Reception (Set the LIN Data Length)



Notes:

- The setting of reception enable bit (SCR:RXE) and transmission enable bit (SCR:TXE) is disregarded during the header reception of the master in the assist mode.
- However, if the reception enable is set (SCR:RXE=1) when LIN Break Field is received, it is judged that the stop bit is "L" level before LIN Break is detected and detects the framing error. Therefore, please set to reception prohibition setting (SCR:RXE=0) when the header is transmitted.
- The Sync Field value when the assist mode operates cannot be stored in the RDR register.

From ID Field reception to DATA Field transmission/reception

Whether DATA Field is transmitted to or received in the master can be selected after ID Field reception.

(When DATA Field is transmitted)

- Please write data in transmission data register (TDR) after reception ID Field. At this time, please set to transmission interrupt enable (SCR:TIE=1).
- Please set LIN data length setting bit (LAMCR:LDL2 to LDL0) from the value of reception ID Field.
- Checksum is operated based on LIN data length setting bit (LAMCR:LDL2 to LDL0), and after final data is transmitted, checksum is transmitted automatically.
- The arithmetic operations of checksum can select the arithmetic operations method by LIN checksum type selection bit (LAMCR:LCSTYP).
- When the arithmetic operations of checksum is completed, checksum arithmetic operations completion flag (LAMCR:LCSC) is set. At this time, when the checksum arithmetic operations completion interrupt enable bit is set (LAMIER:LCSCIE=1), the status interrupt is generated.
- After the response transmission is completed (LAMSR:LCSC=1), transmission prohibition setting (SCR:TXE=0) is done.

Notes:

- The response transmission data (Data Field and checksum) when the assist mode operates cannot be stored in the RDR register.
- Please write the dummy value ("don 't care") in the TDR register to operate checksum automatically and transmit it when the LIN data length is set to 0 byte length (LAMCR:LDL3-0="0000") in response transmission. The TDR setting value at this time doesn't influence the checksum arithmetic operations.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - ☐ When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - ☐ When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes an inverted ID Field.

Figure 32-104. From ID Field Reception to DATA Field Transmission (ID Register Use).

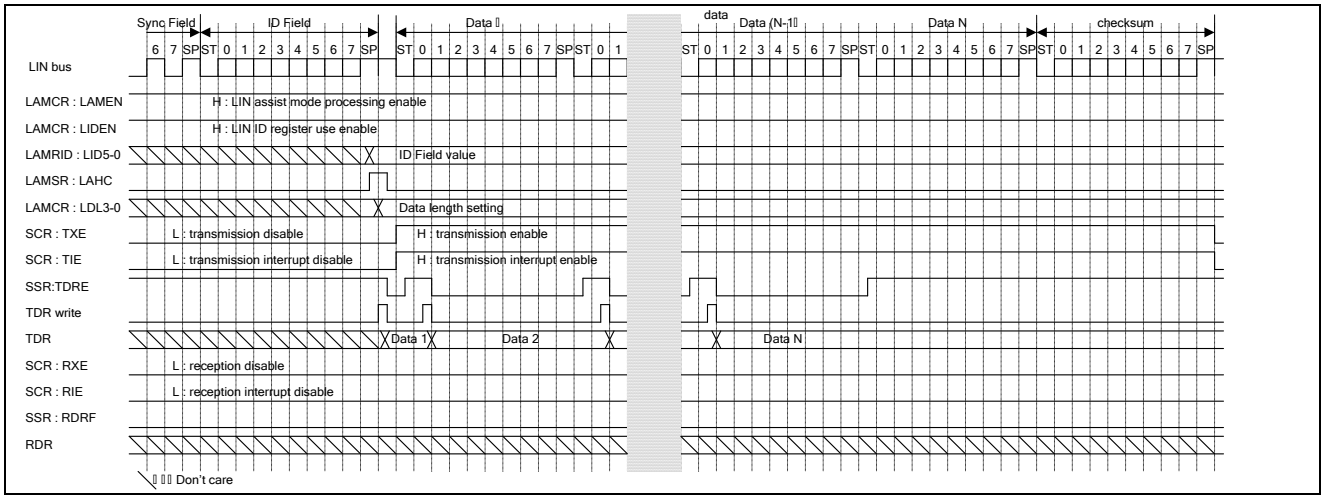
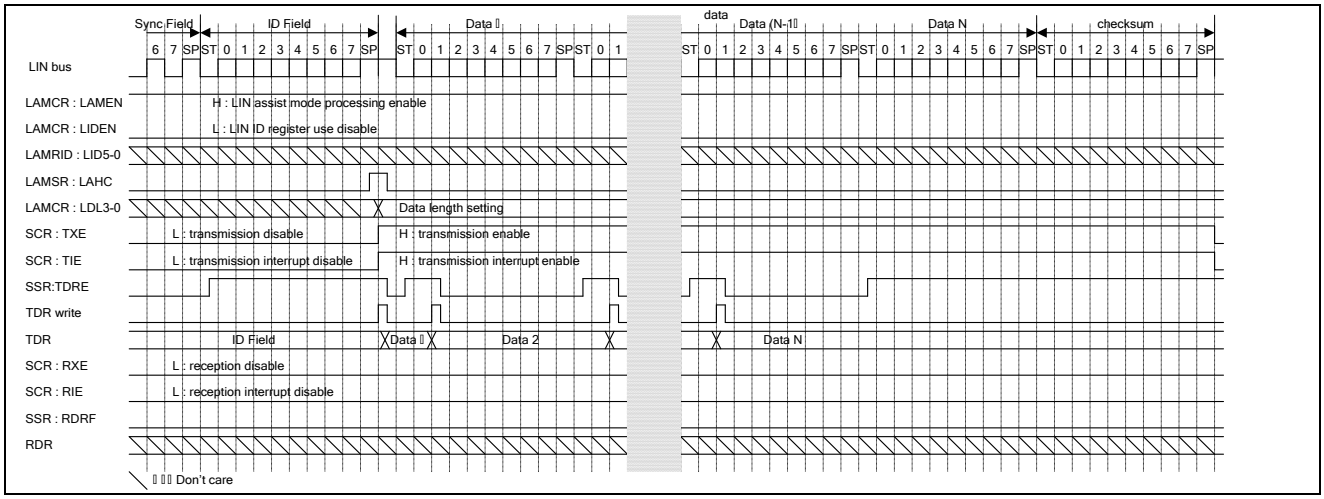


Figure 32-105. From ID Field Reception to DATA Field Transmission (ID Register Unused).



(When receive DATA Field)

- Please set LIN data length setting bit (LAMCR:LDL2 to LDL0) from the value of reception ID Field.
- Please set the reception enable (SCR:RXE=1).
- SSR:RDRF is set to "1" at each DATA Field reception. At this time, if reception interrupt enable (SCR:RDIE=1) is done, the reception interrupt is generated.
- Detection of the start bit is as follows; the falling edge is detected after passing through the noise filter (which samples serial data input in 3 bus clock and decides the value by majority), and the data "L" is detected after the noise filter at the sampling point.
- Checksum is operated based on LIN data length setting bit (LAMCR:LDL3 to LDL0), and the normality of the reception checksum is confirmed automatically. The arithmetic operations result of checksum can be confirmed by LIN checksum error flag bit (LAMESR:LCSER). When LCSER is "1", the checksum error is detected. At this time, when LIN checksum error interrupt enable bit (LAMIER:LCSERIE) is "1", the interrupt is generated.

- When the checksum arithmetic operations is completed, the checksum arithmetic operations completion flag bit (LAMSR:LCSC) becomes "1". At this time, when LIN checksum arithmetic operations completion interrupt enable bit (LAMIER:LCSCIE) is "1", the interrupt is generated.
- After the checksum reception is completed (LAMSR:LCSC=1), reception prohibition setting (SCR:RXE=0) is done.

Figure 32-106. From ID Field Reception to DATA Field Reception (ID Register Use).

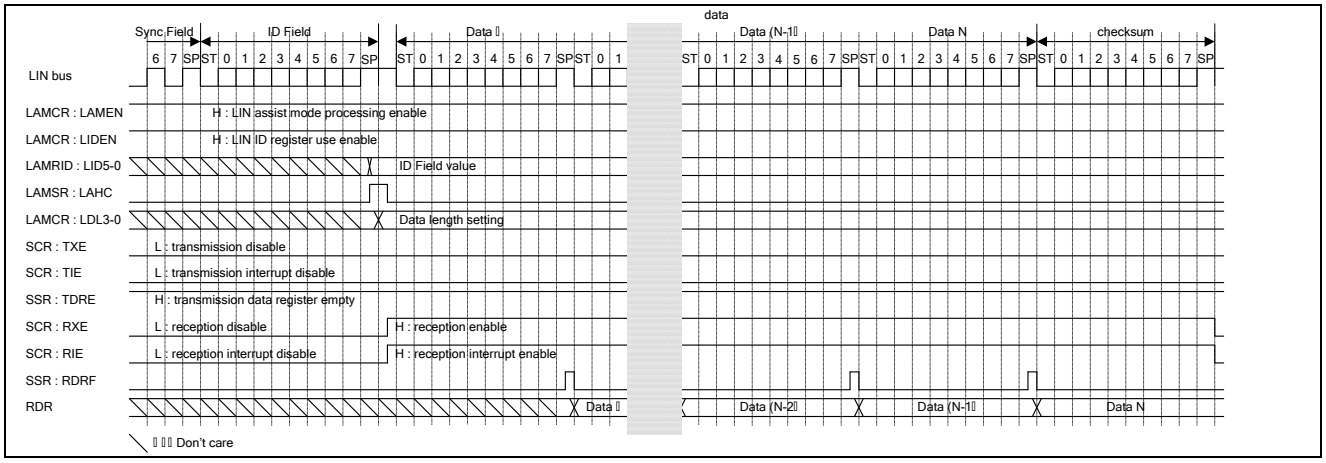
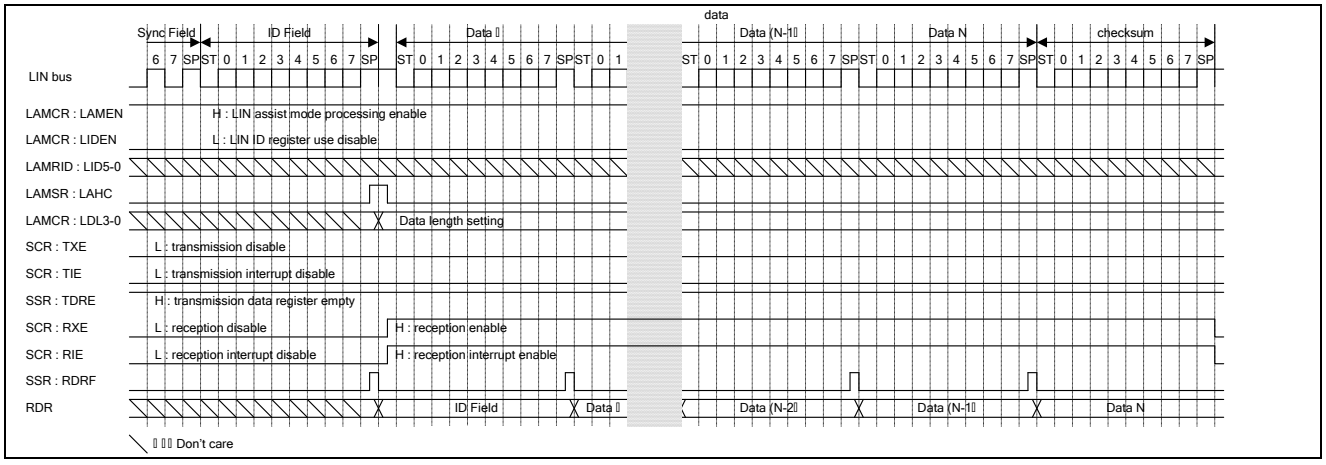


Figure 32-107. From ID Field Reception to DATA Field Reception (ID Register Unused).



Notes:

- Although the noise filter (where the serial data input is sampled three times with the bus clock and decided by majority) is built in, design the board so that the noise should not pass this filter or communicate so that noise passing may not become a problem (for instance, adding data checksum at the end and retransmitting the Data Field if an error occurs).
- If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two bus clocks during reception, the edge becomes invalid and it becomes impossible to receive the next frame normally. If successive frames are to be output, leaving a space between frames is recommended.
- The checksum value of the response reception is not stored in the RDR register when the assist mode operates.
- The checksum value becomes the following when the LIN data length is set by 0 byte length (LAMCR:LDL3-0="0000").
  - ☐ When the standard checksum is set (LAMCR:LCSTYP=0), the checksum value becomes 0xFF.
  - ☐ When the expanded checksum is set (LAMCR:LCSTYP=1), the checksum value becomes reversing ID Field.

LIN Break Field reception processing in assist mode

After LIN Break Field is detected (SSR:LBD=1), the following procedures are needed for the LIN frame of retransmitted LIN Break Field when the assist mode is processed (SSR:RDRF=1 or SCR:TXE=1 or SSR:TBI=0).

- First of all, reception prohibition setting (SCR:RXE=0) and transmission prohibition setting (SCR:TXE=0) are done.
- Cancels the reception data before it is received again.
  - ☐ When reception FIFO is used, the reception FIFO is reset (FCR0:FCL1=1 or FCR0:FCL2=1) after reception FIFO operation is prohibited (FCR0:FE1=0 or FCR0:FE2=0).
  - ☐ Then, to clear the reception data register, the RDR register is read.
- Cancels the transmission data before it is received again.
  - ☐ When transmission FIFO is used, the transmission FIFO is reset (FCR0:FCL1=1 or FCR0:FCL2=1) after transmission FIFO operation is prohibited (FCR0:FE1=0 or FCR0:FE2=0).
  - ☐ Then, the transmission data register clear is executed (LAMCR:LTDRCL=1), and the state is made the transmission bus idle.
- When the automatic header reception in the LIN assist mode completes, the LAMSR:LAHC bit is set to "1". Moreover, please confirm neither the LIN parity error nor the framing error has been detected with ID Field.
- Please set the LIN data length setting bit (LAMCR:LDL3-0) when ID Field is normally received.
- The following processing is equal to that of the preceding section "From ID Field Reception to Data Field Transmission/Reception."

Notes:

- In LIN Break Field received, the framing error is detected before LIN Break Field is detected when reception enable setting (SCR:RXE=1) is done. However, it operates normally without stopping the header reception.
- In the assist mode (LAMCR:LAMEN), the framing error is detected at "L" level of the tenth bit of new LIN Break Field regardless of reception prohibition setting (SCR:RXE=0) when new LIN Break is transmitted continuously from the master between from the detection of LIN Break Field to the ID Field reception completion. However, it operates normally without stopping the header reception.

Slave operation timing chart

Figure 32-108. LIN Bus Timing (DATA Field Transmission: FIFO Unused, AUTE=1 and ID Register Use).

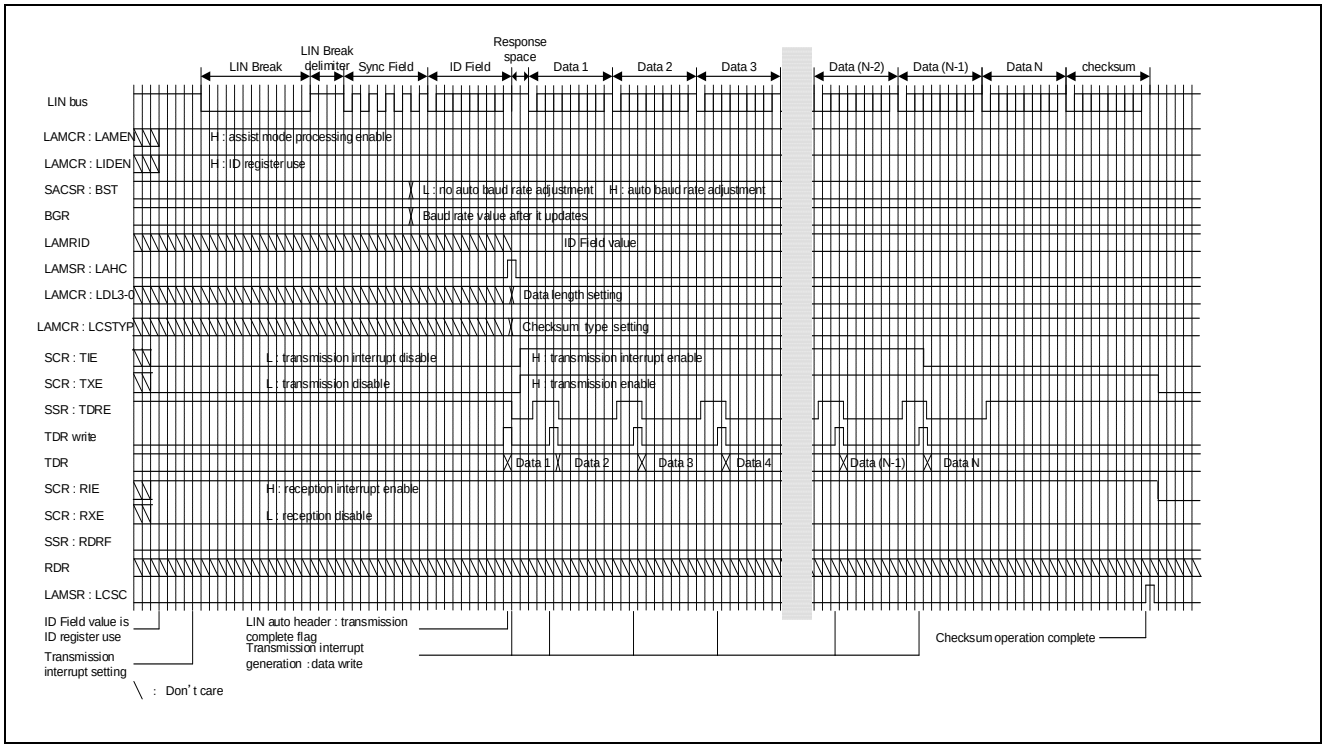


Figure 32-109. LIN Bus Timing (DATA Field Transmission: FIFO Unused, AUTE=1 and ID Register Unused).

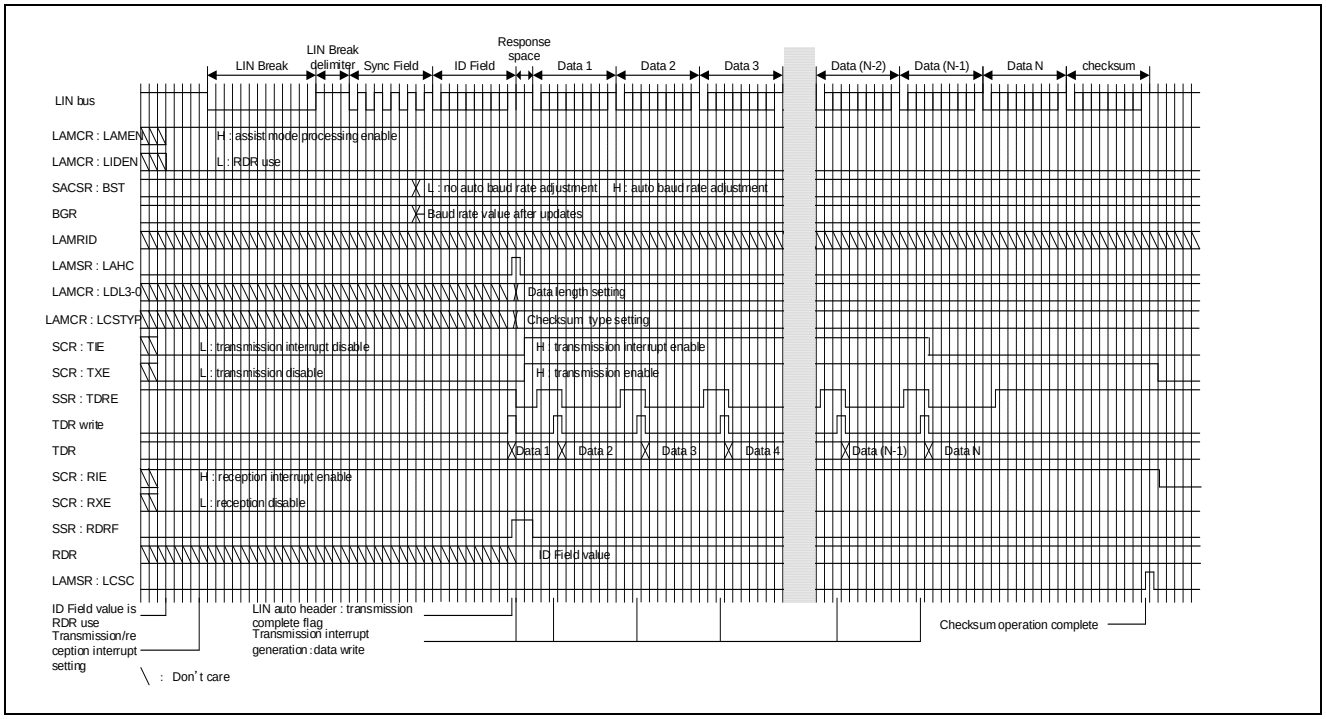


Figure 32-110. LIN Bus Timing (DATA Field Reception: FIFO Unused, AUTE=1 and ID Register Use).

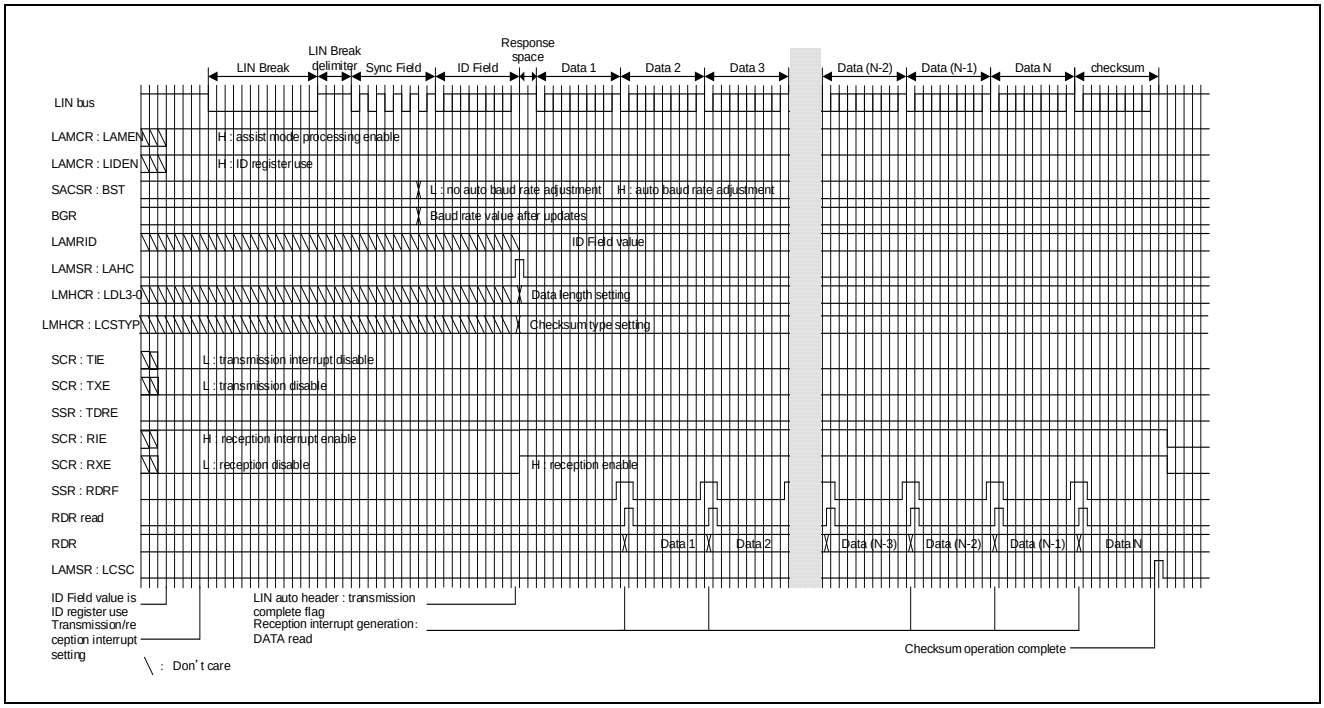


Figure 32-111. LIN Bus Timing (DATA Field Reception: FIFO Unused, AUTE=1 and ID Register Unused).

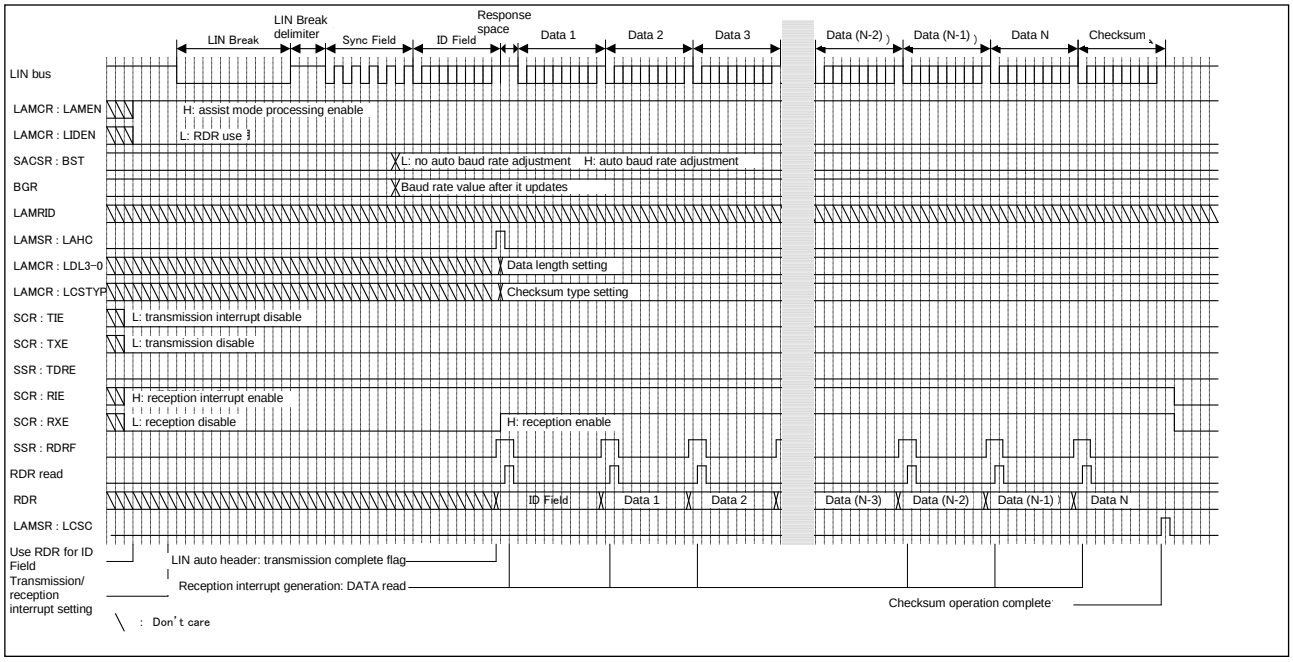


Figure 32-112. LIN Bus Timing (DATA Field Transmission: FIFO Use, AUTE=1 and ID Register Use).

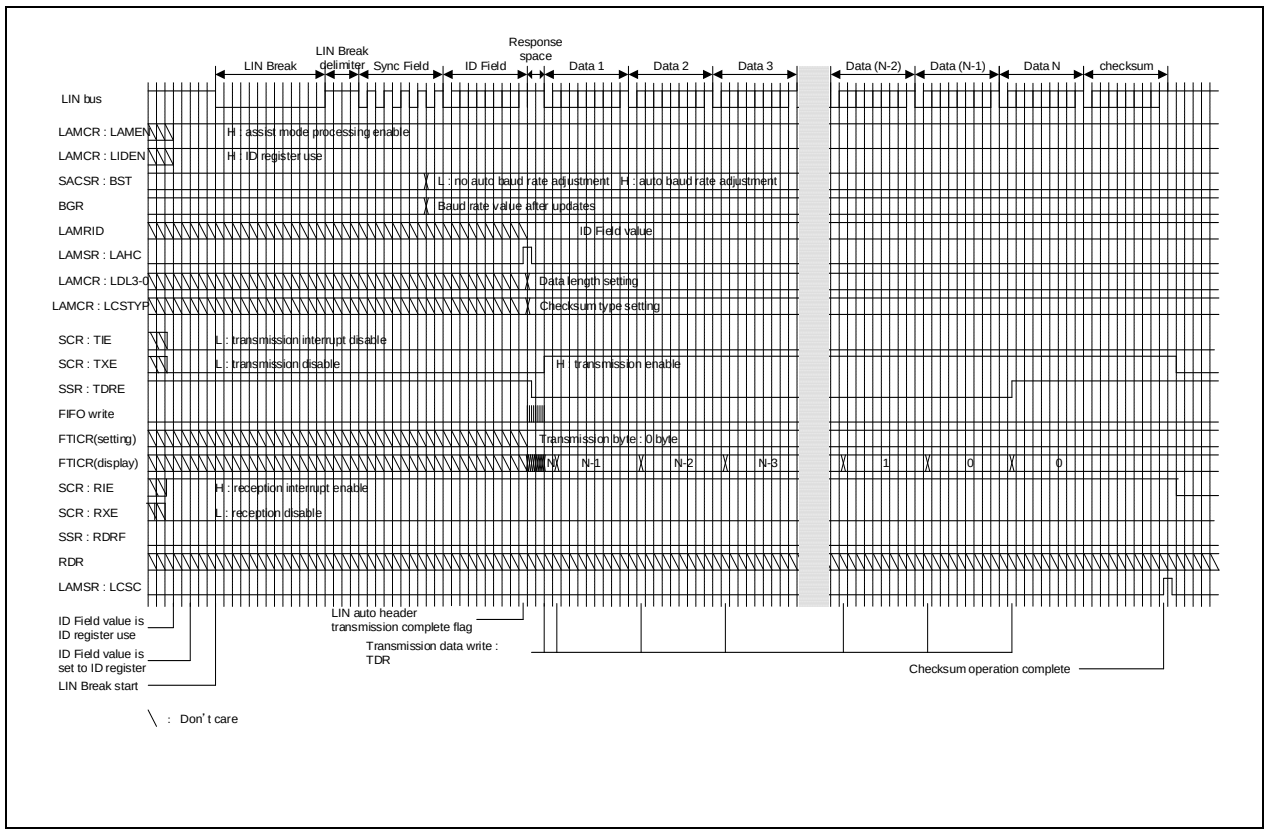


Figure 32-113. LIN Bus Timing (DATA Field Transmission: FIFO Use, AUTE=1 and ID Register Unused).

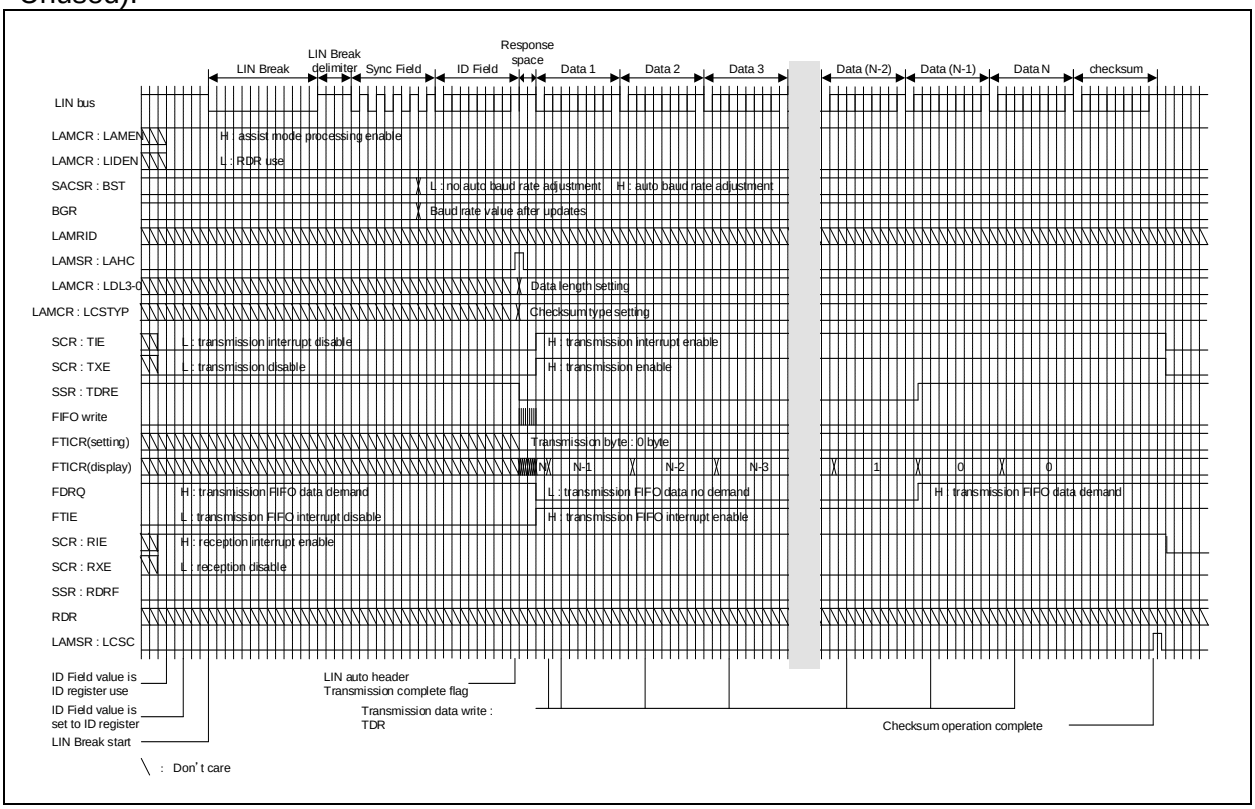


Figure 32-114. LIN Bus Timing (DATA Field Reception: FIFO Use, AUTE=1 and ID Register Use).

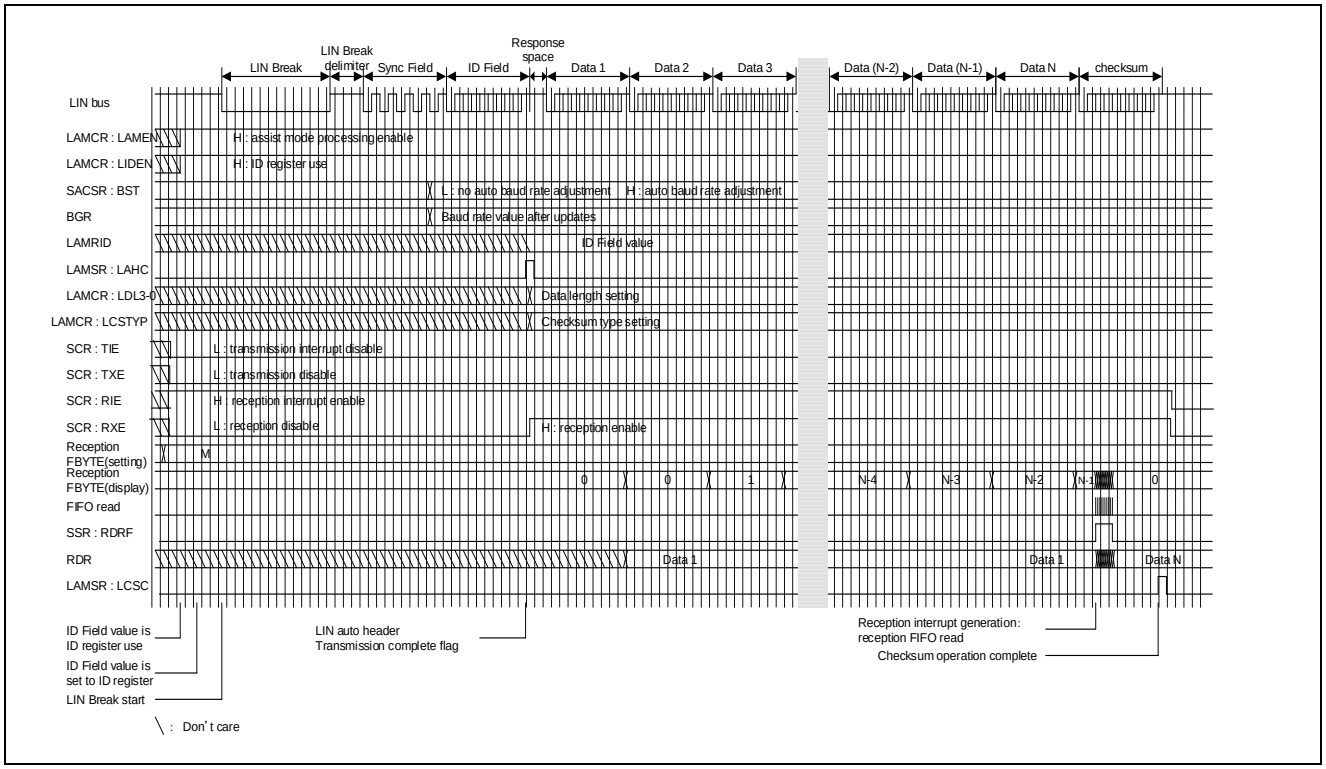
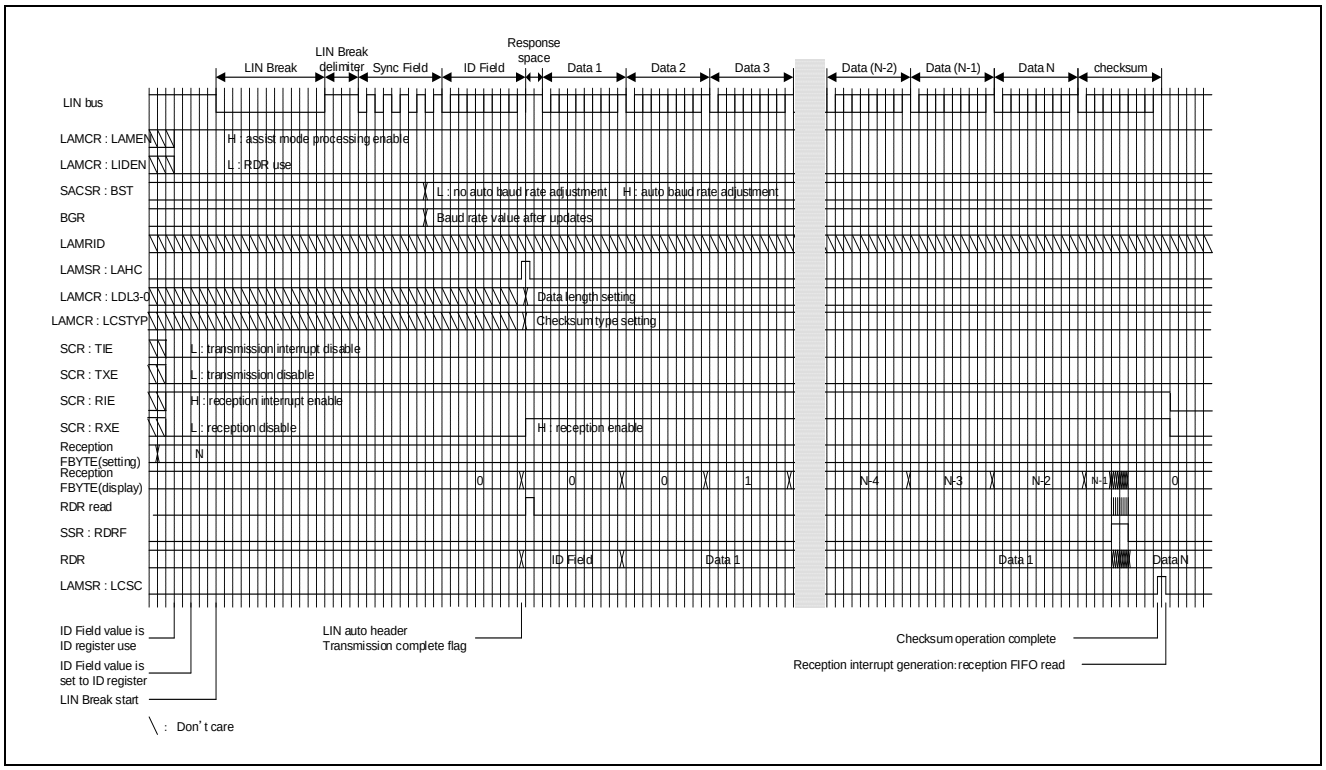


Figure 32-115. LIN Bus Timing (DATA Field Reception: FIFO Use, AUTE=1 and ID Register Unused).



32.7.5.3 LIN Baud Rate Selection/Setting

This section explains the LIN baud rate selection/setting.

The LIN can use:

- Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the internal clock
- Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the external clock

The setting method is the same as the method used in the case of UART (mode 0/1). See "32.5.2.12 UART Baud Rate Selection/Setting."

32.7.6 Setup Procedure and Program Flow

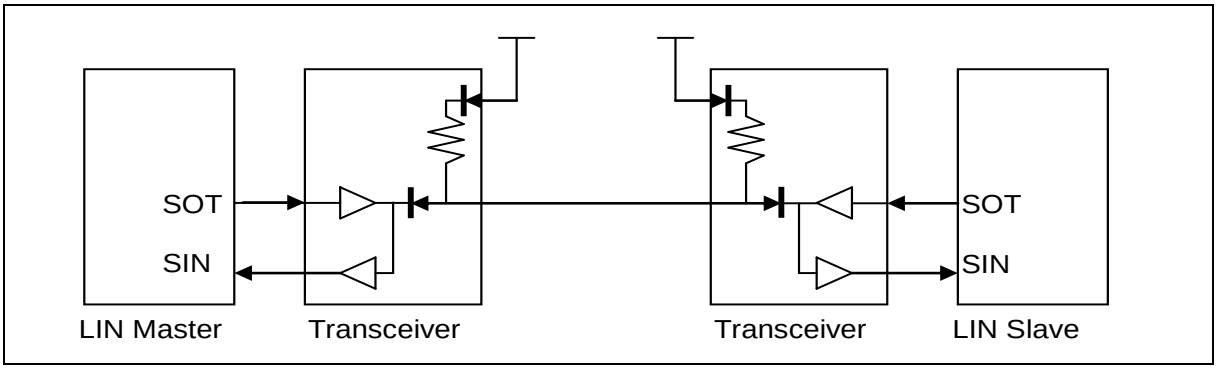
Setup procedure and program flow are shown.

In operation mode 3 (LIN communication mode), the selected baud rate can be used for the LIN master system or LIN slave system.

CPU Interconnection

The following Figure shows a communication system that contains one LIN master and one LIN slave. The multi-function serial interface can work as an LIN master or LIN slave.

Figure 32-116. Example of LIN Bus System Communication



32.7.6.1 Manual mode

This section explains the manual mode.

The example of the flowchart of the master side and the slave side in the manual mode is shown.

Flowchart Example

Master operation

Figure 32-117. Example of a Flowchart in LIN Communication Master Mode (without Using FIFO)

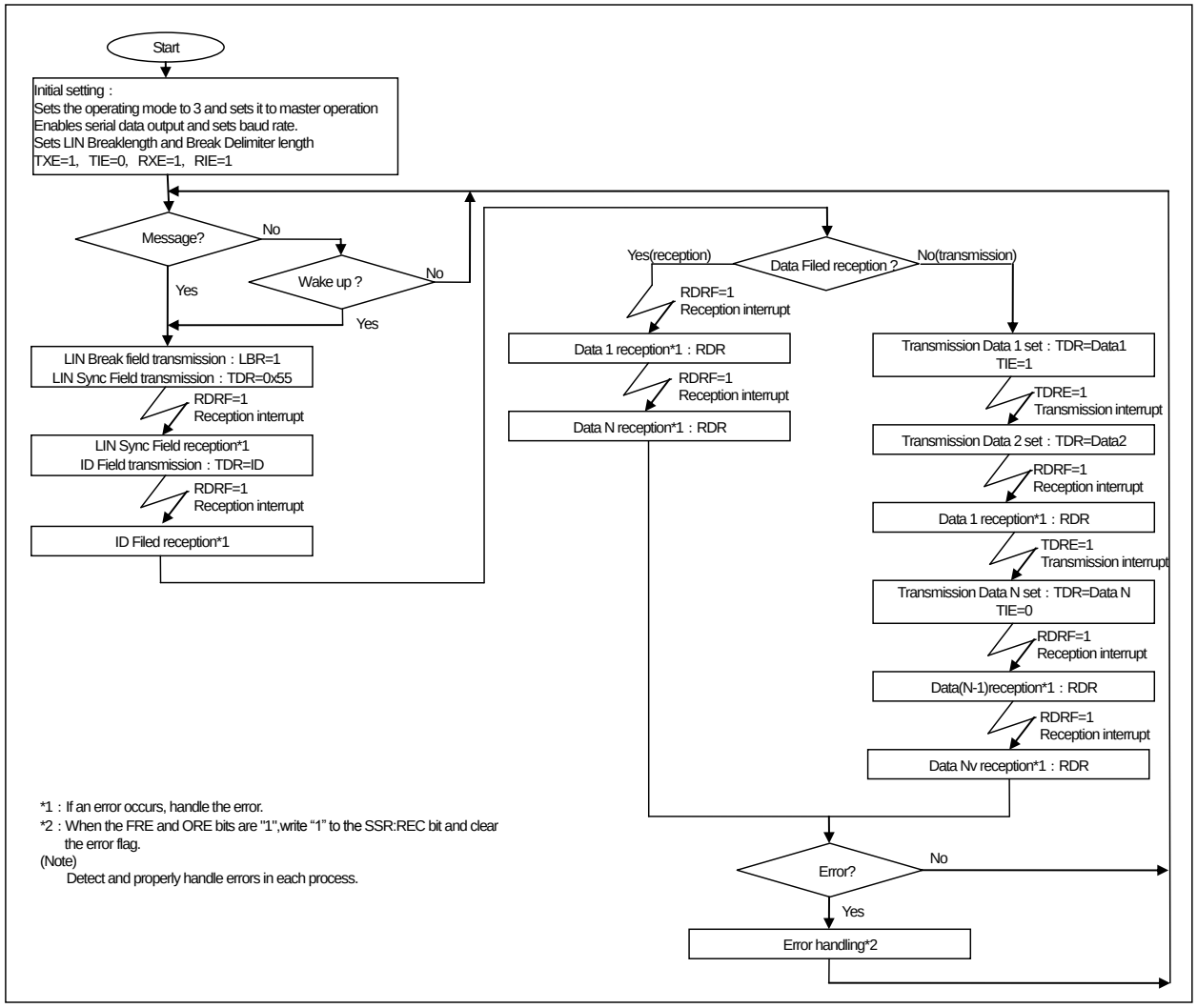
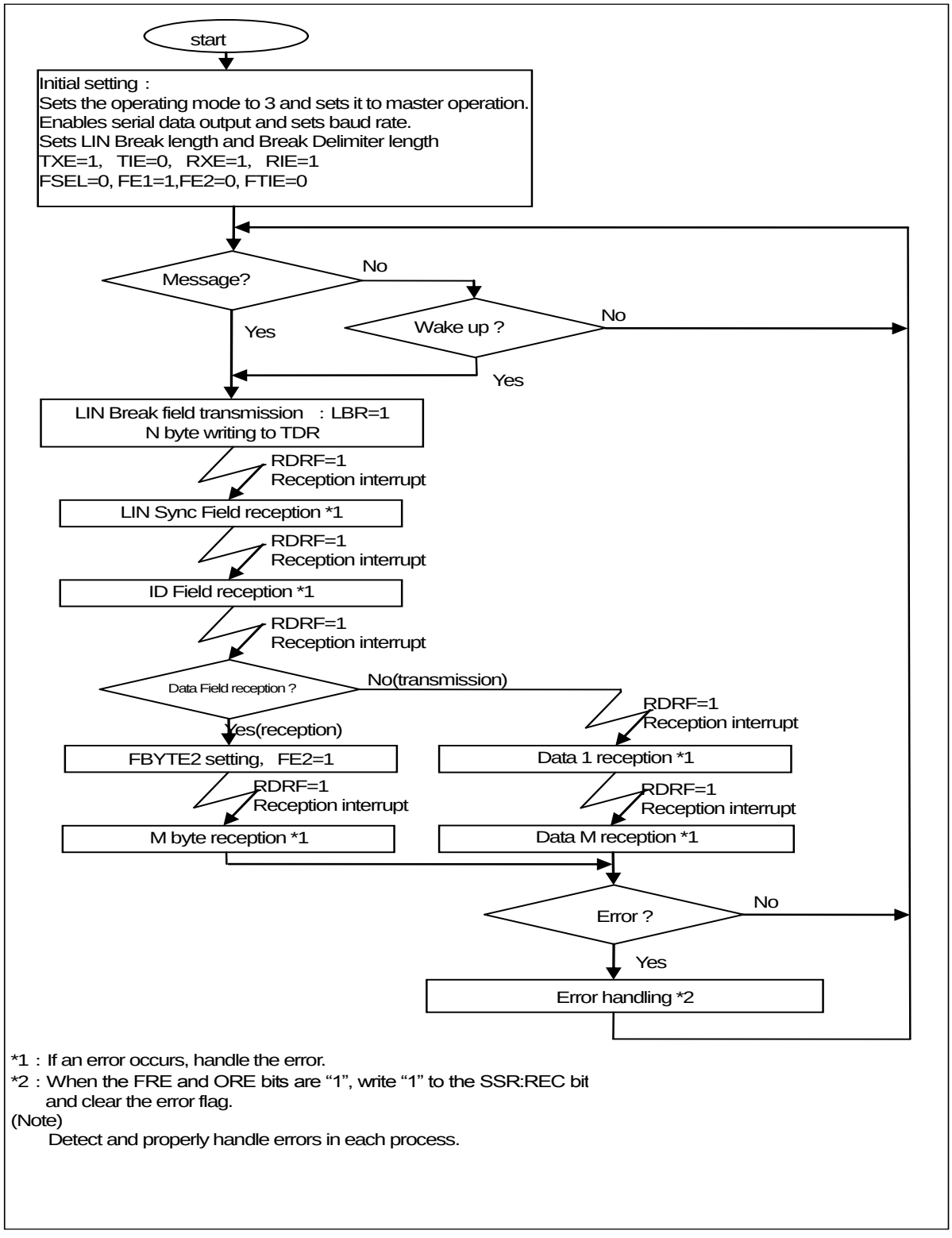


Figure 32-118. Example of a Flowchart in LIN Communication Master Mode (Using FIFO)



*1 : If an error occurs, handle the error.
*2 : When the FRE and ORE bits are “1”, write “1” to the SSR:REC bit and clear the error flag.
(Note)
Detect and properly handle errors in each process.

Slave operation

Figure 32-119. Example of a Flowchart in LIN Communication Slave Mode

(Automatic Baud Rate can be adjusted (SACSR:AUTE=1), without Using FIFO)

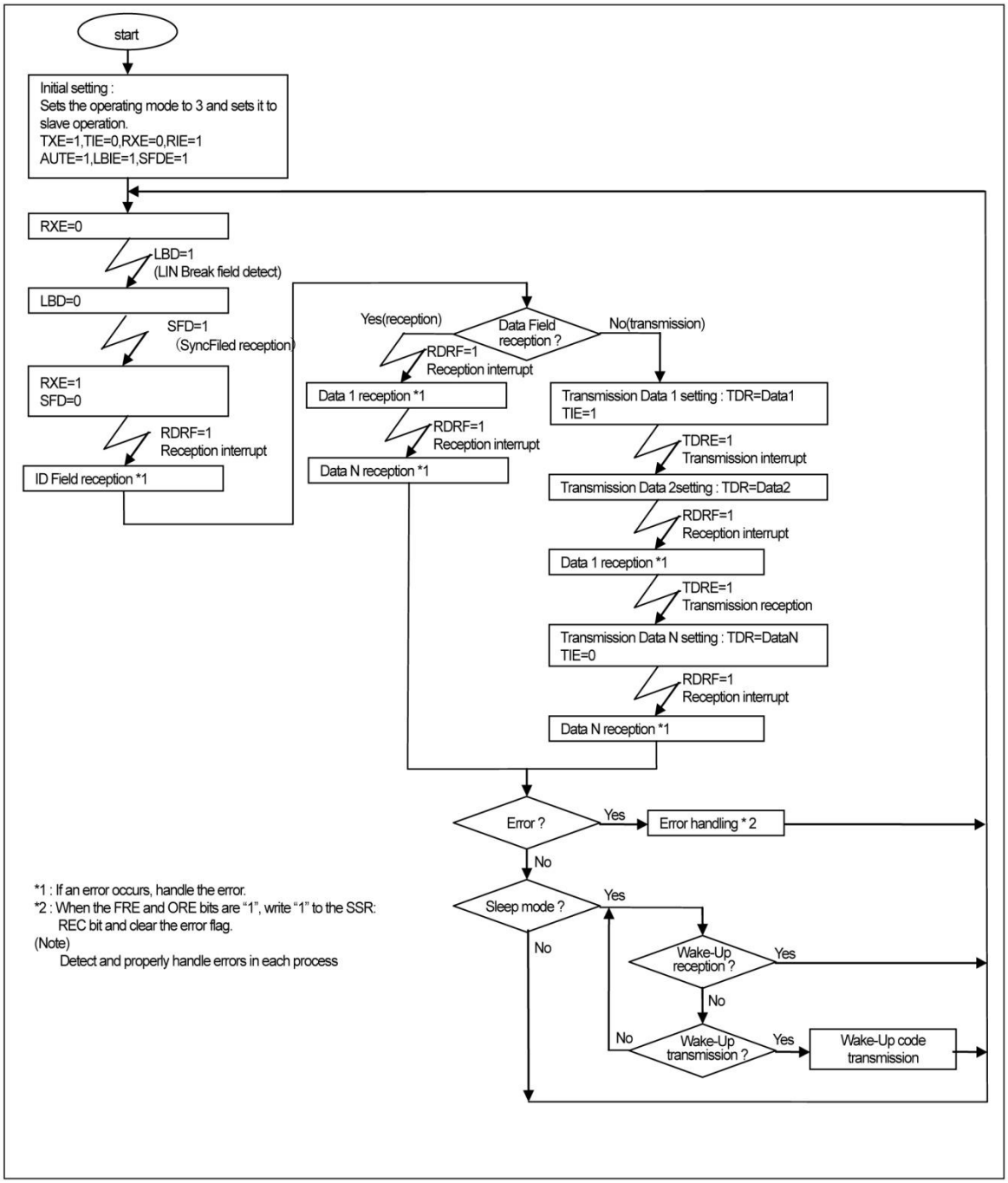
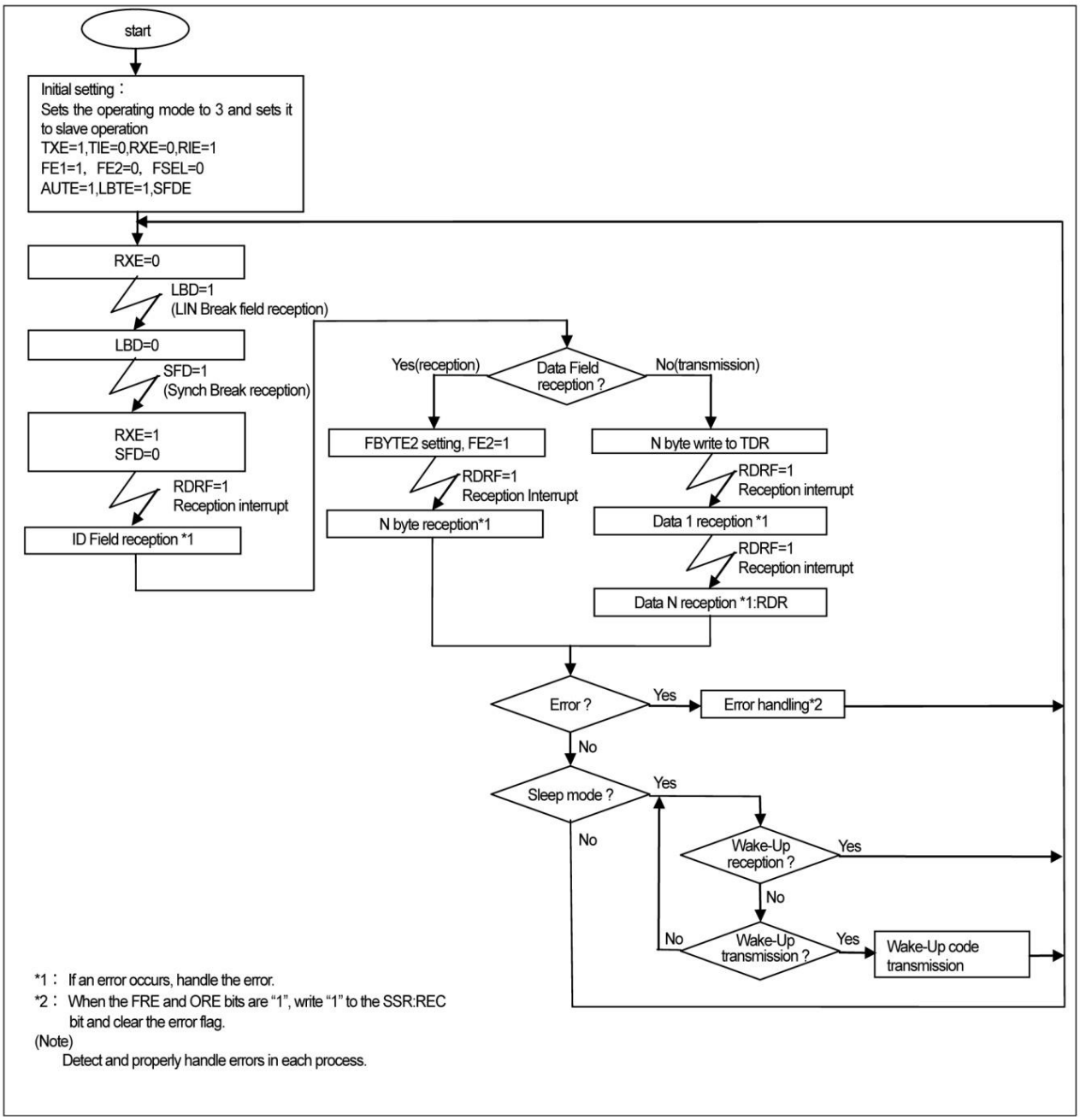


Figure 32-120. Example of a Flowchart in LIN Communication Slave Mode
(Automatic Baud Rate can be adjusted (SACSR:AUTE=1), Using FIFO)



32.7.6.2 Assist mode

This section explains the assist mode.

The example of the flow chart of the master side and the slave side in the assist mode is shown.

Flowchart Example

Master operation

Figure 32-121. Example of a Flowchart in LIN Communication Master Mode

(Assist Mode, without Using FIFO)

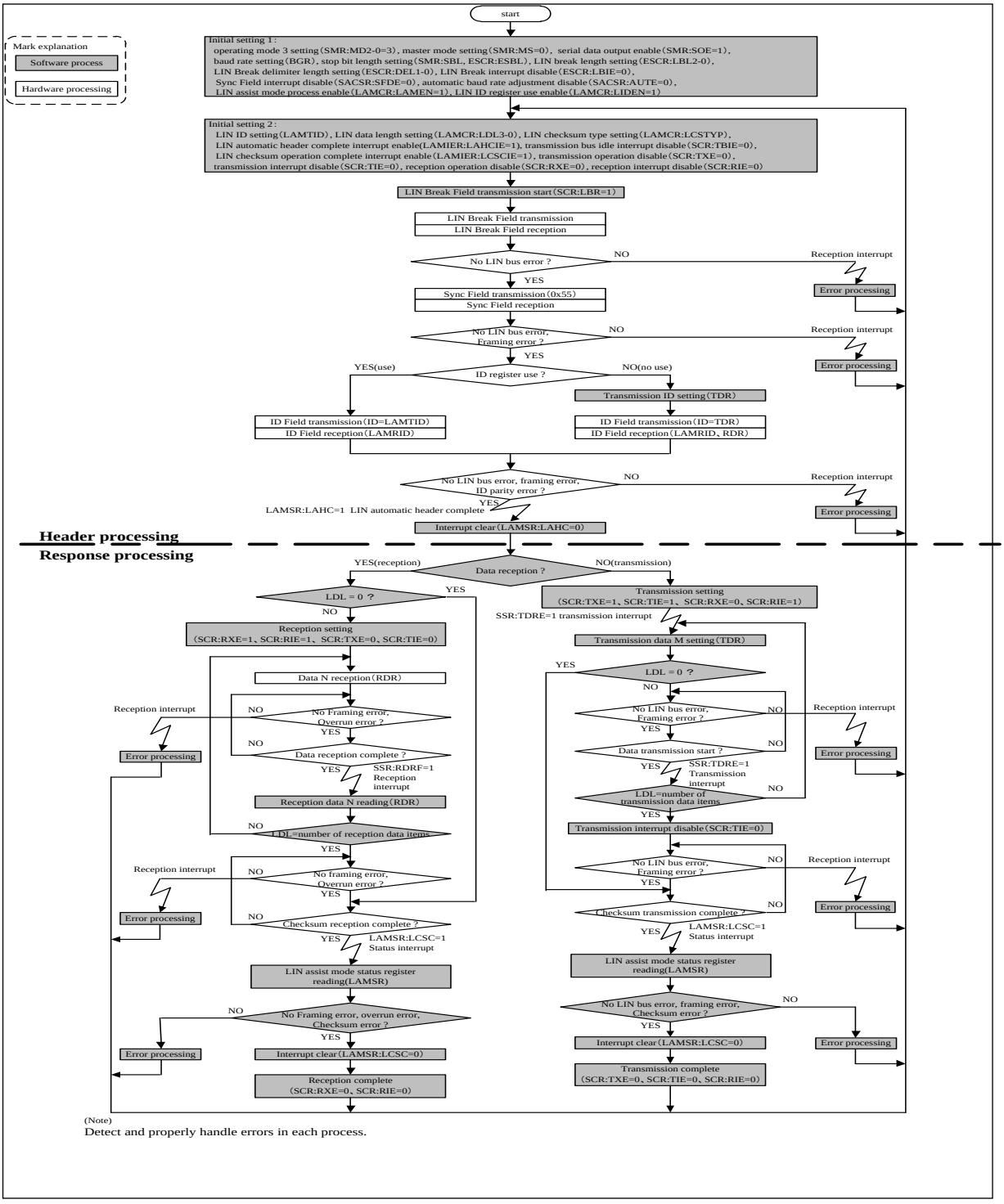
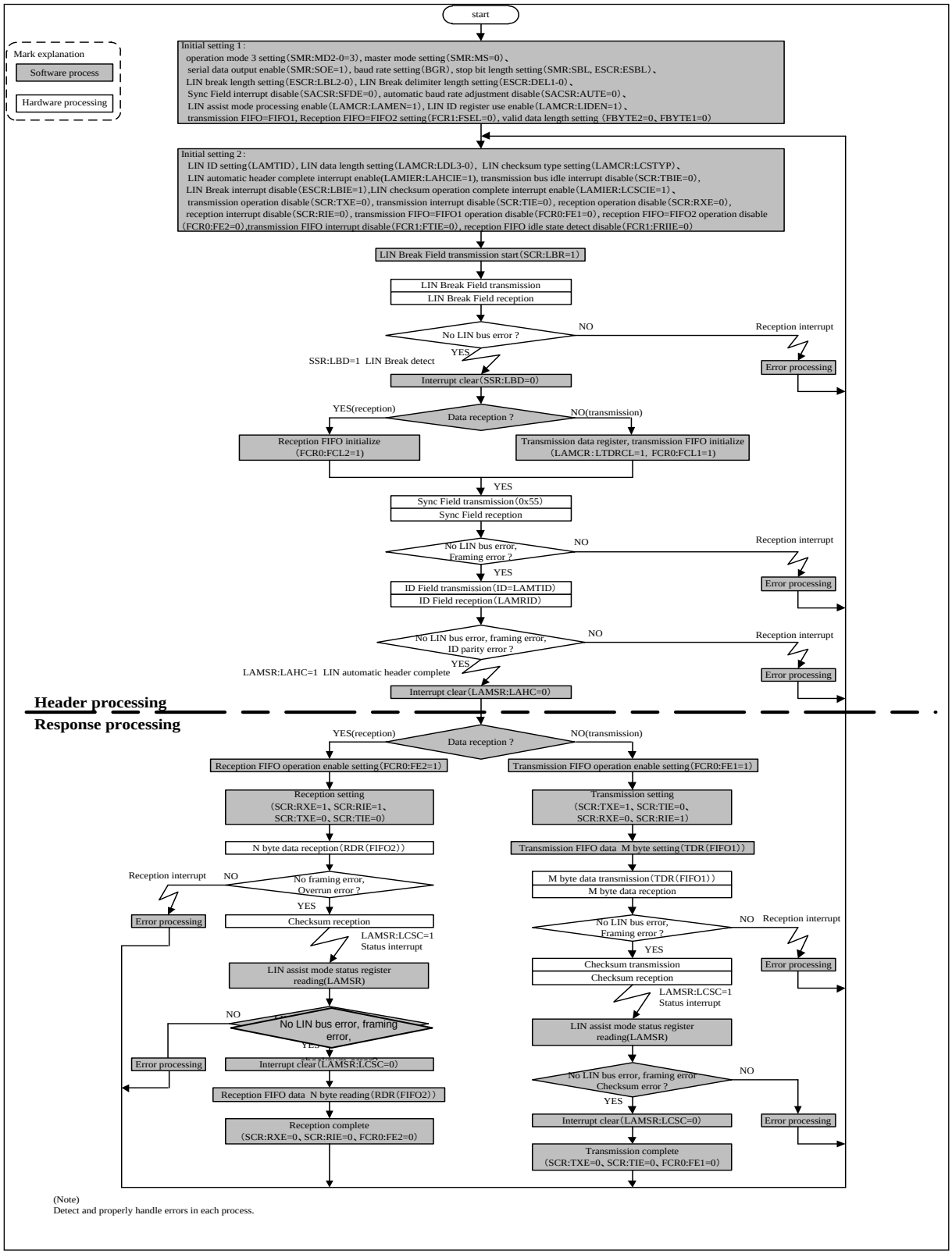


Figure 32-122. Example of a Flowchart in LIN Communication Master Mode

(Assist Mode, Using FIFO)



Slave operation

Figure 32-123. Example of a Flowchart in LIN Communication Slave Mode

(Assist Mode, without Using FIFO)

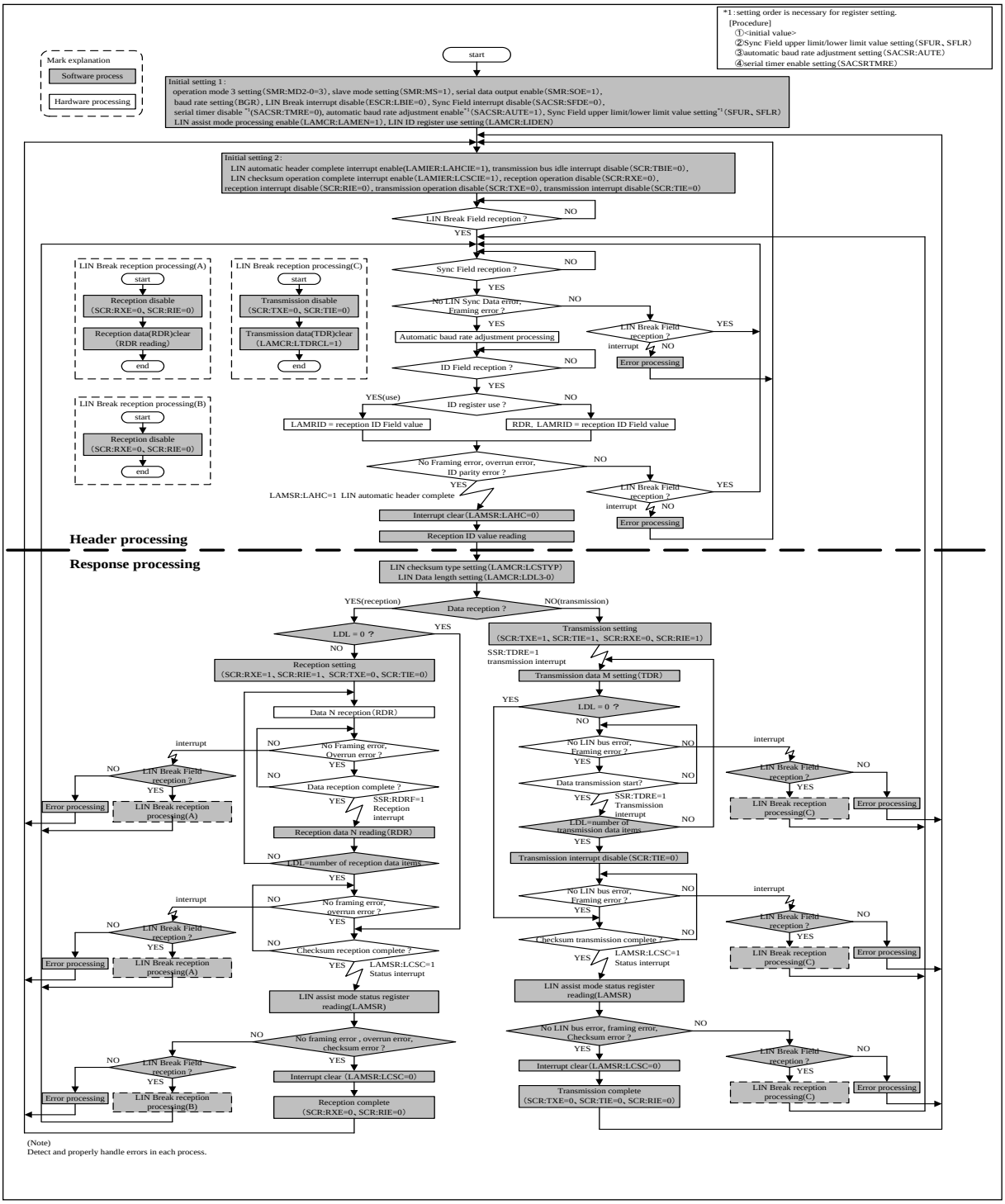
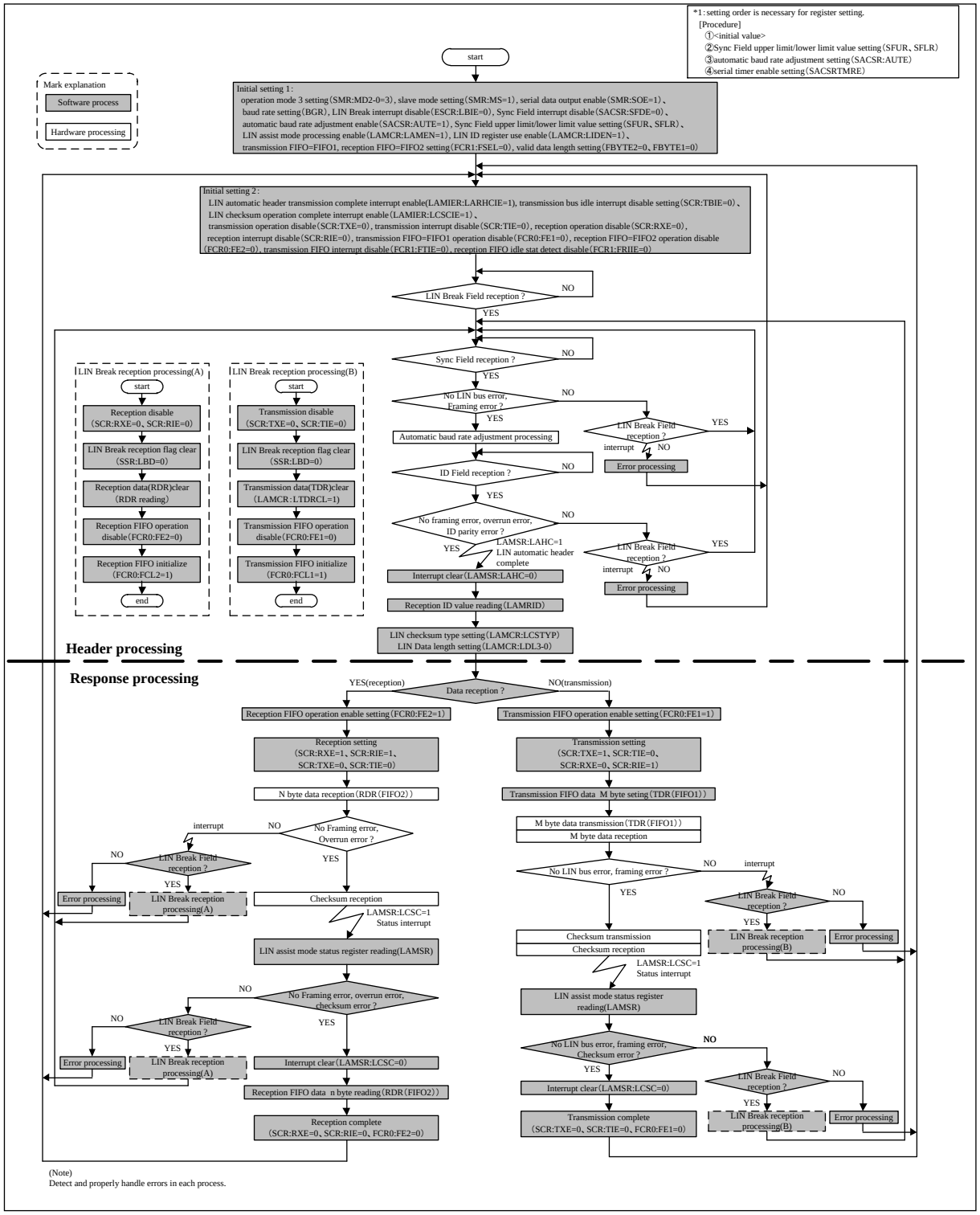


Figure 32-124. Example of a Flowchart in LIN Communication Slave Mode

(Assist Mode, Using FIFO)



33. CAN Controller

This chapter explains the CAN.

33.1 Overview
33.2 Features
33.3 Configuration
33.4 Registers
33.5 Operation
33.6 Limitations

33.1 Overview

This section explains the overview of the CAN.

This series includes 1 CAN channel.

CAN is based on the CAN protocol ver. 2.0A/B which is a standard protocol for serial communication, and is widely used for automobiles, factory automation, and other industrial fields.

33.2 Features

This section explains the features of the CAN.

The CAN of this series has the following features.

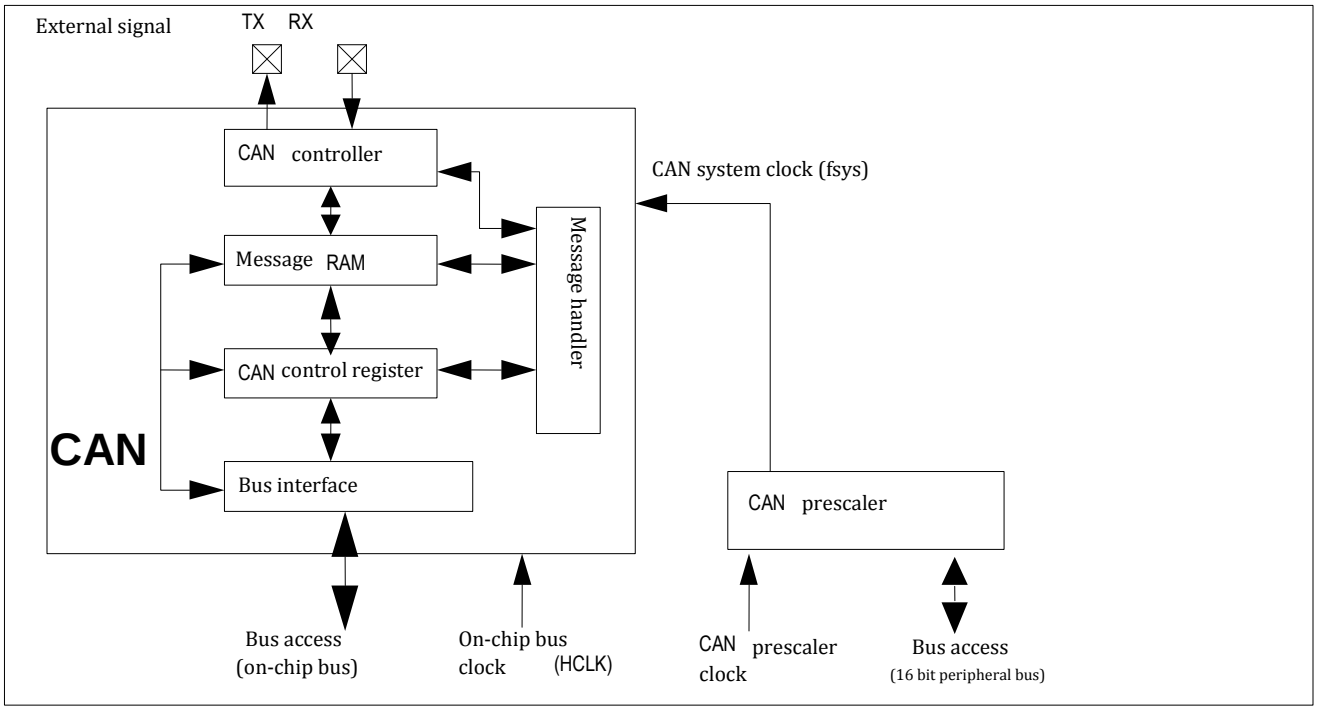
- CAN protocol ver. 2.0A/B is supported.
- Bit rates up to 1 Mbits/s are supported.
- An identification mask is applied to each message object.
- Supports programmable FIFO mode (a chain of message objects).
- Maskable interrupts.
- Programmable loopback mode for self-test operation is supported.
- Read and write from/to the message object using message interface registers.
- Support 64 message objects.

33.3 Configuration

This section explains the configuration of the CAN.

A block diagram of the CAN is shown below.

Figure 33-1. Block Diagram of CAN (for 1 Channel)



CAN controller

The CAN controller controls the CAN protocol and shift registers for serial/parallel conversion to transfer the transmission/reception message.

Message RAM

Stores message objects.

Message handler

Controls the message RAM and CAN controller.

CPU interface

Controls the interface with the FR internal bus.

CAN prescaler

Generates CAN system clocks (fsys).

33.4 Registers

This section explains the registers of the CAN.

33.4.1 Overview

This section shows the overview of the registers.

The CAN includes the following registers.

- CAN control register (CTRLR)
- CAN status register (STATR)
- CAN error counter (ERRCNT)
- CAN bit timing register (BTR)
- CAN interrupt register (INTR)
- CAN test register (TESTR)
- CAN prescaler extension register (BRPER)
- IFx command request registers (IFxCREQ)
- IFx command mask registers (IFxCMSK)
- IFx mask registers 1, 2 (IFxMSK1, IFxMSK2)
- IFx arbitration registers 1, 2 (IFxARB1, IFxARB2)
- IFx message control register (IFxMCTR)
- IFx data registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)
- CAN transmission request registers 1, 2, 3, 4 (TREQR1, TREQR2, TREQR3, TREQR4)
- CAN New Data registers 1, 2, 3, 4 (NEWDT1, NEWDT2, NEWDT3, NEWDT4)
- CAN interrupt pending registers 1, 2, 3, 4 (INTPND1, INTPND2, INTPND3, INTPND4)
- CAN message valid registers 1, 2, 3, 4 (MSGVAL1, MSGVAL2, MSGVAL3, MSGVAL4)

The CAN register is given an address space of 256 bytes (64 words) and accessible in byte or word mode. The CPU accesses the message RAM via a message interface register.

33.4.1.1 List of Base-addresses (Base-addr), External Signals and Buffer Size

This section shows the list of base-addresses (Base-addr), external signals and buffer size.

Channel Number	Base-addr	External Signal Name		Buffer Size
0	0x2100	TX	RX	64-message object

33.4.1.2 List of Overall Control Register

This section shows the list of overall control register.

Table 33-1. List of Overall Control Register

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 00_H	CAN control register (CTRLR)		CAN status register (STATR)		STAR: BOff, EWarn,EPass=Read Only RxOk,TxOk,LEC=Read/Write
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Reserved bits	See the CTRLR.	Reserved bits	See the STATR.	
	Reset: 00_H	Reset: 01_H	Reset: 00_H	Reset: 00_H	
Base-addr + 04_H	CAN error counter (ERRCNT)		CAN bit timing register (BTR)		ERRCNT: Read Only BTR: Write is enabled when Init(CTRLR) = CCE(CTRLR) = "1"
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	RP, REC[6: 0]	TEC[7: 0]	TSeg2[2: 0], TSeg1[3: 0]	SJW[1: 0], BRP[5: 0]	
	Reset: 00_H	Reset: 00_H	Reset: 23_H	Reset: 01_H	
Base-addr + 08_H	CAN interrupt register (INTR)		CAN test register (TESTR)		INTR: Read Only TESTR: Write is enabled when Test(CTRLR) = "1" "Rx" indicates the level at the CAN_RX signal.
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	IntId[15: 8]	IntId[7: 0]	Reserved bits	See the TESTR.	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H & 0br0000000	
Base-addr + $0C_H$	CAN prescaler extension register (BRPER)		Reserved bits		BRPER: Write is enabled when CCE(CTRLR) = "1"
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Reserved bits	BRPE[3: 0]	-	-	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	

33.4.1.3 List of Message Interface Register

This section shows the list of message interface register.

Table 33-2. List of Message Interface Register

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 10_H	IF1 command request register (IF1CREQ)		IF1 command mask register (IF1CMSK)		
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Busy	Mess. No. [5: 0]	Reserved bits	See the IF1CMSK.	
	Reset: 00_H	Reset: 01_H	Reset: 00_H	Reset: 00_H	
Base-addr + 14_H	IF1 mask register 2 (IF1MSK2)		IF1 mask register 1 (IF1MSK1)		
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	MXtd. MDir, Msk[28: 24]	Msk[23: 16]	Msk[15: 8]	Msk[7: 0]	
	Reset: FF_H	Reset: FF_H	Reset: FF_H	Reset: FF_H	
Base-addr + 18_H	IF1 arbitration register 2 (IF1ARB2)		IF1 arbitration register 1 (IF1ARB1)		
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	MsgVal, Xtd, Dir,ID[28: 24]	ID[23: 16]	ID[15: 8]	ID[7: 0]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + $1C_H$	IF1 message control register (IF1MCTR)		Reserved bits		
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	See the IF1MCTR.	See the IF1MCTR.	-	-	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 20_H	IF1 data A register 1 (IF1DTA1)		IF1 data A register 2 (IF1DTA2)		Byte order: Big Endian
	bit[7: 0]	bit[15: 8]	bit[7: 0]	bit[15: 8]	
	Data[0]	Data[1]	Data[2]	Data[3]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 24_H	IF1 data B register 1 (IF1DTB1)		IF1 data B register 2 (IF1DTB2)		Byte order: Big Endian
	bit[7: 0]	bit[15: 8]	bit[7: 0]	bit[15: 8]	
	Data[4]	Data[5]	Data[6]	Data[7]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 30_H	IF1 data A register 2 (IF1DTA2)		IF1 data A register 1 (IF1DTA1)		Byte order: Little Endian
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Data[3]	Data[2]	Data[1]	Data[0]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 34_H	IF1 data B register 2 (IF1DTB2)		IF1 data B register 1 (IF1DTB1)		Byte order: Little Endian
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Data[7]	Data[6]	Data[5]	Data[4]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 40_H	IF2 command request register (IF2CREQ)		IF2 command mask register (IF2CMSK)		
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Busy	Mess. No. [5: 0]	Reserved bits	See the IF2CMSK.	
	Reset: 00_H	Reset: 01_H	Reset: 00_H	Reset: 00_H	
Base-addr + 44_H	IF2 mask register 2 (IF2MSK2)		IF2 mask register 1 (IF2MSK1)		
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	MXtd. MDir, Msk[28: 24]	Msk[23: 16]	Msk[15: 8]	Msk[7: 0]	
	Reset: FF_H	Reset: FF_H	Reset: FF_H	Reset: FF_H	

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 48_H	IF2 arbitration register 2 (IF2ARB2)		IF2 arbitration register 1 (IF2ARB1)		
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	MsgVal, Xtd, Dir,ID[28: 24]	ID[23: 16]	ID[15: 8]	ID[7: 0]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + $4C_H$	IF2 message control register (IF2MCTR)		Reserved bits		
	bit[15: 8]	bit[7: 0]	bit[7: 0]	bit[15: 8]	
	See the IF2MCTR.	See the IF2MCTR.	-	-	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 50_H	IF2 data A register 1 (IF2DTA1)		IF2 data A register 2 (IF2DTA2)		Byte order: Big Endian
	bit[7: 0]	bit[15: 8]	bit[7: 0]	bit[15: 8]	
	Data[0]	Data[1]	Data[2]	Data[3]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 54_H	IF2 data B register 1 (IF2DTB1)		IF2 data B register 2 (IF2DTB2)		Byte order: Big Endian
	bit[7: 0]	bit[15: 8]	bit[7: 0]	bit[15: 8]	
	Data[4]	Data[5]	Data[6]	Data[7]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 60_H	IF2 data A register 2 (IF2DTA2)		IF2 data A register 1 (IF2DTA1)		Byte order: Little Endian
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Data[3]	Data[2]	Data[1]	Data[0]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 64_H	IF2 data B register 2 (IF2DTB2)		IF2 data B register 1 (IF2DTB1)		Byte order: Little Endian
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	Data[7]	Data[6]	Data[5]	Data[4]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	

33.4.1.4 List of Message Handler Register

This section shows the list of message handler register.

Table 33-3. List of Message Handler Register

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 80_H	CAN transmission request register 2 (TREQR2)		CAN transmission request register 1 (TREQR1)		TREQR 1, 2: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	TxRqst[32: 25]	TxRqst[24: 17]	TxRqst[16: 9]	TxRqst[8: 1]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 84_H	CAN transmission request register 4 (TREQR4)		CAN transmission request register 3 (TREQR3)		TREQR3, 4: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	TxRqst[64: 57]	TxRqst[56: 49]	TxRqst[48: 41]	TxRqst[40: 33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + 88_H Base-addr + $8C_H$	Reserved bits				
Base-addr + 90_H	CAN new data register 2 (NEWDT2)		CAN new data register 1 (NEWDT1)		NEWDT1, 2: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	NewDat[32: 25]	NewDat[24: 17]	NewData[16: 9]	NewData[8: 1]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 94_H	CAN new data register 4 (NEWDT4)		CAN new data register 3 (NEWDT3)		NEWDT3, 4: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	NewDat[64: 57]	NewDat[56: 49]	NewData[48: 41]	NewData[40: 33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + 98_H Base-addr + $9C_H$ 1.	Reserved bits				
Base-addr + $A0_H$	CAN interrupt pending register 2 (INTPND2)		CAN interrupt pending register 1 (INTPND1)		INTPND1, 2: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	IntPnd[32: 25]	IntPnd[24: 17]	IntPnd[16: 9]	IntPnd[8: 1]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	

Address	Registers				Note
	+0	+1	+2	+3	
Base-addr + $A4_H$	CAN interrupt pending register 4 (INTPND4)		CAN interrupt pending register 3 (INTPND3)		INTPND3, 4: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	IntPnd[64: 57]	IntPnd[56: 49]	IntPnd[48: 41]	IntPnd[40: 33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + $A8_H$ Base-addr + AC_H	Reserved bits				
Base-addr + $B0_H$	CAN message valid register 2 (MSGVAL2)		CAN message valid register 1 (MSGVAL1)		MSGVAL1,2: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	MsgVal[32: 25]	MsgVal[24: 17]	MsgVal[16: 9]	MsgVal[8: 1]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + $B4_H$	CAN message valid register 4 (MSGVAL4)		CAN message valid register 3 (MSGVAL3)		MSGVAL3, 4: Read Only
	bit[15: 8]	bit[7: 0]	bit[15: 8]	bit[7: 0]	
	MsgVal[64: 57]	MsgVal[56: 49]	MsgVal[48: 41]	MsgVal[40: 33]	
	Reset: 00_H	Reset: 00_H	Reset: 00_H	Reset: 00_H	
Base-addr + $B8_H$ Base-addr + BC_H	Reserved bits				

33.4.2 Overall Control Registers

Overall control registers are shown.

Overall control registers control the CAN protocol and operation modes and provide status information.

- CAN control register (CTRLR)
- CAN status register (STATR)
- CAN error counter (ERRCNT)
- CAN bit timing register (BTR)
- CAN interrupt register (INTR)
- CAN test register (TESTR)
- CAN prescaler extension register (BRPER)

33.4.2.1 CAN Control Register: CTRLR

The bit configuration of the CAN control register is shown.

Controls the operation mode of the CAN controller.

CAN control register (upper byte): Address Base + 00_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	-	-	-	-	-	-	-	-

CAN control register (lower byte): Address Base + 01_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Test	CCE	DAR	Reserved	EIE	SIE	IE	Init
Initial Value	0	0	0	0	0	0	0	1
Attribute	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W

[bit15 to bit8]: Reserved bits

The read value is always "0". Be sure to write "0" to these bits.

[bit7]: Test mode enable bit

Test	Function
0	Normal operation[Initial value]
1	Test mode

Note:

Set "1" to the Test bit only when the INIT bit is "1".

[bit6]: CAN Bit timing register write enable bit

CCE	Function
0	Disable the writing to the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER).[Initial value]
1	Enable the writing to the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER). This bit is valid when the Init bit is "1".

[bit5]: Automatic retransmission disable bit

The CAN controller retransmits the frame automatically when it loses the arbitration or when an error is detected during transfer. To enable automatic retransmission, set "0" to the DAR bit. In order to operate CAN in Time Triggered CAN environments, "1" needs to be set to the DAR bit.

DAR	Function
0	Enable the automatic retransmission of the message when CAN loses the arbitration or when an error is detected. [Initial value]
1	Disable automatic retransmission.

Notes:

When "1" is set to the DAR bit, the values for the TxRqst and NewDat bits of the message objects are as follows. (For message objects, see "33.4.4 Message Object.")

- When frame transmission is started, the TxRqst bit for the message object is cleared to "0", but the NewDat bit remains to be set to "1".
- When frame transmission is completed successfully, the NewDat bit is cleared to "0". When the transmission loses the arbitration or when an error is detected, the NewDat bit remains to be set to "1". To restart the transmission, set "1" to the TxRqst bit.
- When the DAR bit in the CAN control register (CTRLR) is changed from "0" to "1" during frame transmission (TxRqst=1), the frame that is being sent is retried. Thus, change the DAR bit only when the Init bit is "1".

The transmission operations when "1" is set to the DAR bit and several message objects are used are as follows.

- When "1" is set to TxRqst of *other* message objects (when "1" is set to TxRqst of several message objects) before CAN starts frame transmission or during transmission, all TxRqst set are reset to "0" and the data of the highest order message object is sent when frame transmission is started.
- When frame transmission is completed successfully, NewDat of sent message object is reset to "0", and IntPnd of the message object is set to "1" when the TxIE of the message object is "1".
- Other message objects do not send frames at frame transmission start because TxRqst is reset to "0". After the message object sent by NewDat or IntPnd is checked, "1" needs to be set to TxRqst and NewDat again for the message object to be sent.

[bit4] Reserved bit

The read value is always "0". When writing to this bit, set "0".

[bit3]: Error interrupt code enable bit

EIE	Function
0	Disable the interrupt code setting to the CAN interrupt register (INTR) with the bit change for Boff or EWarn of the CAN status register (STATR). [Initial value]
1	Enable the status interrupt code setting to the CAN interrupt register (INTR) with the bit change for Boff or EWarn of the CAN status register (STATR).

[bit2]: Status interrupt code enable bit

SIE	Function
0	Disable the interrupt code setting to the CAN interrupt register (INTR) with the bit change for TxOk, RxOk or LEC of the CAN status register (STATR). [Initial value]
1	Enable the status interrupt code setting to the CAN interrupt register (INTR) with the bit change for TxOk, RxOk or LEC of the CAN status register (STATR). The bit change for TxOk, RxOk and LEC generated by the writing from the CPU is not set to the CAN interrupt register (INTR).

[bit1]: Interrupt enable bit

IE	Function
0	Disable interrupt. [Initial value]
1	Enable interrupt.

[bit0]: Initialization bit

Init	Function
0	Operate after the initialization release of the CAN controller.
1	Initialize the CAN controller and stops the operation. [Initial value]

Notes:

- The bus-off recovery sequence cannot be shortened with the Init bit setting/release. When a device is in the bus-off state, the CAN controller itself sets "1" to the Init bit and stops all bus operations. When the Init bit is cleared to "0" in the bus-off state, the bus operation is stopped until the bus-idle continues 129 times (11-bit recessive is regarded as 1 time). The CAN error counter is reset after the execution of the bus-off recovery sequence.
- When the Init bit is set to "1" and then to "0" during the bus-off recovery sequence, the bus-off recovery sequence runs from the beginning (129 times regarding 11-bit recessive as 1 time).
- To set the CAN bit timing register (BTR), set "1" to the Init and CCE bits.
- When "1" is set to the Init bit during transmission/reception, the transmission/reception is stopped immediately.
- During transmission, to set "1" in the Init bit, set "1" in the bit after the transmission is completed. If "1" has been set in the Init bit during transmission, set "0" in the Init bit and then wait 2 bit times before making a transmission setting (TxRqst="1").
- Before a transition to a low power consumption mode (stop mode, clock mode) or a change to the supplied clock, "1" must be written to the Init bit and the CAN controller must be initialized.
- To change the clock divide ratio which supplies to the CAN interface by the following registers, set "1" to the Init bit and stop the CAN controller.
  - ☐ CAN bit timing register (BTR)
  - ☐ CAN prescaler extension register (BRPER)

33.4.2.2 CAN Status Register: STATR

The bit configuration of the CAN status register is shown.

Displays the CAN and CAN bus statuses.

CAN status register (upper byte): Address Base + 02_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	-	-	-	-	-	-	-	-

CAN status register (lower byte): Address Base + 03_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BOff	EWarn	EPass	RxOk	TxOk	LEC[2: 0]		
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R/W	R/W	R/W	R/W	R/W

[bit15 to bit8]: Reserved bits

The read value is always "0". Be sure to write "0" to these bits.

[bit7]: Bus-off bit

BOff	Function
0	Indicate the CAN controller is not in the bus-off state. [Initial value]
1	Indicate the CAN controller is in the bus-off state.

[bit6]: Warning bit

EWarn	Function
0	Indicate both the transmission and reception counters are below 96.[Initial value]
1	Indicate the transmission or reception counter is 96 or more.

[bit5]: Error passive bit

EPass	Function
0	Indicate both the transmission and reception counters are below 128 (error active state). [Initial value]
1	Indicate the reception counter is the RP bit = "1" and the transmission counter is 128 or more (error passive state).

[bit4]: Successful message reception bit

RxOk	Function
0	Indicate successful message communication is not performed on the CAN bus or the bus is in the idle state. [Initial value]
1	Indicate successful message communication is performed on the CAN bus.

[bit4]: Successful message transmission bit

TxOk	Function
0	Indicate the bus is in the idle state or successful message transmission is not performed. [Initial value]
1	Indicate successful message transmission is performed.

Note:

The RxOk and TxOk bits are cleared only with "0" writing.

[bit2 to bit0]: Last error code bits

The LEC bit holds the code that indicates the last error occurred on the CAN bus. This bit is cleared to "0" when a message transfer (reception/transmission) completes without error. The undetected code "111_B" can be used for checking the code update.

LEC[2: 0]	State	Function
000	Normal	Indicate transmission or reception is performed successfully. [Initial value]
001	Stuff error	Indicate more than 6 bits of dominant or recessive is detected continuously in a message.
010	Form error	Indicate the fixed format segment of a received frame is detected as incorrect.
011	Ack error	Indicate the transmission message is not acknowledged by other nodes.
100	Bit1 error	Indicate dominant was detected even though recessive was sent with the message transmission data other than arbitration field.
101	Bit0 error	Indicate recessive was detected even though dominant was sent with the message transmission data. This bit is set every time 11 bits of recessive is detected during the bus recovery. Reading this bit allows the monitoring of the bus recovery sequence.
110	CRC error	Indicate that CRC data and CRC result calculated for a received message did not match.
111	Undetected	Indicate no transmission or reception is performed during the period when LEC reads "111_B" after "111_B" is set to the LEC bit. (bus idle status)

Notes:

- The status interrupt code (8000_H) is set to the CAN interrupt register (INTR) if the BOff or EWarn bit is changed when the EIE bit is "1" or if RxOk, TxOk or the LEC bit is changed when the SIE bit is "1".
- The flag values for the RxOk and TxOk bits are updated with the program writing, and thus the RxOk and TxOk bit values set by the CAN controller are changed. When using RxOk and TxOk bits, these bits needs to be cleared within (45 x BT) time after the RxOk or TxOk bit is set to "1". BT is 1 bit time.
- Do not write into the CAN status register (STATR) if an interrupt occurs due to the LEC bit change when the SIE bit is "1".
- In the EPass bit change and writing operation into the RxOk, TxOk and the LEC bits, the error code interrupt is not set to the CAN interrupt register (INTR).
- When the BOff bit is "1", the EPass and EWarn bits are "1". In addition, the EWarn bit is "1" when the EPass bit is "1".
- The status interrupt (8000_H) of the CAN interrupt register (INTR) is cleared with the readout of the CAN status register (STATR).

33.4.2.3 CAN Error Counter: ERRCNT

The bit configuration of the CAN error counter is shown.

Indicates reception error passive display, reception error counter and transmission error counter.

CAN error counter register (upper byte): Address Base + 04_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	RP	REC [6: 0]						
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN error counter register (lower byte): Address Base + 05_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TEC [7: 0]							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

[bit15]: Reception error passive display

RP	Function
0	Indicate that it is not the error passive state. [Initial value]
1	Indicate that the error passive state that is defined in the CAN specification has been reached.

[bit14 to bit8]: Reception error counter

Reception error counter value. The range for the reception error counter values is 0 to 127.
When the reception error counter is greater than or equal to 128, "1" is set to the RP bit and the reception error counter is not updated.

Example:

When RP=0 and REC[6:0]=127 is incremented by 8 for reception error, the result is RP=1 and REC[6:0]=127.

When RP=0 and REC[6:0]=126 is incremented by 8 for reception error, the result is RP=1 and REC[6:0]=126.

When RP=0 and REC[6:0]=119 is incremented by 8 for reception error, the result is RP=0 and REC[6:0]=127.

When RP=1, REC[6:0]=126 and reception has ended successfully, the result is RP=0 and REC[6:0]=125.

[bit7 to bit0]: Transmission error counter

Transmission error counter value. The range for the transmission error counter values is 0 to 255.
When the transmission error counter is greater than or equal to 256, "1" is set to the Init bit of the CAN control register and the transmission error counter is not updated.

Example:

When Init=0 and TEC[7:0]=255 is incremented by 8 for transmission error, the result is Init=1 and TEC[7:0]=255.

When Init=0 and TEC[7:0]=254 is incremented by 8 for transmission error, the result is Init=1 and TEC[7:0]=254.

When Init=0 and TEC[7:0]=247 is incremented by 8 for transmission error, the result is Init=0 and TEC[7:0]=255.

33.4.2.4 CAN Bit Timing Register: BTR

The bit configuration of the CAN bit timing register is shown.

Sets the prescaler and bit timing.

CAN bit timing register (upper byte): Address Base + 06_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	TSeg2			TSeg1			
Initial Value	0	0	1	0	0	0	1	1
Attribute	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CAN bit timing register (lower byte): Address Base + 07_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SJW		BRP					
Initial Value	0	0	0	0	0	0	0	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15]: Reserved bit

The read value is always "0". Be sure to write "0" to these bits.

[bit14 to bit12]: Time segment 2 setting bits

Valid setting values are 0 to 7. TSeg2+1 bit value is time segment 2.
Time segment 2 corresponds to the phase buffer segment (Phase_Seg2) based on the CAN specification.

[bit11 to bit8]: Time segment 1 setting bits

Valid setting values are 1 to 15. 0 cannot be set. TSeg1+1 bit value is time segment 1.
Time segment 1 corresponds to the propagation segment (Prop_Seg) and phase buffer segment 1 (Phase_Seg1) based on the CAN specification.

[bit7, bit6]: Resynchronization jump width setting bits

Valid setting values are 0 to 3. The SJW+1 bit value is the resynchronization jump width.

[bit5 to bit0]: Baud rate prescaler setting bits

Valid setting values are 0 to 63. The BRP+1 bit value is the baud rate prescaler.

Divides frequency for system clock (fsys) and determines the basic unit time (tq) of the CAN controller.

Note:

When "1" is set to the CCE and Init bits of the CAN control register (CTRLR), set the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER).

33.4.2.5 CAN Interrupt Register: INTR

The bit configuration of the CAN interrupt register is shown.

Displays the message interrupt and status interrupt codes.

CAN interrupt register (upper byte): Address Base + 08_H (access: half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntId 15 to IntId 8							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt register (lower byte): Address Base + 09_H (access: half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntId 7 to IntId 0							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

If more than one interrupt codes are pending, the CAN interrupt register (INTR) will indicate the interrupt code of the highest priority. If a higher-priority interrupt code is generated when an interrupt code is set to the CAN interrupt register (INTR), the CAN interrupt register (INTR) is updated to the higher-order interrupt code.
Higher orders are given to the status interrupt code (8000_H), message interrupt (0001_H, 0002_H, 0003_H, ..., 0040_H) in descending order.

When the IntId bit is other than 0000_H and the IE bit of the CAN control register (CTRLR) is set to "1", the interrupt signal for CPU is active. When the IntId bit is 0000_H (an interrupt factor is reset) or the IE bit of the CAN control register (CTRLR) is reset to "0", the interrupt signal is inactive.

If the IntPnd bit of the target message objects (for message objects, see "33.4.4 Message Object") is cleared to "0", the message interrupt code will be cleared.

Status interrupt code will be cleared when the CAN status register (STATR) is read.

IntId	Function
0000_H	No interrupt
0001_H to 0040_H	The message object number is indicated as an interrupt factor. (Message interrupt code)
0041_H to $7FFF_H$	Unused
8000_H	Indicate interrupts with the change of the CAN status register (STATR). (Status interrupt code)
8001_H to $FFFF_H$	Unused

Note

To read the CAN Interrupt Register, access it in 16-bit or 32-bit mode.

33.4.2.6 CAN Test Register: TESTR

The bit configuration of the CAN test register is shown.

Monitors the test mode setting and CAN_RX signals. For operation, see "33.5.7 Test Mode."

CAN test register (upper byte): Address Base + $0A_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	-	-	-	-	-	-	-	-

CAN test register (lower byte): Address Base + $0B_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Rx	Tx1	Tx0	LBack	Silent	Basic	Reserved	Reserved
Initial Value	r	0	0	0	0	0	0	0
Attribute	R	R/W	R/W	R/W	R/W	R/W	-	-

Note: The level on the CAN bus is displayed as the initial value (r) of Rx.

[bit15 to bit8]: Reserved bits

The read value is always "0". Be sure to write "0" to these bits.

[bit7] Rx: CAN_RX signal monitor bit

Rx	Function
0	Indicate the CAN bus is dominant.
1	Indicate the CAN bus is recessive.

[bit6, bit5] Tx1, Tx0: CAN_TX signal control bits

Tx1,Tx0	Function
00	Normal operation [Initial value]
01	Sampling points will be output to the CAN_TX signal.
10	Dominant will be output to the CAN_TX signal.
11	Recessive will be output to the CAN_TX signal.

[bit4] LBack: Loopback mode

LBack	Function
0	Disable loopback mode. [Initial value]
1	Enable loopback mode.

[bit3] Silent: Silent mode

Silent	Function
0	Disable silent mode. [Initial value]
1	Enable silent mode.

[bit2] Basic: Basic mode

Basic	Function
0	Disable basic mode. [Initial value]
1	Enable basic mode. The IF1 register will be used as a transmission message, and the IF2 register will be used as a reception message.

[bit1, bit0]: Reserved bits

The read value is always "0". Be sure to write "0" to these bits.

Notes:

- After setting "1" to the Test bit of the CAN control register (CTRLR), write into the register. The test mode is valid when the Test bit of the CAN control register (CTRLR) is set to "1". The CAN controller transits from the test mode to the normal mode when the Test bit of the CAN control register (CTRLR) is set to "0".
- Messages cannot be sent when the Tx bit is set to a value other than "00".

33.4.2.7 CAN Prescaler Extension Register: BRPER

The bit configuration of the CAN prescaler extension register is shown.

Extends the prescaler used in the CAN controller by combining the prescaler set at the CAN bit timing.

CAN prescaler extension register (upper byte): Address Base + $0C_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	-	-	-	-	-	-	-	-

CAN prescaler extension register (lower byte): Address Base + $0D_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	BRPE			
Initial Value	0	0	0	0	0	0	0	0
Attribute	-	-	-	-	R/W	R/W	R/W	R/W

[bit15 to bit4]: Reserved bits

The read value is always "0". Be sure to write "0" to these bits.

[bit3 to bit0] BRPE: Baud rate prescaler extension bits

The baud rate prescaler can be extended up to 1023 by combining the BRP and BRPE bits of the CAN bit timing register (BTR).

The {BRPE (MSB:4 bits), BRP (LSB:6 bits)} + 1 value is the prescaler of the CAN controller.

33.4.3 Message Interface Register

This section shows the message interface register.

2 pairs of message interface registers are provided to control access from the CPU to the message RAM.

There are 2 pairs of message interface registers used to control access from the CPU to the message RAM. These 2 pairs of registers avoid conflict between accesses from the message RAM to the CPU and from the CAN controller by buffering transferred data (message object). The message object (for message object, see "33.4.4 Message Object") transfers messages between the message interface register and the message RAM.

The functions for 2 pairs of message interface registers are the same except the basic mode for tests, and these registers can operate independently. For example, the message interface register of IF2 can be used for readout from the message RAM while the message interface register of IF1 is being written into the message RAM Table 33-1 shows 2 pairs of message interface registers.

The message interface register consists of the command register (command request, command mask registers) and the message buffer register (mask, arbitration, message control and data registers) controlled by this command register. The command mask register indicates data transfer direction and which part of the message object will be transferred. The command request register selects the message number and performs the operation set to the command mask register.

Table 33.4-1 IF1, IF2 Message Interface Registers

Address	IF1 Register Set	Address	IF2 Register Set
Base + 10_H	IF1 command request	Base + 40_H	IF2 command request
Base + 12_H	IF1 command mask	Base + 42_H	IF2 command mask
Base + 14_H	IF1 mask 2	Base + 44_H	IF2 mask 2
Base + 16_H	IF1 mask 1	Base + 46_H	IF2 mask 1
Base + 18_H	IF1 arbitration 2	Base + 48_H	IF2 arbitration 2
Base + $1A_H$	IF1 arbitration 1	Base + $4A_H$	IF2 arbitration 1
Base + $1C_H$	IF1 message control	Base + $4C_H$	IF2 message control
Base + 20_H	IF1 data A1	Base + 50_H	IF2 data A1
Base + 22_H	IF1 data A2	Base + 52_H	IF2 data A2
Base + 24_H	IF1 data B1	Base + 54_H	IF2 data B1
Base + 26_H	IF1 data B2	Base + 56_H	IF2 data B2

33.4.3.1 IFx Command Request Register: IFxCREQ

The bit configuration of the IFx command request register is shown.

This register selects the message number of the message RAM and transfers the message between the message RAM and the message buffer register. In addition, IF1 is used for transmission control and IF2 is used for reception control in the basic mode for tests.

IFx command request register (upper byte): Address Base + 10_H & Base + 40_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BUSY	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	-	-	-	-	-	-	-

IFx command request register (lower byte): Address Base + 11_H & Base + 41_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Message Number							
Initial Value	0	0	0	0	0	0	0	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Immediately after the message number is written into the IFx command request register (IFxCREQ), the message transfer between the message RAM and the message buffer register (mask, arbitration, message control and data register) is started. This writing operation indicates that "1" is set to the BUSY bit and a message is being transferred. When the transfer is completed, the BUSY bit is reset to "0".

When "1" is set to the BUSY bit, the CPU will be kept waiting until the BUSY bit becomes "0" if the CPU accesses to the message interface register (3 to 6 cycles after writing into the command request register).

The BUSY bit is used differently in the basic mode for tests. The IF1 command request register is used as a transmission message, and setting "1" to the BUSY bit directs message transmission start. When the message transfer is completed successfully, the BUSY bit is reset to "0". In addition, resetting the BUSY bit to "0" aborts message transfer at any time.

The IF2 command request register is used as a reception message, and setting "1" to the BUSY bit stores the received message in the IF2 message interface register.

[bit15]BUSY: Busy flag bit

(1) Other than basic mode for tests

BUSY	Function
0	Indicate that data is not being transferred between the message interface register and the message RAM. [Initial value]
1	Indicate that data is being transferred between the message interface register and the message RAM.

(2) Basic mode for tests

IF1 command request register

BUSY	Function
0	Disable the message transmission.
1	Enable the message transmission.

IF2 command request register

BUSY	Function
0	Disable the message reception.
1	Enable the message reception.

[bit14 to bit8]: Reserved bits

The read value is always "0". Be sure to write "0" to these bits.

[bit7 to bit0] Message Number: Message Number

Message Number	Function
00_H	Setting prohibited. If this value is set, it is interpreted as 40_H and 40_H is read out.
01_H to 40_H	Set the message number for processing.
41_H to FF_H	Setting prohibited. If this value is set, it is interpreted as 01_H to $3F_H$ and the value interpreted is read out.

Note:

The BUSY bit is readable/writable. Other than in the basic mode for tests, it does not affect the operation no matter which value is written to this bit. (See "33.5.7 Test Mode" for the details of the basic mode.)

33.4.3.2 IFx Command Mask Register: IFxCMSK

The bit configuration of the IFx command mask register is shown.

This register sets which data to be updated by controlling the direction of transfer between the message interface register and message RAM. The register becomes invalid in the basic mode for tests.

IFx command mask register (upper byte): Address Base + 12_H & Base + 42_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	-	-	-	-	-	-	-	-

IFx command mask register (lower byte): Address Base + 13_H & Base + 43_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	WR/RD	Mask	Arb	Control	CIP	TxRqst/NewDat	DataA	DataB
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit8]: Reserved bits

The read value is always "0". Be sure to write "0" to these bits.

[bit7]WR/RD: Write/read control bit

WR/RD	Function
0	Indicate reading data from message RAM. Reading data from message RAM will be executed by writing data to the IFx command request register (IFxCREQ). Data read from message RAM depends on the settings of Mask, Arb, Control, CIP, TxRqst/NewDat, DataA, and DataB bits. [Initial value]
1	Indicate writing data to message RAM. Writing data to message RAM will be executed by writing data to the IFx command request register (IFxCREQ). Data written to message RAM depends on the settings of Mask, Arb, Control, CIP, TxRqst/NewDat, DataA, and DataB bits.

Note:

Data in message RAM is undefined after reset. Reading data from message RAM is disabled while data in message RAM is undefined.

Bit6 to bit0 of the IFx command mask register (IFxCMSK) has different meanings depending on the settings of transfer direction (WR/RD bit).

(1) When the transfer direction is write (WR/RD="1")

[bit6] Mask: Mask data update bit

Mask	Function
0	Indicate not updating the mask data (ID mask + MDir + MXtd) of message object*. [Initial value]
1	Indicate updating the mask data (ID mask + MDir + MXtd) of message object*.

[bit5] Arb: Arbitration data update bit

Arb	Function
0	Indicate not updating the arbitration data (ID + Dir + Xtd + MsgVal) of message object*. [Initial value]
1	Indicate updating the arbitration data (ID + Dir + Xtd + MsgVal) of message object*.

[bit4] Control: Control data update bit

Control	Function
0	Indicate not updating the control data (IFx message control register (IFxMCTR)) of message object*. [Initial value]
1	Indicate updating the control data (IFx message control register (IFxMCTR)) of message object*.

[bit3] CIP: Interrupt clear bit

Operation of CAN controller will not be affected whether "0" or "1" is set.

[bit2] TxRqst/ NewDat: Message transmission request bit

TxRqst/ NewDat	Function
0	Indicate not changing the TxRqst bit of message object* and CAN transmission request register (TREQR). [Initial value]
1	Indicate that "1" is set to the TxRqst bit of message object* and CAN transmission request register (TREQR) (transmission request).

[bit1] DataA: Data 0 to Data 3 update bit

DataA	Function
0	Indicate not updating Data 0 to Data 3 of message object*. [Initial value]
1	Indicate updating Data 0 to Data 3 of message object*.

[bit0] DataB: Data 4 to Data 7 update bit

DataB	Function
0	Indicate not updating Data 4 to Data 7 of message object*. [Initial value]
1	Indicate updating Data 4 to Data 7 of message object*.

* See "33.4.4 Message Object."

Notes:

- When the TxRqst/NewDat bit of the IFx command mask register (IFxCMSK) is set to "1", the TxRqst bit settings of the IFx message control register (IFxMCTR) becomes invalid.
- The register becomes invalid in the basic mode for tests.

(2) When the transfer direction is read (WR/RD="0")

[bit6] Mask: Mask data update bit

Mask	Function
0	Indicate not transferring data (ID mask +MDir + MXtd) from message object*1 to IFx mask registers 1, 2 (IFxMSK1, IFxMSK2). [Initial value]
1	Indicate transferring data (ID mask +MDir + MXtd) from message object*1 to IFx mask registers 1, 2 (IFxMSK1, IFxMSK2).

[bit5] Arb: Arbitration data update bit

Arb	Function
0	Indicate not transferring data (ID + Dir + Xtd + MsgVal) from message object*1 to IFx arbitration 1, 2 (IFxARB1, IFxARB2). [Initial value]
1	Indicate transferring data (ID + Dir + Xtd + MsgVal) from message object*1 to IFx arbitration 1, 2 (IFxARB1, IFxARB2).

[bit4] Control: Control data update bit

Control	Function
0	Indicate not transferring data from message object*1 to IFx message control register (IFxMCTR). [Initial value]
1	Indicate transferring data from message object*1 to IFx message control register (IFxMCTR).

[bit3] CIP: Interrupt clear bit

CIP	Function
0	Indicate holding the IntPnd bit of message object*1 and CAN interrupt pending register (INTPND). [Initial value]
1	Indicate clearing the IntPnd bit of message object*1 and CAN interrupt pending register (INTPND) to "0".

[bit2] TxRqst/ NewDat: Data update bit

TxRqst/ NewDat	Function
0	Indicate holding the NewDat bit of message object* and CAN data update register. [Initial value]
1	Indicate clearing the NewDat bit of message object* and CAN data update register to "0".

[bit1] DataA: Data 0 to Data 3 update bit

DataA	Function
0	Indicate holding data of message object* and CAN data registers A1, A2. [Initial value]
1	Indicate updating data of message object*1 and CAN data registers A1, A2.

[bit0] DataB: Data 4 to Data 7 update bit

DataB	Function
0	Indicate holding data of message object* and CAN data registers B1, B2. [Initial value]
1	Indicate updating data of message object* and CAN data registers B1, B2.

* See "33.4.4 Message Object."

Notes:

- It is possible to reset the IntPnd and NewDat bits to "0" by reading access to the message object. However, for the IntPnd and NewDat bits of the IFx message control register (IFxMCTR), the IntPnd and NewDat bits are stored before they are reset by reading access.
- It becomes invalid in the basic mode for tests.

33.4.3.3 IFx Mask Registers 1, 2: IFxMSK1, IFxMSK2

The bit configuration of the IFx mask registers 1, 2 is shown.

They are used to write/read message object mask data of message RAM. In the basic mode for tests, the configured mask data becomes invalid.
See "33.4.4 Message Object" for the functions of each bit.

IFx mask register 2 (upper byte): Address Base + 14_H & Base + 44_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MXtd	MDir	Reserved	Msk28 to Msk24				
Initial Value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

IFx mask register 2 (lower byte): Address Base + 15_H & Base + 45_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Msk23 to Msk16							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFx mask register 1 (upper byte): Address Base + 16_H & Base + 46_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Msk15 to Msk8							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFx mask register 1 (lower byte): Address Base + 17_H & Base + 47_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Msk7 to Msk0							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For the reserved bit (bit13 of IFx mask register 2), "1" is read out. Be sure to write "1" to these bits.

33.4.3.4 IFx Arbitration Registers 1, 2: IFxARB1, IFxARB2

The bit configuration of the IFx arbitration registers 1, 2 is shown.

They are used to write/read message object arbitration data of message RAM. They become invalid in the basic mode for tests.

See "33.4.4 Message Object" for the functions of each bit.

IFx arbitration register 2 (upper byte): Address Base + 18_H & Base + 48_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal	Xtd	Dir	ID28 to ID24				
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFx arbitration register 2 (lower byte): Address Base + 19_H & Base + 49_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ID23 to ID16							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFx arbitration register 1 (upper byte): Address Base + $1A_H$ & Base + $4A_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ID15 to ID8							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFx arbitration register 1 (lower byte): Address Base + $1B_H$ & Base + $4B_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ID7 to ID0							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note:

If the MsgVal bit of the message object is cleared to "0" while the transmission is in progress, the TxOk bit of the CAN status register (STATR) will be set to "1" when the transmission has completed. However, the TxRqst bits of the message object and CAN transmission request register (TREQR) will not be cleared to "0". So, make sure to clear the TxRqst bits to "0" using the message interface register.

33.4.3.5 IFx Message Control Register: IFxMCTR

The bit configuration of the IFx message control register is shown.

They are used to write/read message object control data in message RAM. The IF1 message control register will be disabled in the basic mode for tests. NewDat and MsgLst of the IF2 message control register will operate normally and the DLC bits will display the DLC of message received. Other control bits will operate as disabled ("0").
See "33.4.4 Message Object" for the functions of each bit.

IFx message control register (upper byte): Address Base + $1C_H$ & Base + $4C_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFx message control register (lower byte): Address Base + $1D_H$ & Base + $4D_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EoB	Reserved	Reserved	Reserved	DLC3 to DLC0			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	-	-	-	R/W	R/W	R/W	R/W

See "33.4.4 Message Object" for explanation of bits.

Note:

TxRqst, NewDat, and IntPnd bits operate differently depending on the settings of the WR/RD bit in the IFx command mask register (IFxCMSK).

- If the transfer direction is "write" (IFx command mask register (IFxCMSK): WR/RD="1").
- The TxRqst bit of this register will only be enabled when TxRqst/NewDat in the IFx command mask register (IFxCMSK) is set to "0".
- If the transfer direction is "read" (IFx command mask register (IFxCMSK): WR/RD="0").
- The IntPnd bit before it has been reset will be stored to this register when the message object and the IntPnd bit of the CAN interrupt pending register (INTPND) are reset by a write operation to the IFx command request register (IFxCREQ) after setting the CIP bit of the IFx command mask register (IFxCMSK) to "1".
- The NewDat bit before it has been reset will be stored to this register when the message object and the NewDat bit of the CAN data update register are reset by a write operation to the IFx command request register (IFxCREQ) after setting the TxRqst/NewDat bit of the IFx command mask register (IFxCMSK) to "1".

33.4.3.6 IFx Data Registers A1, A2, B1, B2: IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2

The bit configuration of the IFx data registers A1, A2, B1, B2 are shown.

They are used to write/read message object transmission/reception data in message RAM. Only used for transmitting/receiving data frames, and not for transmitting/receiving remote frames.

	addr+0	addr+1	addr+2	addr+3
IFx Message Data A1 (addresses 20_H & 50_H)	Data 0	Data 1		
IFx Message Data A2 (addresses 22_H & 52_H)			Data 2	Data 3
IFx Message Data B1 (addresses 24_H & 54_H)	Data 4	Data 5		
IFx Message Data B2 (addresses 26_H & 56_H)			Data 6	Data 7
IFx Message Data A2 (addresses 30_H & 60_H)	Data 3	Data 2		
IFx Message Data A1 (addresses 32_H & 62_H)			Data 1	Data 0
IFx Message Data B2 (addresses 34_H & 64_H)	Data 7	Data 6		
IFx Message Data B1 (addresses 36_H & 66_H)			Data 5	Data 4

IFx data register:

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Data							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transmission message data setting

Data set starts from MSB (bit7, bit15) and will be transmitted in the order of Data (0), Data (1), ..., Data (7).

Reception message data

Reception message data starts from MSB (bit7, bit15) and will be stored in the order of Data (0), Data (1), ..., Data (7).

Notes:

- If the reception message data is less than 8 bytes, undefined data will be written to the remaining bytes of the data register.
- Data transfer to the message object will be in units of 4 bytes of DataA or DataB. It is therefore not possible to update only a part of the 4-byte data.

33.4.4 Message Object

Message object is shown.

The message RAM has 64 message objects. In order to prevent conflict between accesses to message RAM from CPU and CAN controller, the CPU cannot access the message object directly. These accesses are performed via the IFx message interface register.
This section explains the configuration and function of the message objects.

33.4.4.1 Configuration of Message Object

The configuration of the message object is shown.

The configuration of the message object is shown below:

Table 33.4-2 Configuration of Message Object

UMask	Msk28 to Msk0	MXtd	MDir	EoB	NewDat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID28 to ID0	Xtd	Dir	DLC3 to DLC 0	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

Note:

The message object will not be initialized by the Init bit of the CAN control register (CTRLR) or hardware reset. In the case of hardware reset, after its release, initialize message RAM by the CPU or set the MsgVal of message RAM to "0".

33.4.4.2 Functions of Message Object

The functions of the message object are shown.

When transmitting a message, ID28 to ID0, Xtd and Dir bits will be used as the ID and type of the message. When receiving a message, they will be used with Msk28 to Msk0, MXtd and MDir bits in the acceptance filter.

ID, IDE, RTR, DLC and DATA for data frame or remote frame passing through the acceptance filter will be stored in the ID28 to ID0, Xtd, Dir, DLC3 to DLC0, Data7 to Data0 of the message objects. Xtd indicates whether the message object is an extension frame or standard frame, a 29-bit ID (extension frame) will be received if Xtd is "1", and an 11-bit ID (standard frame) will be received if Xtd is "0".

If the received data frame or remote frame matches 1 or more message objects, it will be stored to the lowest matched message number. (See "Reception Message Acceptance Filter" in "33.5.3 Message Reception Operation" for details.)

MsgVal: Valid message bit

MsgVal	Function
0	Message object is invalid. Message transmission/reception will not be performed.
1	Message object is valid. Message transmission/reception will become possible.

Notes:

- Be sure to initialize the MsgVal bit of the message object before resetting the Init bit of the CAN control register (CTRLR) to "0" and changing the value of ID28 to ID0, Xtd, Dir, and DLC3 to DLC0.
- If the MsgVal bit is cleared to "0" while the transmission is in progress, the TxOk bit of the CAN status register (STATR) will become "1" as soon as the transmission ends. However, the message object and the TxRqst bit of the CAN transmission request register (TREQR) will not be cleared to "0". So be sure to clear the TxRqst bit to "0" by the message interface register.

UMask: Acceptance mask enable bit

UMask	Function
0	Do not use Msk28 to Msk0, MXtd, and MDir.
1	Use Msk28 to Msk0, MXtd, and MDir.

Notes:

- Change the UMask bit while the Init bit of the CAN control register (CTRLR) is "1" or while the MsgVal bit is "0".
- When the Dir bit is "1" and the RmtEn bit is "0", it will operate differently depending on the UMask bit setting.
  - ☐ If the UMask bit is "1", the TxRqst bit will be reset to "0" when the remote frame is received through the acceptance filter. At this time, the received ID, IDE, RTR and DLC will be stored to the message object, the NewDat bit will be set to "1", and the data will remain unchanged (treated as a data frame).
  - ☐ If the UMask bit is "0", the TxRqst bit will remain unchanged by the remote frame reception; and it will ignore the remote frame.

D28 to ID0: Message ID

	Function
ID28 to ID0	Instruct a 29-bit ID (extended frame).
ID28 to ID18	Instruct an 11-bit ID (standard frame).

Msk28 to Msk0: ID Mask

Msk	Function
0	Mask the bit corresponding to the message object ID.
1	Do not mask the bit corresponding to the message object ID.

Xtd: Extended ID enable bit

Xtd	Function
0	An 11-bit ID (standard frame) is used for the message object.
1	A 29-bit ID (extended frame) is used for the message object.

MXtd: Extended ID mask bit

MXtd	Function
0	Do not compare the values between those set to the Xtd of the message object and those for the IDE bit in the received frame. The IDE in the received frame determines whether to compare it as a standard frame ID or an extended frame ID.
1	Compare the values between those set to the Xtd of the message object and those for the IDE in the received frame.

Note:

If an 11-bit ID (standard frame) is set to the message object, ID of the received data frame will be written to ID28 to ID18. Msk28 to Msk18 are used for ID masks.

Dir: Message direction bit

Dir	Function
0	Indicate the reception direction. The remote frame will be transmitted when the TxRqst is set to "1", and the data frame that has passed through the acceptance filter will be received when TxRqst is set to "0".
1	Indicate the transmission direction. Data frame will be transmitted when the TxRqst is set to "1". If the TxRqst is "0" and the RmtEn is set to "1", the CAN controller itself sets the TxRqst bit to "1" by receiving the remote frame that has passed through the acceptance filter.

MDir: Message direction mask bit

MDir	Function
0	Mask the message direction bit (Dir) in the acceptance filter.
1	Do not mask the message direction bit (Dir) in the acceptance filter.

Note:

Always set the MDir bit to "1".

EoB: End of Buffer bit (see "33.5.4 . FIFO Buffer Function" for details)

EoB	Function
0	Indicate that the massage object is used as FIFO buffer and is not the final message.
1	Indicate a single message object or the final message object of FIFO buffer.

Notes:

- The EoB bit is used to configure the FIFO buffer of 2 to 64 messages.
- Always set the EoB bit to "1" in the case of a single message object (when FIFO is not used).

NewDat: Data update bit

NewDat	Function
0	Valid data does not exist.
1	Valid data exists.

MsgLst: Message lost

MsgLst	Function
0	No message lost occurs.
1	Message lost occurs.

Note:

The MsgLst bit is only enabled when the Dir bit is "0" (reception direction).

RxIE: Reception interrupt flag enable bit

RxIE	Function
0	The IntPnd remains unchanged after successful frame reception.
1	The IntPnd is set to "1" after successful frame reception.

TxIE: Transmission interrupt flag enable bit

TxIE	Function
0	The IntPnd remains unchanged after successful frame transmission.
1	The IntPnd is set to "1" after successful frame transmission.

IntPnd: Interrupt pending bit

IntPnd	Function
0	No interrupt factor exists.
1	Interrupt factor exists. If no other high priority interrupt exists, the IntId bit of the CAN interrupt register (INTR) will indicate this message object.

RmtEn: Remote enable

RmtEn	Function
0	The TxRqst remains unchanged by remote frame reception.
1	The TxRqst will be set to "1" if a remote frame is received while the Dir bit is "1".

Note:

When the Dir bit is "1" and the RmtEn bit is "0", it will operate differently depending on the UMask setting.

- If the UMask is "1", the TxRqst bit will be reset to "0" when the remote frame is received through the acceptance filter. At this time, the received ID, IDE, RTR and DLC will be stored in the message object, the NewDat bit will be set to "1", and the data will remain unchanged (treated as a data frame).
- If the UMask is "0", the TxRqst bit will remain unchanged by the remote frame reception; and it will ignore the remote frame.

TxRqst: Transmission request bits

TxRqst	Function
0	Indicate the transmission idle state (neither transmission is in progress nor in the transmission wait state).
1	Indicate that transmission is in progress or in the transmission wait state.

DLC3 to DLC0: Data length code

DLC3 to 0	Function
0 to 8	Data frame length is 0 to 8 bytes.
9 to 15	Setting prohibited. If set, it will be 8 bytes in length.

Note:

The received DLC will be stored in the DLC bit when the data frame is received.

Data 0 to 7: Data 0 to 7

	Function
Data 0	1st data byte of the CAN data frame
Data 1	2nd data byte of the CAN data frame
Data 2	3rd data byte of the CAN data frame
Data 3	4th data byte of the CAN data frame
Data 4	5th data byte of the CAN data frame
Data 5	6th data byte of the CAN data frame
Data 6	7th data byte of the CAN data frame
Data 7	8th data byte of the CAN data frame

Notes:

- Serial output to the CAN bus is output from MSB (bit7 or bit15).
- If the received message data is less than 8 bytes, the remaining byte data of the data register will be undefined.
- Data transfer to the message object will be in units of 4 bytes of DataA or DataB. It is therefore not possible to update only a part of the 4-byte data.

33.4.5 Message Handler Registers

Message handler registers are shown.

All message handler registers are for reading only. The TxRqst, NewDat, IntPnd, MsgVal, and IntId bits of the message object are used to display a status.

- CAN transmission request registers 1 to 4 (TREQR1 to TREQR4)
- CAN data update registers 1 to 4 (NEWDT1 to NEWDT4)
- CAN interrupt pending registers 1 to 4 (INTPND1 to INTPND4)
- CAN message valid registers 1 to 4 (MSGVAL1 to MSGVAL4)

33.4.5.1 CAN Transmission Request Registers: TREQR1 to TREQR4

The bit configuration of the CAN transmission request registers is shown.

Displays the TxRqst bit of all message objects. It is possible to check which message objects transmission request is pending by reading the TxRqst bits.

CAN transmission request register 4 (upper byte): Address Base + 84_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst64 to TxRqst57							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN transmission request register 4 (lower byte): Address Base + 85_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst56 to TxRqst49							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN transmission request register 3 (upper byte): Address Base + 86_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst48 to TxRqst41							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN transmission request register 3 (lower byte): Address Base + 87_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst40 to TxRqst33							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN transmission request register 2 (upper byte): Address Base + 80_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst32 to TxRqst25							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN transmission request register 2 (lower byte): Address Base + 81_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst24 to TxRqst17							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN transmission request register 1 (upper byte): Address Base + 82_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TxRqst16 to TxRqst9							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN transmission request register 1 (lower byte): Address Base + 83_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TxRqst8 to TxRqst1							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

TxRqst64 to TxRqst1: Transmission request bits

TxRqst64 to TxRqst1	Function
0	Indicate the transmission idle state (neither transmission is in progress nor in the transmission wait state).
1	Indicate that transmission is in progress or in the transmission wait state.

Set/reset conditions of the TxRqst bits are shown below.

Set condition

When the WR/RD is set to "1" and the TxRqst is set to "1" for the IFx command mask register (IFxCMSK), it is possible to set the TxRqst of a specific object by writing to the IFx command request register (IFxCREQ).

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the TxRqst is set to "0", the Control is set to "1", and the TxRqst of the IFx message control register (IFxMCTR) is set to "1", it is possible to set the TxRqst of a specific object by writing data to the IFx command request register (IFxCREQ).

The bit will be set by a reception of remote frame that has passed the acceptance filter when the Dir bit and RmtEn bit are set to "1" respectively.

Reset condition

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the TxRqst is set to "0", the Control is set to "1", and the TxRqst of the IFx message control register (IFxMCTR) is set to "0", it is possible to reset the TxRqst of a specific object by writing data to the IFx command request register (IFxCREQ).

When frame transmission is completed successfully, the bit will be reset.

The bit will be reset by a reception of remote frame that has passed the acceptance filter when the Dir bit is set to "1", the RmtEn bit is set to "0", and the UMask is set to "1".

Notes:

When the message object with the lowest priority is used for transmission and the TxRqst is set to "1" and then to "0" to cancel transmission, setting the TxRqst to "1" again may not, depending on the timing, result in transmission of a message until one of the following events occurs.

- A valid message is transmitted on the CAN bus.
- A transmission request is issued to other message object.
- CAN is initialized by the Init bit.

If there is a situation in which transmission is canceled due to system reasons, either do not use the message object with the lowest priority as the transmission message object or, after transmission cancellation, generate one of the above events and then set the TxRqst to "1" again.

When the TxRqst bit is "1", do not change the message objects of ID28 to ID0, DLC3 to DLC0, Xtd, and Data7 to Data0. Otherwise, message objects before and after the change may be transmitted in a mixed way or message objects after the change may not be transmitted. Change them when the TxRqst bit is "0".

33.4.5.2 CAN Data Update Registers: NEWDT1 to NEWDT4

The bit configuration of the CAN data update registers is shown.

Displays the NewDat bit of all message objects. It is possible to check which message objects data has been updated by reading the NewDat bit.

CAN data update register 4 (upper byte): Address Base + 94_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat64 to NewDat57							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R,	R	R

CAN data update register 4 (lower byte): Address Base+ 95_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat56 to NewDat49							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN data update register 3 (upper byte): Address Base + 96_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat48 to NewDat41							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN data update register 3 (lower byte): Address Base+ 97_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat40 to NewDat33							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN data update register 2 (upper byte): Address Base + 90_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat32 to NewDat25							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN data update register 2 (lower byte): Address Base+ 91_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat24 to NewDat17							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN data update register 1 (upper byte): Address Base + 92_H (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	NewDat16 to NewDat9							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN data update register 1 (lower byte): Address Base+ 93_H (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	NewDat8 to NewDat1							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

NewDat64 to NewDat1: Data update bits

NewDat64 to NewDat1	Function
0	Indicate no valid data exists.
1	Indicate valid data exists.

Set/reset conditions of the NewDat bits are shown below.

Set condition

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the Control is set to "1", and the NewDat of the IFx message control register (IFxMCTR) is set to "1", it is possible to set a specific object by writing data to the IFx command request register (IFxCREQ).

The bit will be set by a reception of data frame that has passed the acceptance filter.

When the Dir is set to "1", the RmtEn is set to "0", and the UMask is set to "1", the bit will be set by a reception of remote frame that has passed the acceptance filter.

Reset condition

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "0" and the NewDat is set to "1", it is possible to reset the NewDat of a specific object by writing data to the IFx command request register (IFxCREQ).

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the Control is set to "1", and the NewDat of the IFx message control register (IFxMCTR) is set to "0", it is possible to reset the NewDat of a specific object by writing data to the IFx command request register (IFxCREQ).

It will be reset after data has been transferred to the transmission shift register (internal register).

33.4.5.3 CAN Interrupt Pending Registers: INTPND1 to INTPND4

The bit configuration of the CAN interrupt pending registers is shown.

Displays the IntPnd bit of all message objects. It is possible to check which message objects interrupt is pending by reading the IntPnd bit.

CAN interrupt pending register 4 (upper byte): Address Base + $A4_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd64 to IntPnd57							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt pending register 4 (lower byte): Address Base + $A5_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd56 to IntPnd49							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt pending register 3 (upper byte): Address Base + $A6_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd48 to IntPnd41							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt pending register 3 (lower byte): Address Base + $A7_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd40 to IntPnd33							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt pending register 2 (upper byte): Address Base + $A0_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd32 to IntPnd25							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt pending register 2 (lower byte): Address Base + $A1_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd24 to IntPnd17							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt pending register 1 (upper byte): Address Base + $A2_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd16 to IntPnd9							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN interrupt pending register 1 (lower byte): Address Base + $A3_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd8 to IntPnd1							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

IntPnd64 to IntPnd1: Interrupt pending bits

IntPnd64 to IntPnd1	Function
0	No interrupt factor exists.
1	Interrupt factor exists.

Set/reset conditions of the IntPnd bits are shown below.

Set condition

If the TxIE is set to "1", the IntPnd bit will be set after the frame transmission has ended successfully.

If the RxIE is set to "1", the bit will be set after the frame reception that has passed the acceptance filter completed successfully.

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "1", the Control is set to "1", and the IntPnd of the IFx message control register is set to "1", it is possible to set the IntPnd to "1" in a specific object by writing data to the IFx command request register (IFxCREQ).

Reset condition

When the WR/RD of the IFx command mask register (IFxCMSK) is set to "0" and the CIP is set to "1", it is possible to reset the IntPnd of a specific object by writing data to the IFx command request register (IFxCREQ). When the WR/RD of the IFx command mask register is set to "1", the Control is set to "1", and the IntPnd of the IFx message control register (IFxMCTR) is set to "0", it is possible to reset the IntPnd to "0" in a specific object by writing data to the IFx command request register.

33.4.5.4 CAN Message Valid Registers: MSGVAL1 to MSGVAL4

The bit configuration of the CAN message valid registers is shown.

Displays the MsgVal bit of all message objects. It is possible to check which message object is valid by reading the MsgVal bit.

CAN message valid register 4 (upper byte): Address Base + $B4_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal64 to MsgVal57							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN message valid register 4 (lower byte): Address Base + $B5_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal56 to MsgVal49							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN message valid register 3 (upper byte): Address Base + $B6_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal48 to MsgVal41							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN message valid register 3 (lower byte): Address Base + $B7_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal40 to MsgVal33							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN message valid register 2 (upper byte): Address Base + $B0_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal32 to MsgVal25							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN message valid register 2 (lower byte): Address Base + $B1_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal24 to MsgVal17							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN message valid register 1 (upper byte): Address Base + $B2_H$ (access: byte, half-word, word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MsgVal16 to MsgVal9							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

CAN message valid register 1 (lower byte): Address Base + $B3_H$ (access: byte, half-word, word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MsgVal8 to MsgVal1							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R	R	R	R	R	R	R	R

MsgVal64 to MsgVal1: Message valid bits

MsgVal64 to MsgVal1	Function
0	Message object is invalid. Message will not be transmitted/received.
1	Message object is valid. Message transmission/reception is possible.

Set/reset conditions of the MsgVal bits are shown below.

Set condition

When the WR/RD of the IFx command mask register is set to "1", the Arb is set to "1", and the MsgVal bit of the IFx arbitration register 2 is set to "1", it is possible to set the MsgVal bit of a specific object by writing data to the IFx command request register (IFxCREQ).

Reset condition

When the WR/RD of the IFx command mask register is set to "1", the Arb is set to "1", and the MsgVal bit of the IFx arbitration register 2 is set to "0", it is possible to clear the MsgVal bit of a specific object by writing data to the IFx command request register (IFxCREQ).

33.5 Operation

This section explains the operation of the CAN.

The CAN has the following functions:

- Message object
- Message transmission operation
- Message reception operation
- FIFO buffer function
- Interrupt function
- Bit timing
- Test mode
- Software initialization

33.5.1 Message Object

Message object is shown.

This section explains the message object and interface of message RAM.

33.5.1.1 Message Object

Message object is shown.

The configuration of message objects in the message RAM (excluding the MsgVal, NewDat, IntPnd, and TxRqst bits) is not initialized by a hardware reset. Initialize the message objects by one of the following steps after hardware reset.

1. Initialize all message objects by the CPU.
2. Initialize only the used message objects by CPU. Clear MsgVal bit to "0" in the other message objects.

Configure the CAN Bit Timing Register while the Init bit in the CAN Control Register is "1".

A message object must be configured by programming message interface registers (the IFx Command Mask Register, IFx Mask Register, IFx Arbitration Register, IFx Message Control Register, and IFx Data Register), and then writing a message number to the corresponding IFx Command Request Register. By writing the message number, the message interface register data will be transferred to the addressed message object.

When the Init bit in the CAN Control Register is cleared to "0", the CAN controller starts operation. The received data that have passed acceptance filtering are stored into the message RAM. Messages with pending transmission requests are transferred from the message RAM to the shift register in the CAN controller, and then sent to the CAN bus.

The CPU reads the received messages and updates outgoing messages via message interface registers. The CPU is interrupted according to the configuration of the CAN Control Register and IFx Message Control Register (message object).

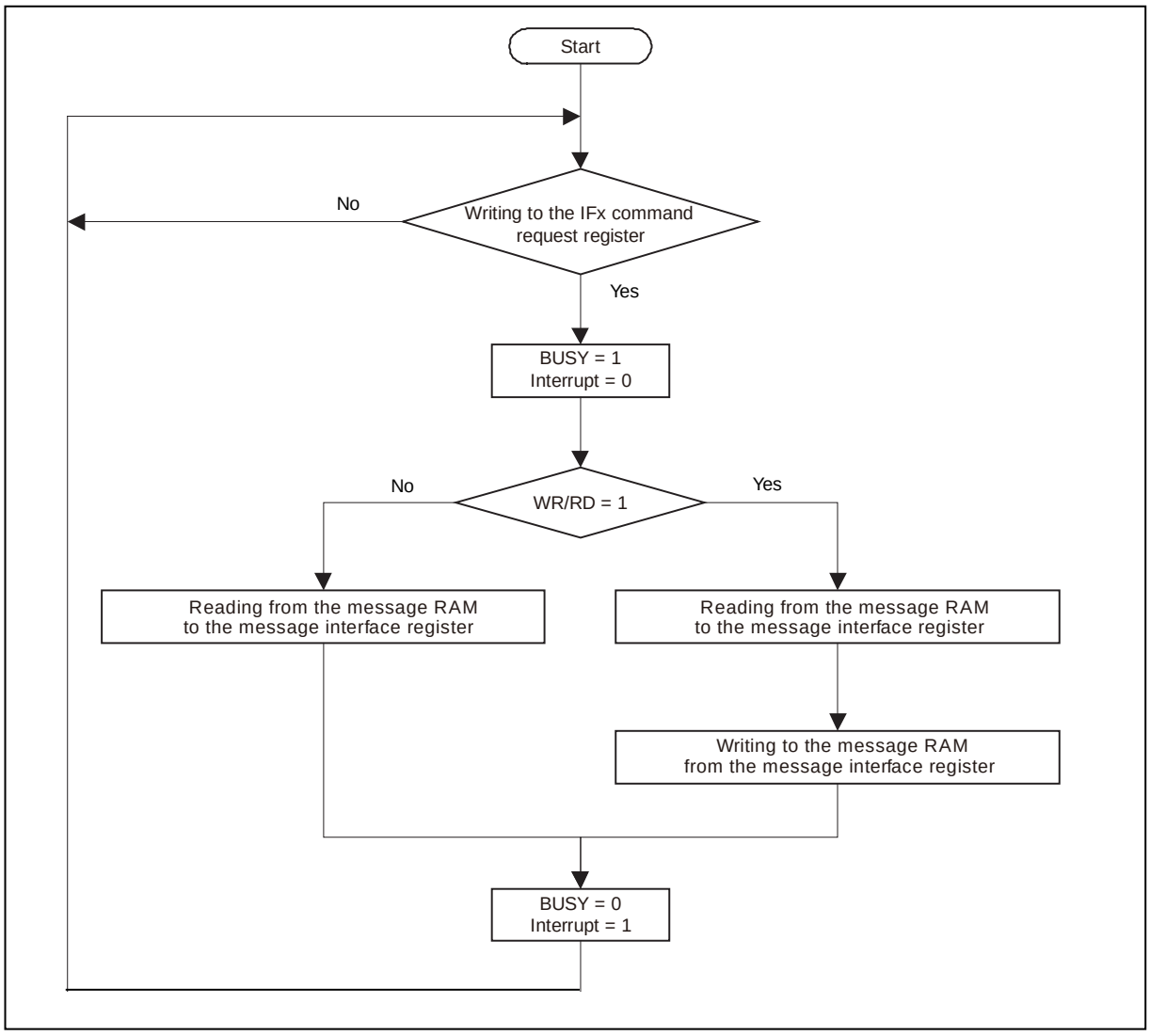
33.5.1.2 Data Transmission/Reception with Message RAM

Data transmission/reception with message RAM is shown.

The BUSY bit of the IFx command request register (IFxCREQ) will be set to "1" when the data transfer between the message interface register and message RAM is started. After the completion of the transfer, the BUSY bit is cleared to "0" (see "Figure 33.5-1 Data Transfer between Message Interface Register and Message RAM").

The IFx command mask register (IFxCMSK) sets whether to transfer the entire or partial data of a message object. Due to the structure of message RAM, it is not possible to write a single bit/byte of the message object to message RAM. The entire data of a single message object is always written to message RAM. Data transfer from the message interface register to message RAM therefore requires a read-modify-write cycle.

Figure 33.5-1 Data Transfer between Message Interface Register and Message RAM



33.5.2 Message Transmission Operation

Message transmission operation is shown.

This section explains the setting method and transmission operation of the transmission message object.

33.5.2.1 Message Transmission

Message transmission is shown.

If there is no data transfer between the message interface register and message RAM, the MsgVal bit of the CAN message valid register (MSGVAL) and the TxRqst bit of the CAN transmission request register (TREQR) will be evaluated. Of all message objects with pending transmission request, a valid message object having the highest priority will be transferred to the transmission shift register. The NewDat bit of the message object will be cleared to "0" at this time.

The TxRqst bit will be reset to "0" if there is no new data in the message object (NewDat=0) when the transmission has ended successfully. If the TxIE is set to "1", the IntPnd bit will be set to "1" after the transmission has ended successfully. If the CAN controller has lost the arbitration on the CAN bus or an error has occurred during the transfer, message will be retransmitted immediately when the CAN bus becomes idle.

33.5.2.2 Transmission Priority

Transmission priority is shown.

Transmission priority of a message object is determined by its message number. Message object 1 has the highest priority; and message object 64 (or the maximum equipped message object number) has the lowest priority. Therefore, if 2 or more transmission requests are pending, message objects will be transferred in the order starting from the message object having the smallest corresponding message number.

Notes:

- When the message object with the lowest priority is used for transmission and the TxRqst is set to "1" and then to "0" to cancel transmission, setting the TxRqst to "1" again may not, depending on the timing, result in transmission of a message until one of the following events occurs:
  - ☐ A valid message is transmitted on the CAN bus.
  - ☐ A transmission request is issued to other message object.
  - ☐ CAN is initialized by the Init bit.

  If there is a situation in which transmission is canceled due to system reasons, either do not use the message object with the lowest priority as the transmission message object or, after transmission cancellation, generate one of the above events and then set the TxRqst to "1" again.
- When the TxRqst bit is "1", do not change the message objects of ID28 to ID0, DLC3 to DLC0, Xtd, and Data7 to Data0. Otherwise, message objects before and after the change may be transmitted in a mixed way or message objects after the change may not be transmitted. Change them when the TxRqst bit is "0".

33.5.2.3 Transmission Message Object Setting

Transmission message object setting is shown.

The initialization method for the transmission object is shown below:

Table 33.5-1 Transmission Message Object Initialization

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The IFx arbitration register (ID28 to ID0 and Xtd bit) is provided by the application, and it defines the ID and type of the transmission message.

ID28 to ID18 will be used and ID17 to ID0 will be disabled if standard frame (11-bit ID) has been set. ID28 to ID0 will be used if extended frame (29-bit ID) has been set.

If the TxIE bit is set to "1", the IntPnd bit will be set to "1" after the transmission of the message object has ended successfully.

If the RmtEn bit is set to "1", the TxRqst bit will be set to "1" and the data frame will be transmitted automatically after receiving the matching remote frame.

Settings for the data registers (DLC3 to 0, Data0 to 7) are provided by the application.

When UMask=1, the IFx mask register (Msk28 to 0, UMask, MXtd and MDir bits) will receive the remote frame having the ID that has been grouped by the mask setting, and then will be used to allow the transmission (sets the TxRqst bit to "1"). See the heading "Remote Frame" in "33.5.3 Message Reception Operation" for details.

Note:

Mask is not allowed for the Dir bit of the IFx mask register.

33.5.2.4 Update of Transmission Message Object

Update of transmission message object is shown.

CPU can update the data of the transmission message object via the message interface register.

Data of the transmission message object will be written in units of 4 bytes of the corresponding IFx data register (IFxDTx) (in unit of the IFx data register A (IFxDTAx) or IFx data register B (IFxDTBx)). Therefore, it is not possible to change only 1 byte of the transmission message object.

0087_H will be written to the IFx command mask register (IFxCMSK) first when updating 8-byte data. Then, data of the transmission message object (8-byte data) will be updated and "1" will be written to the TxRqst bit at the same time when a message number is written to the IFx command request register (IFxCREQ).

If the NewDat bit and TxRqst bit are both "1", the NewDat bit will be reset to "0" when the transmission starts.

Notes:

When updating data, perform it in units of 4 bytes of the IFx data register A (IFxDTAx) or IFx data register B (IFxDTBx).

When the TxRqst bit is "1", do not change the message objects of ID28 to 0, DLC3 to 0, Xtd, and Data7 to 0. Otherwise, message objects before and after the change may be transmitted in a mixed way or message objects after the change may not be transmitted. Change them when the TxRqst bit is "0".

33.5.3 Message Reception Operation

Message reception operation is shown.

This section explains the setting method and reception operation of the reception message object.

33.5.3.1 Reception Message Acceptance Filter

Reception message acceptance filter is shown.

When the arbitration/control field (ID + IDE + RTR + DLC) of the message is completely shifted to the CAN controller reception shift register, scanning of message RAM for a match comparison with the valid message object will be started.

The arbitration field and mask data (including MsgVal, UMask, NewDat and EoB) will be loaded from the message object in message RAM at this time, and the arbitration fields of the message object and shift register will be compared (including mask data).

This operation will be repeated until a match is detected between the arbitration fields of the message object and shift register or until the final word of message RAM is reached. When a match is detected, scanning of message RAM will be stopped and CAN controller will perform different processes according to the type of the reception frame (data frame or remote frame).

33.5.3.2 Reception Priority

Reception priority is shown.

Reception priority of a message object is determined by its message number. Message object 1 has the highest priority; and message object 64 (or the maximum equipped message object number) has the lowest priority. If 2 or more message objects match in the acceptance filter, the one having the smaller message number will be the reception message object.

33.5.3.3 Data Frame Reception

Data frame reception is shown.

CAN controller transfers and stores the reception message from the shift register to message RAM of the message object that matched the acceptance filter. This stored data not only contains data bytes but also all arbitration fields and data length codes. This operation will be performed even if the IFx mask register is set as a mask (stored in order to hold the ID and data bytes).

The NewDat bit will be set to "1" when a new data is received. Reset the NewDat bit to "0" when a message object is read by the CPU. If the NewDat bit is already set to "1" when the message is received, the previous data will be treated as lost and the MsgLst bit will be set to "1".

If the RxIE bit is set to "1", the IntPnd bit of the CAN interrupt pending register (INTPND) will be set to "1" when a message object is received. The TxRqst bit of the message object will be cleared to "0" at this time. This operation is performed to prevent a transmission process from starting when a request data frame is received while the remote frame transmission process is in progress.

33.5.3.4 Remote Frame

Remote frame is shown.

The following 3 processes are performed when the remote frame is received. The appropriate process will be selected from the setting of the matching message object.

1. Dir="1" (Transmission direction), RmtEn="1", UMask="1" or "0"
   The matched remote frame will be received, only the TxRqst bit of this message object will be set to "1", and the automatic reply (transmission) of the data frame in response to the received remote frame will be performed. (The message object will remain unchanged except for the TxRqst bit.)
2. Dir="1" (Transmission direction), RmtEn="0", UMask="0"
   Even if the received remote frame matches the message object, it will not be processed as being received and will be disabled. (The TxRqst bit of the message object will remain unchanged.)
3. Dir="1" (Transmission direction), RmtEn="0", UMask="1"
   If the received remote frame matches the message object, the TxRqst bit of this message object will be reset to "0", and the remote frame will be processed as a reception data frame. The received arbitration field and control field (ID + IDE + RTR + DLC) will be stored to the message object in message RAM, and the NewDat bit of this message object will be set to "1". Data field of the message object will be unchanged.

33.5.3.5 Reception Message Object Setting

Reception message object setting is shown.

The initialization method for the reception message object is shown below:

Table 33.5-2 Reception Message Object Initialization

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The IFx arbitration register (ID28 to 0 and Xtd bit) is provided by the application; and it defines the ID and type of the reception message to be used in the acceptance filter.

ID28 to ID18 will be used and ID17 to ID0 will be disabled if standard frame (11-bit ID) has been set. ID17 to ID0 will be reset to "0" when a standard frame is received. ID28 to ID0 will be used if extended frame (29-bit ID) has been set.

If the RxIE bit is set to "1", the IntPnd bit will be set to "1" when the reception data frame is stored to the message object.

Data length code (DLC3 to 0) is provided by the application. Reception data length code and an 8-byte data will be stored when the CAN controller stores the reception data frame to the message object. If the data length code is less than 8 bytes, undefined data will be written to the remaining data of the message object.

When UMask="1", the IFx mask register (Msk28 to 0, UMask, MXtd and MDir bits) will be used to allow the reception of the data frame having the ID that has been grouped by the mask setting. See "Data Frame Reception" in "5.3 Message Reception Operation" or details.

Note:

The Dir bit of the IFx mask register cannot be set as a mask.

33.5.3.6 Reception Message Processing

Reception message processing is shown.

CPU can read reception messages at any time via the message interface register.

Generally, "007F$_H$" is written to the IFx command mask register (IFxCMSK). Message number of the message object will then be written to the IFx command request register (IFxCREQ). By using this procedure, reception message of the specified message number will be transferred from message RAM to the message interface register. At this time, the NewDat bit and IntPnd bit of the message object can be cleared to "0" by the setting of the IFx command mask register (IFxCMSK).

The message will be received if it matches the acceptance filter. If the acceptance filter mask is used in the message object, the data that has been set as a mask will be excluded from the acceptance filter, and the decision of whether or not to receive the message will be made.

The NewDat bit indicates whether a new message has been received after the message object was last read.

The MsgLst bit indicates that the next reception data has been received before the previously received data is read from the message object, resulting in the loss of the previous data. The MsgLst bit will not be reset automatically.

The TxRqst bit will be cleared to "0" automatically when a data frame matching the acceptance filter is received while the remote frame transmission is being processed.

33.5.4 FIFO Buffer Function

FIFO buffer function is shown.

This section explains the configuration and operation of the FIFO buffer of the message object in the reception message processing.

33.5.4.1 Configuration of FIFO Buffer

The configuration of FIFO buffer is shown.

The configuration of the reception message objects in the FIFO buffer is the same as that of other reception message objects, except for the EoB bit (see "33.5.3 Message Reception Operation" for the reception message object setting).

FIFO buffer is used by linking 2 or more reception message objects. When using the ID and mask of the reception message object, it is necessary to match those settings in order to store the reception message to this FIFO buffer.

The first reception message object of the FIFO buffer will be the message object having the highest priority (smallest message number). The EoB bit of the final reception message object of the FIFO buffer must be set to "1" to indicate the end of the FIFO buffer (Set the EoB bit to "0" for message objects other than the final message object that uses the configuration of the FIFO buffer).

Notes:

- Always make the same settings for ID and mask setting of the message object to be used in the FIFO buffer.
- Always set the EoB bit to "1" when FIFO buffer is not used.

33.5.4.2 Message Reception by FIFO Buffer

Message reception by FIFO buffer is shown.

When a received message matches with the FIFO buffer ID, it is stored into the message object with the lowest message number of the FIFO buffer.

The NewDat bit of this reception message object will be set to "1" when the message is stored to the reception message object in the FIFO buffer. When the NewDat bit is set to the reception message object whose EoB bit is "0", a write operation to the FIFO buffer by the CAN controller will not be performed as the reception message object will be protected until the final reception message object (EoB bit = "1") is reached.

When both of the following conditions are met, the next incoming message is written to the last message object and therefore overwrites the previous message.

- Valid data is stored into the last FIFO buffer
- The NewDat bit of the receive message object is not written by "0" (to release the write protect)

33.5.4.3 Reading from FIFO Buffer

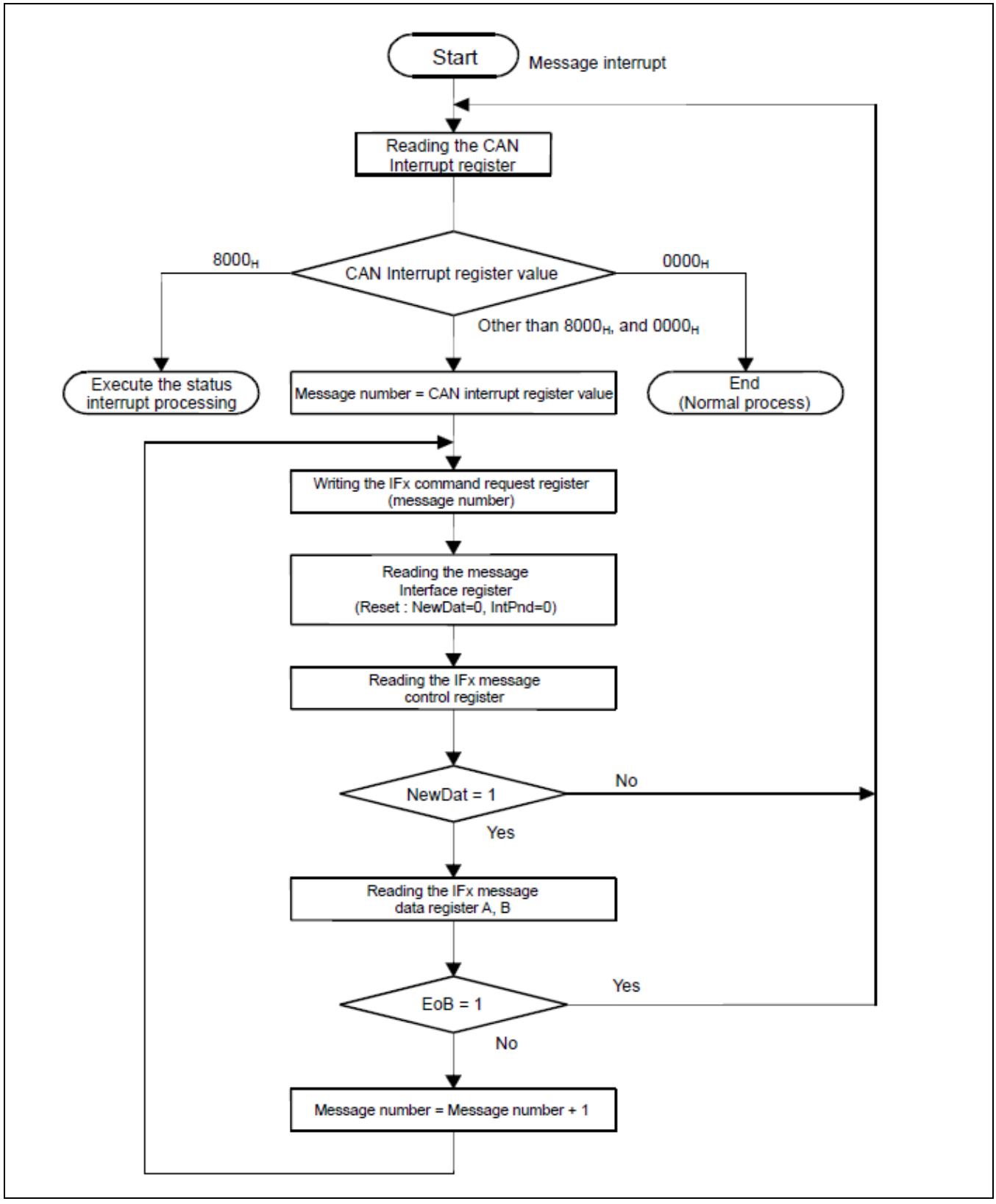
Reading from FIFO buffer is shown.

CPU can read the content of the received message object by writing the reception message number to the IFx command request register (IFxCREQ) that will cause the message object to be transferred to the message interface register. Set the WR/RD to "0" (read), set the TxRqst/NewDat and the IntPnd to "1" and reset NewDat and IntPnd bits to "0" in the IFx command mask register (IFxCMSK) at this time.

In order to guarantee the function of the FIFO buffer, always read the reception message objects in the FIFO buffer starting from the one having the smallest message number.

The figure below shows the CPU processing method for the message objects that are linked in the FIFO buffer.

Figure 33.5-2 CPU Processing of FIFO Buffer



33.5.5 Interrupt Function

Interrupt function is shown.

This section explains the processing of status interrupt (IntId=8000_H) and message interrupt (IntId=message number).
If 2 or more interrupts are pending, the CAN interrupt register (INTR) will indicate the pending interrupt code of the highest priority interrupt. High priority interrupt codes will always be displayed, ignoring the chronological order in which the interrupt codes were set. Interrupt code will be held until it is cleared by CPU.
Status interrupt (IntId bit = 8000_H) has the highest priority.
Priority of message interrupts becomes higher as the message number gets smaller, and vice versa.

Message interrupt will be cleared when the IntPnd bit of the message object is cleared. Status interrupt will be cleared when the CAN status register (STATR) is read.

The IntPnd bit of the CAN interrupt pending register (INTPND) indicates whether any interrupt exists. The IntPnd bit will indicate "0" if there is no pending interrupt.

The interrupt signal to the CPU will become active when the IntPnd bit becomes "1" while the IE bit of the CAN control register (CTRLR) and TxIE and RxIE bits of the IFx message control register (IFxMCTR) are set to "1". The interrupt signal maintains its active state until the CAN interrupt pending register (INTPND) is cleared to "0" (interrupt factor reset) or until IE bit of the CAN control register (CTRLR) is reset to "0".

The CAN interrupt register (INTR) being set to "8000_H" indicates an update of the CAN status register (STATR) by the CAN controller; and this interrupt will have the highest priority. The interrupt generated by updating the CAN status register (STATR) can allow or prohibit the setting of the CAN interrupt register (INTR) by using EIE and SIE bits of the CAN control register (CTRLR). Interrupt signal to the CPU can be controlled by the IE bit of the CAN control register (CTRLR).

The RxOk bit, TxOk bit and LEC bit of the CAN status register (STATR) can be updated (reset) by a write from the CPU. However, interrupt cannot be set or reset by the write operation.
The CAN interrupt register (INTR) set to other than "8000_H" and "0000_H" indicates that the message interrupt is currently pending and that it has a high priority.
The CAN interrupt register (INTR) will be updated even when IE has been reset.

Message interrupt factor to the CPU can be confirmed in the CAN interrupt register (INTR) or CAN interrupt pending register (INTPND). See "4.5 Message Handler Registers." When clearing a message interrupt, it is possible to read the message data at the same time. When the message interrupt specified by the CAN interrupt register (INTR) is cleared, the next priority interrupt will be set to the CAN interrupt register (INTR), waiting for the next interrupt process. The CAN interrupt register (INTR) will indicate "0000_H" if there is no interrupt.

Notes:

- Status interrupt (IntId=8000_H) will be cleared by a read access from the CAN status register (STATR).
- Status interrupt (IntId=8000_H) by a write access to the CAN status register (STATR) will not be generated.

33.5.6 Bit Timing and CAN System Clock (fsys) Generation

Bit timing and CAN system clock (fsys) generation is shown.

This section explains the overview of bit timing and its role in the CAN controller.

Each CAN node of the CAN network has a clock oscillator (normally a crystal oscillator). Time parameter of bit time can be configured individually for each CAN node. A common bit rate can be produced even if the oscillation cycle of each CAN node is different.

Frequency of these oscillators differ slightly by temperature/voltage change or component deterioration. CAN node can compensate different bit rates by resynchronizing to the bit stream, as long as this fluctuation falls within the tolerance range of the oscillator.

The bit time is divided into the following 4 segments (see "Figure 5-4 Bit Timing") according to the CAN specification: synchronization segment (Sync_Seg), transmission time segment (Prop_Seg), phase buffer segment 1 (Phase_Seg1) and phase buffer segment 2 (Phase_Seg2). Each segment consists of a programmable time quantum (see "Table 5-3 CAN Bit Time Parameters"). Basic unit time (tq) of the bit time is defined by the system clock fsys of the CAN controller and baud rate prescaler (BRP).

tq = BRP / fsys

CAN system clock fsys will be generated as shown in the figure 3-1. Sync_Seg of the synchronization segment will be the timing within the bit time expecting the edge of the CAN bus. Prop_Seg of the transmission time segment compensates the physical delay time in the CAN network. Phase_Seg1 and Phase_Seg2 of the phase buffer segment specify the sampling point. Resynchronization jump width (SJW) defines the displacement of the sampling point at resynchronization in order to compensate the edge phase error.

Figure 33.5-3 Schematic Diagram of CAN System Clock (fsys) Generation

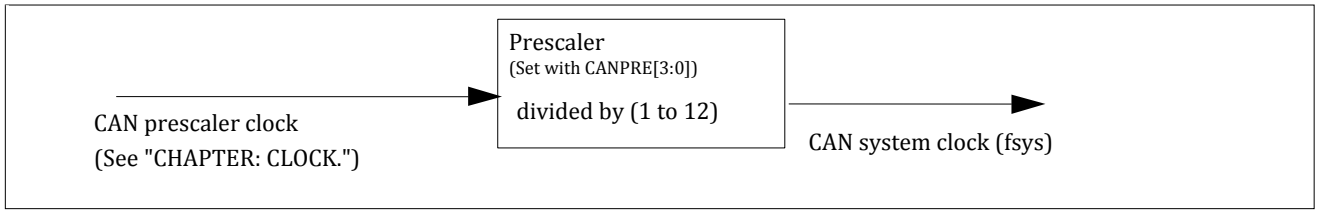


Figure 33.5-4 Bit Timing

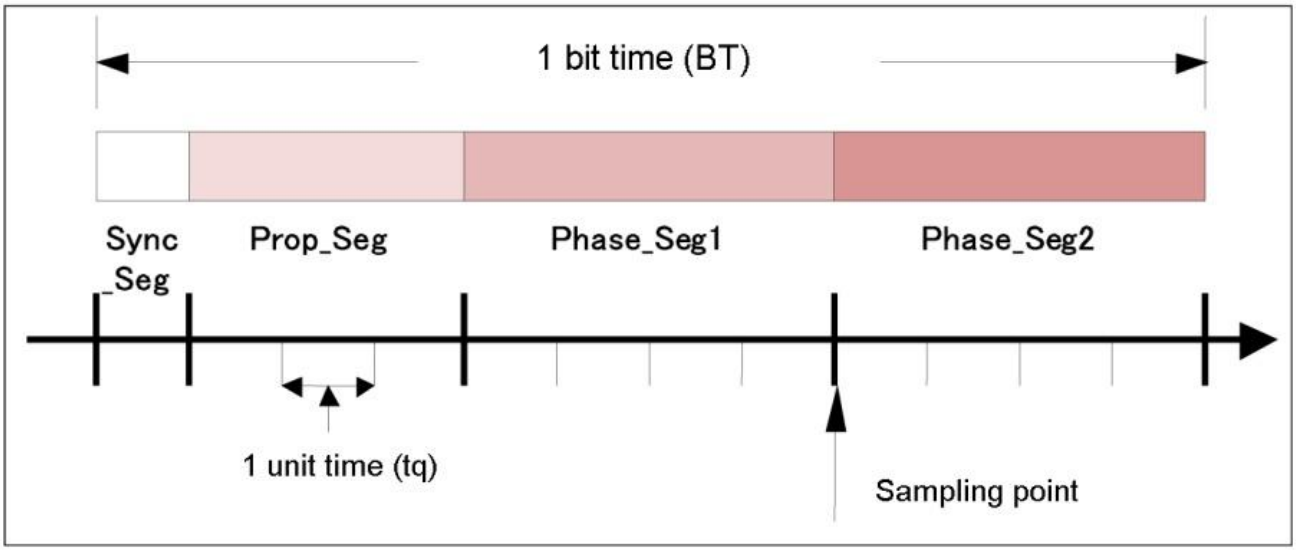


Table 33.5-3 CAN Bit Time Parameters

Parameter	Range	Function
BRP	[1 to 32]	Defines the time quantity tq
Sync_Seg	1 tq	Fixed length. Synchronizes the bit time with the system clock.
Prop_Seg	[1 to 8] tq	Compensates for physical delay time.
Phase_Seg1	[1 to 8] tq	Guarantees identification of edge-phase errors prior to the sample point. The bit time may be temporarily prolonged due to synchronization.
Phase_Seg2	[1 to 8] tq	Guarantees identification of edge-phase errors subsequent to the sample point. The bit time may be temporarily shortened due to synchronization.
SJW	[1 to 4] tq	Defines the resynchronization jump width. It will not be greater than either of the phase buffer segments.

The bit timing effected by the CAN controller is shown in the following.

Figure 33.5-5 Bit Timing Effected by the CAN Controller

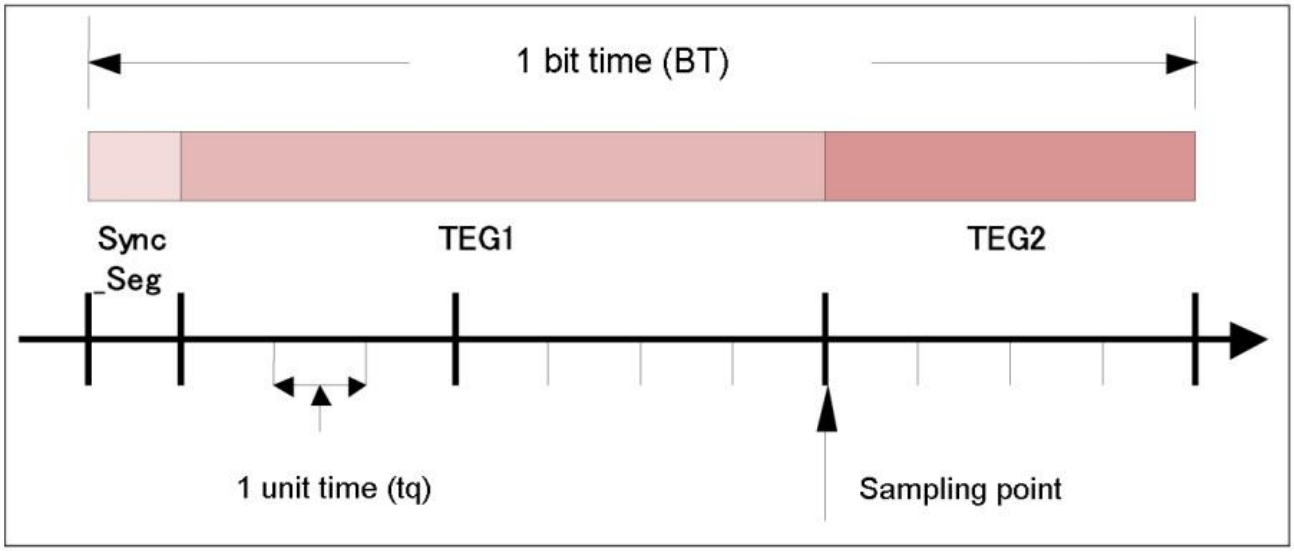


Table 33.5-4 CAN Controller Parameters

Parameter	Range	Function
BRPE, BRP	[0 to 1023]	Defines the length of time quantum tq. Can extend the prescaler by up to 1024 by the CAN Bit Timing Register and the CAN Prescaler Extension Register.
Sync_Seg	1 tq	Synchronizes the bit time with the system clock. Fixed length
TSEG1	[1 to 15] tq	Time segment prior to the sample point. This corresponds to Prop_Seg and Phase_Seg1. This width can be controlled using the bit timing register.
TSEG2	[0 to 7] tq	Time segment subsequent to the sample point. This corresponds to Phase_Seg2. This width can be controlled using the bit timing register.
SJW	[0 to 3] tq	Defines the resynchronization jump width. This width can be controlled using the bit timing register.

The relationships among the parameters are as follows.

Tq = ([BRPE,BRP]+1) / fsys

BT = Sync_Seg + TEG1 + TEG2

= (1 + (TSeg1 + 1) + (TSeg2 + 1)) x tq

= (3 + TSeg1 + TSeg2) x tq

33.5.7 Test Mode

Test mode is shown.

This section explains the test mode setting method and operation.

33.5.7.1 Test Mode Setting

Test mode setting is shown.

The CAN controller enters test mode when the Test bit of the CAN control register (CTRLR) is set to "1". In test mode, the bits Tx1, Tx0, LBack, Silent, and Basic of the CAN test register (TESTR) are valid.

When the Test bit in the CAN Control Register is set to "0", all CAN Test Register functions (except Rx bit) are disabled.

33.5.7.2 Silent Mode

Silent mode is shown.

The CAN controller enters silent mode when the Silent bit of the CAN test register (TESTR) is set to "1".

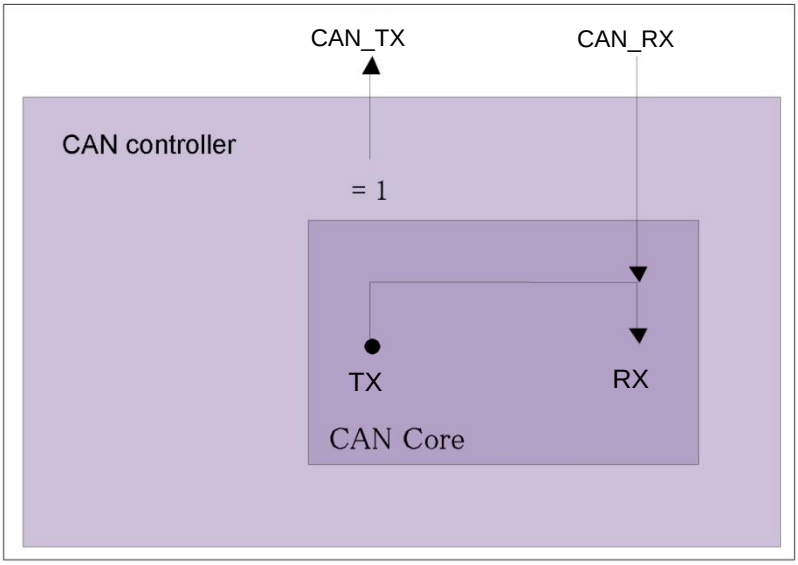
In silent mode, the CAN controller can receive data frames and remote frames, but only outputs a recessive level to the CAN bus and does not send messages or ACKs.

When the CAN controller is requested to send a dominant bit (the ACK bit, overload flag, or active error flag), it sends the dominant bit to the RX end through a loopback circuit within the CAN controller. During this operation, the receiving end can receive the dominant bit that is sent through the loopback circuit within the CAN controller even if the CAN bus is in the recessive-level state.

In silent mode, traffic over the CAN bus can be analyzed without influence from the transmission of dominant bits (ACK bits and error flags).

The figure below shows how signals TX and RX are connected to the CAN controller in silent mode:

Figure 33.5-6 CAN Controller in Silent Mode



33.5.7.3 Loopback Mode

Loopback mode is shown.

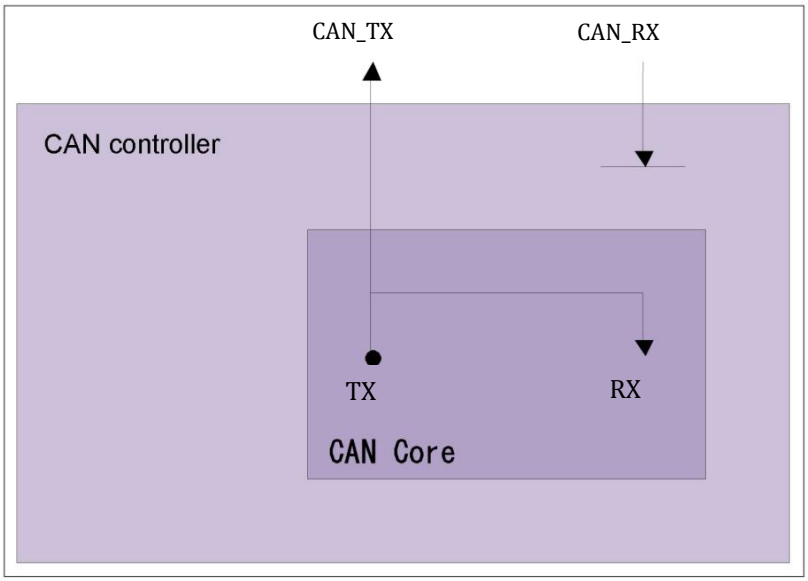
The CAN controller enters loopback mode when the LBack bit of the CAN test register (TESTR) is set to "1".

Loopback mode can be used for self-diagnostics.

In loopback mode, the TX end and the RX end are connected within the CAN controller, messages sent by the CAN controller are handled as messages received by the RX end, and messages that have passed through the acceptance filter are stored in the receive buffer.

The figure below shows how signals CAN_TX and CAN_RX are connected to the CAN controller in loopback mode:

Figure 33.5-7 CAN Controller in Loopback Mode



Note:

Dominant bits from the acknowledge slot of data/remote frames are not sampled to ensure that they are left independent of external signals. Therefore, the CAN controller will not generate acknowledge errors in test mode although it may generate these errors in other mode.

33.5.7.4 Combination of Silent and Loopback Modes

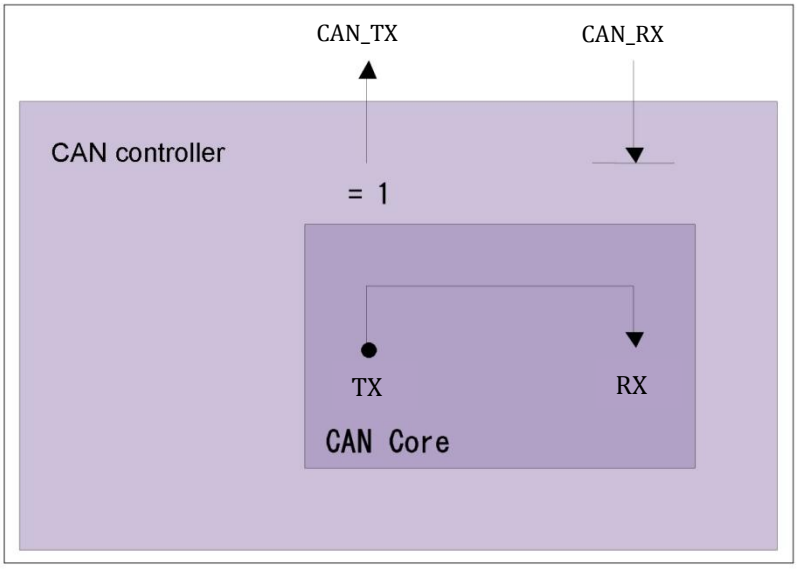
Combination of silent and loopback modes is shown.

The CAN controller can work in a mode that combines loopback and silent modes, when the LBack and Silent bits of the CAN test register (TESTR) are set to "1" simultaneously.

This combined mode can be used for hot self-tests. "Hot self-test" means that when the CAN controller is in process of tests in loopback mode, CAN system operation receives no influence from these tests because a fixed recessive-level output is at the CAN_TX signal and the input from the CAN_RX signal is invalid.

The figure below shows how signals TX and RX are connected to the CAN controller in the silent and loopback combined mode:

Figure 33.5-8 CAN Controller in the Silent and Loopback Combined Mode



33.5.7.5 Basic Mode

Basic mode is shown.

The CAN controller enters basic mode when the Basic bit of the CAN test register (TESTR) is set to "1".

In basic mode, the CAN controller works without using the message RAM.

The IF1 message interface register is used for transmission control.

The message transmission procedure begins with the setting of the send data in the IF1 message interface register. The next step is to set the BUSY bit of the IF1 command request register to "1" to issue a transmission request. While the BUSY bit is set to "1", the IF1 message interface register is locked or transmission is held.

When the BUSY bit is set to "1", the CAN controller performs the following operation:

As soon as the CAN bus becomes idling, the CAN controller begins transmission by loading the content of the IF1 message interface register to the transmission shift register. When transmission ends normally, the BUSY bit is reset to "0", and the locked IF1 message interface register is released.

While transmission is held, it can be suspended anytime by resetting the BUSY bit of the IF1 command request register to "0". When the BUSY bit is reset to "0" during transmission, retransmission that would be initiated after an arbitration loss or error will not be initiated.

The IF2 message interface register is used for reception control.

When the BUSY bit is set to "1", the CAN controller performs the following operation:

The CAN controller stores the received message (content of the reception shift register) in the IF2 message interface register without using the acceptance filter.

If the CAN controller has stored a new message in the IF2 message interface register, it sets the NewDat bit to "1". If the CAN controller receives a further new message when the NewDat bit is "1", it sets the MsgLst bit to "1".

Notes:

- In basic mode, all message objects relating to the control/status bits and the control mode settings on the IFx command mask register (IFxCMSK) are invalidated.
- The message number in the command request register is invalid.
- On the IF2 message control register, the NewDat and MsgLst bits work as usual, the DLC3 to 0 bits identify the received DLC, and the other control bits are read as "0".

33.5.7.6 Software Control of the CAN_TX signal

Software control of the CAN_TX signal is shown.

The CAN_TX signal, which is the CAN transmission pin, has 4 output functions as follows.

- Serial data output (ordinary output)
- CAN sampling point signal output for CAN controller bit timing monitoring
- Fixed dominant output
- Fixed recessive output

Fixed dominant and recessive outputs can be used to check the physical layer of the CAN bus together with the RX monitoring function of the CAN reception pin.

The CAN_TX signal output mode can be controlled using the Tx1 and Tx0 bits of the CAN test register (TESTR).

Note:

For CAN message transmission or operation in loopback, silent, or basic mode, the CAN_TX signal must be configured for serial data output.

33.5.8 Software Initialization

Software initialization is shown.

Software-controlled initialization is as follows:

The sources of software initialization are as follows:

- Hardware reset
- Setting the Init bit in the CAN Control Register
- Shift to a busoff state

A hardware reset resets all other than the message objects (excluding the MsgVal, NewDat, IntPnd, and TxRqst bits). Initialize the message objects by one of the following steps after hardware reset.

1. Initialize all message objects by the CPU.
2. Initialize only the used message objects by CPU. Clear MsgVal bit to "0" in the other message objects.

The CAN Bit Timing Register must be configured before clearing the Init bit in the CAN Control Register to "0".

The Init bit in the CAN Control Register is set to "1" in the following conditions:

- Writing "1" from the CPU
- Hardware reset
- In a busoff state

When the Init bit is set to "1", all message transfer from/to the CAN bus is stopped, and the CAN_TX siganl in the CAN bus output is in a recessive state (excludes software control of the CAN_TX signal).

Setting the Init bit to "1" does not change the error counter and any register.

When the Init bit and CCE bit in the CAN Control Register are set to "1", the CAN Bit Timing Register for baud rate control and CAN Prescaler Extension Register can be configured.

The software initialization is completed by resetting the Init bit to "0".

By waiting for the occurrence of a consecutive 11 recessive bits (i.e., bus idle) after the Init bit is reset to "0", the message is transferred after synchronization with data transfer on the CAN bus.

Before changing message object masks ID, Xtd, EoB, and RmtEn during normal operation, the MsgVal must be disabled.

33.5.9 CAN Wake Up Function

CAN Wake Up function is shown.

It can be wake up in the reception operation of the CAN by connecting the CAN_RX signal of the CAN with the external interrupt signal.

About the signal used by the CAN wake up function

Because the CAN_RX signal and the INT0 signal are shared, the wake up function can be used.

Table 5-5 shows the relation among the CAN wake up function, the CAN_RX signal, and the INT signal.

Table 33.5-5 Signals of CAN Wake Up Function

	CAN_RX signal	Interrupt Function
CAN	RX(57)	INT0

About CAN wake up function

It is possible to return from the sleep mode or the standby mode by the reception data of the CAN.

Note:

It is necessary to set an external interrupt before it shifts to the sleep mode or the standby mode when the wake up function is used.

33.6 Limitations

This section explains limitations of the CAN.

33.6.1 Init Bit

33.6.1 Init Bit

This section shows the Init bit.

33.6.1.1. Limitations

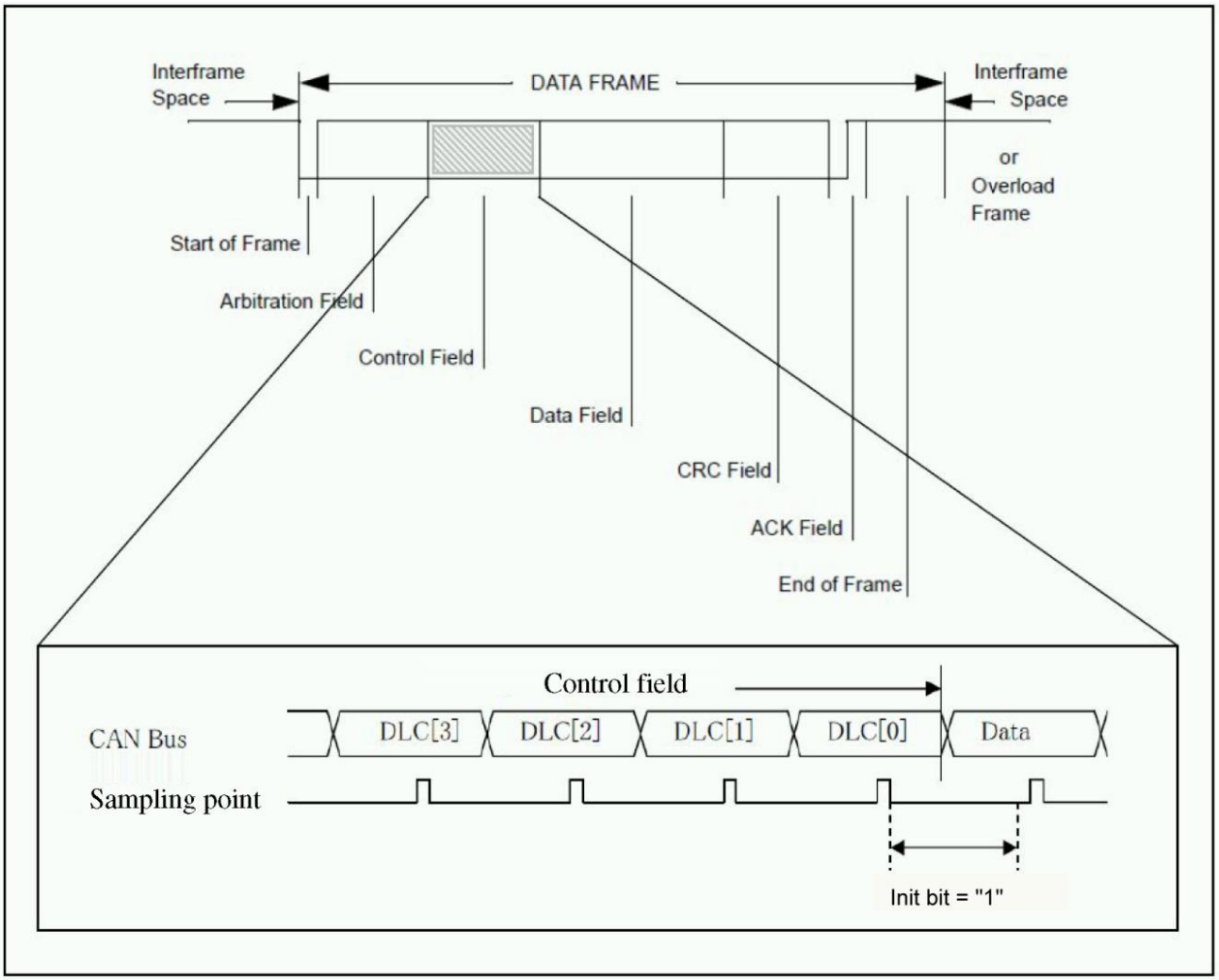
33.6.1.2. Workarounds

33.6.1.1 Limitations

This section explains limitations.

Setting Init bit = "1" (Figure 6-1) in the CAN control register (CTRLR) during transmission of the last bit of a control field has the following effect. In the first frame transmitted after the Init bit is cleared, the data field is shifted by 1 bit to the left.
All subsequent messages are transmitted correctly.
Remote frames and data frames with a data length of 0 are not affected by this timing of the setting of the Init bit.

Figure 33.6-1 Operation with Limitations



33.6.1.2 Workarounds

This section explains workarounds.

Avoid this problem in one of the following ways.

1. Setting "1" in the Init bit in the CAN control register (CTRLR)
   Immediately after the completion of transmission, set "1" in the Init bit in the CAN control register (CTRLR).
2. Setting "1" in the Init bit in the CAN control register (CTRLR) during transmission, and then setting "0" in the Init bit at the desired transmission time
   First, set "1" in the Init bit. Respond to the message object whose transmission request bit (TxRqst) is set to "1", by canceling transmission (set "0" in the TxRqst bit). Then, set "0" in the Init bit. After that, wait 2 bit times of the CAN before setting "1" in the transmission request bit (TxRqst) of the transmitted message object.

34. CAN Clock Prescaler

This chapter explains the CAN clock prescaler.

34.1 Overview

34.2 Features

34.3 Configuration

34.4 Registers

34.1 Overview

This section gives an overview of the CAN clock prescaler.

This module generates clocks (fsys) supplied from each clock source to the CAN macro. Figure 34-1 shows CAN, the CAN interface, the CAN clock prescaler and clock source selector circuit.

34.2 Features

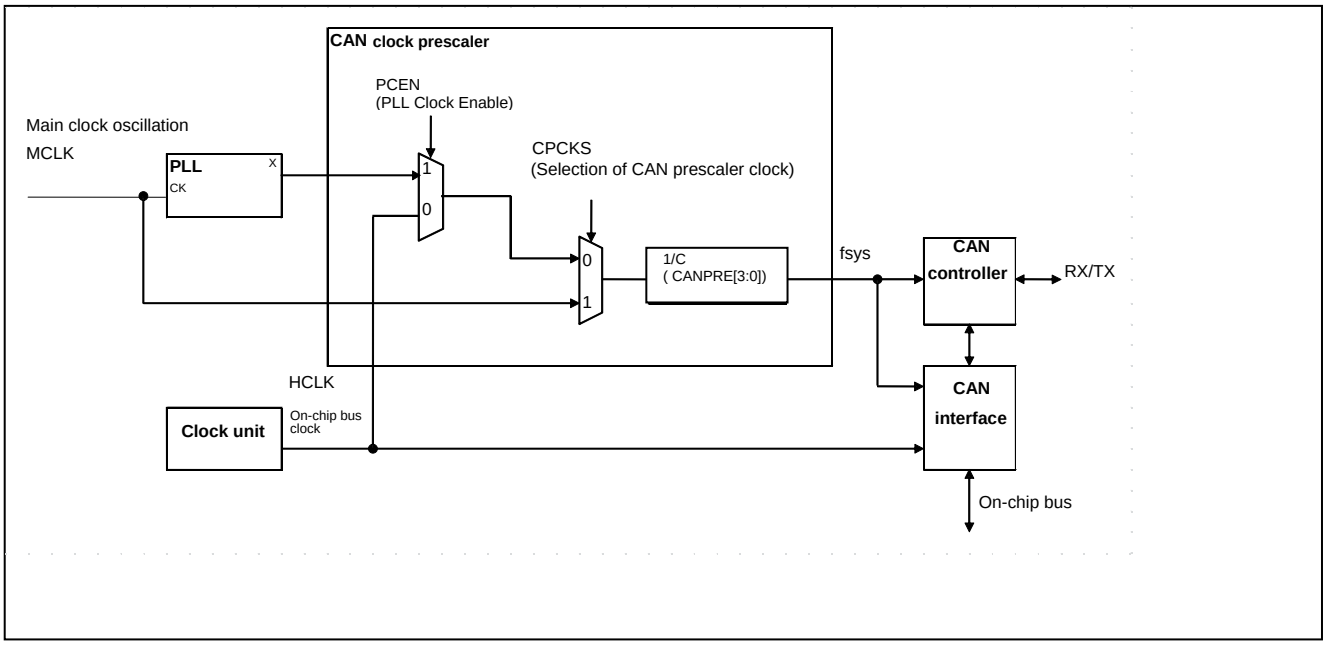
This section explains features of the CAN clock prescaler.

- As a source clock of the CAN clock prescaler, PLL clock/HCLK or main oscillation can be selected.
- PLL clock/HCLK is changed by PLL oscillation enable (PCEN).
- The counter which can divide CAN system clock frequency (fsys) by C (C=1 to12) is installed.

34.3 Configuration

This section shows configuration of the CAN clock prescaler.

Figure 34-1. Block Diagram of CAN Clock Prescaler



34.4 Registers

This section shows registers of the CAN clock prescaler.

Table 34-1. Register Map

Address	Register				Register Function
	+0	+1	+2	+3	
0x04A4	CANPRE	Reserved	Reserved	Reserved	CAN clock prescaler control register

34.4.1 CAN Prescaler Register: CANPRE

The bit configuration of CAN prescaler register is shown.

This register sets the CAN system clock (fsys) generation prescaler. For details, see "5.6. Bit Timing and CAN System Clock (fsys) Generation" in "Chapter: CAN."

When changing the value of this register, set the initialization bit (Init) in the CAN control register (CTRLR) to "1" in order to stop all the bus operations.

CAN Prescaler Register: Address $04A4_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved		CPCKS	CANPRE3	CANPRE2	CANPRE1	CANPRE0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W

[bit7] Reserved

Be sure to write "0" to this bit.

[bit6, bit5] Reserved

The value read is always "0". Be sure to write "0" to these bits.

[bit4] CAN prescaler clock select bits

CPCKS	CAN Prescaler Source Clock
0	PLL clock/HCLK(on-chip bus clock)
1	Main oscillation (MCLK)

[bit3 to bit0] CAN prescaler setting bits

CANPRE [3:0]	Functions	Input CAN Prescaler Clock: 80 MHz	Input CAN Prescaler Clock: 64 MHz	Input CAN Prescaler Clock: 48 MHz
0000	Selects 1/1 period of the system clock as the CAN clock. (Initial value: CANPRE[3:0]=0000)	80 MHz	64 MHz	48 MHz
0001	Selects 1/2 period of the system clock as the CAN clock.	40 MHz	32 MHz	24 MHz
001x	Selects 1/4 period of the system clock as the CAN clock.	20 MHz	16 MHz	12 MHz
01xx	Selects 1/8 period of the system clock as the CAN clock	10 MHz	8 MHz	6 MHz
1000	Selects 2/3 period of the system clock as the CAN clock. The duty of the clock is 67 %.	53.3 MHz	42.7 MHz	32 MHz
1001	Selects 1/3 period of the system clock as the CAN clock.	26.7 MHz	21.4 MHz	16 MHz
1010	Selects 1/6 period of the system clock as the CAN clock.	13.3 MHz	10.7 MHz	8 MHz
1011	Selects 1/12 period of the system clock as the CAN clock.	6.7 MHz	5.4 MHz	4 MHz

CANPRE [3:0]	Functions	Input CAN Prescaler Clock: 80 MHz	Input CAN Prescaler Clock: 64 MHz	Input CAN Prescaler Clock: 48 MHz
110x	Selects 1/5 period of the system clock as the CAN clock.	16.0 MHz	12.8 MHz	9.6 MHz
111x	Selects 1/10 period of the system clock as the CAN clock.	8.0 MHz	6.4 MHz	4.8 MHz

Notes:

- Before changing the CAN prescaler setting bit, set the initialization bit in the CAN control register (CTRLR) to "1" in order to stop all the bus operations.
- By setting the register, be sure to keep the frequency of a clock, supplied to the CAN interface,16 MHz or less.

35. 12-bit A/D Converter

This chapter explains the 12-bit A/D converter.

35.1 Overview

35.2 Features

35.3 Configuration

35.4 Registers

35.5 Operation

35.6 Notes

35.1 Overview

This section explains the overview of the 12-bit A/D converter.

The 12-bit A/D converter can convert analog input voltages to 12-bit digital values using the RC successive approximation conversion method. It begins A/D conversion when an A/D activation trigger is entered. When an A/D activation trigger is entered again during A/D conversion, another sequence of A/D conversion starts. The converter also supports forced stop due to an A/D conversion cancel input signal.

35.2 Features

This section explains features of the 12-bit A/D converter.

35.2.1 Function of A/D Activation Compare

35.2.2 Function of A/D Activation Arbitration

35.2.3 Functions of 12-bit A/D Converter Control

35.2.1 Function of A/D Activation Compare

The function of A/D activation compare is explained.

Analog input control

This function can enable/disable max. 8 channels of analog inputs with the A/D converter of 1 unit.

Activation channel

- It performs operation for A/D activation request control and A/D conversion data storage with each activation channel.
- The A/D activation channel corresponds to 12-bit A/D converter. The correspondence is as follows.

Table 35-1. Channel Allocation

MB91F552	ch.0 to ch.7

- Each activation channels is composed of the following registers.
  - ☐ Compare buffer register/Compare register
  - ☐ A/D activation trigger control status register
  - ☐ A/D data register
  - ☐ A/D activation trigger extend control register
- A/D activation trigger control status extend register

A/D activation request

- Each activation channel issues an A/D activation request by one of the following methods: software, external trigger (falling), reload timer (rising), and PWM. A given activation channel cannot reissue an A/D activation request during A/D conversion that was initiated by that channel.
- For software activation, external trigger, reload timer, and PWM activation, any activation channel can be selected.
- For each activation channel, either single mode or repeat mode can be specified as the activation request method.
- In single mode, one activation request is issued when one activation factor is encountered. One sequence of A/D conversion is performed and the activation request is reset when the A/D conversion completes.
- In repeat mode, activation requests are issued in succession as triggered by one activation factor. A/D conversion is performed repeatedly and the activation request continues its effect as long as repeat mode prevails.

A/D conversion data

- When A/D conversion completes, conversion-result data is stored in the A/D data register. One A/D data register is assigned to each activation channel.
- Each A/D data register contains error flag and error status bits, whose values indicate the status of the A/D conversion data.

Interrupt request

Each activation channel can generate an interrupt request when A/D conversion completes.

Data protection function

- Each A/D data register can be configured to set the data protection function.
- If the data protection function is enabled, A/D activation requests are masked until data is read from the A/D data register and the interrupt flag is cleared. Data reading and interrupt flag clearing may be in any sequence. In addition, whether clearing of the interrupt flag is included in the condition can be configured.
- Whether an A/D activation request or a conversion operation is in progress can be reported using the A/D activation request bit. Moreover, the current A/D conversion request or conversion operation can be forced to terminate by resetting the A/D activation request bit to "0".

35.2.2 Function of A/D Activation Arbitration

The function of A/D activation arbitration is explained.

- A/D activation arbitration is provided for each 12-bit A/D converter unit. For details on assignments for A/D activation arbitration for each A/D activation channel, see "Table 35-1 Channel Allocation ."
- The A/D activation arbitration consists of arbitration circuit, an A/D activation trigger generation section, and an analog channel number select section.
- Activation requests from A/D activation compare is arbitrated and the activation triggers, A/D conversion cancel signals, and analog channel numbers are generated.
- An activation trigger is generated for one selected activation request from an A/D activation compare channel. If a contention occurs between activation requests from A/D activation compare channels, A/D activation arbitration uses priority control. The priority order is as follows: "Lower activation channel numbers are assigned higher priorities (priority control based on channel numbers)" or "external trigger/reload timer/PWM activation by activation request > software activation (priority control based on activation factors)." Activation requests that are not selected are made to wait. When the current A/D conversion completes, arbitration restarts.
  Priority control based on activation factors is also performed during A/D conversion. In this case, the current conversion is suspended and the activation factor assigned a higher priority is serviced. The suspended activation factor will be activated again after the higher-priority conversion is processed if the re-arbitration process does not find a lower channel number of a higher-priority activation factor.
  - ☐ If an activation factor with the same priority is encountered during A/D conversion suspension:
    The request from the activation channel with the lower number is processed first.
  - ☐ If an activation factor with a different priority is encountered during A/D conversion suspension:
    The request based on the higher-priority activation factor is processed first.
  - ☐ If an activation factor with a higher priority is encountered during A/D conversion:
    The current conversion is suspended, and the higher-priority activation factor is processed. After this processing completion, arbitration is performed again. The suspended activation factor is then processed.
  - ☐ If an activation factor with a lower priority is encountered during A/D conversion:
    After the current conversion completions, arbitration is performed again. The activation factor with the lower priority is then processed.
  - ☐ If an activation factor with the same priority is encountered during A/D conversion:
    After the current conversion completes, arbitration is performed again. The activation factor with the same priority is then processed.
- A conversion cancel signal is generated to force the current conversion processing to termination when the activation factor that is in process of conversion becomes inactive and there is no other active activation factor.
- For the analog channel number, the activation request analog number entered from the activation channel of the activation request arbitration result is selected.

35.2.3 Functions of 12-bit A/D Converter Control

The function of 12-bit A/D converter control is explained.

- There are 12-bit A/D converters, to which analog input pins are assigned. Please refer to "Table 35-1 Channel Allocation " for the correspondence of each input pin and the unit.

A function used to A/D-convert analog voltages (input voltages) input to the analog input pins to digital values is provided. Its features are as follows.

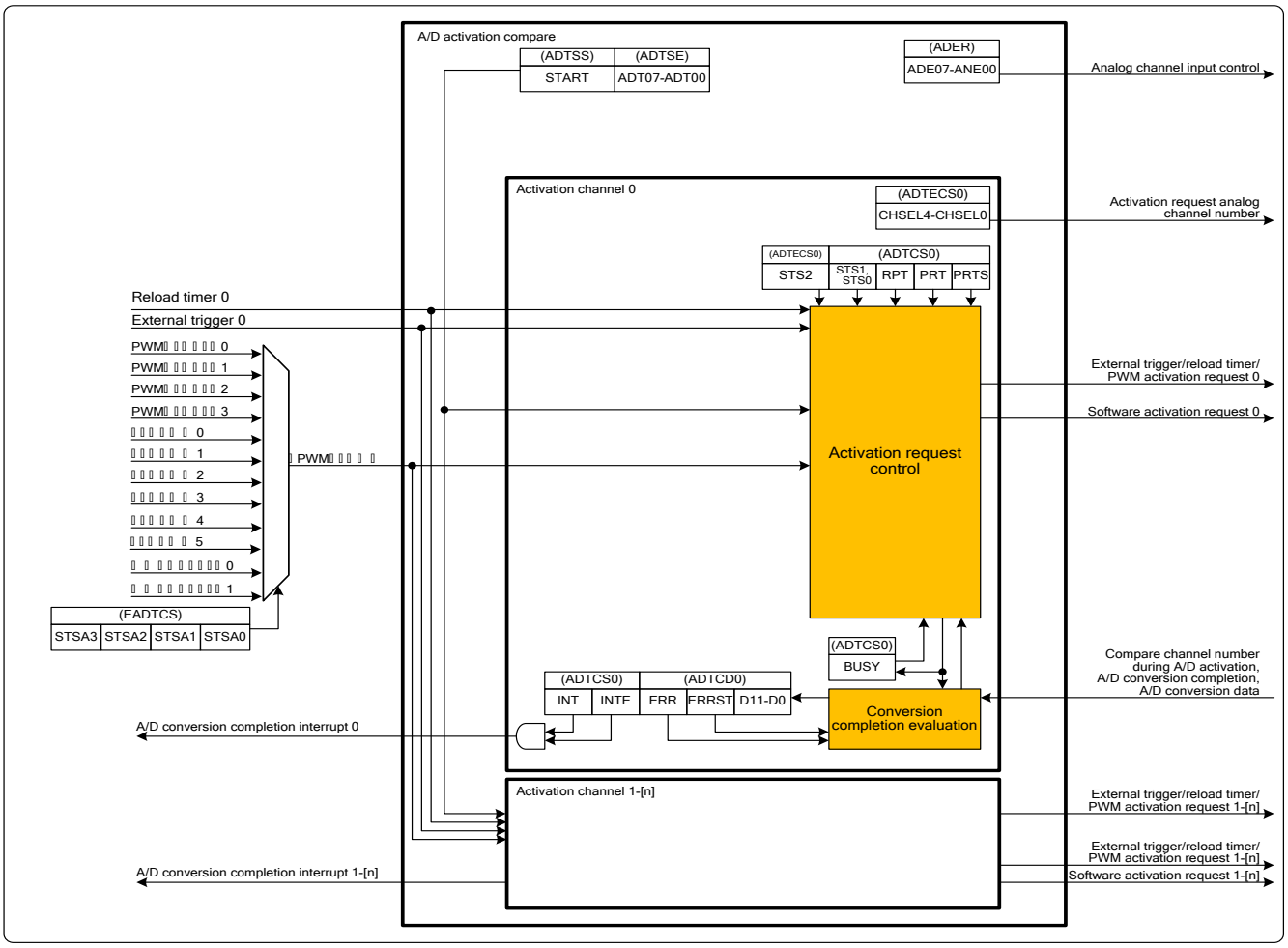
- The conversion time is at least 1.0 µs (including sampling time).
- The conversion method is the RC successive approximation conversion method with a sample hold circuit.
- The analog input pins can be selected under program control. (This program is configured in the A/D activation compare section.)
- Activation signals are entered as pulse signals.
- One sequence of A/D conversion is performed for one activation factor.
- If an activation signal is entered again during A/D conversion, reactivation is performed. (Reactivation function)
- If an A/D conversion cancel signal is received during A/D conversion, the current processing is stopped and initialization takes place. (Forced stop function)
- As for the setting of the sampling time, a common sampling time setting to all channels and the sampling time settings of each channel can be selected.

35.3 Configuration

This section explains the configuration of the 12-bit A/D converter.

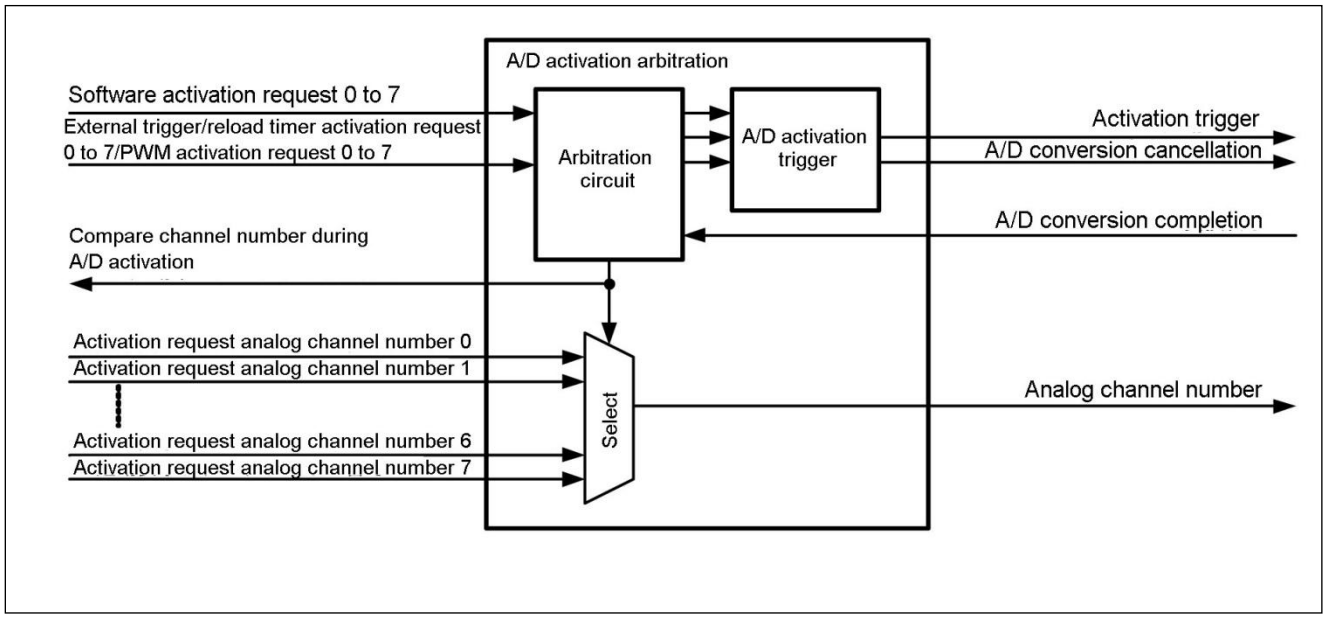
Configuration of A/D activation compare

Figure 35-1. Configuration of A/D Activation Compare (n=7 A/D Converter Unit)



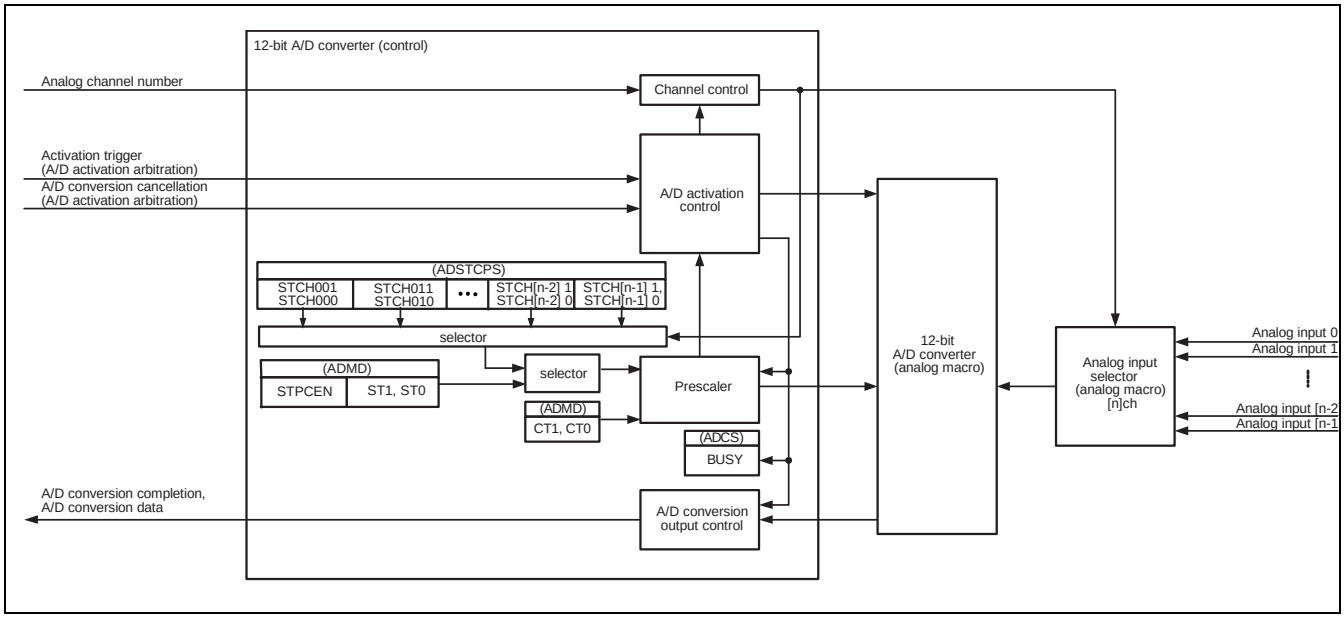
Configuration of A/D activation arbitration

Figure 35-2. Configuration of A/D Activation Arbitration



Configuration of 12-bit A/D converter control

Figure 35-3. Configuration of 12-bit A/D Converter Control (n=8 AD Converter Unit)



35.4 Registers

This section explains registers of the 12-bit A/D converter.

List of analog input control registers

Table 35-2. List of Analog Input Control Registers

Address	+0	+1	+2	+3
0x04AC	Reserved		Analog input control register lower (ADERL0) key code target register)	
0x04B0	Reserved			

List of A/D activation compare registers

Table 35-3. List of A/D Activation Compare Registers

Address	+0	+1	+2	+3
0x1304	A/D software activation egister 0 (ADTSS0)	Reserved	Reserved	Reserved
0x1308	A/D software activation channel select register 0 (ADTSE0)			
0x130C | 0x1348	Reserved			
0x134C	A/D activation trigger control status register 0 (ADTCS0)		A/D activation trigger control status register 1 (ADTCS1)	
0x1350	A/D activation trigger control status register 2 (ADTCS2)		A/D activation trigger control status register 3 (ADTCS3)	
0x1354	A/D activation trigger control status register 4 (ADTCS4)		A/D activation trigger control status register 5 (ADTCS5)	
0x1358	A/D activation trigger control status register 6 (ADTCS6)		A/D activation trigger control status register 7 (ADTCS7)	
0x135C | 0x1388	Reserved			
0x138C	A/D data register 0 (ADTCD0)		A/D data register 1 (ADTCD1)	
0x1390	A/D data register 2 (ADTCD2)		A/D data register 3 (ADTCD3)	
0x1394	A/D data register 4 (ADTCD4)		A/D data register 5 (ADTCD5)	
0x1398	A/D data register 6 (ADTCD6)		A/D data register 7 (ADTCD7)	
0x139C | 0x13C8	Reserved			
0x13CC	A/D activation trigger extend control register 0 (ADTECS0)		A/D activation trigger extend control register 1 (ADTECS1)	
0x13D0	A/D activation trigger extend control register 2 (ADTECS2)		A/D activation trigger extend control register 3 (ADTECS3)	

Address	+0	+1	+2	+3
0x13D4	A/D activation trigger extend control register 4 (ADTECS4)		A/D activation trigger extend control register 5 (ADTECS5)	
0x13D8	A/D activation trigger extend control register 6 (ADTECS6)		A/D activation trigger extend control register 7 (ADTECS7)	
0x13DC | 0x1454	Reserved			
0x1458	Data protection status flag register 0 (ADPRTF0)			
0x145C	Reserved			
0x146C	A/D activation trigger control status extend register 0 (EADTCS0)	A/D activation trigger control status extend register 1 (EADTCS1)	A/D activation trigger control status extend register 2 (EADTCS2)	A/D activation trigger control status extend register 3 (EADTCS3)
0x1470	A/D activation trigger control status extend register 4 (EADTCS4)	A/D activation trigger control status extend register 5 (EADTCS5)	A/D activation trigger control status extend register 6 (EADTCS6)	A/D activation trigger control status extend register 7 (EADTCS7)

List of 12-bit A/D converter control registers

Table 35-4. List of 12-bit A/D Converter Control Registers

Address	+0	+1	+2	+3
0x1460	A/D control status register 0 (ADCS0)		A/D channel status register 0 (ADCH0)	A/D mode setting register 0 (ADMD0)
0x1464	Sampling time setting register 0 for each A/D channel (ADSTPCS0)	Sampling time setting register 1 for each A/D channel (ADSTPCS1)	Reserved	Reserved

35.4.1 Register of Analog Input Control

The register of the analog input control is explained.

The analog input control register is used to control the analog input.

35.4.1.1 Analog Input Control Register: ADER

The bit configuration of the analog input control register is shown.

The analog input control register (ADERL0) controls the analog input.

ADERL0: Address 04AE$_H$ (Access: Byte, Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADE07	ADE06	ADE05	ADE04	ADE03	ADE02	ADE01	ADE00
Initial value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit8] Undefined

"1" is always read. Writing has no influence on an operation.

[bit7 to bit0] ADE07 to ADE00: Analog input enable bits

ADE07 to ADE00	Function
0	Analog input disabled
1	Analog input enabled

- This bit controls analog input pin.
- If these bits are "0", the analog input is disabled.
- If these bits are "1", the analog input is enabled.

Note:

This register is a key code target register. Key code setting is required for writing.
For the setting method, refer to sections "KEY CoDe Register: KEYCDR" and "Key Code Register Function Settings" in "Chapter: I/O Ports." In addition, word access to this register is disabled.

35.4.2 Registers of A/D Activation Compare

The registers of the A/D activation compare is explained.

35.4.2.1 A/D Software Activation Register: ADTSS0

The bit configuration of the A/D software activation register is shown.

The A/D software activation register (ADTSS) issues a 12-bit A/D converter A/D activation request. The activation channel is specified by the A/D software activation channel select register (ADTSE).

ADTSS0: Address 1304_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							START
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W

[bit7 to bit1] Reserved

Always write 0 to these bits.

[bit0] START: A/D conversion activation bit (software)

START	Function
0	Does not activate the A/D conversion function.
1	Activates the A/D conversion function.

- This bit activates the A/D conversion operation under software control.
- The A/D conversion activates when this bit is set to "1". The activation channel is specified by the A/D software activation channel select register (ADTSE).
- The A/D conversion cannot be reactivated by changing this bit.

35.4.2.2 A/D Software Activation Channel Select Register: ADTSE0

The bit configuration of the A/D software activation channel select register is shown.

The A/D software activation channel select register (ADTSE) is a register to select the activation channel that issues an A/D activation request.

ADTSE0: Address 1308_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	-							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	-							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-							
Initial value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADT07	ADT06	ADT05	ADT04	ADT03	ADT02	ADT01	ADT00
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit8] Undefined

"1" is always read. Writing has no influence on an operation.

[bit7 to bit0] ADT07 to ADT00: Software activation channel select bits

ADT07 to ADT00	Function
0	Software activation disabled
1	Software activation enabled

- These bits control the software activation from the activation channel.
- If these bits are "0", the software activation is disabled.
- If these bits are "1", the software activation is enabled.

35.4.2.3 A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS7

The bit configuration of the A/D activation trigger control status register is shown.

The A/D activation trigger control status register (ADTCS) verifies A/D activation requests, enables/disables interrupt requests, verifies an interrupt request status, selects an activation factor, selects a conversion mode, controls the protection function, and selects an analog input channel.

ADTCS0 to ADTCS7: Address $134C_H$ to $135A_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BUSY	INT	INTE	STS1	STS0	RPT	PRT	PRTS
Initial value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R(RM1),W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial value	0	0	1	0	0	0	0	0
Attribute	R0/W0	R0/W0	R1/W1	R0/W0	R0/W0	R0/W0	R0/W0	R0/W0

[bit15] BUSY: A/D activation request bit

BUSY	Function	
	Read	Write
0	A/D activation is not requested	A/D activation request forced stop
1	A/D activation is requested or A/D conversion is operating	This bit value does not change and there is no influence on an operation by writing.

- This bit shows the operation of the A/D activation request or conversion.
- When the reading value of this bit is "0", it is shown that A/D conversion is not requested. When the reading value of this bit is "1", it is shown that A/D conversion is requested or A/D conversion is operating.
- The A/D activation request or conversion forced stops by writing "0" in this bit.
  This bit value does not change and there is no influence on an operation by writing "1".

Note:

If the read-modify-write (RMW) instruction is executed, "1" will be read out.

[bit14] INT: Interrupt request flag bit

INT	Function	
	Read	Write
0	A/D conversion is not completed.	Clear of bit
1	A/D conversion was completed.	This bit value does not change and there is no influence on an operation by writing.

- When data is set in A/D data register (ADTCD) by the A/D conversion, this bit is set to "1".
- When this bit and interrupt request enable bit (ADTCS:INTE) are "1", an interrupt request is generated.
- This bit is cleared by writing "0". This bit value does not change and there is no influence on an operation by writing "1".
- When the A/D conversion completion interrupt clear signal is "H", this bit is cleared.

Note:

If the read-modify-write (RMW) instruction is executed, "1" will be read out.
If the software clear (INT="0" writing) or the clear due to an interrupt clear signal ("H") occurs at the same time with the hardware set, the hardware set takes priority.

[bit13] INTE: Interrupt request enable bit

INTE	Function
0	Interrupt request output disabled
1	Interrupt request output enabled

- This bit enables/disables the interrupt output to CPU.
- When this bit and interrupt request flag bit (ADTCS:INT) are "1", the interrupt request is generated.

[bit12, bit11] STS1, STS0: A/D activation factor select bit

ADTECS: STS2	STS1	STS0	Function
0	0	0	Software activation
0	0	1	External pin trigger activation (falling edge)
0	1	0	Reload timer activation (rising edge)
0	1	1	Setting disable
1	0	0	PWM activation (rising edge)
1	0	1	Setting disable
1	1	0	
1	1	1	

- The activation factor of the A/D conversion is selected by a combination of the STS1, STS0 bit, and bit8 (STS2) of the A/D activation trigger extend control register (ADTECS).

Notes:

- Since the A/D activation factor select bit changes immediately when the bits are rewritten, change these bits while activation factors of the current target and of the target to be changed are inactive and A/D conversion is not being requested (ADTCS:BUSY=1).
- Please set these bits including ADTECS.STS2 as software activation ("000_B"), and set a corresponding bit of ADTSE (activation channel) to the software activation disable (ADT bit=0) when A/D conversion is not being requested.

[bit10] RPT: Repeat conversion select bit

RPT	Function
0	Single conversion
1	Repeat conversion

- This bit specifies an A/D conversion mode.
- To select single conversion mode, set this bit to "0". In this mode, one activation factor leads to the issuance of one A/D conversion request. The A/D conversion is performed once.
- To select repeat conversion mode, set this bit to "1". In this mode, one activation factor leads to a sequence of A/D conversion requests. The A/D conversion is performed repeatedly until single conversion mode is selected.

[bit9] PRT: A/D data register protection enable bit

PRT	Function
0	Protection disabled
1	Protection enabled

- If this bit is set to "1", the A/D data register is protected against being overwritten. The protection function works if the activation factor is not compare-match activation (STS1, STS0=11).
- After conversion data is stored in the A/D data register, the next activation request will be masked to protect the A/D data register against being overwritten until the factor specified by the A/D data register protection clear select bit (PRTS) occurs.

Note:

Please set the A/D data register protection enable bit before operating the A/D conversion. Please do not change the A/D data register protection enable bit with the A/D conversion requested or the A/D data register protected.

[bit8] PRTS: A/D data register protection clear select bit

PRTS	Function
0	Data read and interrupt flag clear
1	Data read

- This bit selects a condition for clearing the activation request mask if the A/D data register protection function is enabled (PRT=1).
- If this bit is set to "0", the A/D data register (ADTCD) reading and the interrupt request flag bit (INT) clearing become the protection release conditions (random order).
- If this bit is set to "1", the A/D data register (ADTCD) reading becomes the protection release condition.

Note:

Please set the A/D data register protection clear select bit before operating the A/D conversion.
Please do not change the A/D data register protection clear select bit with the A/D conversion requested or the A/D data register protected.

[bit7, bit6] Reserved

Be sure to write "0" to these bits.

[bit5] Reserved

Be sure to write "1" to this bit.

[bit4 to bit0] Reserved

Be sure to write "0" to these bits.

35.4.2.4 A/D Activation Trigger Control Status Extension Register: EADTCS0 to EADTCS7

The bit configuration of the A/D activation trigger control status extension register is shown.

For the A/D activation trigger control status extension register (EADTCS), the PWM selects the activation factor.

EADTCS0 to EADTCS7: Address $146C_H$ to 1473_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	STSA3	STSA2	STSA1	STSA0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W

[bit7 to bit4] Undefined

"0" is always read. Writing has no influence on an operation.

[bit3 to bit0] STSA3 to STSA0: A/D PWM activation factor select bit

STSA3	STSA2	STSA1	STSA0	Function
0	0	0	0	PWM edge trigger 0
0	0	0	1	PWM edge trigger 1
0	0	1	0	PWM edge trigger 2
0	0	1	1	PWM edge trigger 3
0	1	0	0	Fault result 0
0	1	0	1	Fault result 1
0	1	1	0	Fault result 2
0	1	1	1	Fault result 3
1	0	0	0	Fault result 4
1	0	0	1	Fault result 5
1	0	1	0	Special event trigger 0
1	0	1	1	Special event trigger 1
1	1	0	0	Setting disable
:				
1	1	1	1	

- The STSA3 to STSA0 bits are enabled when bit8 (STS2) is "1" in the A/D activation trigger extension control register (ADTECS) and bit12 (STS1) and bit11 (STS0) are "00" in the A/D activation trigger control status register (ADTCS).

35.4.2.5 A/D Data Register: ADTCD0 to ADTCD7

The bit configuration of the A/D data register is shown.

The A/D data register (ADTCD) stores A/D conversion results.

ADTCD0 to ADTCD7: Address138C$_H$ to 139A$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ERR	ERRST	Reserved		D11	D10	D9	D8
Initial value	1	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15] ERR: Conversion data error flag bit

ERR	Function
0	The conversion data is normal
1	The conversion data is not normal.

- This bit indicates the presence of an error contained in the A/D conversion data. If this bit is "1", the value of the ERRST bit indicates the content of the error.
- When this bit is read, it is set to "1".
- When new conversion result are written in this register, this bit is cleared to "0".
- If the A/D data register protection function is enabled (ADTCS.PRT=1), this bit is read as "0".

[bit14] ERRST: Conversion data error status bit (only when ERR=1)

ERRST	Function
0	The conversion data is the old result.
1	The conversion data is overwritten by the new data.

- If the ERR bit is "1", this flag indicates the content of the error in the A/D conversion data.
- If the ERR bit is "1" and this bit is "0", the conversion result read by the CPU are old data.
- If the ERR bit and this bit are "1", the previous conversion results have been overwritten by new conversion results and are lost before they have not been completely read by CPU.
- This bit is set to "1" if the previous conversion results have been overwritten by new conversion results and lost before they have not been completely read by the CPU.
- When this bit is read, it is cleared to "0".
- If the A/D data register protection function is enabled (ADTCS.PRT=1), this bit is read as "0".

[bit13, bit12] Reserved

Be sure to write "0" to these bits.

[bit11 to bit0] D11 to D0: A/D data bits

	Function
D11 to D0	Conversion data

- A/D conversion results are stored in this register, when one sequence of conversion is completed, the register is rewritten.
- In general, the register holds the last conversion value.

Note:

Be sure to avoid writing data to these bits.

35.4.2.6 A/D Activation Trigger Extend Control Register: ADTECS0 to ADTECS7

The bit configuration of the A/D activation trigger extend control register is shown.

The A/D activation trigger extend control register (ADTECS) selects the activation factor select and the analog input channel.

ADTECS0 to ADTECS7: Address 13CC$_H$ to 13DA$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							STS2
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W

[bit15 to bit9] Reserved

Be sure to write "0" to these bits.

[bit8] STS2: A/D activation factor select bit

The activation factor of the A/D conversion is selected by a combination of this bit and bit12, bit11 (STS1, STS0) of the A/D activation trigger control status register (ADTCS). Please refer to "35.4.2.3 A/D Activation Trigger Control Status Register: ADTCS0 to ADTCS7" for details.

Notes:

- Since the A/D activation factor select bit changes immediately when the bit is rewritten, change this bit while activation factors of the current target and of the target to be changed are inactive and A/D conversion is not being requested (ADTCS:BUSY=1).
- Please set this bit including ADTCS.STS1 and ADTCS.STS0 as software activation ("000_B"), and set a corresponding bit of ADTSE (activation channel) to the software activation disable (ADT bit=0) when A/D conversion is not being requested.

[bit7 to bit5] Reserved

Be sure to write "0" to these bits.

[bit4 to bit0] CHSEL4 to CHSEL0: Analog channel select bit

CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	Explanation
0	0	0	0	0	Channel 0
0	0	0	0	1	Channel 1
:					:
0	0	1	1	0	Channel 6
0	0	1	1	1	Channel 7
0	1	0	0	0	Setting disable
:					:
1	1	1	1	1	Setting disable

These bits select the analog channel with the specified value.

- Do not change the analog channel select bit when A/D conversion is being requested.

35.4.2.7 Data Protection Status Flag Register: ADPRTF0

The bit configuration of the data protection status flag register is shown.

The data protection status flag register (ADPRTF) indicates the protection status of A/D data register of each activation channel.

ADPRTF0: Address 1458_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	-							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	-							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PRTF7	PRTF6	PRTF5	PRTF4	PRTF3	PRTF2	PRTF1	PRTF0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit8] Undefined

"0" is always read. Writing has no influence on an operation.

[bit7 to bit0] PRTF7 to PRTF0: Data protection status flag bits

PRTF	Explanation
0	Not in data protection state
1	In data protection state

- These bits indicate the data protection status for the A/D data registers of each activation channel.
- The writing operation doesn't influence the data protection status.

35.4.3 Registers of 12-BIT A/D Converter Control

The registers of the 12-bit A/D converter control is explained.

The 12-bit A/D converter control uses A/D control status register, A/D channel status register, A/D mode setting register, and A/D sampling time setting register.

35.4.3.1 A/D Control Status Register: ADCS0

The bit configuration of the A/D control status register is shown.

The A/D control status register (ADCS) provides the function to confirm conversion.

ADCS0: Address 1460_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BUSY	Reserved						
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

[bit15] BUSY: A/D conversion operating bit

BUSY	Function
0	The A/D conversion is stopping.
1	The A/D conversion is operating.

- This bit indicates the A/D converter operation state.
- When the reading value of this bit is "0", it is shown that A/D conversion is stopping. When the reading value of this bit is "1", it is shown that A/D conversion is operating.
- This bit value does not change and there is no influence on an operation by writing.

[bit14 to bit0] Reserved

Be sure to write "0" to these bits.

35.4.3.2 A/D Channel Status Register: ADCH

The bit configuration of the A/D channel status register is shown.

The A/D channel status register (ADCH) shows the analog channel number of conversion target while the A/D conversion is operating.

ADCH0: Address 1462_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			CH4	CH3	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7 to bit5] Reserved

Be sure to write "0" to these bits.

[bit4 to bit0] CH4 to CH0: Analog channel bits

CH4	CH3	CH2	CH1	CH0	Function
					ADCH0
0	0	0	0	0	ch.0
0	0	0	0	1	ch.1
:					:
0	0	1	1	0	ch.6
0	0	1	1	1	ch.7
0	1	0	0	0	Setting disable
:					:
1	1	1	1	1	Setting disable

- These bits show the analog channel number of the conversion target while the A/D conversion is operating.

35.4.3.3 A/D Mode Setting Register: ADMD

The bit configuration of the A/D mode setting register is shown.

The A/D mode setting register (ADMD) sets the compare time and the sampling time of the A/D conversion.

ADMD0: Address 1463$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STPCEN	Reserved			CT1	CT0	ST1	ST0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit7] STPCEN: Sampling time setting per channel enable bit

STPCEN	Explanation
0	Sampling time setting common to all channels
1	Sampling time setting of each channel

- This bit selects setting of each channel or setting common to all channels as the sampling time setting in the A/D conversion.
- When STPCEN="0", the sampling time common to all channels is enabled. The sampling time is set by the sampling time set bit.
- When STPCEN="1", the sampling time of each channel is enabled. The sampling time of each channel is set by sampling time setting bit per A/D channel.

[bit6 to bit4] Reserved

Be sure to write "0" to these bits.

[bit3, bit2] CT1, CT0: Compare time setting bits

CT1	CT0	Function
0	0	28 peripheral clock cycle (A/D clock output: Peripheral clock/2)
0	1	42 peripheral clock cycle (A/D clock output: Peripheral clock/3)
1	0	56 peripheral clock cycle (A/D clock output: Peripheral clock/4)
1	1	112 peripheral clock cycle (A/D clock output: Peripheral clock/8)

- These bits select the compare time in the A/D conversion.
- After the analog input is taken (sampling time passage), the data of the conversion result is fixed when the time set to these bits has elapsed.

Setting examples

Peripheral clock (MHz)	CT1	CT0	Compare time (ns)
40	0	0	700
32	0	0	875
24	0	0	1166.7
16	0	0	1750

Note:

Set the CT1 and CT0 bits to be 700 ns or more at the compare time. When the compare time is set to 700 ns or less, a normal value of the analog conversion value might not be obtained.
Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

[bit1, bit0] ST1, ST0: Sampling time setting bits

ST1	ST0	Function
0	0	12 peripheral clock cycle (A/D clock output: Peripheral clock/2)
0	1	18 peripheral clock cycle (A/D clock output: Peripheral clock/3)
1	0	24 peripheral clock cycle (A/D clock output: Peripheral clock/4)
1	1	48 peripheral clock cycle (A/D clock output: Peripheral clock/8)

- These bits select the sampling time in the A/D conversion.
- The analog input is taken during the period set in these bits after A/D is activated.

Setting examples (Use conditions: AVcc=4.5 V to 5.5 V)

Peripheral Clock (MHz)	ST1	ST0	Sampling Time (ns)
40	0	0	300
32	0	0	375
24	0	0	500
16	0	0	750

Note:

Set the ST1 and ST0 bits to be 300 ns (4.5 V to 5.5 V) or more at the sampling time. When the sampling time is set to 300 ns or less, a normal value of the analog conversion value might not be obtained.
Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

35.4.3.4 A/D Sampling Time Setting Per Channel Register: ADSTPCS

The bit configuration of the A/D sampling time setting per channel register is shown.

The A/D sampling time setting per channel register (ADSTPCS) sets the sampling time of the A/D conversion for each channel.

ADSTPCSm (m=0 to 1): Address 1464_H to 1465_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STCH (m*4+3)1	STCH (m*4+3)0	STCH (m*4+2)1	STCH (m*4+2)0	STCH (m*4+1)1	STCH (m*4+1)0	STCH (m*4)1	STCH (m*4)0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to 0] STCHn1, STCHn0 (n=00 to 07): Sampling time setting bits

STCHn1	STCHn0	Explanation
0	0	12 peripheral clock cycle (A/D clock output: Peripheral clock/2)
0	1	18 peripheral clock cycle (A/D clock output: Peripheral clock/3)
1	0	24 peripheral clock cycle (A/D clock output: Peripheral clock/4)
1	1	48 peripheral clock cycle (A/D clock output: Peripheral clock/8)

- These bits select the sampling time in the A/D conversion for each channel.
- If the sampling time setting permission bit (ADMD.STPCEN) is set to "1" for each channel, the sampling time setting (STCHn1, STCHn0) for each channel is enabled.
- The correspondence of a set channel and an analog channel is shown below.

Table 35-5. Correspondence between Sampling Time Setting for Each Channel and Analog Channel (A/D Converter Unit 0)

Sampling Time Setting per Channel Enable Bit	Analog Channel Number
	ADSTPCS0 to 1
STCH001, STCH000	Channel 0
STCH011, STCH010	Channel 1
:	:
STCH061, STCH060	Channel 6
STCH071, STCH070	Channel 7

Note:

Rewrite these bits while the A/D conversion operations are still stopped.

35.5 Operation

This section explains the operation.

35.5.1 Interrupt of A/D Activation Compare

35.5.2 A/D Activation Compare Operation

35.5.3 A/D Activation Arbitration Operation

35.5.4 12-bit A/D Converter Operation

35.5.1 Interrupt of A/D Activation Compare

This section explains the interrupt control bit and the interrupt factor of A/D activation compare.

The interrupt control bit and the interrupt factor of A/D activation compare are shown.

35.5.1.1 A/D Conversion Completion Interrupt

The A/D conversion completion interrupt is explained.

Table 35-6. Interrupt Control Bit and Interrupt Factor by A/D Conversion Completion Interrupt

	A/D Conversion Completion Interrupt
Interrupt request flag bit	INT:bit14 of A/D activation trigger control status register (ADTCS)
Interrupt request enable bit	INTE:bit13 of A/D activation trigger control status register (ADTCS)
Interrupt factor	Writing of A/D conversion result in A/D data register

The A/D conversion completion interrupt request can be generated when the A/D conversion of compare channel that activates the A/D converter completes. Moreover, the A/D conversion completion interrupt can be controlled in units of activation channel.
When the A/D conversion result is set in the A/D data register (ADTCD), the INT bit of A/D activation trigger control status register (ADTCS) is set to "1". At this time, when the interrupt request enable bit has been enabled (ADTCS.INTE="1"), the interrupt request is output to the interrupt controller.

35.5.2 A/D Activation Compare Operation

This section explains the A/D activation compare operation.

The A/D activation can be requested by any of the software, the external trigger, the reload timer, or PWM.

35.5.2.1 A/D Activation

The A/D activation is explained.

An activation request for the A/D converter is issued. The activation request can be generated at each analog channel (max. 8 channels).
The activation request signal is generated to the A/D activation arbitration with any of the software, the external trigger (falling), the reload timer (rising), or PWM. There are three A/D activation request signals, any of the "software activation request", "external trigger/reload timer/PWM activation request" or "compare match activation request" becomes active in each activation channel.
The activation request is cleared on completion of the A/D conversion of the corresponding channel and the conversion data is stored in the A/D data register. In that case, the interrupt can be generated.
Even if the activation factor is received in the activation request, the activation request is not reactivated in the activation channel.

- The activation channels are assigned to the A/D converter. The assignment is as shown in the table below.

Table 35-7. Activation Channel Correspondence

	Activation Channel
MB91F552	ch.0 to ch.7

35.5.2.2 A/D Activation Enable

The A/D activation enable is explained.

The A/D activation factor is selected by the A/D activation trigger control factor select bits (ADTECS.STS2, ADTCS.STS1, STS0) and the A/D PWM activation factor select bits (EADTCS.STSA3 to EADTCS.STSA0). Any of the software, the external trigger, the reload timer, or PWM is selected. When the selected activation factor is generated, the A/D activation request signal is generated for the A/D activation arbitration.
For the activation channel that does not activate A/D, it is possible to disable the A/D activation request by selecting the software activation (ADTECS.STS2="0", ADTCS.STS1, STS0="00_B") and disabling the software activation of a corresponding channel of A/D software activation channel select register (ADTSE).

35.5.2.3 Analog Channel Select

The analog channel select is explained.

An analog channel that does the A/D conversion can be selected by the CHSEL bit of A/D activation trigger extend control register (ADTECS).

35.5.2.4 Software Activation

The software activation is explained.

The A/D activation trigger control factor select bit is set to software activation (ADTECS.STS2="0", ADTCS.STS1, STS0="00_B").
The channel desired for the software activation is set to activation enable according to the A/D software activation channel select register (ADTSE). Two or more channels set to the ADTSE register can generate the activation request at the same time.
And, the software activation request signal is set by writing "1" in the START bit of the A/D software activation register (ADTSS).

35.5.2.5 External Trigger Activation

The external trigger activation is explained.

The A/D activation trigger control factor select bit is set to external trigger activation (ADTECS.STS2="0", ADTCS.STS1, STS0="01_B").
The external trigger corresponds to 12-bit A/D converter activation channels 0 to 7. For details on the A/D converter activation channels in this series, see "Table 35-7 Activation Channel Correspondence."
When the falling edge of the external trigger is detected, the activation request signal of external trigger/reload timer/PWM is set.

35.5.2.6 Reload Timer Activation

The reload timer activation is explained.

The A/D activation trigger control factor select bit is set to reload timer activation (ADTECS.STS2="0", ADTCS.STS1, STS0="10_B").
The reload timer is input in the 12-bit A/D converter, and reload timer 0 corresponds to activation channels 0 to 7. For details on the A/D converter activation channels in this series, see "Table 35-7. Activation Channel Correspondence."
When the rising edge of the reload timer is detected, the activation request signal of external trigger/reload timer/PWM is set.

35.5.2.7 PWM Activation

The PWM activation is explained.

The A/D activation trigger control factor select bit is set to PWM activation (ADTECS.STS2="1", ADTCS.STS1, STS0="00B").
The PWM activation factor is selected from the A/D PWM activation factor select bit. The PWM activation is input independently in each activation channel.
When the rising edge of the set signal in the A/D PWM activation factor select bit is detected, the activation request signal of external trigger/reload timer/PWM is set.

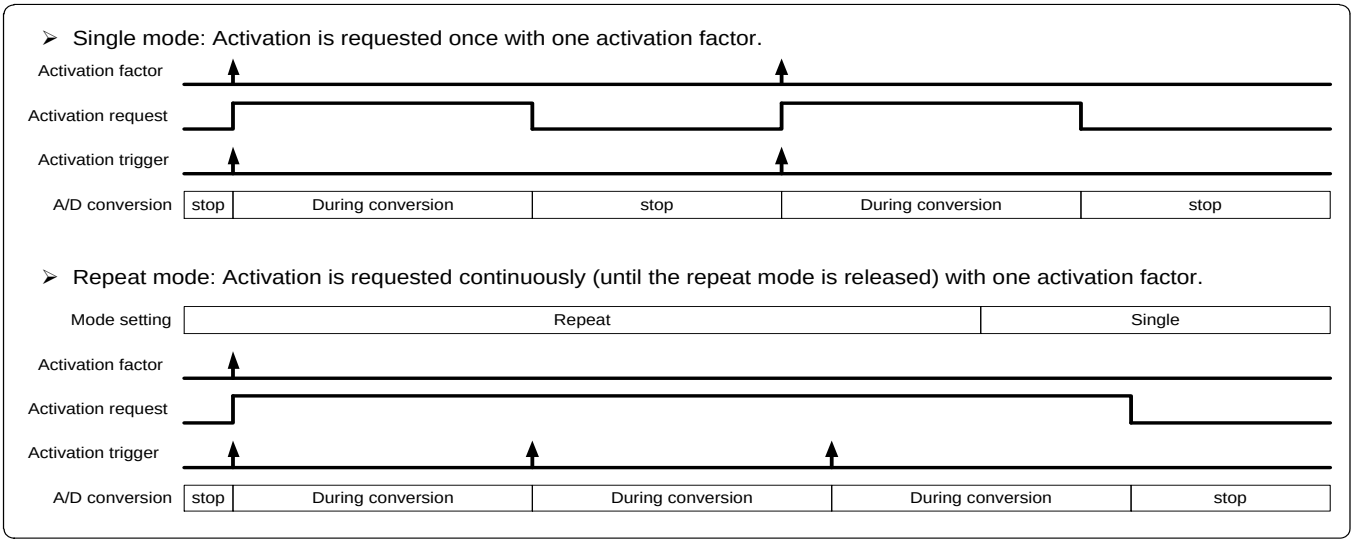
35.5.2.8 Activation Request Mode

The activation request mode is explained.

The activation request mode of each activation channel can be set. There are two activation request modes (the single mode and the repeat mode). The mode is set by the RPT bit of A/D activation trigger control status register (ADTCS).

- In single mode, activation is requested once with one activation factor. The A/D conversion is done once, and the activation request is released by the A/D conversion completion.
- In repeat mode, activation is requested continuously with one activation factor. The A/D conversion is repeatedly executed, and the activation request continues until the repeat mode is released.

Figure 35-4. Activation Request Mode



35.5.2.9 A/D Conversion Data

The A/D conversion data is explained.

The A/D conversion result data is stored in A/D data bit (ADTCD.D11 to D0) at each activation channel.
Moreover, when the data protection function is disabled (ADTCS.PRT="0"), the state of the A/D conversion data stored in A/D data bit (ADTCD.D11 to D0) can be confirmed by the conversion data error flag bit (ADTCD.ERR) and the conversion data error status bit (ADTCD.ERRST).
When the data protection function is enabled (ADTCS.PRT="1"), the conversion data error flag bit (ADTCD.ERR) and the conversion data error status bit (ADTCD.ERRST) are fixed to "0".

Table 35-8. Status Confirmation of A/D Conversion Data (When the Data Protection Function is Disabled)

ADTCD:ERR	ADTCD:ERRST	State of A/D conversion data
0	0	The latest data (read out not yet)
0	1	- (It does not have a meaning)
1	0	The old data (already read out) (Note) Initial value
1	1	The latest data/ The overwriting is generated (read out not yet) (Note) There is lost data

35.5.2.10 Protection Function

The protection function is explained.

Each A/D data register can set the data protection function. The protection function is set by the PRT bit of A/D activation trigger control status register (ADTCS).

When the protection function is effective, the data becomes the protection state if the conversion result is stored in the A/D data register. The condition to release the data protection state can be selected by the PRTS bit of the ADTCS register.

- When the PRTS bit is "0", the activation request is masked until the data of the A/D data register is read and the interrupt flag is cleared. The data reading and the interrupt flag clear are in random order.
- When the PRTS bit is "1", the activation request is masked until the data of the A/D data register is read.
- In the state of data protection, even if the next activation factor is generated, the activation request signal is masked (not active). Therefore, the data (read out not yet) of the A/D data register is not overwritten by the next A/D conversion data.

35.5.2.11 Forced Termination of Activation Request

The forced termination of activation request is explained.

Whether an A/D activation request or a conversion operation is in progress can be reported using the A/D activation request bits (ADTCS0 to 7.BUSY) of the A/D activation trigger control status register. The current A/D activation request or conversion operation can be terminated forcibly by resetting these bits to "0".

35.5.3 A/D Activation Arbitration Operation

This section explains the A/D activation arbitration operation.

A/D activation requests from A/D activation compare are arbitrated to generate an A/D activation trigger. In addition, the analog channel to be A/D-converted is determined.

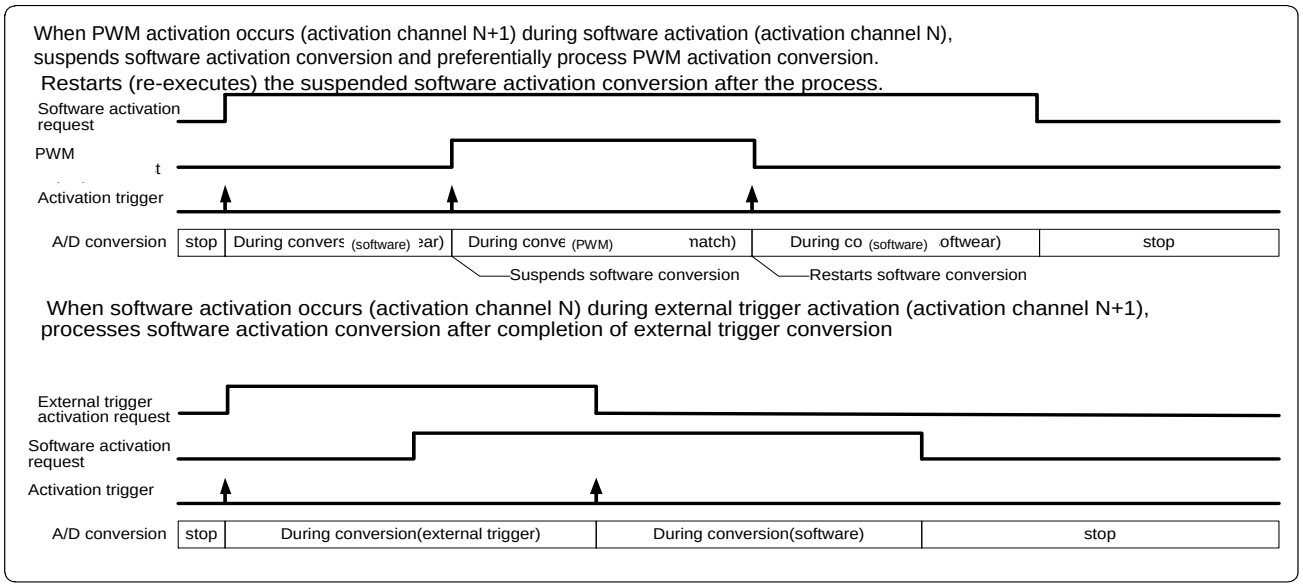
35.5.3.1 A/D Activation Trigger Arbitration

The A/D activation trigger arbitration is explained.

An A/D activation trigger is generated after one of activation requests from A/D activation compare channels is selected. Two types of activation requests from individual A/D activation compare channels are entered: software activation request, and external trigger/reload timer activation request/PWM activation. A/D activation trigger signals are thus generated. If contention occurs among multiple activation requests, the activation request with the lowest compare channel number takes precedence. Activation requests that are not selected are made to wait. When the current A/D conversion completes, arbitration restarts.
The sequence of activation arbitration priorities based on the activation factor is as follows: External trigger/reload timer activation request/PWM activation request > software activation request. If two activation requests are caused by activation factors assigned the same priority, the request with the lower activation channel number takes precedence.

- If an activation factor with the same priority is encountered during A/D conversion suspension:
  The request from the activation channel with the lower number is processed first.
- If an activation factor with a different priority is encountered during A/D conversion suspension:
  The request based on the higher-priority activation factor is processed first.
- If an activation factor with a higher priority is encountered during A/D conversion:
  The current conversion is suspended, and the higher-priority activation factor is processed. After this processing completes, arbitration is performed again. The suspended activation factor is then processed.
- If an activation factor with a lower priority is encountered during A/D conversion:
  After the current conversion completes, arbitration is performed again. The activation factor with the lower priority is then processed.
- If an activation factor with the same priority is encountered during A/D conversion:
  After the current conversion completes, arbitration is performed again. The activation factor with the same priority is then processed.

Figure 35-5. Activation Arbitration



35.5.3.2 Analog Channel Select

The analog channel select is explained.

The A/D activation request and the A/D conversion target analog channel number are entered from the A/D activation compare.
The A/D activation arbitration selects the activation request analog channel number of the selected A/D activation compare channel.

35.5.3.3 A/D Conversion Cancel Function

The A/D conversion cancel function is explained.

When the activation request from the request source becomes inactive during the A/D conversion, an A/D conversion cancel signal is generated to abort the current conversion processing. If the activation factor from another activation channel is active when the activation request from the request source becomes inactive, an A/D conversion cancel signal is not generated, but an A/D activation trigger based on the active activation factor is generated.

35.5.4 12-bit A/D Converter Operation

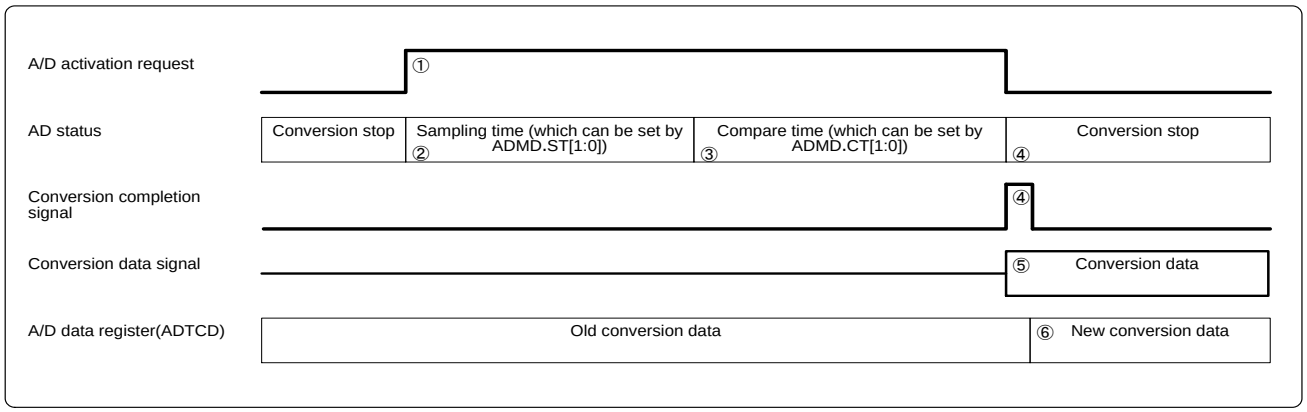
This section explains the 12-bit A/D converter operation.

The 12-bit A/D converter controls the A/D conversion.

35.5.4.1 Operation Timing

The operation timing is shown.

Figure 35-6. 12-bit A/D Converter Operation Timing



1. With the specified activation factor, the A/D conversion is started.
2. After the activation request of (1) is received, the sampling operation is started.
3. After the sampling time specified with ADMD.ST[1:0] passes, the compare operation is started.
4. After the compare time specified with ADMD.CT[1:0] passes, the conversion completion signal rises, and the conversion is completed.
5. A/D conversion data is output.
6. New conversion data is stored in the A/D data register (ADTCD).

35.5.4.2 Activation Factors

The activation factor is shown.

A/D conversion activation factors include the software activation, external trigger activation, reload timer activation, and PWM activation. It is selected with ADTECS.STS2, ADTCS.STS[1:0], and EADTCS.STSA[3:0].
If you do not want to issue an A/D activation request, set these bits to specify software activation ("000_B") and set the corresponding bit of ADTSE to disable software activation (ADTSE.ADT=0).

35.5.4.3 A/D Conversion

The A/D conversion is shown.

One sequence of A/D conversion is performed in response to one activation trigger.

35.5.4.4 Re-activation

The re-activation is shown.

When an activation trigger signal is input during the A/D conversion, the current conversion is stopped and initialized and the A/D conversion is re-activated.
For this reason, the A/D conversion re-activation starts with a delay of several cycles of the clock (12-bit A/D converter clock) in comparison with ordinary activation (start of A/D conversion while the A/D conversion is stopped).

35.5.4.5 A/D Conversion Cancel

The A/D conversion cancel is shown.

If an A/D conversion cancel signal is received during the A/D conversion, the current conversion stops and initialization takes place.

35.5.4.6 Analog Channel Select Control

The analog channel select control is shown.

Information on the analog channel to be A/D-converted is input in addition to the activation trigger.
The analog channel information, which is effective when the activation trigger is active, is used to select an analog channel.

35.5.4.7 A/D Conversion Time

The A/D conversion time is shown.

The A/D conversion time consists of the sampling time and the compare time.

Sampling time

The sampling time selection can be set to either each channel or to a common setting by using the sampling time setting per channel enable bit (ADMD.STPCEN).

- When ADMD.STPCEN="0", the sampling time common to all channels is enabled. The sampling time is set by the sampling time set bit (ADMD.ST1, ST0).
- When ADMD.STPCEN="1", the sampling time of each channel can be set. The sampling time of each channel is set by the sampling time setting per channel bit (ADSTPCS.STCHn1, STCHn0: n=00 to 07).

Table 35-9. Compare Time to Peripheral Clock Frequency

ST1, STCHn1	ST0, STCHn0	Function	Sampling Time (Peripheral Clock Frequency)			
			(40 MHz)	(32 MHz)	(24 MHz)	(16 MHz)
0	0	12 peripheral clock cycle	300 ns	375 ns	500 ns	750 ns
0	1	18 peripheral clock cycle	450 ns	562.5 ns	750 ns	1125 ns
1	0	24 peripheral clock cycle	600 ns	750 ns	1000 ns	1500 ns
1	1	48 peripheral clock cycle	1200 ns	1500 ns	2000 ns	3000 ns

Note:

Set the ST1 and ST0 bits to be 300 ns (4.5 V to 5.5 V) or more at the sampling time. When the sampling time is set to 300 ns or less, a normal value of the analog conversion value might not be obtained.
Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

Compare time

The compare time is set to compare time set bit (ADMD.CT1, CT0).

Table 35-10. Compare Time to Peripheral Clock Frequency

CT1	CT0	Function	Compare Time (Peripheral Clock Count)			
			(40 MHz)	(32 MHz)	(24 MHz)	(16 MHz)
0	0	28 peripheral clock cycle	700 ns	875 ns	1166.7 ns	1750 ns
0	1	42 peripheral clock cycle	1050 ns	1312.5 ns	1750 ns	2625 ns
1	0	56 peripheral clock cycle	1400 ns	1750 ns	2333.4 ns	3500 ns
1	1	112 peripheral clock cycle	2800 ns	3500 ns	4666.7 ns	7000 ns

Note:

Set the CT1 and CT0 bits to be 700 ns or more at the compare time. When the compare time is set to 700 ns or less, a normal value of the analog conversion value might not be obtained.
Perform rewriting of the bit when the A/D operation which has not been converted yet is stopped.

35.5.4.8 A/D Conversion Completion and A/D Data Retrieval

The A/D conversion completion and A/D data retrieval is shown.

When the A/D conversion completes normally without re-activation or cancel (the specified number of cycles have passed), the received conversion data is retrieved and output. At this time, an A/D conversion completion signal is generated.

35.5.4.9 Power Down

The power down is shown.

Power down is in effect during the standby mode.

Note:

Stop the A/D conversion before the transition to standby mode occurs.

35.6 Notes

This section explains notes.

Notes on using A/D activation compare

Configuration of protection of A/D data register

It is necessary to set the PRT bit and the PRTS bit before the A/D conversion starts. It is not allowed to set these bits during the A/D conversion and the state that the A/D data register is being protected.
When canceling the protection function of the A/D data register, execute the protection cancellation which is set in the PRTS bit after the A/D conversion stops, or disable the protection function by the PRT bit.
If the PRTS bit is changed while the A/D data register is being protected, it is necessary to execute the protection cancellation operation which is set in the PRTS bit (for example reading of the A/D data register, or clearing operation by writing "0" to an interrupt request flag bit) after changing the PRTS bit in order to cancel the A/D data register protection. For example, PRTS is set to 0 after clearing an interrupt request flag bit while PRT=1 and PRTS=1, and the A/D data register is being protected, it is needed to read the A/D data register and clear by writing "0" to the interrupt request flag bit again in order to cancel the protection.

About the setting of the sampling time and the compare time

Set the ST1, ST0/STCHn1, and STCHn0 (n=00 to 07) bits to be 300 ns (4.5 V to 5.5 V) or more at the sampling time. When the sampling time is set to 300 ns or less, a normal value of the analog conversion value might not be obtained.
Set the CT1 and CT0 bits to be 700 ns or more at the compare time. When the compare time is set to 700 ns or less, a normal value of the analog conversion value might not be obtained.

About the setting of the ADMD register and ADSTPCS

Please rewrite the bits of the A/D mode setting register (ADMD) and the sampling time setting register (ADSTPCS) when the A/D operation has stopped before A/D conversion operation.

36. Flash Memory

This chapter explains the flash memory.

36.1 Overview

36.2 Features

36.3 Configuration

36.4 Registers

36.5 Operation

36.1 Overview

This section explains the overview of the flash memory.

The size of the flash memory built in this series is up to 192 Kbytes (128 Kbytes + 64 Kbytes). Error correction codes (ECC) are attached.

36.2 Features

This section explains features of the flash memory.

Usable capacity:
MB91F552: 128 Kbytes + 64 Kbytes (large sectors 128 Kbytes x 1 + small sectors16 Kbytes x 4)

For ECC code storage in this model, there are 6 bits of built-in flash memory in addition to the above for every 4 bytes.
Note: The sector itself has 64 Kbytes (large sector)/8 Kbytes (small sector) capacity, but the pair of sectors (odd and even) is alternately mapped word-by-word in the address space so that the sectors are to be used in 128 Kbytes (large sectors)/16 Kbytes (small sectors) in real use.

- High speed operation:
  Reading on a word-by-word basis (32 bit) is possible by 80 MHz x 1 cycle.
- Write from external:
  Possible from ROM writer
- Operation mode:
  1. CPU-ROM mode
     (CPU/DMA accesses the flash memory. Read-only)
  2. CPU programming mode
     (CPU accesses the flash memory. Read/Write/Erase)
  3. Flash memory mode (flash memory accessible from external)
- Can be read, written, or erased (automatic algorithm*) by CPU
- Can be read, written, or erased (automatic algorithm*) by ROM writer
- Security function
  - ☐ Operations after instruction fetch from external and write/erase except for chip erase are inhibited when security is on to prevent an outsider from reading out flash memory data.
  - ☐ The use of on-chip debugger (OCD) enables read from external by using OCD, even if security is on after password authentication.
- Error correction code (ECC) function
  - ☐ There is an error correction code (ECC) function that corrects errors of up to 1 bit in each word. (A function for detecting 2-bit errors is not provided.) Errors are automatically corrected during read. Furthermore, ECC codes are automatically added during writing to flash memory. Because there is no read cycle penalty due to error correction, no consideration needs to be given to error correction penalties during software development.
  - ☐ An error is detected when data is read in the chip erase/sector erase state.
    If data in the erase state (FFFF) needs to be read correctly, be sure to first write "FFFF" before reading it.

*: Automatic algorithm = Embedded Algorithm™

36.3 Configuration

This section explains the configuration of the flash memory.

36.3.1 Block Diagram

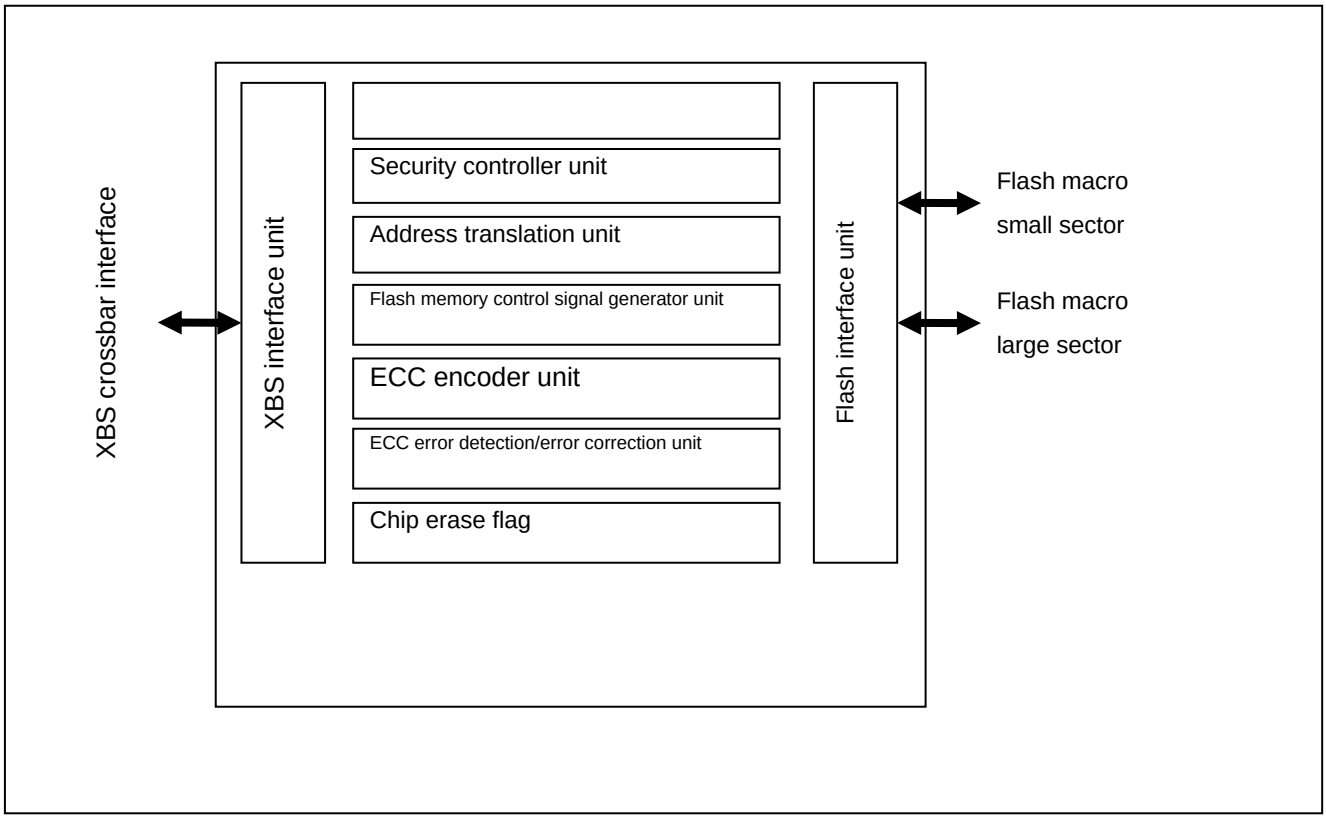
36.3.2 Sector Configuration Diagram

36.3.3 Sector Number and Flash Macro Number Correspondence Chart

36.3.1 Block Diagram

This section shows the block diagram of the flash memory.

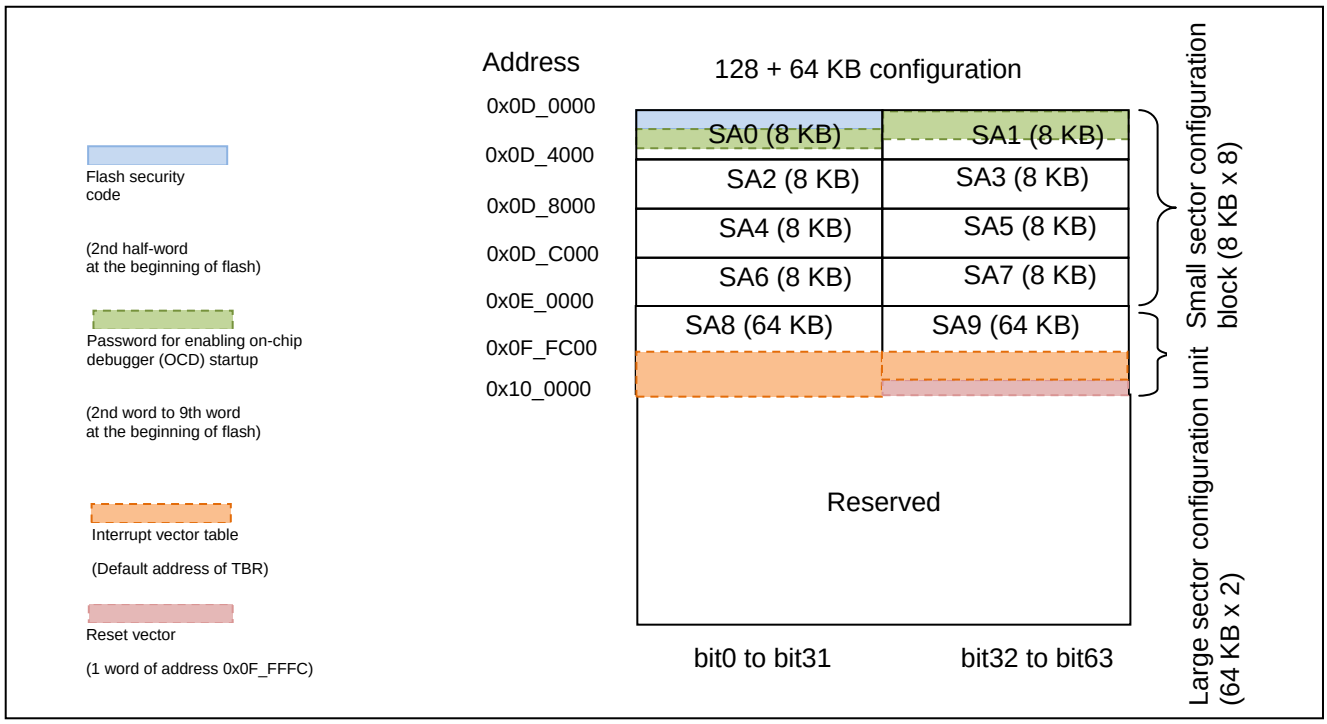
Figure 36-1. Block Diagram



36.3.2 Sector Configuration Diagram

The sector configuration diagram of the flash memory is shown below.

Figure 36-2. Sector Configuration Diagram (MB91F552)



Note:

The FixedVector function returns the start address of flash memory + 0x0024 instead of the value written in address 0x0F_FFFC as the reset vector. For details, see "Chapter: Fixedvector Function."
As for a password setting for enabling on-chip debugger (OCD) to start, see "Chapter: On Chip Debugger (Ocd)." If it is unnecessary to use the security function for on-chip debugger (OCD), do not write anything to the area and keep the initial state just after the flash erase (all bits=1).

36.3.3 Sector Number and Flash Macro Number Correspondence Chart

Sector number and flash macro number correspondence charts are shown below.

Table 36-1. Sector Number Table MB91F552 (128 + 64 KB Models)

Sector Number	Address	Sector Size	Remark
SA0	0x0D_0000 to 0x0D_3FFB (Lower 32 bits)	8 KB	Flash security code area (0x0D_0002 to 0x0D_0003) Password area for enabling on-chip debugger (OCD) startup (0x0D_0008 to 0x0D_000B, 0x0D_0010 to 0x0D_0013, 0x0D_0018 to 0x0D_001B, 0x0D_0020 to 0x0D_0023)
SA1	0x0D_0004 to 0x0D_3FFF (Upper 32 bits)	8 KB	Password area for enabling on-chip debugger (OCD) startup (0x0D_0004 to 0x0D_0007, 0x0D_000C to 0x0D_000F, 0x0D_0014 to 0x0D_0017, 0x0D_001C to 0x0D_001F)
SA2	0x0D_4000 to 0x0D_7FFB (Lower 32 bits)	8 KB	
SA3	0x0D_4004 to 0x0D_7FFF (Upper 32 bits)	8 KB	
SA4	0x0D_8000 to 0x0D_BFFB (Lower 32 bits)	8 KB	
SA5	0x0D_8004 to 0x0D_BFFF (Upper 32 bits)	8 KB	
SA6	0x0D_C000 to 0x0D_FFFB (Lower 32 bits)	8 KB	
SA7	0x0D_C004 to 0x0D_FFFF (Upper 32 bits)	8 KB	
SA8	0x0E_0000 to 0x0F_FFFB (Lower 32 bits)	64 KB	
SA9	0x0E_0004 to 0x0F_FFFF (Upper 32 bits)	64 KB	
-	0x0F_FC00 to 0x0F_FFFF	-	Interrupt vector table position (Default of TBR) (0x0F_FC00 to 0x0F_FFFB) Reset vector position (0x0F_FFFC to 0x0F_FFFF)

36.4 Registers

This section explains registers of the flash memory.

Table 36-2. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0840	FCTLR		Reserved	FSTR	Flash control register Flash status register
0x2308	FLIFCTLR	Reserved	FLIFFER1	FLIFFER2	Flash interface control register Flash interface feature extension register 1 Flash interface feature extension register 2

36.4.1 Flash Control Register: FCTLR (Flash ConTroL Register)

The bit configuration of the flash control register is shown below.

This register configures the access control to flash.

FCTLR: Address 0840_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	FWE	Reserved		FSZ[1:0]		FAW[1:0]	
Initial Value	1	0	-	-	1	0	0	0
Attribute	R1,WX	R/W	RX,W0	RX,W0	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FDSBL	Reserved		RDYF	Reserved			
Initial Value	0	-	-	0	-	-	-	-
Attribute	R/W	RX,W0	RX,W0	R/W	RX,W0	RX,W0	RX,W0	RX,W0

[bit15] Reserved

This bit is reserved. This bit always reads out as "1". Writing has no effect on the operation.

[bit14] FWE (Flash Write Enable): Flash write enable

This bit enables writing of flash memory. Setting this bit configures CPU programming mode. Use the FSTR:FRDY bit to check whether writing is enabled.

If this bit is set, the ECC error detection and data correcting function will be disabled for data fetching to the flash memory.

FWE	Description
0	Disable writing to flash (initial value).
1	Enable writing to flash.

Note:

When flash memory is being written, instruction fetch from the flash memory is prohibited.

[bit13, bit12] Reserved

These bits are reserved. The read value is undefined. When writing, always write "0" to these bits.

[bit11, bit10] FSZ[1:0] (Flash write access SiZe): Flash write access size setting

The FLASH write access size at CPU mode is specified. Be sure to write in the specified bit count of the access width. These bits do not influence the reading access size. 32-bit read is done to the flash macro whenever it is read. When the wait cycle is inserted by the FAW bit, it becomes 64-bit read access.

FSZ[1:0]	Description
00	8-bit
01/10/11	16-bit

[bit9, bit8] FAW[1:0] (FLASH Access Wait): Flash access wait setting

The wait cycle to the flash access at CPU mode is set. Because the reading time of the flash memory is 12.5ns, access to the flash memory at over 80MHz is disabled. Please set it to FAW=1(1wait) when you access it at over 80MHz.

Please set these bits before making the clock high-speed when you insert the wait cycle by FAW. Moreover, please set these bits after setting the clock low-speed when you delete the wait cycle.

FAW[1:0]	Description
00	0 cycle (initial value)
01	1 cycle
10/11	Setting is prohibited.

Note:

When 1 wait cycle is set by these bits, the wild register function cannot be used. Please make the core operation speed to 80 MHz or less, and set value of the FAW bits to 2'b00(0 cycle) when you use the wild register function.

[bit7] FDSBL (Flash DiSaBLe): Flash Disable directive

This bit configures the flash access disabled state (both reads and writes).

FDSBL	Description
0	Enable flash access (Initial value).
1	Disable flash access.

[bit6, bit5] Reserved

These bits are reserved. The read value is undefined. When writing, always write "0" to these bits.

[bit4] RDYF (ReaDY Flag): RDY negating instruction during branch access

This bit directs the wait cycle insertion during branch access. During branch access, the wait cycle is inserted when this bit is set to "1". The purpose of this is to match the processing cycle when branching. When the branch access is generated, the control at the wait cycle is made by an internal state of the flash interface when this bit is "0". If the cycle time is not necessary to be secured when the branch access is accepted, the wait cycle is not inserted. When it is necessary to secure the cycle time, the wait cycle is inserted.

RDYF	Description
0	It depends on the state of FLASH I/F (initial value).
1	Insert wait cycle.

[bit3 to bit0] Reserved

These bits are reserved. The read value is undefined. When writing, always write "0" to these bits.

36.4.2 Flash Status Register: FSTR (Flash STatus Register)

The bit configuration of the flash status register is shown below.

This register indicates the flash memory state.

FSTR: Address 0843_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					FECCERR	FHANG	FRDY
Initial Value	-	-	-	-	-	0	0	1
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	R,W	R,WX	R,WX

[bit7 to bit3] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] FECCERR (Flash ECC Error coRRection): Data read ECC correction occurred

This bit is set if an ECC error correction occurs while reading flash memory other than CPU instruction read. This bit is cleared by writing "0".

FECCERR	Read	Write
0	An error correction by ECC has not occurred during data read (initial value).	Clear this bit.
1	ECC error correction occurred during data read.	No effect

If there are errors in 2-bit or more in a single word, the read value of this bit is undefined.
When reading a CPU instruction, this bit is not set even if an ECC error correction occurs.
When both an ECC error and 0 writing occur simultaneously, the 0 writing will take priority.

[bit1] FHANG (Flash HANG): Flash HANG state

This bit indicates the flash memory HANG state.

FHANG	Description
0	Normal state
1	HANGUP state

If there is a timing overrun (See "[bit5] TLOV: (Timing limit exceeded flag bit)"), the flash memory will go into the HANG state. If this bit becomes "1", issue a reset command (See "36.5.3.1 Command Sequence").
The correct value might not be read out immediately after a command of automatic algorithm has been issued. Therefore, ignore the 1st read value of this bit after the command issuance.

[bit0] FRDY (Flash ReaDY): Flash write enable

This bit indicates whether the flash memory write/erase operation by automatic algorithm is currently running or finished. Flash memory cannot be written or erased while the operation is in progress.

FRDY	Description
0	During operation (write/erase disabled, read status enabled)
1	Completion of operation (write/erase enabled, read enabled)

The correct value might not be read out immediately after a command of automatic algorithm has been issued. Therefore, ignore the 1st read value of this bit after the command issuance.

36.4.3 Flash Interface Control Register: FLIFCTLR (Flash I/F Control Register)

The bit configuration of the flash interface control register is shown below.

This register controls the flash interface. This register is shared among program flash and WorkFlash.

FLIFCTLR: Address 2308_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			DFWDSBL	Reserved		ECCDSBL1	ECCDSBL0
Initial Value	-	-	-	0	-	0	0	0
Attribute	RX,WX	RX,WX	RX,WX	R/W	RX,WX	R/W0	R/W	R/W

[bit7 to bit5] Reserved

These bits are reserved bits. The read value is undefined. Writing has no effect on the operation.

[bit4] DFWDSBL (Data Fetch Wait cycle Disable): Data fetch wait cycle disabled

If this bit is set to "1", the wait cycle inserted when setting wait at data fetch is disabled. However, you cannot disable the wait cycle to guarantee the cycle time.

DFWDSBL	Description
0	Enable wait cycle (initial value).
1	Disable wait cycle.

Note:

When this bit is changed from "1" to "0", always set FCTLR.FAW="00" before changing this bit.

[bit3] Reserved

This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] Reserved

This bit is reserved. Always write "0" to this bit.

[bit1] ECCDSBL1 (ECC Disable1): ECC function disable 1

This bit configures enable/disable for the ECC function when write access and data fetch is performed to WorkFlash memory in the CPU mode.

ECCDSBL1	Description
0	Enable ECC function (initial value).
1	Disable ECC function.

[bit0] ECCDSBL0 (ECC Disable0): ECC function disable 0

This bit configures enable/disable for the ECC function when write access and data fetch is performed to program flash memory in the CPU mode.

ECCDSBL0	Description
0	Enable ECC function (initial value).
1	Disable ECC function.

36.4.4 Flash Interface Feature Extension Register 1: FLIFFER1 (FLash I/F Feature ExtensiONRegister 1)

The bit configuration of the flash interface feature extension register 1 is shown below.

This register is the spare register. If the register is written, please write 0xFF.

FLIFFER1: Address $230A_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1

[bit7 to bit0] Reserved

These bits are reserved. Always write 0xFF to these bits.

36.4.5 Flash Interface Feature Extension Register 2: FLIFFER2 (FLash I/F Feature ExtensiONRegister 2)

The bit configuration of the flash interface feature extension register 2 is shown below.

This register is the spare register. If the register is written, please write 0xFF.

FLIFFER2: Address 230B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1

[bit7 to bit0] Reserved

These bits are reserved. Always write 0xFF to these bits.

36.5 Operation

This section explains the operation of the flash memory.

This section explains how to access the flash memory area.

36.5.1 Access Mode Setting
36.5.2 Writing Flash Memory by CPU
36.5.3 Automatic Algorithm
36.5.4 Reset Command
36.5.5 Write Command
36.5.6 Chip Erase Command
36.5.7 Sector Erase Command
36.5.8 Sector Erase Suspend Command
36.5.9 Security Function
36.5.10 Notes on Using Flash Memory

36.5.1 Access Mode Setting

Access mode setting is shown below.

The flash memory in this model has the following 3 modes. Methods of setting the modes (1) and (2) are explained in this section. As for the mode (3), see the instruction manual of the ROM writer you are using for details.

(1) CPU-ROM mode
(CPU accesses flash memory. For only read, Byte/Half-word/Word access)

(2) CPU programming mode
(CPU accesses flash memory. For read/write, only Half-word access)

(3) Flash memory mode
(Flash memory accessible from external)

36.5.1.1 Configuring CPU-ROM Mode

Configuring CPU-ROM mode is shown below.

When the FWE bit of the flash control register (FCTLR) is "0", it is CPU-ROM mode. In the CPU-ROM, when the FRDY bit of the flash status register (FSTR) is "1", read from the flash memory is enabled. In the CPU-ROM mode, write to the flash memory is disabled. After reset is released, the mode will be the CPU-ROM mode.

36.5.1.2 Configuring CPU Programming Mode

Configuring CPU programming mode is shown below.

When the FWE bit of the flash control register (FCTLR) is "1", it is CPU programming mode. When the FRDY bit of the flash status register (FSTR) is "1", read/write from/to the flash memory is enabled in this mode.

36.5.2 Writing Flash Memory by CPU

Writing the flash memory by the CPU is shown below.

After configuring CPU programming mode, perform erasing and programming using the automatic algorithm.
In this model, because error correction codes (ECC) are added to each single word, programming needs to be performed for each single word.
In the following procedure, each word is programmed by 2 operations to write 1 half-word. If this procedure is not followed, the written values will not be read correctly because the values will be written to the flash memory without ECC calculation.

1. Set the flash write access size to 16 bits. (FCTLR.FSZ[1:0]=01)
2. Issue the write command. Write address = PA, write data = PD[31:16]
   See "36.5.5 Write Command" for details on the write command.
3. Read the hardware sequence flag until the write has finished. See "36.5.3.2 Automatic Algorithm Execution State" for details on hardware sequence flag read.
4. Issue the write command. Write address = PA+2, write data = PD[15:0]
   At this time, the hardware automatically calculates the ECC codes by combining with PD[31:16] from (2), and writing of ECC codes is also performed automatically at the same time.
5. Read the hardware sequence flag until the write has finished.
6. If there is more data to write, return to (2). Continue to (7) when all writes have finished.
7. Set CPU-ROM mode.
8. Read the value which has already been written, and check that the correct value can be read. Even if the correct value can be read, check the FSTR:FECCERR bit to make sure that there was no ECC correction.
   If ECC correction occurs, follow the same procedure again, starting with erasing the flash memory.

PA: Write target address (word alignment)
PD[31:0]: Write data
PD[31:16]: Write data upper 16-bit
PD[15:0]: Write data lower 16-bit

36.5.3 Automatic Algorithm

The automatic algorithm is shown below.

When using CPU programming mode, write and erase of flash memory are performed by starting the automatic algorithm. This section explains the automatic algorithm.

36.5.3.1 Command Sequence

The command sequence is shown below.

The automatic algorithm starts when half-word (16-bit data) is written to flash memory once to 6 times in a row. This is called a command. The command sequences are shown below.

Table 36-3. Command Sequence

Command	Number of Writing	1st Time		2nd Time		3rd Time		4th Time		5th Time		6th Time	
		Address [11:0]	Data [7:0]	Address [11:0]	Data [7:0]	Address [11:0]	Data [7:0]	Address [11:0]	Data [7:0]	Address [11:0]	Data [7:0]	Address [11:0]	Data [7:0]
Reset	1	arbitrary	$F0_H$										
Read	1	RA	RD										
Write	4	$x554_H$	AA_H	$yAA8_H$	55_H	$x554_H$	$A0_H$	PA	PD				
Chip Erase	6	$x554_H$	AA_H	$yAA8_H$	55_H	$x554_H$	80_H	$x554_H$	AA_H	$yAA8_H$	55_H	$x554_H$	10_H
Sector Erase	6	$x554_H$	AA_H	$yAA8_H$	55_H	$x554_H$	80_H	$x554_H$	AA_H	$yAA8_H$	55_H	SA	30_H
Sector Erase Suspend	1	arbitrary	$B0_H$										
Sector Erase Resume	1	arbitrary	30_H										

- The data written in the table only shows the lower 8-bit. The upper 8-bit can be any value. The commands must be written as bytes or half-words.

- The addresses written in the table only show the lower 16-bit. Set the upper 16-bit to any address within the address range of the target flash macro.

x: 1,3,5,7,9,B,D,F
y: 0,2,4,6,8,A,C,E
PA: Write address (half-word alignment)
PD: Write data (Write as 16-bit.)
SA: Sector address (specify an arbitrary address within the address range of the sector to erase.)
RA: Read address
RD: Read data (the read width is arbitrary.)

Notes:

- When the wrong address value and data value are written or writing is performed in the wrong sequence, commands that have been written are cleared.
- Do the following to the LSB 2-bit for the command address, and for the sector address (SA) issued at the generation of the sector erase command.
  - When half-word access: 2'b00
  - When byte access: 2'b01 or 2'b11

  Example 1:
  During byte access, if command address = (LSB 2-bit of the standard command address changed to 2'b01.)
  $yAA8_H$ -> $yAA9_H$, $x554_H$ -> $x555_H$, SA -> {SA[31:2], 2'b01}
  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector)
  Example 2:
  During byte access, if command address = (LSB 2-bit of the standard command address changed to 2'b11.)
  $yAA8_H$ -> $yAAB_H$, $x554_H$ -> $x557_H$, SA -> {SA[31:2], 2'b11}
  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector)

Reset command

Sending the reset command to the target flash memory enables the sequential input of each command shown in Table 36-1 to be canceled, and commands can be input again from the 1st time.
However, when each command is input to the last minute and automatic algorithm starts, automatic algorithm cannot be discontinued by this reset command.
If the execution of the automatic algorithm exceeds the timing limit, the flash memory returns to the reset state if a reset command is input.

Read command

The flash memory can be read by sending read commands to the target sector. If a read command is issued, the flash memory stays in read state until another command is issued.

Programming (write) command

If a write command is sent to the target sector 4 times in a row, the automatic algorithm starts and writes data to the flash memory. Programming (writing) of data can be performed in any order of addresses or across a sector boundary.
In the CPU programming mode, data is written in half-words or bytes. Once the 4th write has finished, the automatic algorithm starts and the automatic write to flash memory is started.
After executing the automatic write algorithm command sequence, there is no need to control the flash memory externally.
See "36.5.5 Write Command" for details on the actual operation.

Notes:

- When writing in half-word, if the 4th command (write data cycle) is written in the odd address, writing is not performed correctly. Always write in even address.
- With 1 write command sequence, only a single half-word data can be written. If you want to write multiple data, issue 1 write command sequence for each data.
- While security is on, the sector erase procedure of flash is limited. See "36.5.9.4 Flash Access Restrictions When Security Is ON" for details.

Chip erase command

If the chip erase command is sent to the target sector 6 times in a row, all sectors of the flash memory can be erased in one step. Once the 6th write has finished, the automatic algorithm starts and the chip erase operation is started.
When the automatic erase algorithm is started, "0" is written to all of the cells in the flash memory chip before erasing the entire chip, and there is no need to write to the flash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.
See "36.5.6 Chip Erase Command" for details on the actual operation.

Sector erase command

If the sector erase command is sent to the target sector 6 times in a row, the sector of the flash memory can be erased. When 40 µs elapses (timeout period) after the 6th write has finished, the automatic algorithm starts and the sector erase operation is started.
If you want to erase multiple sectors, write the erase code (30H) to the address of the sector to erase within the 40 µs (timeout period). If the next sector is not input within the timeout period, the sector erase command may become invalid.
When the automatic erase algorithm is started, "0" is written to the cells in the sector to erase in the flash memory before erasing the sector, and there is no need to write to the flash memory before erasing the sector to verify the margins (preprogramming).

Furthermore, while verifying the margin, there is no need to control the flash memory externally.
See "36.5.7 Sector Erase Command" for details on the actual operation.

Note:

While security is on, the sector erase procedure of flash is limited. See "36.5.9.4 Flash Access Restrictions When Security Is ON" for details.

Sector erase suspend command

It is possible to shift to the sector erase suspend condition (state of the sector erase suspension) by sending the sector erase suspend command in the command time-out or while executing the sector erase.
In the sector erase suspend condition, the reading operation of the memory cell of the sector that is not the erase target becomes possible. However, a new neither writing nor erase command is accepted.
To restart the interrupting erase operation from the sector erase suspend condition, the erase restart command is sent. When the flash memory accepts the erase resume command, it goes back to sector erase state and starts erase operation again.
It does not change to the state of the command time-out when the erase resume command is normally written even if it is time when it changes from the state of the command time-out in this state, it changes to the state of the sector erase, and the sector erase operation is restarted at once.

See "36.5.8 Sector Erase Suspend Command" for details on the actual operation.

Note:

16.7µs + 2 cyc or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector that is not the erase target becomes possible.
Whether it entered the state that can be read is confirmed with the FRDY bit of the flash status register (FSTR) or TOGG1 of the hardware sequence flag.

36.5.3.2 Automatic Algorithm Execution State

The automatic algorithm execution state is shown below.

Because writing and erasing flash memory is performed by an automatic algorithm, whether or not the automatic algorithm is executing can be checked by the FRDY bit of the flash status register (FSTR), and the operating state can be checked by the hardware sequence flag.

Hardware sequence flag

This flag indicates the state of the automatic algorithm. When the FRDY bit of the flash status register (FSTR) is "0", the operating state can be checked by reading from an arbitrary address in flash memory. The bit configuration of the hardware sequence flag is shown below.

Figure 36-3. Bit Configuration of Hardware Sequence Flag

For half-word access

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
bit7	**bit6**	**bit5**	**bit4**	**bit3**	**bit2**	**bit1**	**bit0**
DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

For byte access

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

Notes:

- It is impossible to read by word access. Always read using half-word or byte access in CPU programming mode.
- In the CPU-ROM mode, the hardware sequence flag cannot be read no matter which address is read.

Each bit and flash memory status

The following table shows the correspondence between the status of each bit of the hardware sequence flag and the flash memory status.

Table 36-4. Correspondence between Flags and Flash Memory Status

Status		DPOLL	TOGG1	TLOV	SETI	TOGG2
Running	Writing	Inverted data*	Toggle	0	0	-
	Sector/Chip erasing	0	Toggle	0	1	-
Time limit exceed	Write command	Inverted data*	Toggle	1	0	-
	Sector erase/Chip erase command	0	Toggle	1	1	-
Sector erase suspend	Erase target sector	-	-	-	-	Toggle

*: See "Bit descriptions" for the values that are read out.

Bit descriptions

[bit15 to bit8] Undefined bits

[bit7] DPOLL (Data polling flag bit)

When the hardware sequence flag is read by specifying the write/erase target address, this bit indicates whether or not the automatic algorithm is running using a data polling function.
The value that is read differs depending on the operating state.

1. When writing

During execution of writing	Reads out the opposite value (inverted data) of the value of bit7 of the last data to be written. The address specified for reading the hardware sequence flag is not accessed.
After writing finished	Reads out the value of bit7 of the address specified for reading the hardware sequence flag.

2. During sector erase

During execution of sector erase	Reads "0" from the sector being erased.
After sector erase	This bit always reads out as "1".

3. During chip erase

During execution of chip erase	This bit always reads out as "0".
After chip erase	This bit always reads out as "1".

4. During sector erase suspend

State of suspend (incomplete end)	Reads "0" from the sector erase suspend sector.
Sector erase operation completion	Reads "1" from the sector erase suspend sector.

Note:

When the automatic algorithm is running, the data for the specified address cannot be read. Read data after using this bit to check whether the automatic algorithm operation has finished.

[bit6] TOGG1 (Toggle flag 1 bit)

When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the automatic algorithm is running.
The value that is read differs depending on the operating state.

During write/sector erase/chip erase

During write/sector erase/ chip erase	If this bit is read sequentially, "1" and "0" are read alternatively (toggle operation). The address specified for reading the hardware sequence flag is not accessed.
After write/sector erase/ chip erase	Reads out the value of bit6 of the address specified for reading the hardware sequence flag.

[bit5] TLOV: (Timing limit exceeded flag bit)

When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the execution time of the automatic algorithm exceeded the time specifying internally within the flash memory (number of internal pulses).
The value that is read differs depending on the operating state.

During write/sector erase/chip erase

One of the following values is read.

"0"	Within the rated time
"1"	Exceed rated time.

When this bit is "1", if the DPOLL bit and TOGG1 bit indicate that the automatic algorithm execution is in progress, the write or erase has failed.
For example, because data written to "0" cannot be rewritten with "1" in flash memory, when an attempt is made to write "1" to an address already written with "0", the flash memory is locked and the automatic algorithm does not end. In this case, the value of the DPOLL bit remains invalid and the value read from the TOGG1 bit continues to alternate between "1" and "0". If the rated time is exceeded in this state, this bit changes to "1". If this bit becomes "1", issue a reset command.

Note:

When this bit is "1", this indicates that the flash memory was not used correctly. The flash memory is not faulty. Perform the appropriate processing after issuing the reset command.

[bit4] Undefined bit

[bit3] SETI (Sector erase timer flag bit)

During sector erase, a timeout period of 40 µs is required from when the sector erase command is issued until the sector erase actually starts. When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the sector erase command is within the timeout period.
The value that is read differs depending on the operating state.

When erasing sectors

When erasing sectors, you can check whether the next sector erase code is ready to be accepted by checking this bit before inputting the next sector erase code. One of the following values is read out without accessing the address specified for reading the hardware sequence flag.

"0"	Within sector erase wait period (the next sector erase code (0x30) can be accepted.)
"1"	Exceed the sector erase wait period. (If the DPOLL bit and TOGG1 bit indicate that the automatic algorithm is executing at this time, the flash memory internal erase is started. In this case, commands other than the sector erase code (0x30) are ignored until the flash memory internal erase finishes.)

[bit2] TOGG2 (Toggle flag 2 bit)

In the sector erase suspend state, non target sector for erase can be read, but target sector for erase cannot be read. This flag indicates that output data is toggled and target sector for erase when read address is the target sector for erase during sector erase suspend.

Read out target erase sector	If this bit is read sequentially, "1" and "0" are read alternatively (toggle operation). The address specified for reading the hardware sequence flag is not accessed.
Read out non target erase sector	Read data from specified address

[bit1, bit0] Undefined bits

36.5.4 Reset Command

The reset command is shown below.

The flash memory can be reset by sending reset commands to the target flash memory. Because this state is the flash memory initial state, the flash memory always returns to the reset state when the power is turned on or a command finishes successfully.
When the power is turned on, there is no need to issue a reset command.
Furthermore, in reset state, data can be read using normal read access and programs can be accessed by the CPU; thus there is no need to issue the reset command when reading data.

36.5.5 Write Command

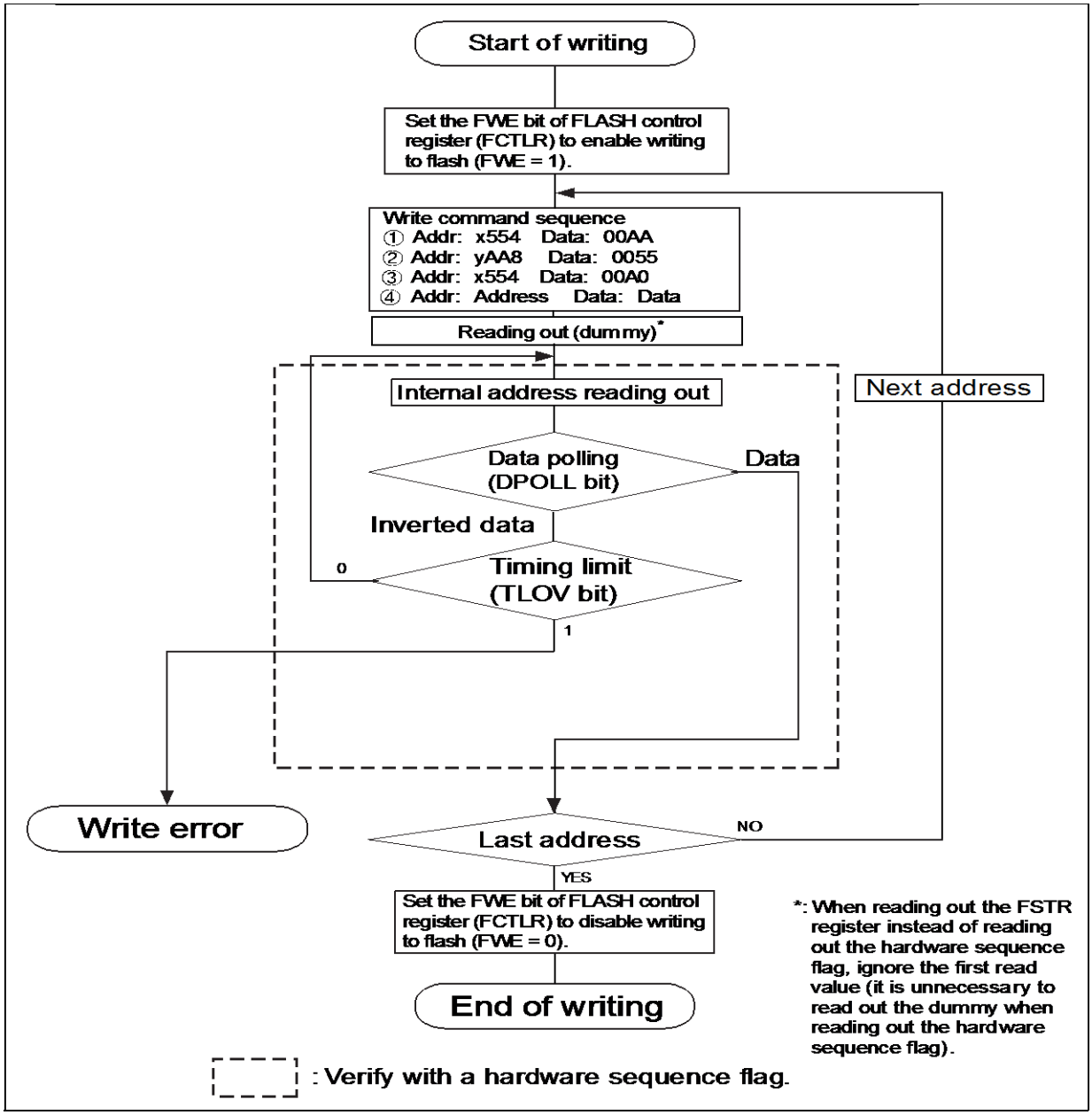
The write command is shown below.

Writes are performed in the following order.

1. Send write commands sequentially to the target sector.
   The automatic algorithm is started and data is written to the flash memory.
   After issuing a write command, there is no need to control the flash memory externally.
2. Perform read access to the written address.
   The read data is the hardware sequence flag. Therefore, if bit7 (DPOLL bit) of the read data matches the written data, the write to the flash memory has finished.
   If the write has not finished, the opposite value (inverted data) of the value of bit7 of the last written data is read out.

The following figure shows an example of write operation to the flash memory.

Figure 36-4. Example of Write Procedure



Notes:

- Once the write has finished, the write address is not accepted because the flash memory returns to read mode.
- See "36.5.3 Automatic Algorithm" for details on write commands.
- Because the value of the DPOLL bit of the hardware sequence flag changes concurrently with the TLOV bit, check this bit again even if the value of the TLOV bit is "1".
- The moment when the TOGG1 bit of the hardware sequence flag and TLOV bit change to "1", the toggle operation stops. Therefore, even if the TLOV bit is "1", checking the TOGG1 bit again must be needed.
- Although flash memory can be written to in any order of addresses, even if it crosses a sector boundary, only a single half-word data can be written in each write command sequence. If you want to write multiple data, issue 1 write command sequence for each data.
- Data that has been written to "0" once cannot be returned to "1". If "0" is rewritten with "1", one of the following occurs.
  - ☐ The element is judged as faulty by the data polling algorithm.
  - ☐ The write rated time is exceeded, and the TLOV bit of the hardware sequence flag changes to "1".
  - ☐ It appears to have been written as "1".

However, even if it appears to have been written as "1", the actual data remains "0" and "0" will be read out when the data is read in read/reset mode. If you want to return data to "1", perform a chip erase or sector erase.

- During write operations, all commands written to flash memory are ignored.
- If this device is reset during a write, the data that was written cannot be guaranteed.
- Because this model has the ECC bit added, data always needs to be written as 32-bit by 2 16-bit writes. See "36.5.2 Writing Flash Memory by CPU" for the procedure.

36.5.6 Chip Erase Command

The chip erase command is shown below.

The erase target flash macros in the flash memory can be erased in one step using the chip erase command.

If the chip erase command is sent to the target flash memory sequentially, the automatic algorithm starts and all sectors of the flash memory can be erased in one step. See "36.5.3 Automatic Algorithm" for details on the chip erase command.

Chip erase is performed in the following order.

1. Send chip erase commands sequentially to a sector in the WorkFlash macro to erase.
   The automatic algorithm is started and data is written to the flash memory.

2. Perform a read access to an arbitrary address in the flash macro to erase.
   The read data is the hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", the chip erase has finished.

The time required to erase the chip is [sector erasure time x total no. of sectors + chip write time (preprogram)]. When the chip erase operation finishes, the flash memory returns to the read/reset state.

Note:

- When the automatic erase algorithm is started, "0" is written to all of the cells in the flash memory chip before erasing the entire chip, and there is no need to write to the flash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.
- When security is on, there are restrictions in the procedure for erasing the flash. See "36.5.9.3 Flash Security Unlocking Method" for details.

36.5.7 Sector Erase Command

The sector erase command is shown below.

A sector in the flash memory can be selected and only data in the selected sector can be deleted. Multiple sectors can also be specified at the same time.

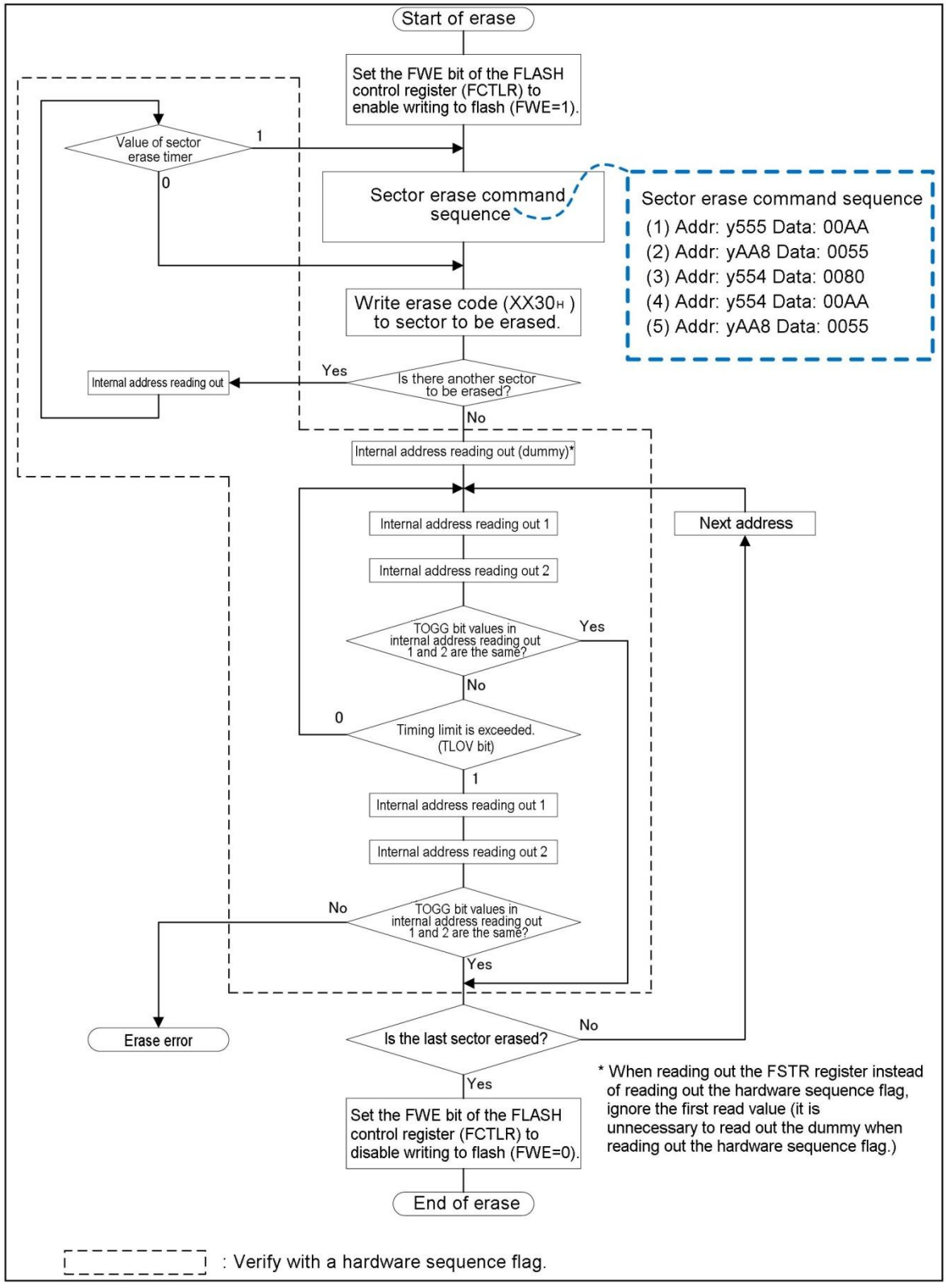
Sector erase is performed in the following order.

1. Send sector erase commands sequentially to the target sector.
   Once 40 µs has elapsed (timeout period), the automatic algorithm starts and the sector erase operation is started. If you want to erase multiple sectors, write the erase code (30H) to the address of the sector to erase within the 40 µs (timeout period). If the write is performed after the timeout period has elapsed, the sector erase command may be invalid.

2. Perform read access to an arbitrary address.
   The read data is the hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", the sector erase has finished. Furthermore, you can use the TOGG1 bit to check whether or not the sector erase has finished.

The following shows an example of the sector erase procedure taking the example of using the TOGG1 bit in the check operation.

Figure 36-5. Example of Sector Erase Procedure



Notes:

- The time required to erase the sector is [(sector erasure time + sector write time (preprogram)) x no. of sectors].
- When the sector erase operation finishes, the flash memory returns to the read/reset mode.
- See "36.5.3 Automatic Algorithm" for details on the sector erase command.
- Because the value of the DPOLL bit of the hardware sequence flag changes concurrently with the TLOV bit, check this bit again even if the value of the TLOV bit is "1".
- When the TOGG1 bit and the TLOV bit of the hardware sequence flag change to "1", toggle operation stops at the same time. Therefore, even if the TLOV bit is "1", checking the TOGG1 bit again must be needed.
- In the command timeout state, the only accepted write commands are the erase sector add command and erase suspend command. In the sector erase state, the only accepted command is the erase suspend command.
- When the automatic erase algorithm is started, "0" is written to the cells to erase in the flash memory before erasing the sector, and there is no need to write to the flash memory before erasing the sector to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

36.5.8 Sector Erase Suspend Command

The sector erase suspend command is shown below.

The sector erase can temporarily be stopped in the command time-out or while executing the sector erase.

In the sector erase suspend condition, the reading operation of the memory cell of the sector that is not the erase target becomes possible. However, a new neither writing nor erase command is accepted.

The sector erase suspend command is sent to an arbitrary address of target FLASH macro to suspend the sector erase.

After the sector erase stops, the reading operation from target flash macro is permitted. At this time, the hardware sequence flag is read from the sector which is under the sector erase suspend condition.

It enters the following states when entering the sector erase suspend condition.

- The TOGG1 bit that does the toggle while erasing the sector does not toggle while being suspended from the sector erase operation.
- FRDY of the flash status register becomes "1".

The thing that entered the sector erase suspend condition can be confirmed by using these.

Note:

The maximum cycles required from the issuing of the sector erase suspend command until the sectors that are not erase targets can be read after the sector erase operation stops is 16.7 µs + 2 cycles.

After resume the sector erase, before run the sector erase suspend command, you should wait 2ms at least.

Because bit2:TOGG2 of the hardware sequence flag does the toggle while the sector erase is temporarily stopping, the sector can be confirmed while stopping by using this bit.

To restart the interrupting erase operation from the sector erase suspend condition, the sector erase restart command in Table 36-3 Command Sequence is sent.

The sector erase restart command is accepted only by the sector erase suspend condition.

Send the command after confirming becoming the sector erase suspend condition.

When the flash memory accepts the erase resume command, it goes back to sector erase state and starts erase operation again.

36.5.9 Security Function

The security function is shown below.

This flash memory is equipped with a security function.

When the security function is off, the flash memory can be used without limits. However, when the security function is on, writes and erases other than chip erase are suppressed. See "36.5.9.4 Flash Access Restrictions When Security Is ON" for details on the restrictions.

36.5.9.1 Flash Security On/Off Determination When Reset Released

Flash security on/off determination when reset released is shown below.

The flash interface of this model reads 2 bytes from the flash security code area after reset is released. If this value is 0x0001, security is turned on and access restrictions are imposed on subsequent accesses to flash memory. For any other value, the security is turned off.

36.5.9.2 Flash Security Setting Method

The flash security setting method is shown below.

When reset is input and released after writing 0x0001 to the flash security code area (see "Figure 36-2. Sector Configuration Diagram (MB91F552)"), security is turned on. Once security has been turned on, the security is not turned off unless the entire flash memory area is erased.

36.5.9.3 Flash Security Unlocking Method

The flash security unlocking method is shown below.

The chip erase command can be performed on all flash macros in the following order.

1. Erase WorkFlash.
2. Erase program flash which contains the flash security code.

Erase program flash last, as shown above. Otherwise, the erase command to program flash is ignored. Furthermore, if a reset is input between erases, repeat from step (1).

Note:

In the user-mode (internal FLASH activation), the erase command can be issued to an arbitrary flash macro, and data in the flash macro be deleted. The recommended order of the chip erase to each WorkFlash macro is as shown above, from the viewpoint of the data protection stored in the WorkFlash macro.

36.5.9.4 Flash Access Restrictions When Security Is ON

Flash access restrictions when security is on are shown below.

When security is on, the restrictions shown below are created by the start mode.

Table 36-5. Access Restrictions when Security Is ON

Operating Mode	Access Restriction
User	In normal mode (the state where there are no access restrictions due to the following flash security violations), writing in the security information area (1st 9 words of the flash memory) is canceled. Moreover, a sector erase command to sector 0/sector 1 is ignored. If an instruction fetch is performed to the on-chip bus area, a reset request is issued by the flash security violation reset factor. Accesses to the flash memory are not accepted thereafter. The flash memory returns to the normal state by reset.
Other than aforementioned (Writer, etc.)	Access to flash memory is restricted. The data from reads is masked and 0xFFFF_FFFF is returned. Write commands and sector erase commands are ignored. Chip erase commands are accepted. See "36.5.9.3 Flash Security Unlocking Method."

Furthermore, while the security is on, when a data read is performed to the security information storage area (1st 9 words of the flash memory),

- A data access error will occur, and an illegal instruction exception or data access error interrupt will occur. (See "FR Family FR81 32-bit microcontroller programming manual" and "10.3.1 MPU Control Register: MPUCR" in "Chapter: CPU" for details.)
- 0xFFFFFFFF is returned as the read value.

However, when the OCD tool is connected, this restriction does not apply to access from OCDU or read during the debug state.

36.5.10 Notes on Using Flash Memory

Notes on using flash memory are shown below.

- If this device is reset during a write, the data that was written cannot be guaranteed.
- If CPU programming mode is set (FWE=1) using the FWE bit of the flash control register (FCTLR), do not execute the program in the flash memory. The program runs out of control without fetching the correct values.
- If CPU programming mode is set (FWE=1) using the FWE bit of the flash control register (FCTLR) and the interrupt vector table is in flash memory, do not generate interrupt requests. The program runs out of control without fetching the correct values.
- Because this model has the ECC bit added, data always needs to be written as 32-bit by 2 16-bit writes. See "36.5.2 Writing Flash Memory by CPU" for the procedure.
- Do not issue commands to multiple macros simultaneously (i.e., in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit.
- If authentication by password of on-chip debugger (OCD) completes, you can read the content of flash memory from external by using OCD even if security is on. When you want to prohibit reading by an outsider, password for on-chip debugger (OCD) activation approval must be configured.
- Changing to the state of the standby is prohibited during flash memory program/erase.
- Because of the build-in ECC in this flash memory, the data superscription to the address where some values have already been written cannot be done.

37. WorkFlash Memory

This chapter explains the WorkFlash memory.

37.1 Overview

37.2 Features

37.3 Configuration

37.4 Registers

37.5 Operation

37.1 Overview

This section explains the overview of the WorkFlash memory.

The size of the WorkFlash in this series is 64 Kbytes. Error correction codes (ECC) are attached.

37.2 Features

This section explains features of the WorkFlash memory.

- Usable capacity:
  MB91F552: 64 Kbytes (8 Kbytes x 8 sectors)
  For ECC code storage in this model, there are 6 bits of built-in WorkFlash memory in addition to the above for every 4 bytes.
- High-speed operation: Reading on a word-by -word basis (32 bit) is possible by 80 MHz x 2 cycle.
- Write from external:
  Possible from ROM writer
- Operation mode:

1. CPU-ROM mode
   (The CPU/DMA accesses WorkFlash memory. Only read)
   Only data access is enabled. Instruction fetch is not enabled.
2. CPU programming mode
   (The CPU accesses WorkFlash memory. Read/Write/Erase)
3. Flash memory mode
   (WorkFlash memory accessible from external)

- Security function
  - ☐ Operations after instruction fetch from external and write/erase except for chip erase are inhibited when security is on to prevent an outsider from reading out WorkFlash memory data.
  - ☐ The use of on-chip debugger (OCD) enables read from external by using OCD, even if security is on after password authentication.
- Error correction code (ECC) function
  - ☐ There is an error correction code (ECC) function that corrects errors of up to one bit in each word. (A function for detecting 2-bit errors is not provided.) Errors are automatically corrected during read. Furthermore, ECC codes are automatically added when writing to the flash memory. Because there is no read cycle penalty due to error correction, no consideration needs to be given to error correction penalties during software development.
  - ☐ An error is detected when data is read in the chip erase/sector erase state.

    If data in the erase state (FFFF) needs to be read correctly, be sure to first write "FFFF" before reading it..

37.3 Configuration

This section explains the configuration of the WorkFlash memory.

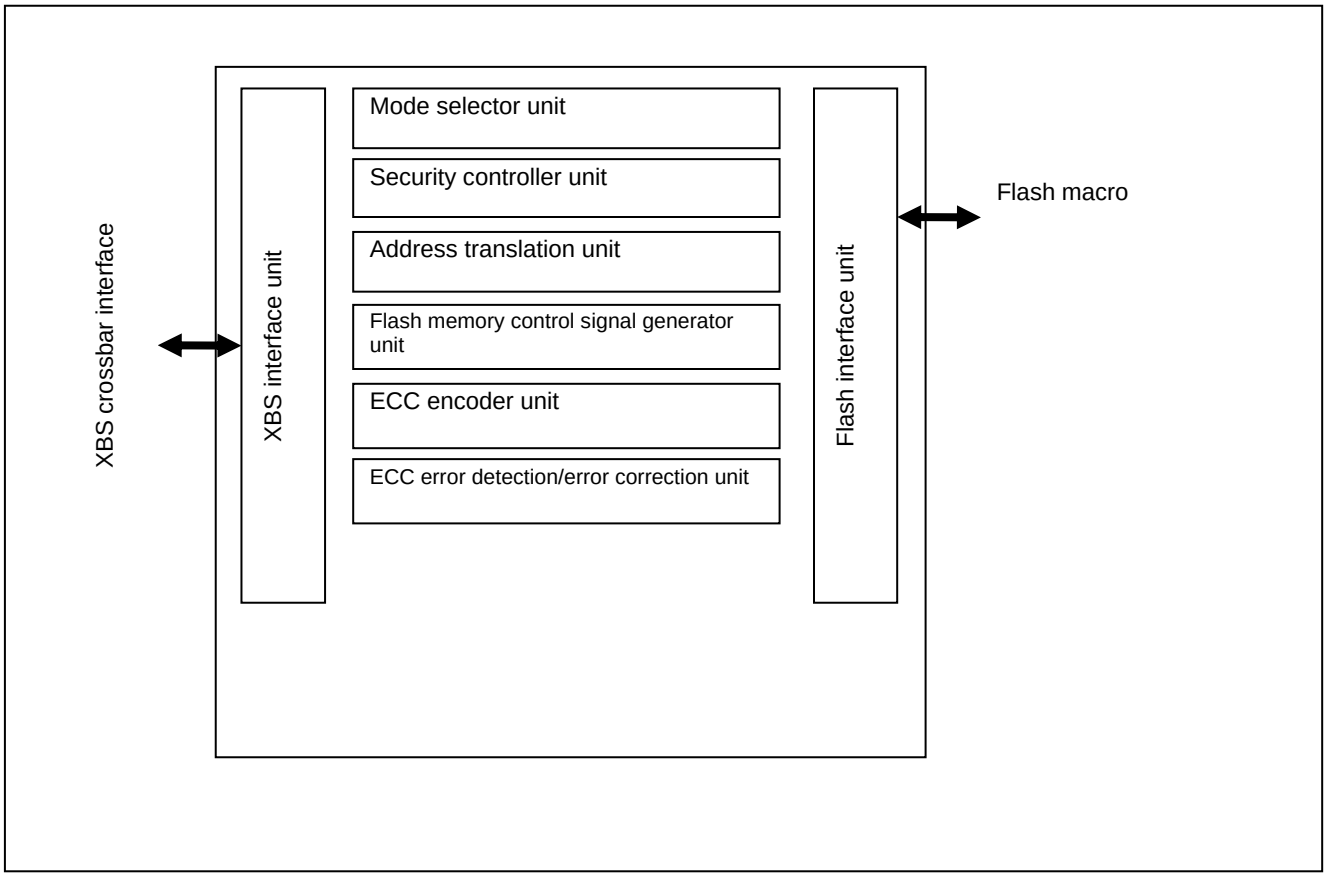
37.3.1 Block Diagram

37.3.2 Sector Configuration Diagram

37.3.1 Block Diagram

This section shows the block diagram of the WorkFlash memory.

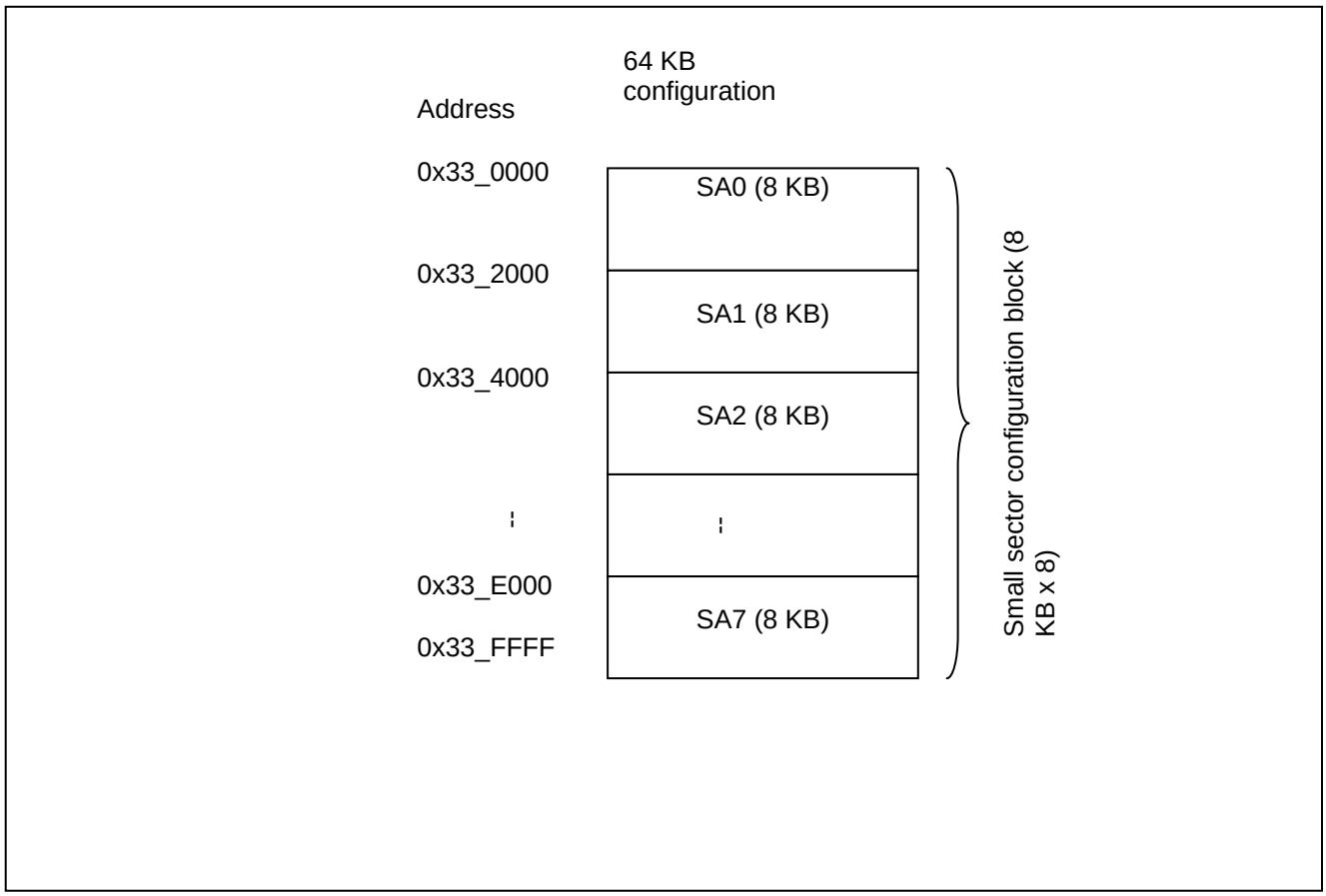
Figure 37-1. Block Diagram (64KB models)



37.3.2 Sector Configuration Diagram

The sector configuration diagram of the WorkFlash memory is shown below.

Figure 37-2. Sector Configuration Diagram



37.4 Registers

This section explains registers of the WorkFlash memory.

Table 37-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x2300	DFCTLR		Reserved	DFSTR	WorkFlash control register WorkFlash status register
0x2308	FLIFCTLR	Reserved	Reserved	Reserved	Flash interface control register

37.4.1 WorkFlash Control Register: DFCTLR (WorkFlash ConTroL Register)

The bit configuration of the WorkFlash control register is shown below.

This register configures the access control to the WorkFlash.

DFCTLR: Address 2300_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	FWE	Reserved					
Initial Value	-	0	-	-	-	-	-	-
Attribute	RX,WX	R/W	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	-	-	-	-	-	-	-	-
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX

[bit15] Reserved

This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit14] FWE (Flash Write Enable): Flash write enable

This bit is a control bit to enable write to the WorkFlash in the CPU mode.

If this bit is set, the ECC error detection and data correcting function will be disabled for data fetching to the WorkFlash memory.

FWE	Description
0	Disable writing to WorkFlash (initial value).
1	Enable writing to WorkFlash.

[bit13 to bit0] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation.

37.4.2 WorkFlash Status Register: DFSTR (WorkFlash STatus Register)

The bit configuration of the WorkFlash status register is shown below.

This register indicates the WorkFlash status.

DFSTR: Address 2303_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					DFECCERR	DFHANG	DFRDY
Initial Value	-	-	-	-	-	0	0	1
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	R/W	R,WX	R,WX

[bit7 to bit3] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] DFECCERR (WorkFlash ECC Error coRRection): Data read ECC correction occurred

This bit indicates that ECC error occurs when reading data of WorkFlash in the CPU mode. This bit is cleared by writing "0". Writing "0" is prioritized when ECC error and writing "0" occur concurrently.

DFECCERR	Read	Write
0	An error correction by ECC has not occurred during data read (initial value).	Clear this bit.
1	ECC error correction occurred during data read.	No effect

If there are errors in 2-bit or more in a single word, the read value of this bit is undefined.

[bit1] DFHANG (WorkFlash HANG): WorkFlash HANG status

This bit indicates the WorkFlash memory HANG status. If there is a timing overrun (See "[bit5] TLOV: (Timing limit exceeded flag bit)"), the flash memory will go into the HANG status. If this bit becomes "1", issue the reset command (See "37.5.3.1 Command Sequence").
The correct value might not be read out immediately after a command of automatic algorithm has been issued. Therefore, ignore the 1st read value of this bit after the command issuance.

DFHANG	Description
0	Normal state
1	HANGUP state

[bit0] DFRDY (WorkFlash ReaDY): WorkFlash write enable

This bit indicates whether the WorkFlash memory write/erase operation by automatic algorithm is currently running or finished. WorkFlash memory cannot be written or erased while the operation is in progress.

DFRDY	Description
0	During operation (write/erase disabled, read status enabled)
1	Completion of operation (write/erase enabled, read enabled)

The correct value might not be read out immediately after a command of automatic algorithm has been issued. Therefore, ignore the 1st read value of this bit after the command issuance.

37.4.3 Flash Interface Control Register: FLIFCTLR (Flash I/F Control Register)

The bit configuration of the flash interface control register is shown below.

This register controls the flash interface. This register is shared among program flash and WorkFlash.

FLIFCTLR: Address 2308_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			DFWDSBL	Reserved		ECCDSBL1	ECCDSBL0
Initial Value	-	-	-	0	-	0	0	0
Attribute	RX,WX	RX,WX	RX,WX	R/W	RX,WX	R/W0	R/W	R/W

[bit7 to bit5] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation.

[bit4] DFWDSBL (Data Fetch Wait cycle Disable): Data fetch wait cycle disabled

If this bit is set to "1", the wait cycle inserted when setting wait at data fetch is disabled. However, you cannot disable the wait cycle to guarantee the cycle time.

DFWDSBL	Description
0	Enable wait cycle (initial value).
1	Disable wait cycle.

Note:

When this bit is changed from "1" to "0", always set FCTLR.FAW="00" before changing this bit.

[bit3] Reserved

This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] Reserved

This bit is reserved. Always write "0" to this bit.

[bit1] ECCDSBL1 (ECC Disable1): ECC function disable 1

This bit configures enable/disable for the ECC function when write access and data fetch is performed to WorkFlash memory in the CPU mode.

ECCDSBL1	Description
0	Enable ECC function (initial value).
1	Disable ECC function.

[bit0] ECCDSBL0 (ECC Disable0): ECC function disable 0

This bit configures enable/disable for the ECC function when write access and data fetch is performed to program flash memory in the CPU mode.

ECCDSBL0	Description
0	Enable ECC function (initial value).
1	Disable ECC function.

37.5 Operation

The section explains the operation of the WorkFlash memory.

This section explains how to access the WorkFlash memory area.

37.5.1 Access Mode Setting
37.5.2 Writing Flash Memory by CPU
37.5.3 Automatic Algorithm
37.5.4 Reset Command
37.5.5 Write Command
37.5.6 Chip Erase Command
37.5.7 Sector Erase Command
37.5.8 Sector Erase Suspend Command
37.5.9 Security Function
37.5.10 Notes on Using WorkFlash Memory

37.5.1 Access Mode Setting

Access mode setting is shown below.

The WorkFlash memory in this model has the following 3 modes. Methods of setting the modes (1) and (2) are explained in this section. As for the mode (3), see the instruction manual of the ROM writer you are using for details.

1. CPU-ROM mode
   (The CPU/DMA accesses WorkFlash memory. For only read, Byte/Half-word/Word access)
2. CPU programming mode
   (The CPU accesses WorkFlash memory. For read/write/erase, only Half-word access)
3. Flash memory mode
   (WorkFlash memory accessible from external)

37.5.1.1 Configuring CPU-ROM Mode

Configuring CPU-ROM mode is shown below.

When the FWE bit of the WorkFlash control register (DFCTLR) is 0, it is CPU-ROM mode. In the CPU-ROM, when the DFRDY bit of the WorkFlash status register (DFSTR) is "1", read from WorkFlash memory is enabled. In the CPU-ROM mode, write to WorkFlash memory is disabled. After reset is released, the mode will be the CPU-ROM mode.

37.5.1.2 Configuring CPU Programming Mode

Configuring CPU programming mode is shown below.

When the FWE bit of the WorkFlash control register (DFCTLR) is 1, it is CPU programming mode. When the DFRDY bit of the WorkFlash status register (DFSTR) is "1", read/write from/to WorkFlash memory is enabled in this mode.

37.5.2 Writing Flash Memory by CPU

Writing the WorkFlash memory by the CPU is shown below.

After configuring CPU programming mode, perform erasing and programming using the automatic algorithm.
In this model, because error correction codes (ECC) are added to each single word, programming needs to be performed for each single word.
In the following procedure, each word is programmed by 2 operations to write 1 half-word. If this procedure is not followed, the written values will not be read correctly because the values will be written to the WorkFlash memory without calculating the ECC.

1. Set the flash write access size to 16 bits. (FCTLR.FSZ[1:0]=01),
   See "Chapter: Flash Memory" For FCTLR.
2. Issue the write command. Write address = PA, write data = PD[31:16]
   See "37.5.5 Write Command" for details on the write command.
3. Read the hardware sequence flag until the write has finished. See "37.5.3.2 Automatic Algorithm Execution State" for details on hardware sequence flag read.
4. Issue the write command. Write address = PA+2, write data = PD[15:0]
   At this time, the hardware automatically calculates the ECC codes by combining with PD[31:16] from (2), and writing of ECC codes is also performed automatically at the same time.
5. Read the hardware sequence flag until the write has finished.
6. If there is more data to write, return to (2). Continue to (7) when all writes have finished.
7. Set CPU-ROM mode.
8. Read the value which has already been written, and check that the correct value can be read. Even if the correct value can be read, check the DFSTR:DFECCERR bit to make sure that there was no ECC correction.
   If ECC correction occurs, follow the same procedure again, starting with erasing the flash memory.

PA: Write target address (word alignment)
PD[31:0]: Write data
PD[31:16]: Write data upper 16-bit
PD[15:0]: Write data lower 16-bit

37.5.3 Automatic Algorithm

The automatic algorithm is shown below.

When using CPU programming mode, write and erase of WorkFlash memory are performed by starting the automatic algorithm. This section explains the automatic algorithm.

37.5.3.1 Command Sequence

The command sequence is shown below.

The automatic algorithm starts when half-word (16-bit data) is written to WorkFlash memory once to 6 times in a row. This is called a command. The command sequences are shown below.

Table 37-2. Command Sequence

Command	Number of Writing	1st Time		2nd Time		3rd Time		4th Time		5th Time		6th Time	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Reset	1	arbitrary	$F0_H$										
Read	1	RA	RD										
Write	4	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	$A0_H$	PA	PD				
Chip Erase	6	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	80_H	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	10_H
Sector Erase	6	$AA8_H$	AA_H	554_H	55_H	$AA8_H$	80_H	$AA8_H$	AA_H	554_H	55_H	SA	30_H
Sector Erase Suspend	1	arbitrary	$B0_H$										
Sector Erase Resume	1	arbitrary	30_H										

- The data written in the table only shows the lower 8-bit. The upper 8-bit can be any value. The commands must be written as bytes or half-words.

- The addresses written in the table only show the lower 12-bit. Set the upper 20-bit to any address within the address range of the target flash macro.

PA: Write address (half-word alignment)
PD: Write data (Write as 16-bit.)
SA: Sector address (specify an arbitrary address within the address range of the sector to erase.)
RA: Read address
RD: Read data (the read width is arbitrary.)

Notes:

- When the wrong address value and data value are written or writing is performed in the wrong sequence, commands that have been written are cleared.
- Do the following to the LSB 2-bit for the command address, and for the sector address (SA) issued at the generation of the sector erase command.
  - When half-word access: 2'b00
  - When byte access: 2'b01 or 2'b11
  Example 1: During byte access, if command address = (LSB 2-bit of the standard command address changed to 2'b01.), $yAA8_H$ -> $yAA9_H$, $x554_H$ -> $x555_H$, SA -> {SA[31:2], 2'b01}
  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector.)
  Example 2: During byte access, if command address = (LSB 2-bit of the standard command address changed to 2'b11.), $yAA8_H$ -> $yAAB_H$, $x554_H$ -> $x557_H$, SA -> {SA[31:2], 2'b11}
  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector.)

Reset command

Sending the reset command to the target WorkFlash memory enables the sequential input of each command shown in Table 37-2. Command Sequence to be canceled, and commands can be input again from the 1st time.
However, when each command is input to the last minute and automatic algorithm starts, automatic algorithm cannot be discontinued by this reset command.
If the execution of the automatic algorithm exceeds the timing limit, the WorkFlash memory returns to the reset state if a reset command is issued.

Read command

The WorkFlash memory can be read by sending read commands to the target sector. If a read command is issued, the WorkFlash memory stays in read state until another command is issued.

Programming (write) command

If a write command is sent to the target sector 4 times in a row, the automatic algorithm starts and writes data to the WorkFlash memory. Programming (writing) of data can be performed in any order of addresses or across a sector boundary.
In the CPU programming mode, data is written in half-words. Once the 4th write has finished, the automatic algorithm starts and the automatic write to WorkFlash memory is started.
After executing the automatic write algorithm command sequence, there is no need to control the WorkFlash memory externally.

See "37.5.5 Write Command" for details on the actual operation.

Notes:

- When writing in half-word, if the 4th command (write data cycle) is written in the odd address, writing is not performed correctly. Always write in even address.
- In the 1st write command sequence, a single half-word data can be written. If you want to write multiple data, issue 1 write command sequence for each data.
- While security is on, writing of flash is limited. See "37.5.9.4 Flash Access Restrictions When Security Is ON" for details.

Chip erase command

If the chip erase command is sent to the target sector 6 times in a row, all sectors of the WorkFlash memory can be erased in one step. Once the 6th write has finished, the automatic algorithm starts and the chip erase operation is started.
When the automatic erase algorithm is started, "0" is written to all of the cells in the WorkFlash memory chip before erasing the entire chip, and there is no need to write to the WorkFlash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the WorkFlash memory externally.

See "37.5.6 Chip Erase Command" for details on the actual operation.

Sector erase command

If the sector erase command is sent to the target sector 6 times in a row, the sector of the WorkFlash memory can be erased. When 40 µs elapses (timeout period) after the 6th write has finished, the automatic algorithm starts and the sector erase operation is started.
If you want to erase multiple sectors, write the erase code (30_H) to the address of the sector to erase within the 40 µs (timeout period). If the next sector is not input within the timeout period, the sector erase command may become invalid.
When the automatic erase algorithm is started, "0" is written to the cells in the sector to erase in the WorkFlash memory before erasing the sector, and there is no need to write to the WorkFlash memory before erasing the sector to verify the margins (preprogramming).
Furthermore, while verifying the margin, there is no need to control the WorkFlash memory externally.

See "37.5.7 Sector Erase Command" for details on the actual operation.

Note:

While security is on, the sector erase procedure of flash is limited. See "37.5.9.4 Flash Access Restrictions When Security Is ON" for details.

Sector erase suspend command

It is possible to shift to the sector erase suspend condition (state of the sector erase suspension) by sending the sector erase suspend command in the command time-out or while executing the sector erase.
In the sector erase suspend condition, the reading operation of the memory cell of the sector that is not the erase target becomes possible. However, a new neither writing nor erase command is accepted.
To restart the interrupting erase operation from the sector erase suspend condition, the erase restart command is sent.

When the flash memory accepts the erase resume command, it goes back to sector erase state and starts erase operation again.
It does not change to the state of the command time-out when the erase resume command is normally written even if it is time when it changes from the state of the command time-out in this state, it changes to the state of the sector erase, and the sector erase operation is restarted at once.

See "37.5.8 Sector Erase Suspend Command" for details on the actual operation.

Note:

- 16.7µs + 2 cycles or less is required until the sector erase operation is stopped from the issue of the sector erase suspend command and reading from the sector that is not the erase target becomes possible.
- Whether it entered the state that can be read is confirmed with the DFRDY bit of the WorkFlash status register (DFSTR) or TOGG1 of the hardware sequence flag.

37.5.3.2 Automatic Algorithm Execution State

The automatic algorithm execution state is shown below.

Because writing and erasing WorkFlash memory is performed by an automatic algorithm, whether or not the automatic algorithm is executing can be checked by the DFRDY bit of the WorkFlash status register (DFSTR), and the operating state can be checked by the hardware sequence flag.

Hardware sequence flag

This flag indicates the state of the automatic algorithm. When the DFRDY bit of the WorkFlash status register (DFSTR) is "0", the operating state can be checked by reading from an arbitrary address in the WorkFlash memory. The bit configuration of the hardware sequence flag is shown below.

Figure 37-3. Bit Configuration of Hardware Sequence Flag

For half-word access

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

For byte access

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DPOLL	TOGG1	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

Notes:

- It is impossible to read by word access. Always read using half-word or byte access in CPU programming mode.
- In the CPU-ROM mode, the hardware sequence flag cannot be read no matter which address is read.

Each bit and WorkFlash memory status

The following table shows the correspondence between the status of each bit of the hardware sequence flag and the WorkFlash memory status.

Table 37-3. Correspondence between Flags and WorkFlash Memory Status

State		DPOLL	TOGG1	TLOV	SETI	TOGG2
Running	Writing	Inverted data*	Toggle	0	0	-
	Sector/Chip erasing	0	Toggle	0	1	-
Time limit exceed	Write command	Inverted data*	Toggle	1	0	-
	Sector erase/Chip erase command	0	Toggle	1	1	-
Sector erase suspend	Erase target sector	-	-	-	-	Toggle

*: See "Bit descriptions" for the values that are read out.

Bit descriptions

[bit15 to bit8] Undefined bits

[bit7] DPOLL (Data polling flag bit)

When the hardware sequence flag is read by specifying the write/erase target address, this bit indicates whether or not the automatic algorithm is running using a data polling function.
The value that is read differs depending on the operating state.

1. When writing

During execution of writing	Reads out the opposite value (inverted data) of the value of bit7 of the last data to be written. The address specified for reading the hardware sequence flag is not accessed.
After writing finished	Reads out the value of bit7 of the address specified for reading the hardware sequence flag.

2. During sector erase

During execution of sector erase	Reads "0" from the sector being erased.
After sector erase	This bit always reads out as "1".

3. During chip erase

During execution of chip erase	This bit always reads out as "0".
After chip erase	This bit always reads out as "1".

4. During sector erase suspend

State of suspend (incomplete end)	Reads "0" from the sector erase suspend sector.
Sector erase operation completion	Reads "1" from the sector erase suspend sector.

Note:

When the automatic algorithm is running, the data for the specified address cannot be read. Read data after using this bit to check whether the automatic algorithm operation has finished.

[bit6] TOGG1 (Toggle flag 1 bit)

When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the automatic algorithm is running.
The value that is read differs depending on the operating state.

During write/sector erase/chip erase

During write/sector erase/chip erase	If this bit is read sequentially, "1" and "0" are read alternatively (toggle operation). The address specified for reading the hardware sequence flag is not accessed.
After write/sector erase/chip erase	Reads out the value of bit6 of the address specified for reading the hardware sequence flag.

[bit5] TLOV: (Timing limit exceeded flag bit)

When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the execution time of the automatic algorithm exceeded the time specifying internally within the WorkFlash memory (number of internal pulses).

The value that is read differs depending on the operating state.

During write/sector erase/chip erase

One of the following values is read.

"0"	Within the rated time
"1"	Exceed rated time.

When this bit is "1", if the DPOLL bit and TOGG1 bit indicate that the automatic algorithm execution is in progress, the write or erase has failed.

For example, because data written to "0" cannot be rewritten with "1" in the WorkFlash memory, when an attempt is made to write "1" to an address already written with "0", the WorkFlash memory is locked and the automatic algorithm does not end. In this case, the value of the DPOLL bit remains invalid and the value read from the TOGG1 bit continues to alternate between "1" and "0". If the rated time is exceeded in this state, this bit changes to "1". If this bit becomes "1", issue a reset command.

Note:

When this bit is "1", this indicates that the WorkFlash memory was not used correctly. The WorkFlash memory is not faulty. Perform the appropriate processing after issuing the reset command.

[bit4] Undefined bit

[bit3] SETI (Sector erase timer flag bit)

During sector erase, a timeout period of 40 µs is required from when the sector erase command is issued until the sector erase actually starts. When the hardware sequence flag is read by specifying an arbitrary address, this bit indicates whether or not the sector erase command is within the timeout period.
The value that is read differs depending on the operating state.

When erasing sectors

When erasing sectors, you can check whether the next sector erase code is ready to be accepted by checking this bit before inputting the next sector erase code. One of the following values is read out without accessing the address specified for reading the hardware sequence flag.

"0"	Within sector erase wait period (the next sector erase code (30_H) can be accepted.)
"1"	Exceed the sector erase wait period. (If the DPOLL bit and TOGG1 bit indicate that the automatic algorithm is executing at this time, the WorkFlash memory internal erase is started. In this case, commands other than the sector erase code (0x30) are ignored until the WorkFlash memory internal erase finishes.)

[bit2] TOGG2 (Toggle flag 2 bit)

In the sector erase suspend state, non target sector for erase can be read, but target sector for erase cannot be read. This flag indicates that output data is toggled and target sector for erase when read address is the target sector for erase during sector erase suspend.

Read out target erase sector	If this bit is read sequentially, "1" and "0" are read alternatively (toggle operation). The address specified for reading the hardware sequence flag is not accessed.
Read out non target erase sector	Read data from specified address.

[bit1, bit0] Undefined bits

37.5.4 Reset Command

The reset command is shown below.

The WorkFlash memory can be reset by sending reset commands to the target WorkFlash memory. Because this state is the WorkFlash memory initial state, the WorkFlash memory always returns to the reset state when the power is turned on or a command finishes successfully.
When the power is turned on, there is no need to issue a reset command.
Furthermore, in reset state, data can be read using normal read access and programs can be accessed by the CPU; thus there is no need to issue the reset command when reading data.

37.5.5 Write Command

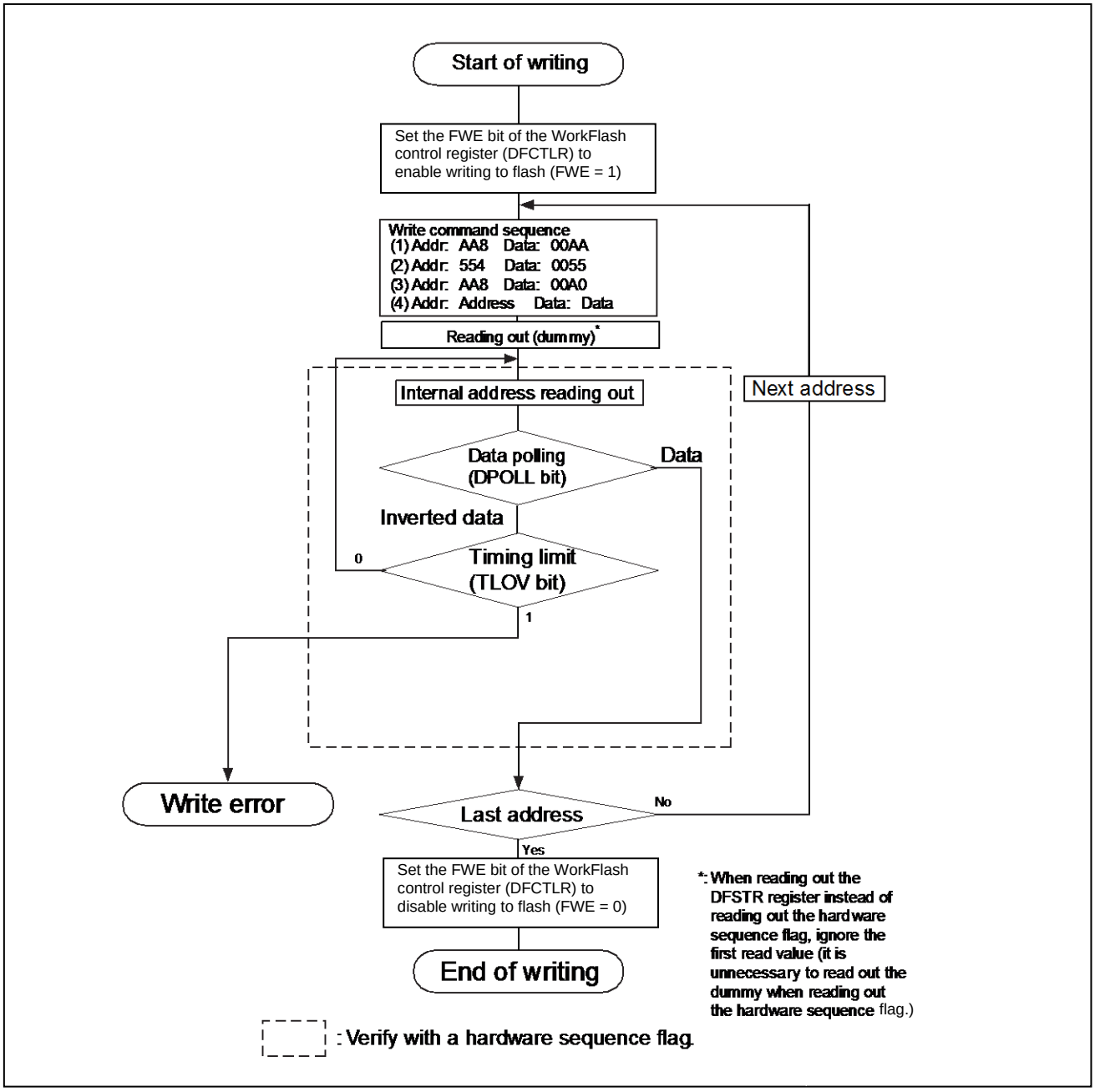
The write command is shown below.

Writes are performed in the following order.

1. Send write commands sequentially to the target sector.
   The automatic algorithm is started and data is written to the WorkFlash memory.
   After issuing a write command, there is no need to control the WorkFlash memory externally.
2. Perform read access to the written address.
   The read data is the hardware sequence flag. Therefore, if bit7 (DPOLL bit) of the read data matches the written data, the write to the WorkFlash memory has finished.
   If the write has not finished, the opposite value (inverted data) of the bit7's value of the last written data is read out.

The following figure shows an example of write operation to the WorkFlash memory.

Figure 37-4. Example of Write Procedure



Notes:

- Once the write has finished, the write address is not accepted because the WorkFlash memory returns to read mode.
- For details on the write command, see Section "37.5.3 Automatic Algorithm" in "Chapter: WorkFlash Memory."
- Because the value of the DPOLL bit of the hardware sequence flag changes concurrently with the TLOV bit, check this bit again even if the value of the TLOV bit is "1".
- The moment when the TOGG1 bit of the hardware sequence flag and TLOV bit change to "1", the toggle operation stops. Therefore, even if the TLOV bit is "1", checking the TOGG1 bit again must be needed.
- Although the WorkFlash memory can be written to in any order of addresses, even if it crosses a sector boundary, only a single half-word data can be written in each write command sequence. If you want to write multiple data, issue 1 write command sequence for each data.
- Data that has been written to "0" once cannot be returned to "1". If "0" is rewritten with "1", one of the following occurs.

  - The element is judged as faulty by the data polling algorithm.
  - The write rated time is exceeded, and the TLOV bit of the hardware sequence flag changes to "1".
  - It appears to have been written as "1".

- However, even if it appears to have been written as "1", the actual data remains "0" and "0" will be read out when the data is read in read/reset mode. If you want to return data to "1", perform a chip erase or sector erase.
- During write operations, all commands written to the WorkFlash memory are ignored.
- If this device is reset during a write, the data that was written cannot be guaranteed.
- Because this model has the ECC bit added, data always needs to be written as 32-bit by 2 16-bit writes. For the procedure, see Section "37.5.2 Writing Flash Memory by CPU" in "Chapter: WorkFlash Memory."

37.5.6 Chip Erase Command

The chip erase command is shown below.

The erase target flash macros in the WorkFlash memory can be erased in one step using the chip erase command.

If the chip erase command is sent to the target sector sequentially, the automatic algorithm starts and all sectors of the WorkFlash memory can be erased in one step. For details on the chip erase command, see Section "37.5.3 Automatic Algorithm" in "Chapter: WorkFlash Memory."

Chip erase is performed in the following order.

1. Send chip erase commands sequentially to a sector in the WorkFlash macro to erase.
   The automatic algorithm is started and data is written to the WorkFlash memory.
2. Perform a read access to an arbitrary address in the flash macro to erase.
   The read data is the hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", the chip erase has finished.

The time required to erase the chip is [sector erasure time x total no. of sectors + chip write time (preprogram)]. When the chip erase operation finishes, the WorkFlash memory returns to the read/reset state.

Notes:

- When the automatic erase algorithm is started, "0" is written to all of the cells in the WorkFlash memory chip before erasing the entire chip, and there is no need to write to the WorkFlash memory before the chip erase to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the WorkFlash memory externally.
- When security is on, there are restrictions in the procedure for erasing the flash. For details, see Section "37.5.9.3 Flash Security Unlocking Method" in Chapter: WorkFlash Memory."

37.5.7 Sector Erase Command

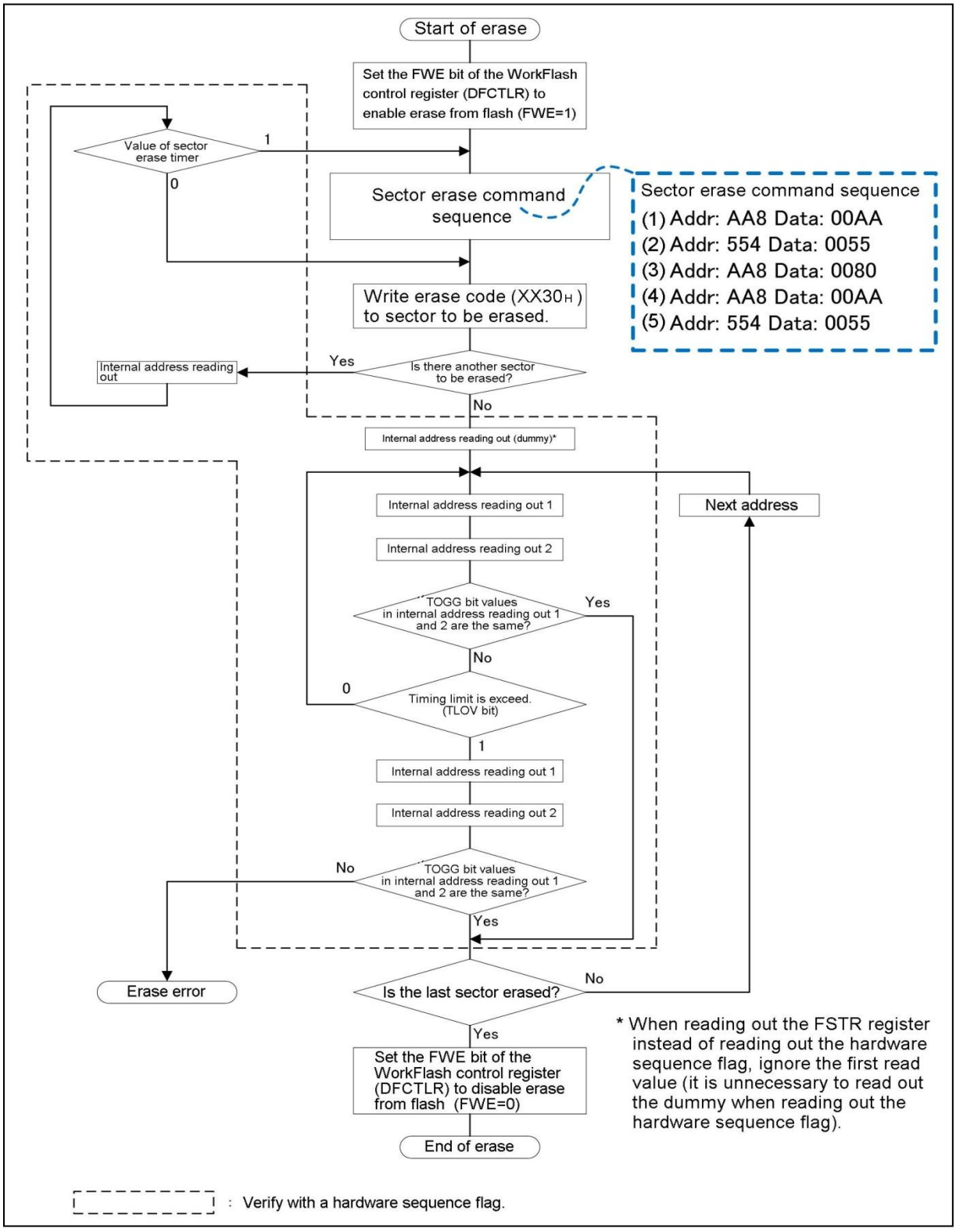
The sector erase command is shown below.

A sector in the WorkFlash memory can be selected and only data in the selected sector can be deleted. Multiple sectors can also be specified at the same time.
Sector erase is performed in the following order.

1. Send sector erase commands sequentially to the target sector.
   Once 40 µs has elapsed (timeout period), the automatic algorithm starts and the sector erase operation is started. If you want to erase multiple sectors, write the erase code (30_H) to the address of the sector to erase within the 40 µs (timeout period). If the write is performed after the timeout period has elapsed, the sector erase command may be invalid.
2. Perform read access to an arbitrary address.
   The read data is the hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", the sector erase has finished. Furthermore, you can use the TOGG1 bit to check whether or not the sector erase has finished.

The following shows an example of the sector erase procedure taking the example of using the TOGG1 bit in the check operation.

Figure 37-5. Example of Sector Erase Procedure



Notes:

- The time required to erase the sector is [(sector erasure time + sector write time (preprogram)) x no. of sectors].
- When the sector erase operation finishes, the WorkFlash memory returns to the read/reset state.
- For details on the sector erase command, see Section "37.5.3 Automatic Algorithm" in "Chapter: WorkFlash Memory."
- Because the value of the DPOLL bit of the hardware sequence flag changes concurrently with the TLOV bit, check this bit again even if the value of the TLOV bit is "1".
- When the TOGG1 bit and the TLOV bit of the hardware sequence flag change to "1", toggle operation stops at the same time. Therefore, even if the TLOV bit is "1", checking the TOGG1 bit again must be needed.
- In the command timeout state, the only accepted write commands are the erase sector add command and erase suspend command. In the sector erase state, the only accepted command is the erase suspend command.
- When the automatic erase algorithm is started, "0" is written to the cells to erase in the WorkFlash memory before erasing the sector, and there is no need to write to the WorkFlash memory before erasing the sector to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the WorkFlash memory externally.

37.5.8 Sector Erase Suspend Command

The sector erase suspend command is shown below.

The sector erase can temporarily be stopped in the command time-out or while executing the sector erase.
In the sector erase suspend condition, the reading operation of the memory cell of the sector that is not the erase target becomes possible. However, a new neither writing nor erase command is accepted.
The sector erase suspend command is sent to an arbitrary address of target FLASH macro to suspend the sector erase.
After the sector erase stops, the reading operation from target FLASH macro is permitted. At this time, the hardware sequence flag is read from the sector which is under the sector erase suspend condition.
It enters the following states when entering the sector erase suspend condition.

- The TOGG1 bit that does the toggle while erasing the sector does not toggle while being suspended from the sector erase operation.
- DFRDY of the WorkFlash status register becomes "1".

The thing that entered the sector erase suspend condition can be confirmed by using these.

Note:

The maximum cycles required from the issuing of the sector erase suspend command until the sectors that are not erase targets can be read after the sector erase operation stops is 16.7 µs + 2 cycles.

Because bit2:TOGG2 of the hardware sequence flag does the toggle while the sector erase is temporarily stopping, the sector can be confirmed while stopping by using this bit.
To restart the interrupting erase operation from the sector erase suspend condition, the sector erase restart command in Table 37-2. Command Sequence is sent.
The sector erase restart command is accepted only by the sector erase suspend condition.
Send the command after confirming becoming the sector erase suspend condition.
When the flash memory accepts the erase resume command, it goes back to sector erase state and starts erase operation again.

37.5.9 Security Function

The security function is shown below.

This WorkFlash memory is equipped with a security function.

When the security function is off, the flash memory can be used without limits. However, when the security function is on, writes and erases other than chip erase are suppressed. See "37.5.9.4 Flash Access Restrictions When Security Is ON" for details of the restrictions.

37.5.9.1 Flash Security On/Off Determination When Reset Released

Flash security on/off determination when reset released is shown below.

The flash interface of this model reads 2 bytes from the flash security code area after reset is released. If this value is 0x0001, security is turned on and access restrictions are imposed on subsequent accesses to WorkFlash memory. For any other value, the security is turned off.

37.5.9.2 Flash Security Setting Method

The flash security setting method is shown below.

When reset is input and released after writing 0x0001 to the flash security code area (see Figure 37-2. Sector Configuration Diagram in "Chapter: WorkFlash Memory"), security is turned on. Once security has been turned on, the security is not turned off unless the entire flash memory area is erased.

37.5.9.3 Flash Security Unlocking Method

The flash security unlocking method is shown below.

The chip erase command can be performed on all flash macros in the following order.

1. Erase WorkFlash.
2. Erase program flash which contains the flash security code.

Erase program flash last, as shown above. Otherwise, the erase command to program flash is ignored. Furthermore, if a reset is input between erases, repeat from step (1).

Note:

In the user-mode (internal FLASH activation), the erase command can be issued to an arbitrary flash macro, and data in the WorkFlash macro can be deleted. The recommended order of the chip erase to each WorkFlash macro is as shown above, from the viewpoint of the data protection stored in the WorkFlash macro.

37.5.9.4 Flash Access Restrictions When Security Is ON

Flash access restrictions when security is on are shown below.

When security is on, the restrictions shown below are created by the start mode.

Table 37-4. Access Restrictions when Security Is ON

Operating Mode	Access Restriction
User	In normal mode (the state where there are no access restrictions due to the following WorkFlash security violations), there are no restrictions on access to WorkFlash memory. If an instruction fetch is performed to the on-chip bus area, a reset request is issued by the flash security violation reset factor. Accesses to the WorkFlash memory are not accepted thereafter. The WorkFlash memory returns to normal state by reset.
Other than aforementioned (Writer, etc.)	Access to WorkFlash memory is restricted. The data from reads is masked and 0xFFFF_FFFF is returned. Write commands and sector erase commands are ignored. Chip erase commands are accepted. See "37.5.9.3 Flash Security Unlocking Method."

37.5.10 Notes on Using WorkFlash Memory

Notes on using the WorkFlash memory are shown below.

- If this device is reset during a write, the data that was written cannot be guaranteed.
- CPU programming mode is set (FWE=1) using the FWE bit of the WorkFlash control register (DFCTLR), do not execute the program in the WorkFlash memory. The program runs out of control without fetching the correct values.
- When CPU programming mode is set (FWE=1) using the FWE bit of the WorkFlash control register (DFCTLR) and the interrupt vector table is in the WorkFlash memory, do not generate interrupt requests. The program runs out of control without fetching the correct values.
- Because this model has the ECC bit added, data always needs to be written as 32-bit by 2 16-bit writes. For the procedure, see Section "37.5.2 Writing Flash Memory by CPU" in "Chapter: WorkFlash Memory."
- Do not issue commands to multiple macros simultaneously (i.e., in parallel). Input a command to the next macro after confirming that the command has completed using either the hardware sequence flag or FRDY bit.
- If authentication by password of on-chip debugger (OCD) completes, you can read the content of WorkFlash memory from external by using OCD even if security is on. When you want to prohibit reading by an outsider, password for on-chip debugger (OCD) activation approval must be configured.
- Changing to the state of the standby is prohibited during WorkFlash memory program/erase.
- Because of the build-in ECC in this WorkFlash memory, the data superscription to the address where some values have already been written cannot be done.

38. On Chip Debugger (OCD)

This chapter explains the on chip debugger (OCD).

38.1 Overview

38.2 Features

38.3 Configuration

38.4 Registers

38.5 Operation

38.1 Overview

This section explains the overview of the on chip debugger (OCD).

This chapter explains an overview of the on chip debugger (OCD) in this series and the related specification restrictions.

OCDU is the device built-in debug support unit that provides the on chip debug function in FR81. OCDU provides the basic debugger functions (CPU execution/break control, CPU register/memory/IO access), small-scale debug support functions (event, execution time measurement, trace, etc.), and security function.

38.2 Features

This section explains features of the on chip debugger (OCD).

One-wire debug tool I/F

- Initial communication type: Manchester code
- MDI wake up and standby function
- Auto negotiation mode function
- The clock calibration is unnecessary on the tool side.

Debug security function

Debug mode control function

Execution control function

- Status display functions (chip status, CPU status, etc.)
- Debug command execution control function
- Small-scale debug main memory (8 bytes=4 instructions)
- CPU register save register (PC/PS)
- PC monitor function
- Reset function
- Chip reset (INT)
- CPU reset (RST)

Semi-hosting function

Break function

- Step execution break
- Event trigger break
- Forced break
- Guarded access break
- Trace end break
- Control on interrupt acceptance immediately after the execution start address

Debug DMA function (DDMA function)

- Support of transfer modes (address mode, verify mode, DEBUG I/F burst transfer)

Event function

- Code event: 8
- Conditional code event: 2
- Data event: 8
- Interrupt event: 2
- User event: 2
- Event sequencer: 2 levels + reset

Execution time measurement timer function

- Go-Break measurement
- Inter-trigger measurement (single measurement/cumulative measurement, MIN/MAX value measurement when accumulation is measured)

Trace function

- Special state trace
- Branch trace
- Data trace
- Trace delay
- Number of trace frames: 512

38.3 Configuration

This section shows the configuration of the on chip debugger (OCD).

Figure 38-1. Block Diagram of OCDU

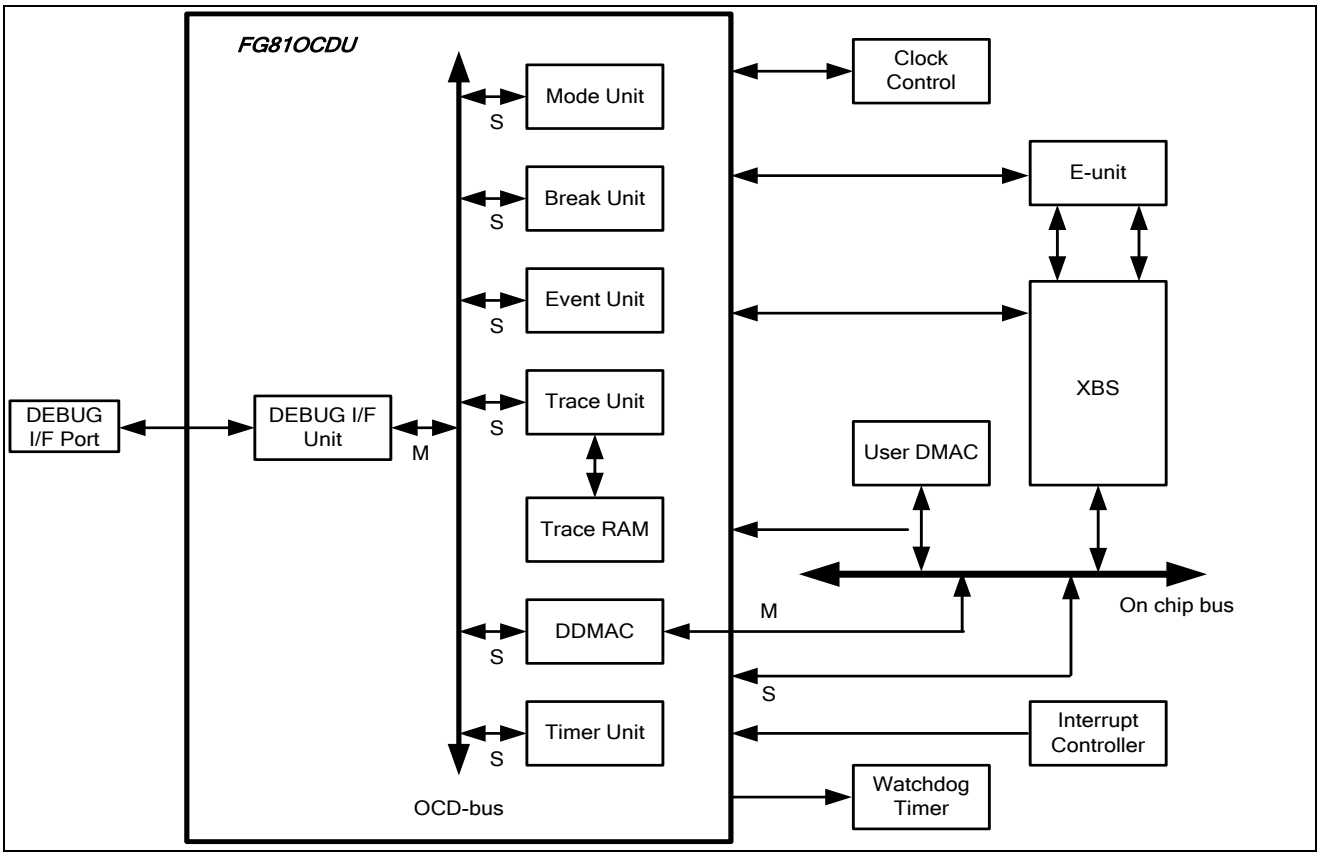
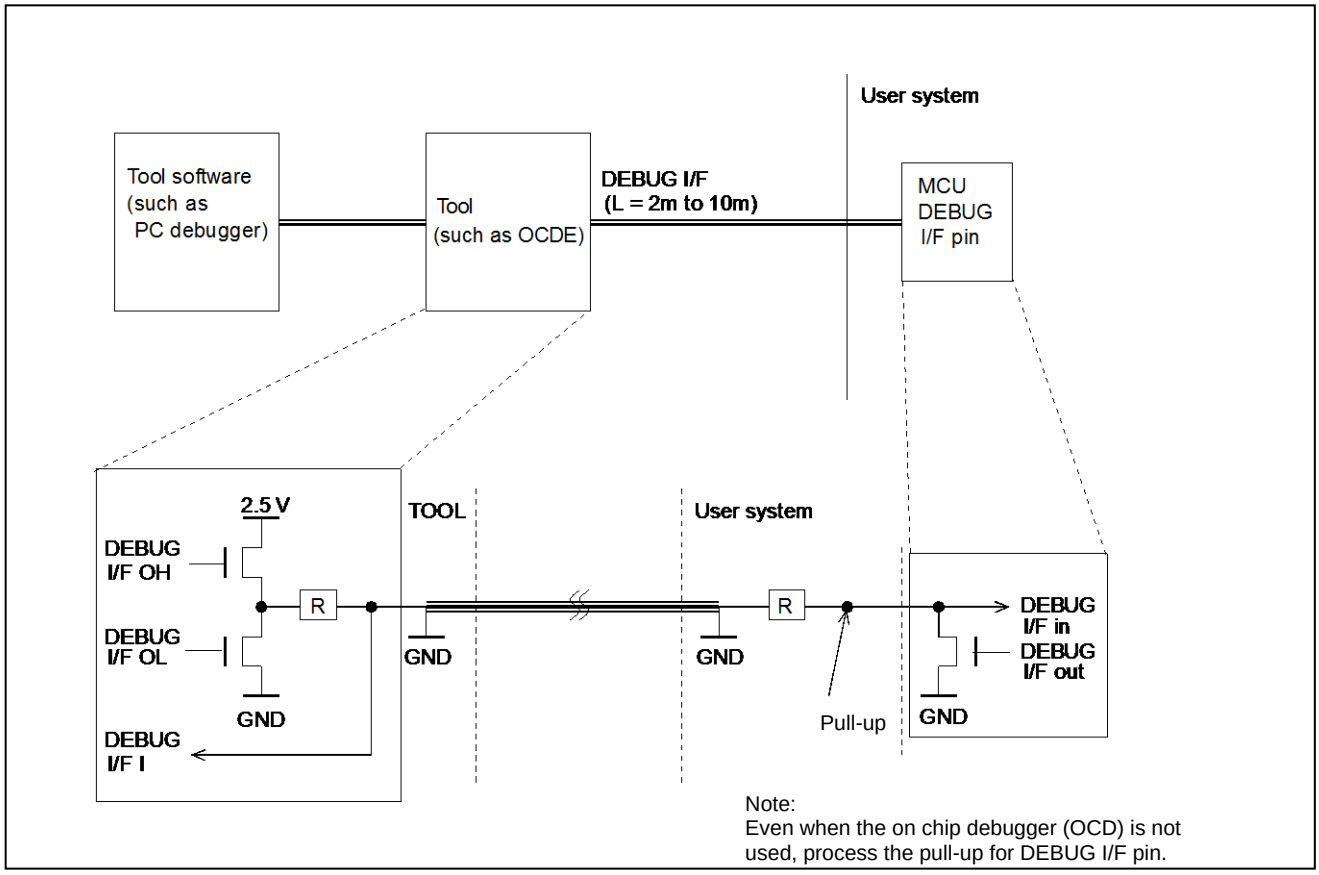


Figure 38-2. OCD Connection Diagram



38.3.1 DEBUG I/F Clock

DEBUG I/F clock is shown.

See "Chapter: Clock" for the clock connection configuration of the DEBUG I/F clock.

38.3.1.1 DEBUG I/F Main Clock: M_MCLK

DEBUG I/F main clock (M_MCLK) is shown.

When OCD tool is connected, the main clock (MCLK) is supplied for DEBUG I/F main clock (M_MCLK).
When OCD tool is not connected, DEBUG I/F main clock (M_MCLK) stops.

38.3.1.2 DEBUG I/F PLL Clock: M_PCLK

DEBUG I/F PLL clock (M_PCLK) is shown.

When the OCD tool is connected and the high-speed UART mode or phase modulation UART mode is selected, the PLL clock (PLLCLK) is supplied for DEBUG I/F PLL clock (M_PCLK).
When the OCD tool is not connected, DEBUG I/F PLL clock (M_PCLK) stops.

38.4 Registers

This section explains the registers of the on chip debugger (OCD).

38.4.1 DBG Register

The DBG register is shown.

Table 38-1. Register Map (DBG Register)

Address	Register				Register Function
	+0	+1	+2	+3	
0xFF00	DSUCR		Reserved		DSU control register

38.4.1.1 DSU Control Register: DSUCR

The bit configuration of DSU control register is shown below.

This register is used to control DSU in the free-run mode.
For details, contact our sales representative.

DSUCR: Address $FF00_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							DSU
Initial value	X	X	X	X	X	X	X	0
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R,W

38.4.2 User IO Register

The User IO register is shown below.

Table 38-2. Register Map (User IO Register)

Address	Register				Register Function
	+0	+1	+2	+3	
0x0BF0	HSCFR				High-speed communication frequency register
0x0BF8	Reserved		MBR		Message buffer
0x0BFC	Reserved		UER		User event register

38.4.2.1 User Event Register: UER

The bit configuration of user event register is shown.

This register is used to detect a user event.
For details, contact our representative.

UER: Address $0BFE_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							UEVT
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W

38.4.2.2 High-Speed Communication Frequency Register: HSCFR

The bit configuration of high-speed communication frequency register is shown.

This register is used to set frequency information on PLL clock used.
For details, contact their representatives.

HSCFR: Address 0BF0$_H$ (Access: Byte, Half-word, Word)

	bit 31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved						FREQ[17:16]	
Initial value	X	X	X	X	X	X	0	0
Attribute	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	FREQ[15:8]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FREQ[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

38.4.2.3 Message Buffer: MBR

The bit configuration of message buffer is shown.

This register is used to control writing and requesting of the message for a semi-hosting.

MBR: Address 0BFA$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BUSY	DMAREQ	Reserved					
Initial value	0	0	X	X	X	X	X	X
Attribute	R,WX	R,W1	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0	RX,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MB[7:0]							
Initial value	X	X	X	X	X	X	X	X
Attribute	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W

[bit15] BUSY (Message Buffer Busy): Message buffer busy

This bit detects the busy state of the message buffer. It becomes "1" by writing in MBR.MB[7:0] bits, and it becomes "0" by the message reading from the tool side. This bit is for reading only, and writing doesn't have the influence in operation.

BUSY	Message Buffer Busy State
0	Non-busy state (initial value)
1	Busy state

[bit14] DMAREQ (DDMA Message Handling Request): DDMA message processing request bit

This bit requests the method of message handling that uses DDMA. Only "1" writing of this bit is effective, and "0" writing doesn't have the influence in operation.

DMAREQ	DDMA Message Processing
0	Do not request (initial value).
1	Request

[bit7 to bit0] MB[7:0] (Message Buffer): Message buffer

These bits are for writing message data of 1 byte. These bits are only for writing. The reading value is undefined.

38.5 Operation

This section explains the operation of the on chip debugger (OCD).

38.5.1 OCDU Operating Mode

38.5.2 Overview of DEBUG I/F

38.5.3 Specification Restrictions at Connection to OCD Tool of This Series

38.5.4 OCD-DSU ID Code and Mount Type Information on This Series

38.5.1 OCDU Operating Mode

OCDU operating mode is shown.

38.5.1.1 Operating Mode

Operating mode is shown.

The OCDU operating mode includes emulator mode and free-run mode.

Emulator mode (debug running status)

The emulator mode consists of the debug state for executing the debug instruction and the user state for executing a user program. If the RETI instruction is executed in the debug state, control transits to the user state. If a break occurs in the user state, control transits to the debug state.

Free-run mode (normal running status)

Mode in which only the user program runs

38.5.1.2 Operating Mode Status Transition

Operating mode status transition is shown.

At INIT releasing (including RST accompanied by INIT), control transits to the debug state of the emulator mode or to the free-run mode according to the mode command from DEBUG I/F in the chip reset sequence.

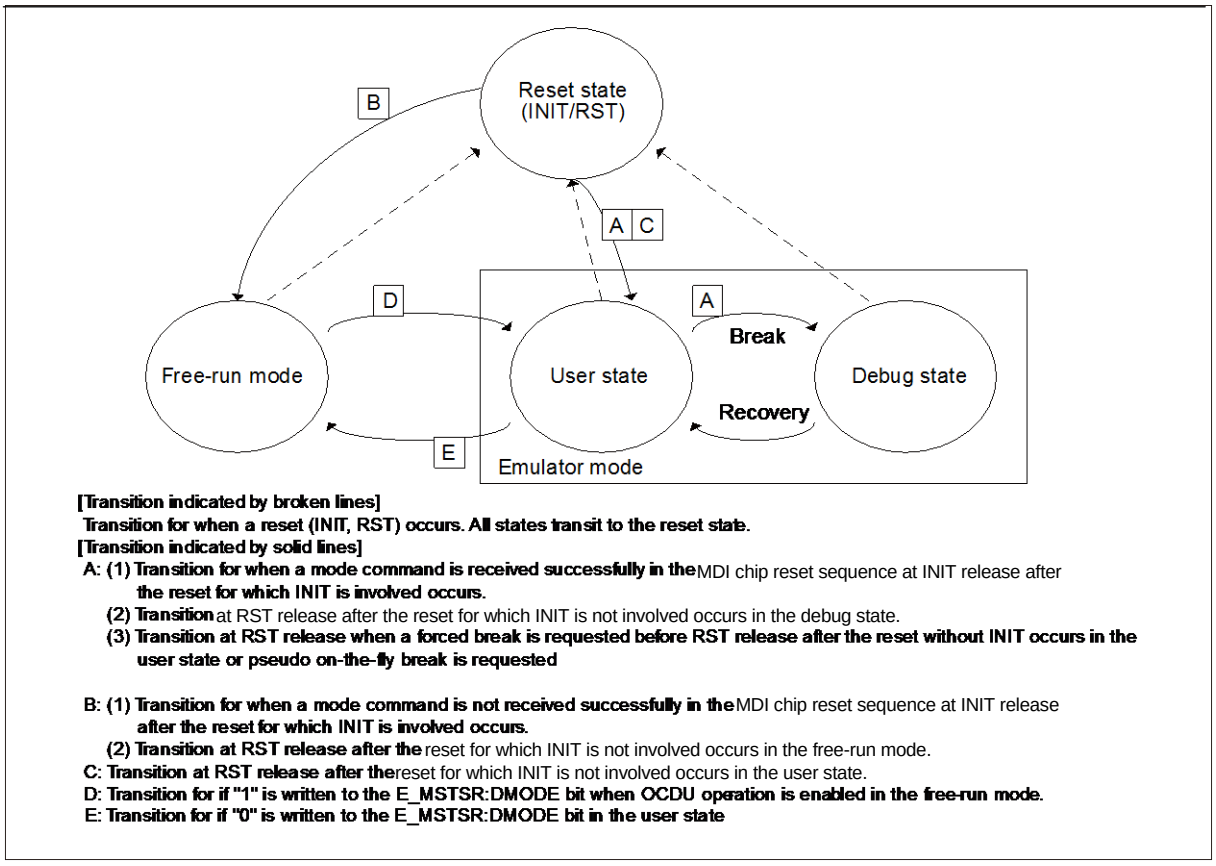
At RST releasing (not accompanied by INIT), control transits to the operating mode before RST generation. However, if a forced break request is issued after RST occurs in the user state, control transits to the debug state of the emulator mode at RST releasing.

Moreover, transition between the free-run mode and user state of emulator mode is enabled by OCD register control.

At transition from the reset status to the debug state, control first transits to the user state. In this case, requesting a break by OCDU makes the following transition: reset status -> user state -> (break) -> debug state.

The transition conditions are shown below.

Figure 38-3. OCDU Operating Mode Transition Diagram



[Transition indicated by broken lines]
Transition for when a reset (INIT, RST) occurs. All states transit to the reset state.
[Transition indicated by solid lines]
A: (1) Transition for when a mode command is received successfully in the MDI chip reset sequence at INIT release after **the reset for which INIT is involved occurs.**
(2) Transition at RST release after the reset for which INIT is not involved occurs in the debug state.
(3) Transition at RST release when a forced break is requested before RST release after the reset without INIT occurs in the user state or pseudo on-the-fly break is requested

B: (1) Transition for when a mode command is not received successfully in the MDI chip reset sequence at INIT release **after the reset for which INIT is involved occurs.**
(2) Transition at RST release after the reset for which INIT is not involved occurs in the free-run mode.
C: Transition at RST release after the reset for which INIT is not involved occurs in the user state.
D: Transition for if "1" is written to the E_MSTSR:DMODE bit when OCDU operation is enabled in the free-run mode.
E: Transition for if "0" is written to the E_MSTSR:DMODE bit in the user state

38.5.2 Overview of DEBUG I/F

The overview of DEBUG I/F is shown.

DEBUG I/F is a single-wire debug interface that connects MCU to a tool via one wire (+GND). MCU uses 1 pin as the one for the debug interface.

DEBUG I/F is a two-way pin and provides the communication function and special sequence function.

Communication uses the serial transmission method (UART). In the normal UART mode, the communication baud rate is obtained by division clocks that are based on the main source oscillation clock of MCU. In the high-speed UART mode and in phase modulation UART (Manchester encode UART), the division clock is based on the PLL clock.

The special sequence includes chip reset sequence and stall. There are the function that MCU notifies the INIT generation and the function to detect the debug mode that activated after releasing INIT in the chip reset sequence. The stall function provides communication stall and forced break requests from the tool, and communication error notification from MCU.

The main DEBUG I/F functions are shown below.

- Chip reset sequence function (INIT notification, mode command)
- UART function (normal UART, high-speed UART, phase modulation UART)
- Stall request (communication stall request, forced break request, communication error notification)
- Auto negotiation mode function (communication form notification)

The two-way pin of DEBUG I/F is accomplished by Nch open-drain output. The DEBUG I/F pin is pulled up on a user system. It is pulled up with a tool during tool connection.

For the tool connection, see "Figure 38-2. OCD Connection Diagram."

38.5.2.1 Chip Reset Sequence

Chip reset sequence is shown.

When INIT is generated, OCDU executes the chip reset sequence according to the specification of DEBUG I/F. A reference clock that executes the chip reset sequence is a sampling clock of the normal UART (8 division clock of the main source oscillation clock).

The chip reset sequence consists of the following 5 phases:

- Start phase
- INIT phase
- Level sense phase
- Mode entry phase
- End phase

Start phase

Start phase is the interval when the generated INIT is released until 32 sampling clock cycles of the normal UART is counted. OCDU does not perform the special operation in this phase.

INIT notification phase

INIT notification phase is the interval when the start phase is ended until 568 sampling clock cycles of the normal UART is counted. OCDU outputs "L" of 280 cycles to DEBUG I/F twice while it is in this phase (The idle of 8 cycles is inserted among), and notifies the tool the generation of INIT.

Level sense phase

Level sense phase is the interval when the INIT notification phase is ended until 256 sampling clock cycles of the normal UART is counted. OCDU does not perform the special operation in this phase.

Mode entry phase

Mode entry phase is the interval when the Idle phase is ended until 256 sampling clock cycles of the normal UART is counted. OCDU starts the reception of the mode command from the tool in this phase.

When starting reception of the mode command is detected (start bit detected in the UART reception) in this phase, OCDU activates in the emulator mode (debug state). Then, if the normal mode command (no reception error and mode command match) is received, OCDU can receive the subsequent register access command after this. If the normal mode command (reception error and no mode command match) is not received, OCDU generates INIT request and executes the chip reset sequence again after INIT is released.

When starting reception of the mode command is not detected (start bit detected in the UART reception) in this phase, OCDU activates in the free-run mode.

If the mode command is received immediately after starting the mode entry phase, the mode command must be received after waiting for input of "H" to MDI for 1 or more cycles width using the UART reception sampling clock. If this condition is not met, the start bit of the mode command reception cannot be detected normally, the mode may not be entered correctly.

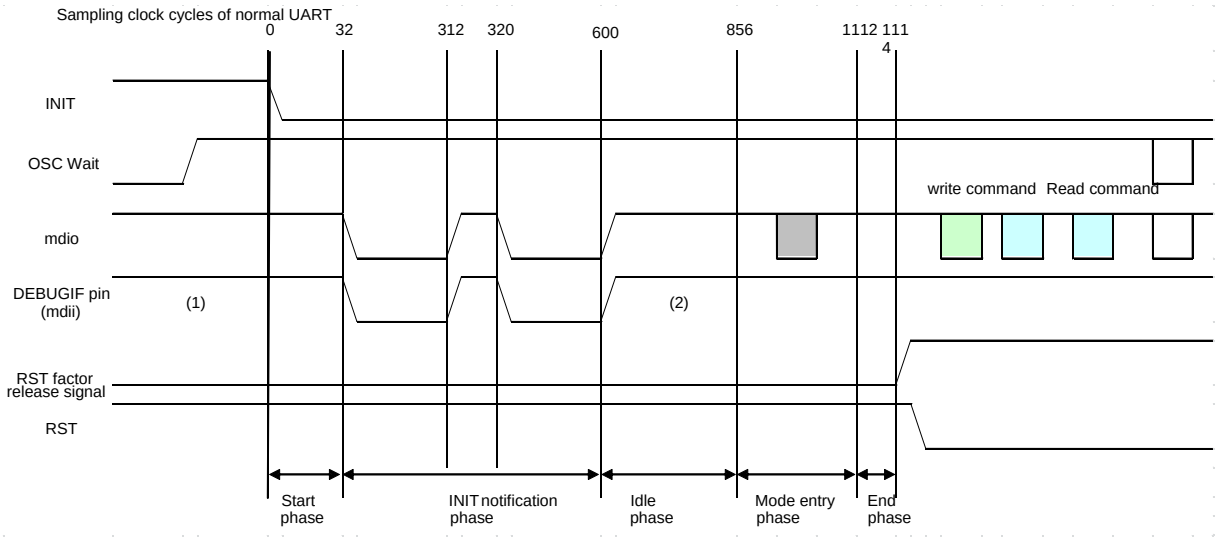
End phase

End phase is the interval when the mode entry phase is ended until 2 sampling clock cycles of the normal UART is counted. OCDU does not perform the special operation in this phase. OCDU executes the reset issuance sequence described in "5.4.3 Reset (RST)" of "Chapter: Reset" when the end phase is ended. The RST factor is released.

The relationship between the number of sampling clock cycles of the normal UART and the phase for the chip reset sequence is as follows.

Phase of Chip Reset sequence	Start Phase	INIT Notification Phase	Idle Phase	Mode Entry Phase	End Phase
Sampling clock cycles of the normal UART from INIT release	1 to 32	33 to 600	601 to 856	857 to 1112	1113, 1114

The following shows the chip reset sequence.



OSC Wait: Oscillation of main source oscillation clock is stabilized. INIT is released after the oscillation stabilization is confirmed.

(1) DEBUGIF is set to H level by pull-up processing of the tool.

(2) DEBUGIF becomes the level of pull-up processing on the user system.

38.5.2.2 Security Function

Security function is shown.

OCDU has the security function. OCDU enables the security function by setting the security information stored in a debug security area of the memory space in CPU. If the security function is enabled, OCDU enters the security lock state. To release this, the security is unlocked by writing a password set in the security information to the E_SLPR register for the number of times and specified length.

Security Information

The debug security area is allocated at 30 bytes of built-in flash start address+4 to +33. OCDU looks up this area in the security sequence.

The following security information is available in the debug security area.

Security password length (PW length)

The security password length is 16-bit data in the start address of the debug security area, and the lower 4 bits are the enabled PW length. The upper 12 bits have no effect on operation. If the PW length is 0x0 or 0xF, the security is disabled. If the PW length is 0x1 to 0xE (1 to 14), the security is enabled.

Security password (PW)

The security password is 16-bit data in the debug security area. There are 1 to 14 passwords. The PW is assigned from an address next to the PW length address, in the order of PW1, PW2,... PW14 (See figure below). If the security is enabled (PW length:1 to 14), the value of the PW length indicates the enabled PW.

(Example: If the PW length is 8, PW1 to PW8 are enabled, and PW9 to PW14 are disabled.)

Address	15 0
ROM/Flash start address +4	PW length
ROM/Flash start address +6	PW1
ROM/Flash start address +8	PW2
...	...
ROM/Flash start address +32	PW14

Note:

If the security function of the on chip debugger (OCD) is not used, nothing is written to this area and the initial state (all bits=1) immediately after flash erase is retained.

38.5.3 Specification Restrictions at Connection to OCD Tool of This Series

Specification restrictions at connection to OCD tool of this series is shown.

The following restrictions are placed at OCD tool connection:

38.5.3.1 Clock Setting

Clock setting is shown.

- When the device is connected to the OCD tool, the main clock does not stop oscillating. It is, however, possible to read from the register CMONR.MCRDY and write to the register CSELR.MCEN, similarly to the case when not connected.
- When the device is communicating via high-speed UART or phase modulation UART, PLL does not stop oscillating. It is, however, possible to read from the register CMONR.PCRDY and write to the register CSELR.PCEN, similarly to the case when not connected. In addition, since PLL continues to oscillate, changes in settings of the registers shown below will have no effect. It is, however, possible to read and write the registers shown below, similarly to the case when not connected.
  - ☐ PLLCR.ODS
  - ☐ PLLCR.PMS
  - ☐ PLLCR.PDS
  - ☐ CCPSDIVR.PODS
  - ☐ CCPLLFBR.IDIV

38.5.3.2 Standby Mode

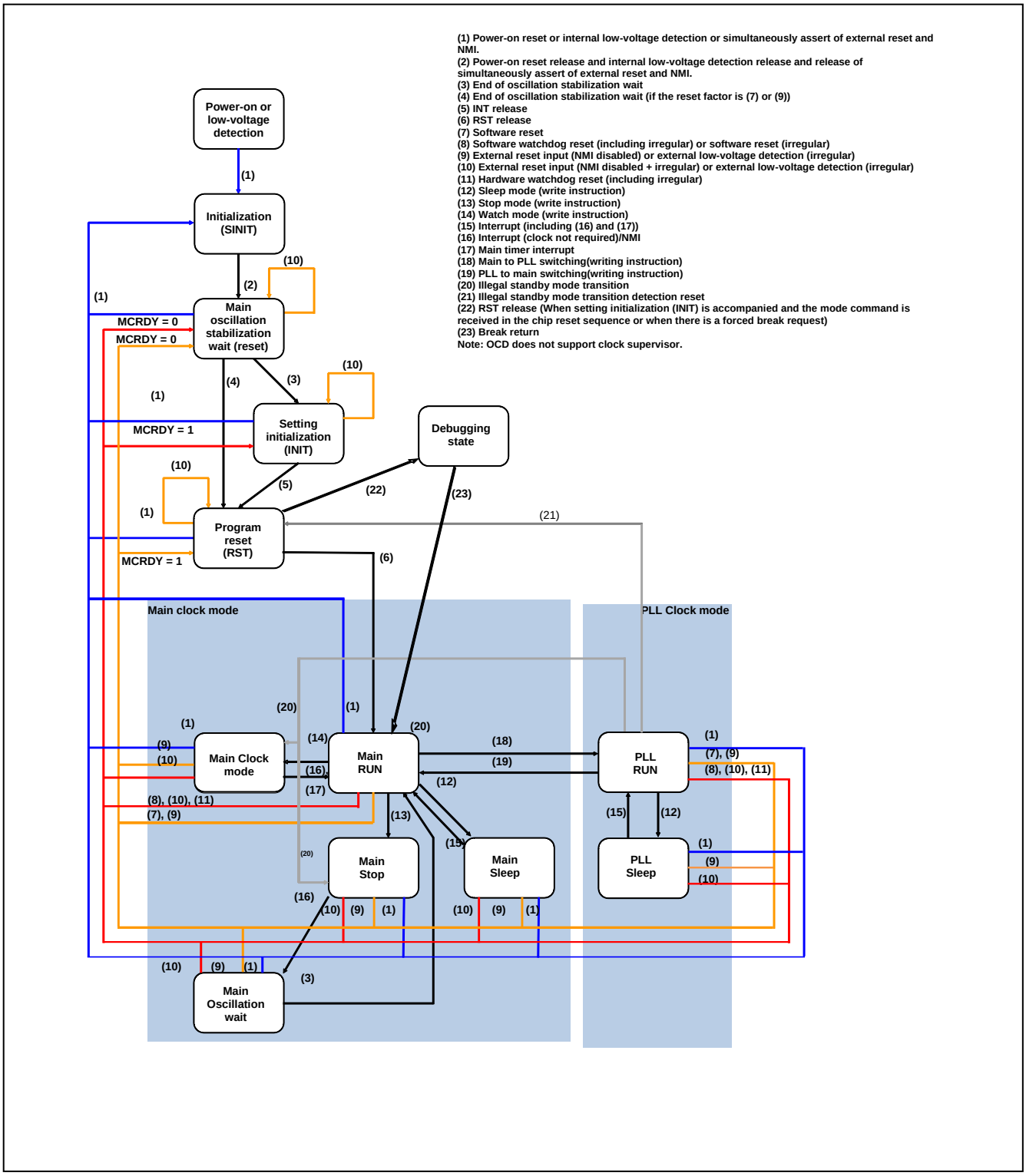
Standby mode is shown.

- Even if the watch mode is entered, PLL does not stop oscillating when OCD high-speed UART and phase modulation UART communication are enabled.
- The main clock does not stop oscillating even if the stop mode is entered. PLL does not stop oscillating when OCD high-speed UART and phase modulation UART communication are enabled.
- The following shows the functions that differ in operation when the OCD tool is not connected, according to the above restrictions:
  - ☐ When PLL is stopped down (CSELR.PCEN=0) or OCD high-speed UART or phase modulation UART communication are enabled,CAN operation continues in the watch mode and stop mode. (This operation is performed within the range in which no CPU processing occurs.)
  - ☐ The counter operation for the WDT1 (calibration) continues in the stop mode.
- The following functions perform the same operations as those when the OCD tool is not connected, with the above restrictions not placed.
  - ☐ The main timer does not run in the stop mode because it is cleared in that mode.
- The power consumption in the watch mode becomes greater than that when the OCD tool is not connected because the PLL clock oscillation continues.
- The power consumption in the stop mode becomes greater than that when the OCD tool is not connected because the PLL clock and main clock oscillation continue.

38.5.3.3 Clock Reset State Transitions

Clock reset state transitions is shown.

Figure 38-4. Device State



38.5.3.4 Summary of Specification Restrictions

Summary of specification restrictions is shown.

1. Communication mode (*1): Normal UART

Reset Factor	Difference from When the OCD Tool Is Not Connected		Remarks
	Initialization Range	Processing Time	
Watchdog reset 0			Causes a transition to the emulator mode (debug state) after reset is released
Watchdog reset 1 (irregular)			
Watchdog reset 1			
External power supply low-voltage detection reset (irregular)		Yes	
External power supply low-voltage detection reset			No voltage step-down circuit switch stabilization wait time (*2) Note: Only recovery from main stop mode or main watch mode
Illegal standby mode transition detection reset (irregular)	No		Causes a transition to the emulator mode (debug state) after reset is released
Illegal standby mode transition detection reset		No	
Low-voltage detection (internal power supply low-voltage detection) reset		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Flash security violation reset (irregular)			
Flash security violation reset		No	
Software reset (irregular)		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Software reset		No	

Interrupt Factor	Processing Time Difference from When the OCD Tool Is Not Connected	Remarks
All interrupts	Yes	No voltage step-down circuit switch stabilization wait time (*2) Note: Only recovery from main stop mode or main watch mode

Device States Other Than Those Related to Reset	Operation Difference from When the OCD Tool Is Not Connected	Remarks
Main RUN/main sleep mode	No	
PLL RUN/PLL sleep mode		
Main stop mode	Yes	Voltage step-down circuit is fixed Main oscillation continues Operation continues (WDT1 calibration counter operation)
Main/sub watch mode		Voltage step-down circuit is fixed Main oscillation continues

2. Communication mode (*1): High-speed UART/phase modulation UART

Reset Factor	Difference from When the OCD Tool Is Not Connected		Remarks
	Initialization Range	Processing Time	
Power-on reset	No	Yes	Causes a transition to the emulator mode (debug state) after reset is released
RSTX pin input (irregular)			
RSTX pin input			No voltage step-down circuit switch stabilization wait time (*2) Note: Only recovery from main stop mode or main watch mode
RSTX pin input (+NMIX pin input)			Causes a transition to the emulator mode (debug state) after reset is released
Watchdog reset 0 (irregular)			
Watchdog reset 0			
Watchdog reset 1 (irregular)			
Watchdog reset 1			
External power supply low-voltage detection reset (irregular)			
External power supply low-voltage detection reset			No voltage step-down circuit switch stabilization wait time (*2) Note: Only recovery from main stop mode or main watch mode
Illegal standby mode transition detection reset (irregular)		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Illegal standby mode transition detection reset		No	
Low-voltage detection (internal power supply low-voltage detection) reset		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Flash security violation reset (irregular)			
Flash security violation reset		No	
Software reset (irregular)		Yes	Causes a transition to the emulator mode (debug state) after reset is released
Software reset		No	

Interrupt Factor	Processing Time Difference from When the OCD Tool Is Not Connected	Remarks
All interrupts	Yes	No voltage step-down circuit switch stabilization wait time (*2) Note: Only recovery from main stop mode or main watch mode

Device States Other Than Those Related to Reset	Operation Difference from When the OCD Tool Is Not Connected	Remarks
Main RUN/main sleep mode	Yes	PLL oscillation continues
PLL RUN/PLL sleep mode	No	
Main stop mode	Yes	Voltage step-down circuit is fixed Main oscillation continues PLL oscillation continues (illegal standby mode transition detection is disabled) Operation continues (CAN, WDT1 calibration counter operation)
Main watch mode		Voltage step-down circuit is fixed Main oscillation continues Operation continues (CAN)

*1 For communication mode settings, see "SOFTUNE Workbench Operating Manual."

*2 Voltage step-down circuit stabilization waits time: about 6 µs

38.5.4 OCD-DSU ID Code and Mount Type Information on This Series

OCD-DSU ID code and mount type information of this series are shown.

Table 38-3. OCD-DSU ID Code of This Series

ID Name	bit Width	Associated ID Register Name	Address in the OCD Space	Value	Remarks
Manufacturer ID	16	E_IDMCR	0x000	0x0400	
CPU family ID	16	E_IDFCR	0x001	0x0200	FR81E/FR81S
DSU type ID	8	E_IDVCR	0x003	0x0X	
DSU version ID	4	E_IDVCR	0x003	0x1	
Device ID	16	E_IDDCR	0x002	0x002F	MB91F552
Device version ID	4	E_IDVCR	0x003	0x1	

Table 38-4. Mount Type Information of This Product Type

Product Name	Number of Code Events	Number of Data Events	Data Event (Compare)	Sequencer Event	Trace
MB91F552	8	8	○	○	512 frames

39. Bus Diagnosis Function

This chapter explains the bus diagnosis function.

39.1 Overview

39.2 Features

39.3 Configuration

39.4 Registers

39.5 Operation

39.1 Overview

This section explains the overview of BUS diagnosis function.

The bus diagnosis function prevents an LSI malfunction by checking data that was output on a bus during access to each resource.

		Function
1	Diagnosis target	Addresses and data to be output to the address and data buses, and control signals (read, write, read-modify-write, and bus access size signals) for controlling buses
2	Diagnosis bus	AHB, APB(PCLK1), Rbus(PCLK1), APB(PCLK2), Rbus(PCLK2)
3	Diagnosis method	Diagnosis based on parity The output side calculates parity for each 8-bit group and outputs that parity. The input side checks that parity.
4	Parity	Odd parity
5	Test function	A parity error is generated to support program debug at bus diagnosis.
6	Error detection	■ Control parity error ■ Address parity error ■ Data parity error At error occurrence, the relevant address is displayed in the register.
7	Effect	■ Detection of a bus disconnection ■ Detection of a bus transistor failure ■ Detection of garbled data resulting from a bus short caused by dirt ■ Detection of garbled data caused by a defective contact
8	NMI notification	Control parity error, address parity error, data parity error

39.2 Features

This section explains the features of BUS diagnosis function.

The internal bus of the CPU consists of AHB, APB(PCLK1), Rbus(PCLK1), APB(PCLK2) and Rbus(PCLK2). The bus diagnosis function is realized by adding parity to address, data, and bus control signal for each of them, and by checking the parities. However, if a parity error occurs at write access to the bus diagnosis register, the NMI request signal cannot be cleared. So, the write access to bus diagnosis register is excluded from diagnosis.
When the parity error is detected at write access, write access to peripheral resource is shut off.

NMI output resulting from bus diagnosis is generated by one of the following factors.

- A control parity error (CNER) was detected.
- A data parity error (DER[3:0]) was detected.
- An address parity error (AER[3:0]) was detected.

39.3 Configuration

This section explains the Configuration of BUS diagnosis function.

Figure 39-1. Configuration Diagram

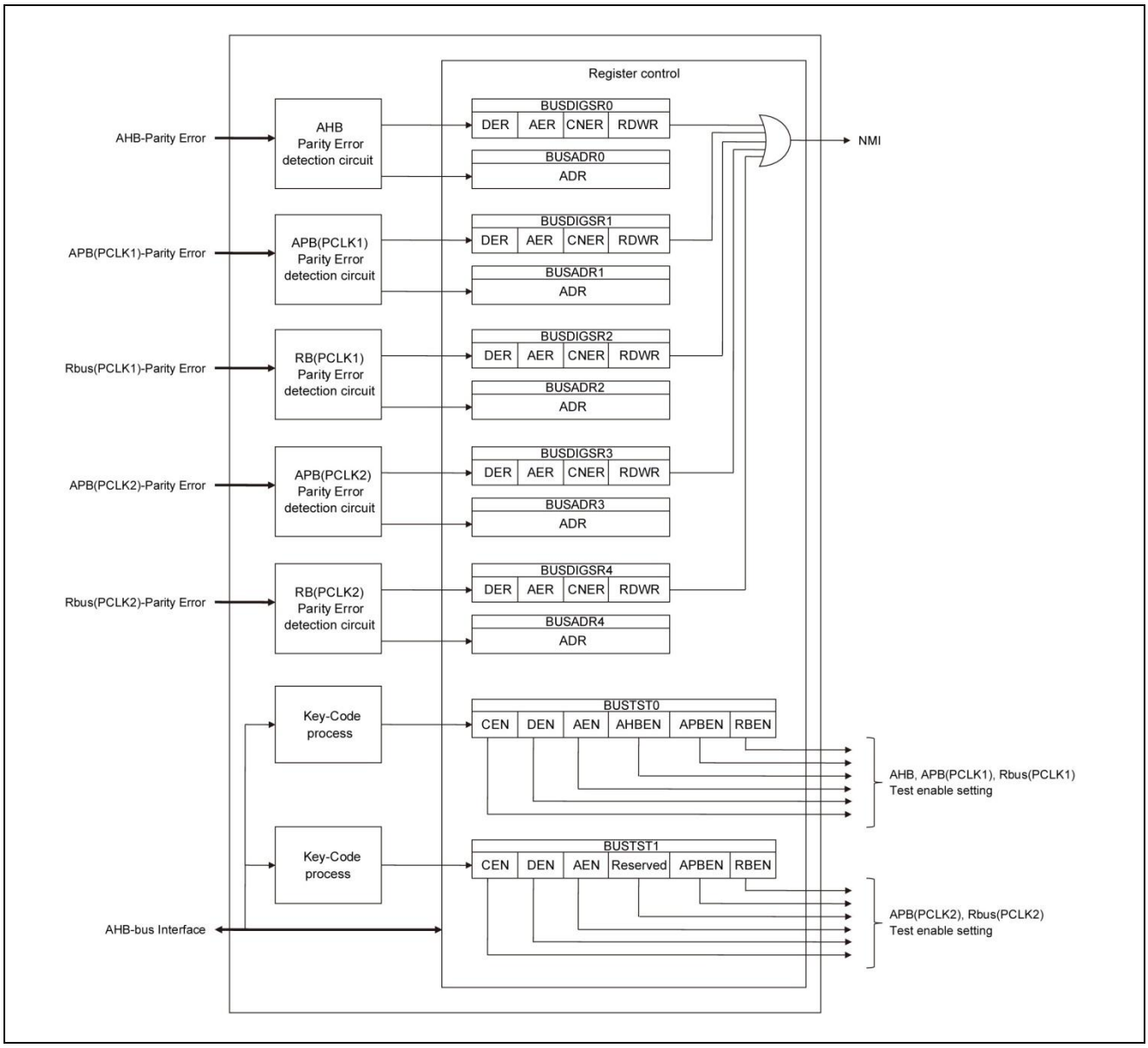
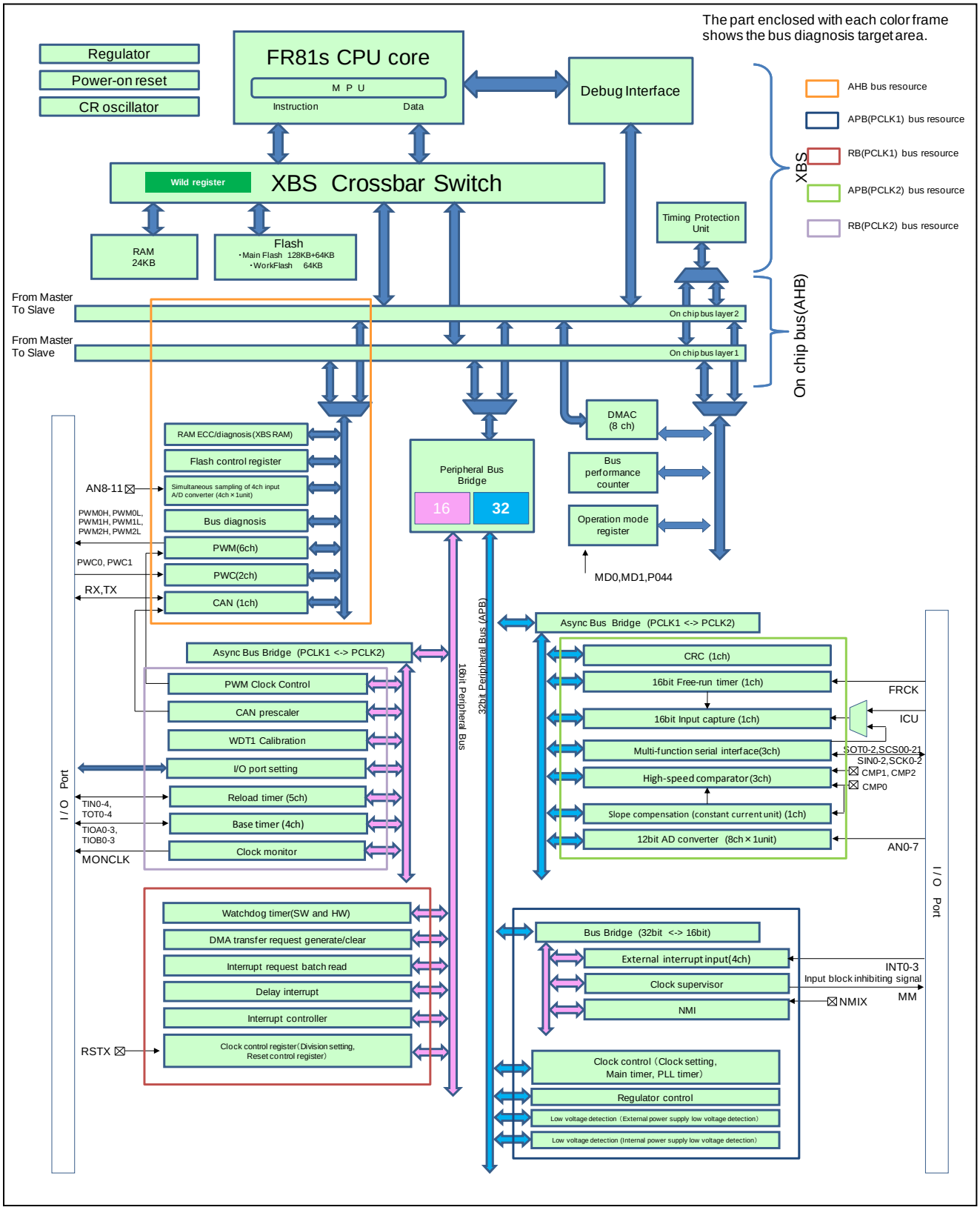


Figure 39-2. Bus Diagnosis Target Area Diagram



Note:

The following address areas are excluded from bus diagnosis target.

Watchdog timer	0x00000038 to 0x0000003F *
Delay interrupt	0x00000044 *
MPU	0x00000310 to 0x000003AC To diagnosis target of reload timer 0/1/2
Interrupt controller	0x00000440 to 0x0000046F *
Reset control/Low-power consumption control	0x00000480 to 0x00000482 *
Clock control	0x00000488 to 0x0000048A *
DMA transfer request by peripheral	0x00000490 to 0x0000049F *
Bus performance counter	0x00000710 to 0x0000071F
Flash memory register	0x00000840 to 0x00000843
Wild register	0x0000085A to 0x0000085B
	0x00000880 to 0x000008FF
OCDU	0x00000BF0 to 0x00000BFF
Synchronous/Asynchronous setting	0x00001000 to 0x00001001
DMA controller	0x00000C00 to 0x00000DFF *
Bus diagnosis	0x00003100 to 0x00003127 (Read data is a diagnosis target.)

* Area connected to R-bus that excluded from diagnosis target may cause error detection under some condition of register setting or failure condition.
For details, see "39.5.3 Notes."

39.4 Registers

This section explains the registers of bus diagnosis function.

Table 39-1. Register Map

Address	bit31 bit16	bit15 bit0	Initial Value
003100_H	BUSDIGSR0 (Bus diagnosis status register 0)	BUSDIGSR1 (Bus diagnosis status register 1)	0x00000000
003104_H	BUSDIGSR2 (Bus diagnosis status register 2)	BUSTSTR0 (Bus diagnosis test register 0)	0x00000000
003108_H	BUSADR0 (Bus diagnosis address register 0)		0x00000000
$00310C_H$	BUSADR1 (Bus diagnosis address register 1)		0x00000000
003110_H	BUSADR2 (Bus diagnosis address register 2)		0x00000000
003114_H	Reserved	BUSDIGSR3 (Bus diagnosis status register 3)	0x00000000
003118_H	BUSDIGSR4 (Bus diagnosis test register 4)	BUSTSTR1 (Bus diagnosis test register 1)	0x00000000
$00311C_H$	Reserved		0x00000000
003120_H	BUSADR3 (Bus diagnosis address register 3)		0x00000000
003124_H	BUSADR4 (Bus diagnosis address register 4)		0x00000000

39.4.1 BUS DIaGnosis Status Register: BUSDIGSR

This section explains the bus diagnosis status register.

The bus diagnosis status register (BUSDIGSR) consists of a data parity error, address parity error, control parity error, data direction, and error flag clear.
Bus diagnosis status register 0 indicates AHB error status. Bus diagnosis status register 1 indicates APB(PCLK1) error status. Bus diagnosis status register 2 indicates Rbus(PCLK1) error status. Bus diagnosis status register 3 indicates APB(PCLK2) error status. Bus diagnosis status register 4 indicates Rbus(PCLK2) error status.

BUSDIGSR0: Address 3100H (Access: Half-word, Word)

BUSDIGSR1: Address 3102H (Access: Half-word, Word)

BUSDIGSR2: Address 3104H (Access: Half-word, Word)

BUSDIGSR3: Address 3116H (Access: Half-word, Word)

BUSDIGSR4: Address 3118H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DER[3]	DER[2]	DER[1]	DER[0]	AER[3]	AER[2]	AER[1]	AER[0]
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PECLR	Reserved					CNER	RDWR
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0/W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX

[bit15 to bit12] DER3 to DER0: Data parity error

Data parity error flag. Parity is calculated for each piece of 8-bit data. If an error occurs, the associated bit is set to "1".

For DER[3]=1, a parity error occurs in bit7 to bit0 of data.
For DER[2]=1, a parity error occurs in bit15 to bit8 of data.
For DER[1]=1, a parity error occurs in bit23 to bit16 of data.
For DER[0]=1, a parity error occurs in bit31 to bit24 of data.

If these bits are "0", this indicates that no error occurs. These bits are not updated while "1" is set in any one of these bits. If these bits are set to "1", NMI occurs.
These bits are read-only. To clear them to "0", write "1" in the PECLR bit.

Note:

- These bits are not updated while any one of them is "1", any one of the AER bits is "1", or the CNER bit is "1".
- Regarding the data parity error, error detection and notification are made only for data with a valid access size.
- Note that it may be led to infinite loop when an error is detected after the status register is read immediately after an error of the bus diagnosis status register is cleared.

[bit11 to bit8] AER3 to AER0: Address parity error

Address parity error flag. Parity is calculated for each piece of 8-bit address. If an error occurs, the associated bit is set to "1".

For AER[3]=1, a parity error occurs in bit7 to bit0 of address.
For AER[2]=1, a parity error occurs in bit15 to bit8 of address.
For AER[1]=1, a parity error occurs in bit23 to bit16 of address.
For AER[0]=1, a parity error occurs in bit31 to bit24 of address.

If these bits are "0", this indicates that no error occurs. If any one of these bits is set to "1", NMI occurs.
These bits are read-only. To clear them to "0", write "1" in the PECLR bit.

Note:

These bits are not updated while any one of them is "1", any one of the DER bits is "1", or the CNER bit is "1".

[bit7] PECLR: Parity error clear

Parity error clear bit. If "1" is written in this bit, the DER, AER, and CNER bits are set to "0". This bit is always "0" during reading. Writing "0" in this bit is ignored. This bit is write-only.

[bit6 to bit2] Reserved

Always write "0" to these bits.

[bit1] CNER: Control parity error

Control parity error bit. Read/write signals for controlling buses and the data size control signal are handled as data. If a parity error occurs, this bit is set to "1".
If this bit is "0", this indicates that no error occurs. This bit is not updated while "1" is set in this bit. If this bit is set to "1", NMI occurs.
This bit is read-only. To clear it to "0", write "1" in the PECLR bit.

Note:

This bit is not updated while this bit is "1", any of the AER bits is "1", or the DER bit is "1".

[bit0] RDWR: Data direction

Data direction flag. If a data or address parity error occurs, this bit indicates that the error has occurred during reading or writing.

The read direction is indicated when this bit is "0".
The write direction is indicated when this bit is "1".

Note:

- Writing "1" in the PECLR bit does not influence this bit.
- This bit is not updated while the DER, AER, or CNER bit is "1".
- This bit is valid when any one of the DER, AER, and CNER bits is "1".

39.4.2 BUS diagnosis TeST Register: BUSTSTR0/1

This section explains the bus diagnosis test register.

The bus diagnosis test register (BUSTSTR0, BUSTSTR1) sets the bus diagnosis test function.

BUSTSTR0: Address 3106H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	KEY1	KEY0	-		CEN	RBEN	APBEN	AHBEN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0/W	R0/W	R0,WX	R0,WX	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DEN[3]	DEN[2]	DEN[1]	DEN[0]	AEN[3]	AEN[2]	AEN[1]	AEN[0]
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 bit14] KEY1, KEY0: Key bits

Key bits. If "00", "01", "10", and "11" are continuously written in these bits, the data is updated to the data written in bit11 to bit0. However, during this continuous writing, data in bit11 to bit0 is not updated unless the same value is written in bit11 to bit0 4 times. Moreover, data is not updated if the bus diagnosis register is read during writing. In this case, writing to this register continuously 4 times is required again.

Example:
Write $07AA_H$ in BUSTSTR.
Next, write $47AA_H$ in BUSTSTR.
Next, write $87AA_H$ in BUSTSTR.
Next, write $C7AA_H$ in BUSTSTR. -> With this writing, BUSTSTR is set to $07AA_H$.

[bit13, bit12] Undefined

"0" is always read. Writing does not affect operation.

[bit11] CEN: Control error

Control error setting bit.

If this bit is "0", the control parity is properly generated.
If this bit is "1", a control parity error occurs.

Note:

For RBEN=0, APBEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit10] RBEN: Rbus parity error generation enable

This bit enables the generation of a Rbus (PCLK1) parity error.

If this bit is "0", the Rbus parity is generated as correct one (odd parity).
If this bit is "1", the Rbus parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]=0000, AEN[3:0]=0000, CEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit9] APBEN: APB parity error generation enable

This bit enables the generation of an APB (PCLK1) parity error.

If this bit is "0", the APB parity is generated as correct one (odd parity).
If this bit is "1", the APB parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]=0000, AEN[3:0]=0000, CEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit8] AHBEN: AHB parity error generation enable

This bit enables the generation of an AHB parity error.

If this bit is "0", the AHB parity is generated as correct one (odd parity).
If this bit is "1", the AHB parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]=0000, AEN[3:0]=0000, CEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to this bit.

[bit7 to bit4] DEN3 to DEN0: Data error

Data error setting bits.

If DEN[3] is "0", parity in bit7 to bit0 of the data bus is properly generated.
If DEN[3] is "1", a parity error for bit7 to bit0 of the data bus is generated.
If DEN[2] is "0", parity in bit15 to bit8 of the data bus is properly generated.
If DEN[2] is "1", a parity error for bit15 to bit8 of the data bus is generated.
If DEN[1] is "0", parity in bit23 to bit16 of the data bus is properly generated.
If DEN[1] is "1", a parity error for bit23 to bit16 of the data bus is generated.
If DEN[0] is "0", parity in bit31 to bit24 of the data bus is properly generated.
If DEN[0] is "1", a parity error for bit31 to bit24 of the data bus is generated.

Note:

- For RBEN=0, APBEN=0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.
- Only value set in valid data range of access size is set as an error.

[bit3 to bit0] AEN3 to AEN0: Address error

Address error setting bits.

If AEN[3] is "0", parity in bit7 to bit0 of the address bus is properly generated.
If AEN[3] is "1", a parity error for bit7 to bit0 of the address bus is generated.
If AEN[2] is "0", parity in bit15 to bit8 of the address bus is properly generated.
If AEN[2] is "1", a parity error for bit15 to bit8 of the address bus is generated.
If AEN[1] is "0", parity in bit23 to bit16 of the address bus is properly generated.
If AEN[1] is "1", a parity error for bit23 to bit16 of the address bus is generated.
If AEN[0] is "0", parity in bit31 to bit24 of the address bus is properly generated.
If AEN[0] is "1", a parity error for bit31 to bit24 of the address bus is generated.

Note:

For RBEN=0, APBEN=0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.

Note:

- Any interrupt is disabled during writing to the bus diagnosis test register.
- The bus diagnosis test register is used for debugging the bus diagnosis functions.
- Writing to the bus diagnosis test register is performed even if diagnosis error occurs, but the error flag of the target status register is not set to "1".
- Continue 2 times the same access in order to generate a parity error against an access to RB-bus resource.
- Set those bits in the bus diagnosis test register only when the test function is used.

BUSTSTR1: Address 311A$_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	KEY1	KEY0	-		CEN	RBEN	APBEN	Reserved
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0/W	R0/W	R0,WX	R0,WX	R/W	R/W	R/W	R/W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DEN[3]	DEN[2]	DEN[1]	DEN[0]	AEN[3]	AEN[2]	AEN[1]	AEN[0]
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15, bit14] KEY1, KEY0: Key bits

Key bits. If "00", "01", "10", and "11" are continuously written in these bits, the data is updated to the data written in bit11 to bit0. However, during this continuous writing, data in bit11 to bit0 is not updated unless the same value is written in bit11 to bit0 4 times. Moreover, data is not updated if the bus diagnosis register is read during writing. In this case, writing to this register continuously 4 times is required again.

Example:
Write 07AA$_H$ in BUSTSTR.
Next, write 47AA$_H$ in BUSTSTR.
Next, write 87AA$_H$ in BUSTSTR.
Next, write C7AA$_H$ in BUSTSTR. -> With this writing, BUSTSTR is set to 07AA$_H$.

[bit13, bit12] Undefined

"0" is always read. Writing does not affect operation.

[bit11] CEN: Control error

Control error setting bit.

If this bit is "0", the control parity is properly generated.
If this bit is "1", a control parity error occurs.

Note:

For RBEN=0, APBEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit.

[bit10] RBEN: Rbus parity error generation enable

This bit enables the generation of a Rbus (PCLK2) parity error.

If this bit is "0", the Rbus parity is generated as correct one (odd parity).
If this bit is "1", the Rbus parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]=0000, AEN[3:0]=0000, CEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit.

[bit9] APBEN: APB parity error generation enable

This bit enables the generation of an APB (PCLK2) parity error.

If this bit is "0", the APB parity is generated as correct one (odd parity).
If this bit is "1", the APB parity is generated as even parity so that an error occurs.

Note:

For DEN[3:0]=0000, AEN[3:0]=0000, CEN=0, this bit is invalid. Under such condition, it will be the same behavior as when "0" is set to the bit.

[bit8] Reserved

Always write "0".

[bit7 to bit4] DEN3 to DEN0: Data error

Data error setting bits.

If DEN[3] is "0", parity in bit7 to bit0 of the data bus is properly generated.
If DEN[3] is "1", a parity error for bit7 to bit0 of the data bus is generated.
If DEN[2] is "0", parity in bit15 to bit8 of the data bus is properly generated.
If DEN[2] is "1", a parity error for bit15 to bit8 of the data bus is generated.
If DEN[1] is "0", parity in bit23 to bit16 of the data bus is properly generated.
If DEN[1] is "1", a parity error for bit23 to bit16 of the data bus is generated.
If DEN[0] is "0", parity in bit31 to bit24 of the data bus is properly generated.
If DEN[0] is "1", a parity error for bit31 to bit24 of the data bus is generated.

Note:

- For RBEN=0, APBEN=0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.
- Only value set in valid data range of access size is set as an error.

[bit3 to bit0] AEN3 to AEN0: Address error

Address error setting bits.

If AEN[3] is "0", parity in bit7 to bit0 of the address bus is properly generated.
If AEN[3] is "1", a parity error for bit7 to bit0 of the address bus is generated.
If AEN[2] is "0", parity in bit15 to bit8 of the address bus is properly generated.
If AEN[2] is "1", a parity error for bit15 to bit8 of the address bus is generated.
If AEN[1] is "0", parity in bit23 to bit16 of the address bus is properly generated.
If AEN[1] is "1", a parity error for bit23 to bit16 of the address bus is generated.
If AEN[0] is "0", parity in bit31 to bit24 of the address bus is properly generated.
If AEN[0] is "1", a parity error for bit31 to bit24 of the address bus is generated.

Note:

For RBEN=0, APBEN=0, these bits are invalid. Under such condition, it will be the same behavior as when "0000" is set to these bits.

39.4.3 BUS diagnosis ADdRess Register: BUSADR

This section explains the bus diagnosis address register.

If an address parity error, data parity error, or control parity error is detected, the bus diagnosis address register (BUSADR) stores the relevant address. This register is valid when the DER, AER, or CNER bit of the bus diagnosis status register (BUSDIGSR) is "1".

Bus diagnosis address register 0 indicates an address in which an AHB diagnosis error was detected. Bus diagnosis address register 1 indicates an address in which an APB (PCLK1) diagnosis error was detected. Bus diagnosis address register 2 indicates an address in which an Rbus (PCLK1) diagnosis error was detected. Bus diagnosis address register 3 indicates an address in which an APB (PCLK2) diagnosis error was detected. Bus diagnosis address register 4 indicates an address in which an Rbus (PCLK2) diagnosis error was detected.

BUSADR0: Address 3108H (Access: Word)

BUSADR1: Address 310CH (Access: Word)

BUSADR2: Address 3110H (Access: Word)

BUSADR3: Address 3120H (Access: Word)

BUSADR4: Address 3124H (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	ADR[31]	ADR[30]	ADR[29]	ADR[28]	ADR[27]	ADR[26]	ADR[25]	ADR[24]
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	ADR[23]	ADR[22]	ADR[21]	ADR[20]	ADR[19]	ADR[18]	ADR[17]	ADR[16]
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADR[15]	ADR[14]	ADR[13]	ADR[12]	ADR[11]	ADR[10]	ADR[9]	ADR[8]
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADR[7]	ADR[6]	ADR[5]	ADR[4]	ADR[3]	ADR[2]	ADR[1]	ADR[0]
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit0] ADR31 to ADR0: Bus address

If an address or data parity error is detected, these bits indicate an address that is accessed at that detection.
This register is read-only.

Note:

- This register is valid when any one of the DER, AER, or CNER bit of the bus diagnosis status register is "1".
- This register is not updated when any one of the DER, AER, or CNER bit of the bus diagnosis status register is "1".

39.5 Operation

This section explains the operation of bus diagnosis function.

If an access is made from AHB, APB, or Rbus, the bus diagnosis performs parity check for the address, data, and control buses to diagnose the address, data, and control bus as being correct.
If a bus is determined as failure by AHB, APB, or Rbus, the content of each error is notified to the bus diagnosis status register and address of resource to the bus diagnosis address register, and the content of failure can be determined.
During writing, no writing is made to a peripheral function if an address parity error, data parity error and control parity error are generated.

39.5.1 Error Detection

This section explains the Error detection of bus diagnosis function.

When the error is detected by the bus diagnosis, access address and access direction at the time of error detection are stored in the RDWR bit of the bus diagnosis address register (BUSADR) and of the bus diagnosis status register (BUSDISR), respectively.
Moreover, when the error by the write access is detected, the write to the resource is not done.

Address error detection

When the parity operation result of the bus address is an error, the address error detection sets "1" to AER[3:0] bit in the bus diagnosis status register (BUSDISR).
AER[3:0] bit can be cleared by writing "1" to PECLR bit in this register.

Control error detection

When the parity operation result of the bus control is an error, the control error detection sets "1" to CNER bit in the bus diagnosis status register (BUSDISR).
The CNER bit can be cleared by writing "1" to PECLR bit in this register.

Data error detection

When the parity operation result of the bus data is an error, the data error detection sets "1" in the corresponding DER[3:0] bit of the bus diagnosis status register (BUSDISR) according to the access size of the word, the half-word, and the byte.
DER[3:0] bit can be cleared by writing "1" to PECLR bit in this register.

The error detection part of the bus diagnosis status register (BUSDIGSR).DER[3:0] is shown as follows.
(○: Error detection is done. -: Error detection is not done.)

Access Size	Address	BUSDIGSR0 (AHB: On-chip Bus) BUSDIGSR1/3 (APB: 32bit Peripheral Bus) BUSDIGSR2/4 (R-bus: 16bit Peripheral Bus)			
		DER[0]	DER[1]	DER[2]	DER[3]
		Data bit31-24	Data bit23-16	Data bit15-8	Data bit7-0
Word access	Addr+0	○ *	○ *	○ *	○ *
Half-word access	Addr+0	○	○	-	-
Half-word access	Addr+2	-	-	○	○
Byte access	Addr+0	○	-	-	-
Byte access	Addr+1	-	○	-	-
Byte access	Addr+2	-	-	○	-
Byte access	Addr+3	-	-	-	○

* R-bus: The word access to 16bit peripheral bus are treated as 2-times half-word access. Therefore, only the half-word access error detected first is notified to the register. It is accessed in following order usually.
Upper half-word access (data bit31-16) --> lower half-word access (data bit15-0)

NMI request generation/stop

NMI request is continued while either error (address error (AER[3:0]), control error (CNER) or data error (DER)) of each bus has been detected.
NMI request is discontinued when all errors (address error (AER[3:0]), control error (CNER) and data error (DER)) of each bus are cleared.

Note:

BUS diagnostic function generates NMI interrupt upon detecting of bus error. NMI is an interrupt that cannot be masked (i.e., cannot be suppressed).
On software side, you must always configure an NMI process routine. If NMI process routine is undefined, and if bus failure occurs, program execution runs out of control after NMI is generated.

39.5.2 Test Function

This section explains the test function of bus diagnosis function.

In this function, a pseudo error can be generated if bus diagnosis test register (BUSTSTR) is used.
The key code processing is necessary for being set to this register.
If "00", "01", "10", and "11" are not continuously written in the KEY1 and KEY0 bits, the register is not set.
At this time, if the same value is not written 4 times, the register value is not updated.
However, even if the bus diagnosis test register is set, the pseudo data error is not detected when the reading data is all "1".

Note:

ALL "1" read might detect the error according to the register setting and the access requirement.
Please see Notes in "Data error setting" for details.
Only when the test function is used, the foregoing limitation is applied.

Bus error setting

A pseudo error can be caused for the set bus by setting "1" to RBEN, APBEN, and AHBEN.
However, it is necessary to set "1" to either of AEN[3:0], CEN or DEN[3:0] bit at the same time.

Address error setting

The pseudo address error can be caused in the corresponding address bit by setting "1" to AEN[3:0].
However, it is necessary to set "1" to either of RBEN, APBEN or AHBEN bit at the same time.

Control error setting

The pseudo control error can be caused in the control bit by setting "1" to CEN.
However, it is necessary to set "1" to either of RBEN, APBEN or AHBEN bit at the same time.

Data error setting

The pseudo data error can be caused in the corresponding data bit by setting "1" to DEN[3:0].
However, it is necessary to set "1" to either of RBEN, APBEN or AHBEN bit at the same time.

Notes:

- Set only the bit corresponding to the access size when you set DEN[3:0].
  When DEN[3:0] is set to the bit that doesn't correspond to the access size, the error of the data not accessed might be detected.
- When you set DEN[3:0] to the bit that doesn't correspond to the access size, it might be different from the method of detecting the data error described in "Data error detection."

Bus diagnosis pseudo error generation procedure

The procedure that causes a pseudo error is the following.

1. Set the type of the bus error to be diagnosed in the bus diagnosis test register (BUSTSTR).

- In the key code, write the same error setting 4 continuous times, "00" -> "01" -> "10" -> "11".

2. Access the resource in the diagnosis area with the bus that does the pseudo error setting.

- When you set the address error and the control error (When you do not set the data error.)

  => A pseudo error occurs by accessing the resource in the diagnosis area.

- When you set the data error

  => A pseudo error occurs by accessing the resource in the diagnosis area by the access size corresponding to DEN[3:0]. (Please see Notes in "Data error setting" for details.)

39.5.3 Notes

This section explains the notes of bus diagnosis function.

- When bus diagnosis test function of R-bus is used

  When it accesses the following resource area in the state (RBEN was set to "1" by the bus diagnosis test register, and either AEN[3:0] and CEN or DEN[3:0] was set to "1"), the error is detected. At this time, it might seem that the bus was diagnosed.

At this time, the write access to the resource is executed.

- Watchdog timer
- Delay interrupt
- Interrupt controller
- Reset control/power consumption control
- Clock control
- DMA transfer request by peripheral
- DMA controller

39.5.4 Example of Operating Bus Diagnosis

This section explains the example of operating bus diagnosis.

The example of operating the bus diagnosis is shown as follows.

- Data error detection operation
  The data bus breaks down when the write (read) is accessed to the resource by byte access[7:0].
  => Only DER[3] of the bus diagnosis status register corresponding to the resource is set to "1".

Notes:

- Even if the bus is out of order at this time by data bus[31:8], neither DER[2], DER[1] nor DER[0] are set to "1".
- Only the bit corresponding to the access size is similarly set as for the word and the half-word access.

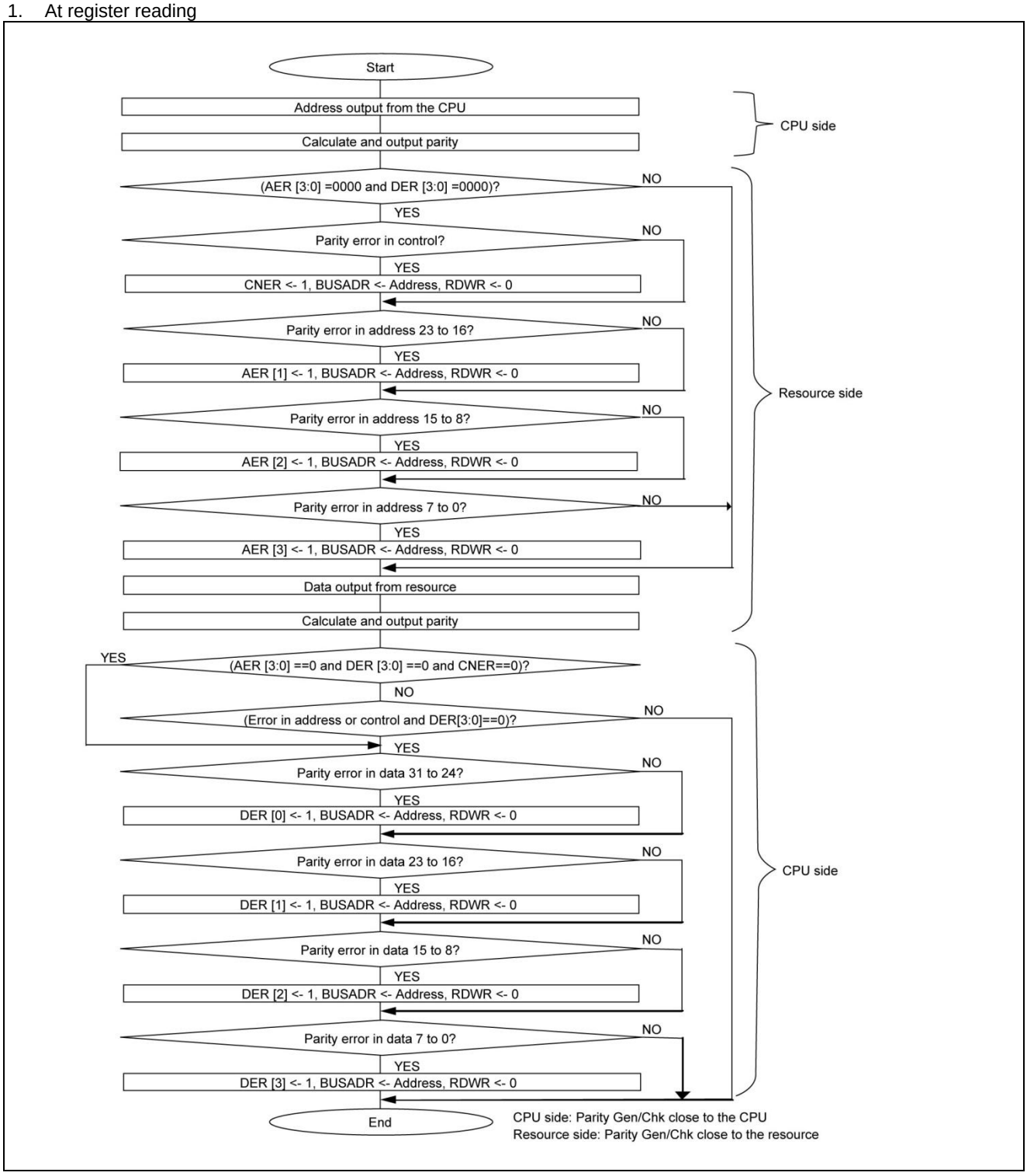
- Pseudo data error setting operation
  When you want to make the error detected in bus diagnosis status register DER[0] in the test function
  => Set "1" to bus error setting and data error setting DER[0] by the key code access, and do byte access[31:24].

When DEN[3:1] is set, and byte access [31:24] is done, the error might be detected in bus diagnosis status register DER[3:1].

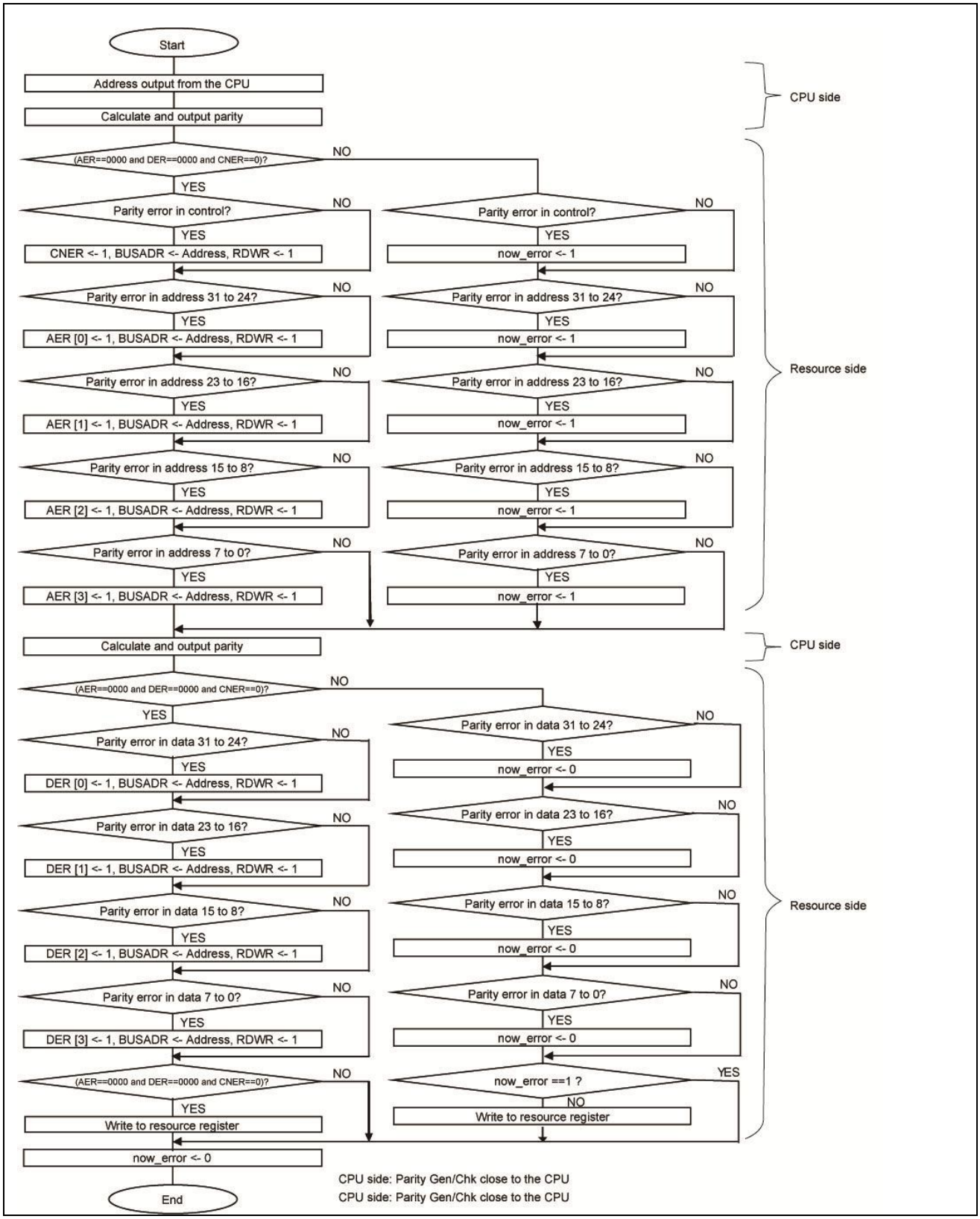
Bus Diagnosis Operation Flow

The operation flow of the bus diagnosis is shown below.

1. At register reading



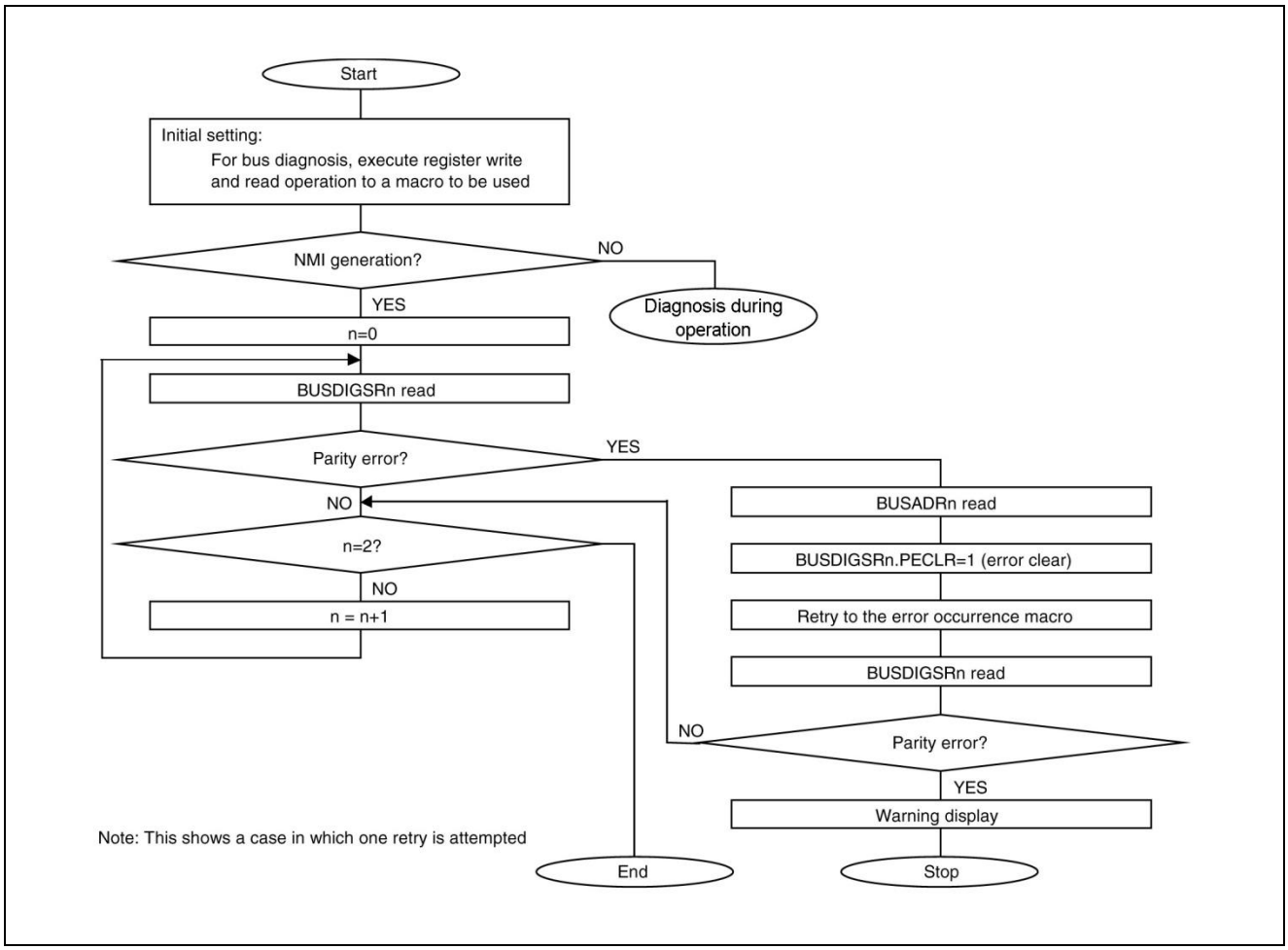
2. At register writing



3. Processing at error detection

The following gives an example of processing at error detection.

- At Initialization



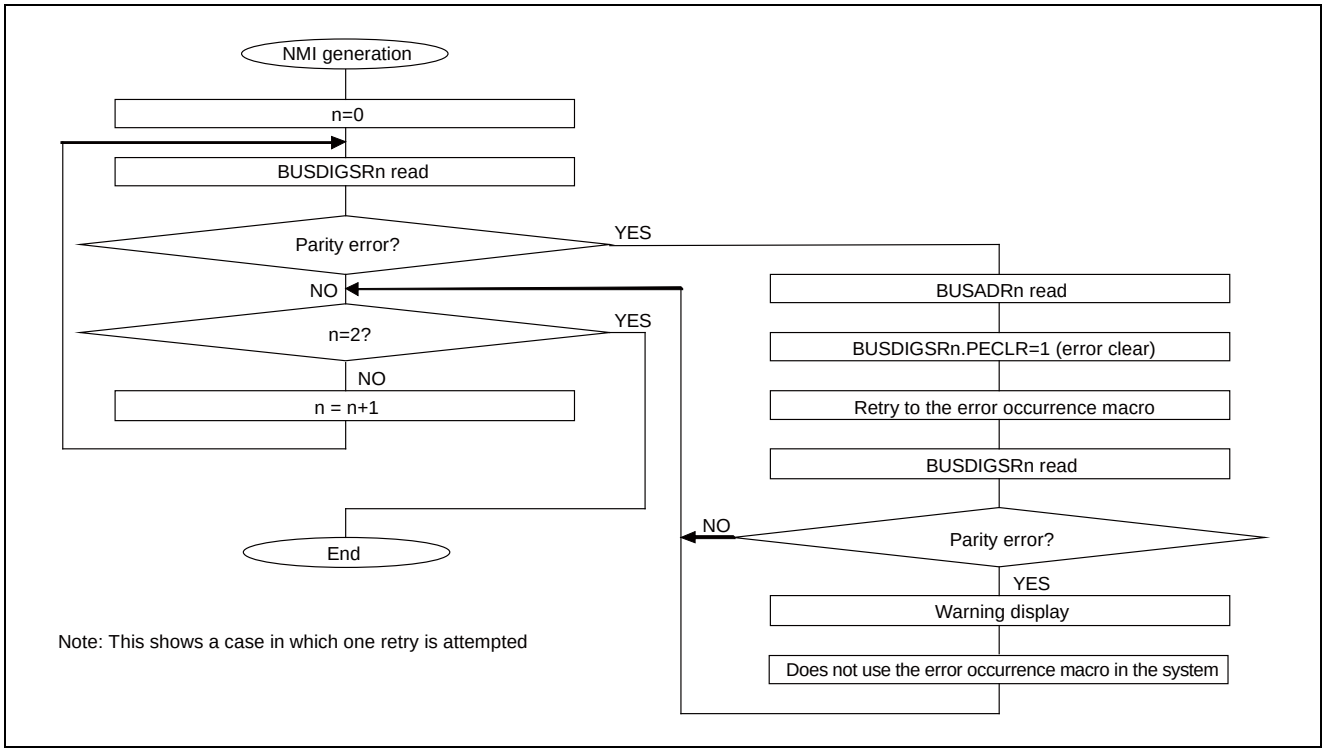
Note:

Before setting the ILM interrupt level mask register to 16 (10000B) or more, set the address of the interrupt function to the interrupt vector.
If the program is executed in following steps while the bus signal continuously malfunctions, CPU refers to the interrupt function address immediately after the ILM setting.

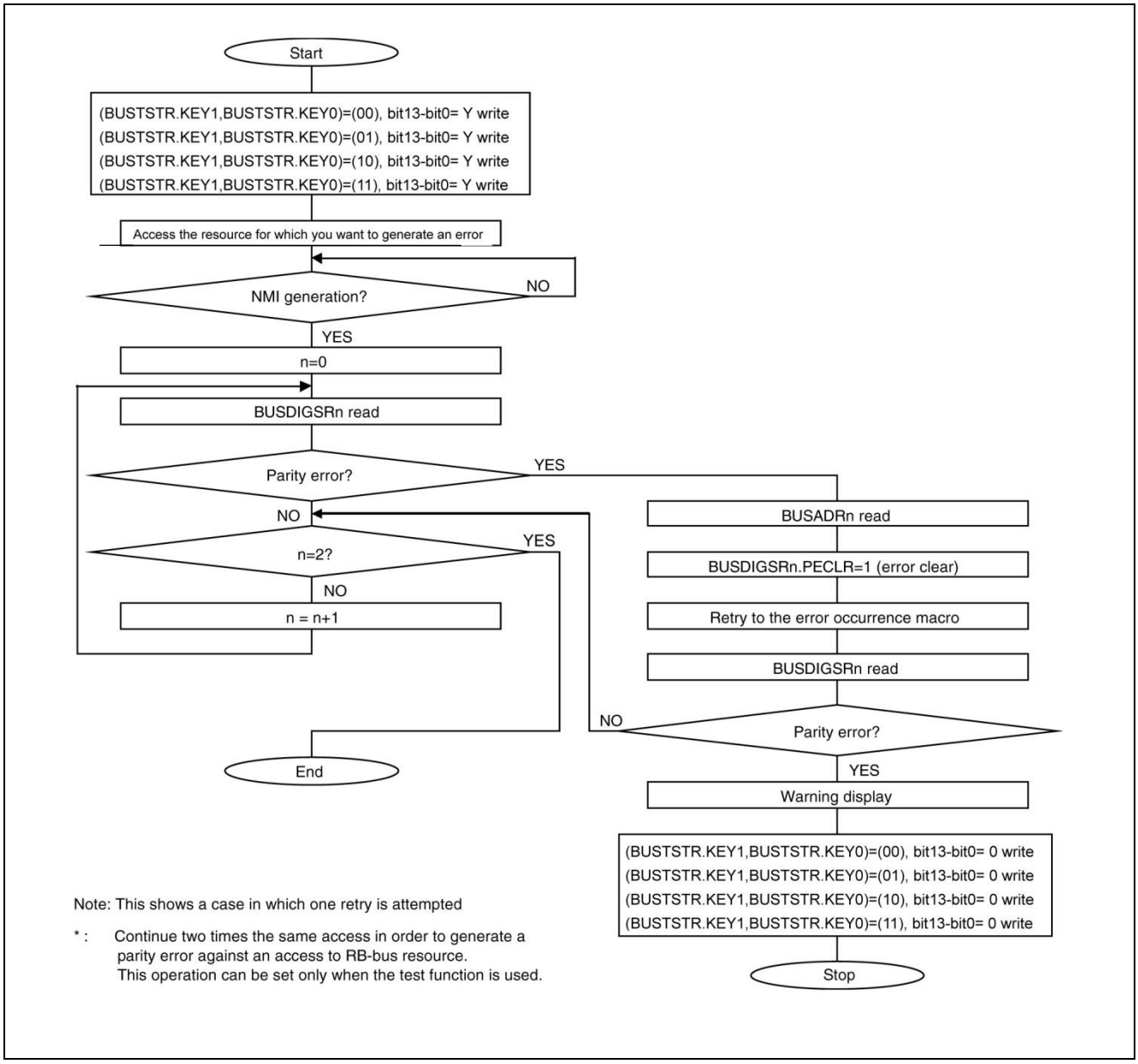
Example:

1. Define the vector area.
2. Clear the Stack.
3. Set the pin register.
4. Set the interrupt mask register (ILM: interrupt mask register).
   The NMI of the bus diagnosis is detected in this step, then the program is terminated.
5. Set the interrupt function address for the interrupt vector.

- During operation



- In the test mode



40. RAM Diagnosis Function

This chapter explains the RAM diagnosis function.

40.1 Overview

40.2 Features

40.3 Configuration

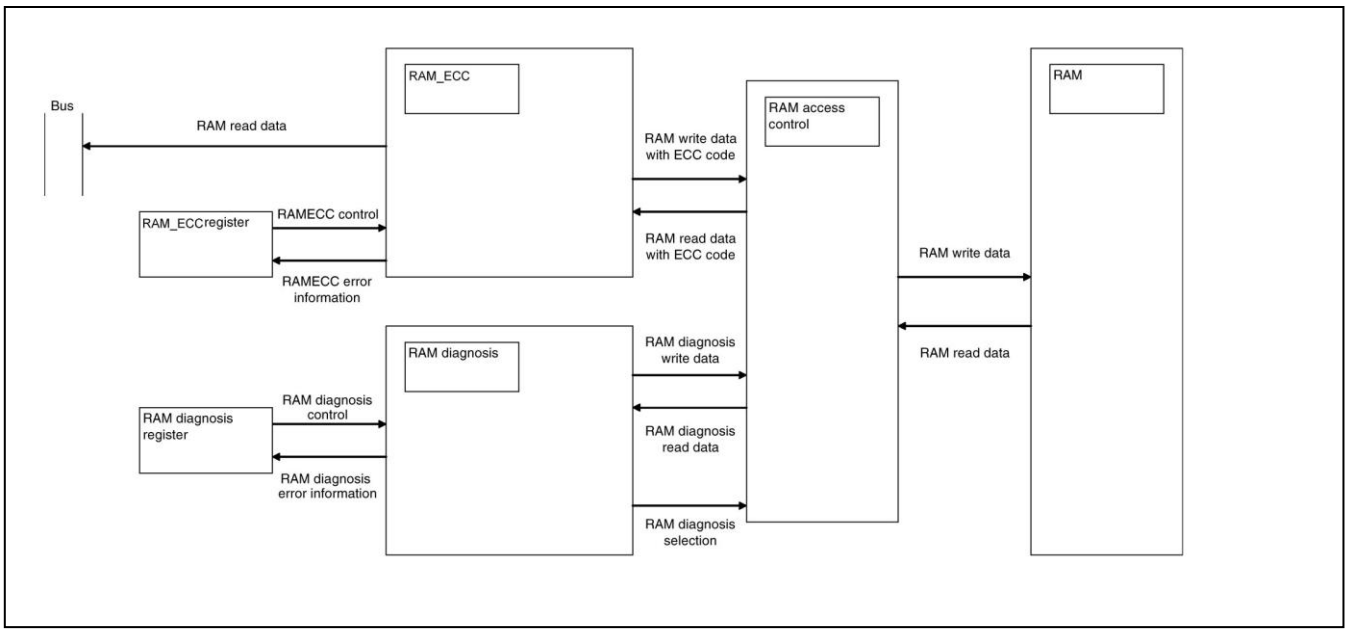
40.4 Registers

40.5 Operation

40.1 Overview

This section explains overview of RAM diagnosis function.

The RAM diagnosis block performs diagnosis of RAM and initialization of RAMs.



For RAMECC functions, see "Chapter: RAMECC Function."

40.2 Features

This section explains feature of RAM diagnosis function.

Target RAM

- XBS RAM

  MB91F552: 24 KB

RAM Diagnosis

The RAM diagnoses listed below can be selected and executed. (Two or more items selectable.)

Unique (unique data is {Address [3:0],{6{Address [7:0]}}})
Checker
March (all "0" -> all "1" are executed in that order.)

Interrupt function

An interrupt signal for a diagnosis end factor is generated. (RAM diagnosis end interrupt)
An interrupt signal is generated when an error is detected. (RAM diagnosis error interrupt)

RAM Initialization

One of the following RAM initializations is selected and executed.

Write all "0"
Write all "1"

Interrupt function

An interrupt signal for an initialization end factor is generated. (RAM initialization complete interrupt)

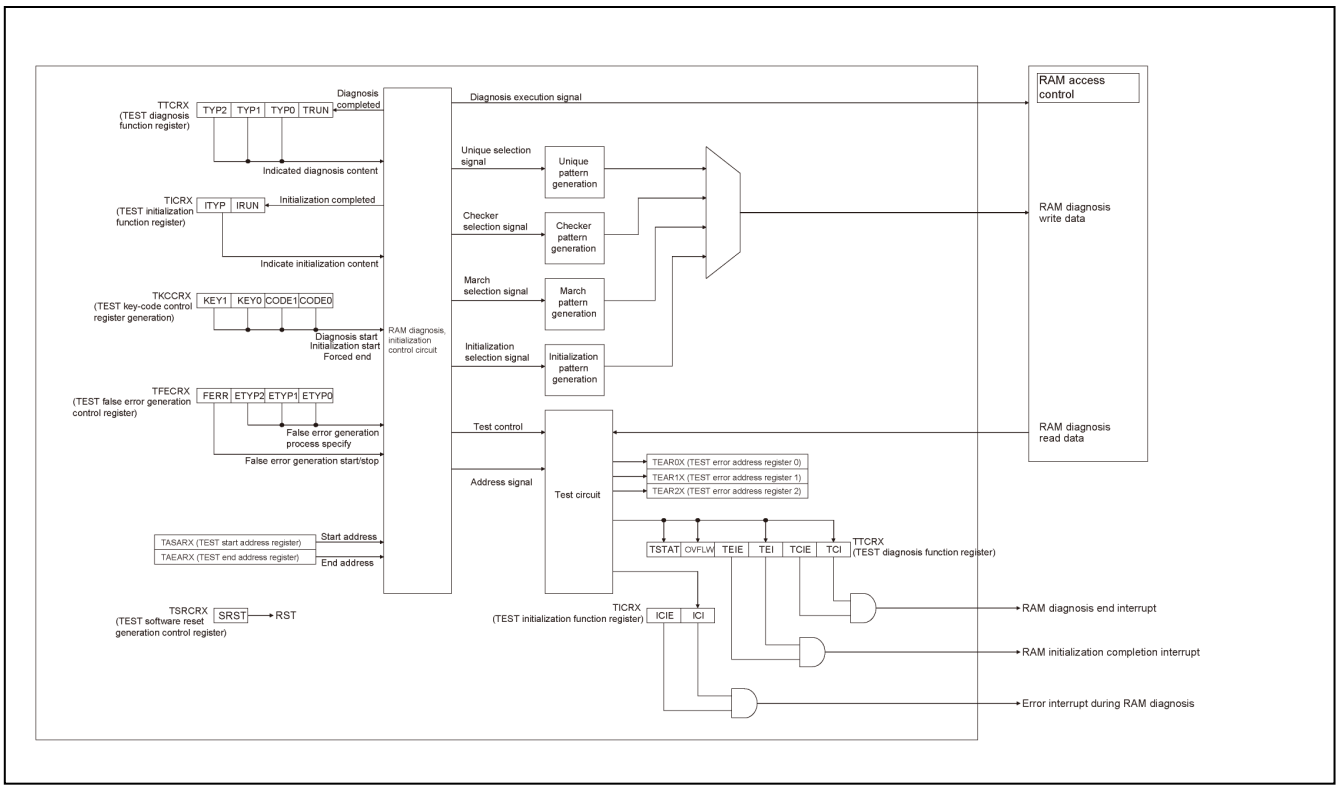
Test mode

In this mode, fake TEST error will be generated for software debugging.

40.3 Configuration

This section explains the configuration of RAM diagnosis function.

Figure 40-1. Block Diagram of XBS RAM Diagnosis Function (Configuration)



40.4 Registers

This section explains the registers of RAM diagnosis function.

Table 40-1. Register Map

Address	Registers				Register functions
	+0	+1	+2	+3	
0x300C	TEAR0X				TEST error address register 0 XBS RAM
0x3010	TEAR1X				TEST error address register 1 XBS RAM
0x3014	TEAR2X				TEST error address register 2 XBS RAM
0x3018	TAEARX		TASARX		TEST end address register XBS RAM TEST start address register XBS RAM
0x301C	TFECRX	TICRX	TTCRX		TEST fake error generation control register XBS RAM TEST initialization function register XBS RAM TEST diagnosis function register XBS RAM
0x3020	TSRCRX	Reserved	Reserved	TKCCRX	TEST software reset generation control register XBS RAM TEST key code control register XBS RAM

40.4.1 TEST Error Address Register 0 XBS RAM: TEAR0X

This section explains the bit structure of TEST Error Address Register 0 XBS RAM.

If an error occurs during RAM diagnosis for XBS RAM, TEST error address register 0 (TEAR0X) holds the relevant address.

TEAR0X: Address 300C$_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	TER2	TER1	TER0	Reserved				
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit29] TER2 to TER0: Diagnosis error factor specification bits

During RAM diagnosis for XBS RAM, these bits hold a diagnosis pattern for which the error occurred. D14 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D14 to D0 are invalid when no error is generated.
-	-	1	An error occurs during March diagnosis.
-	1	-	An error occurs during checker diagnosis.
1	-	-	An error occurs during unique diagnosis.

These bits are initialized (cleared to "000") by hardware, the RAM diagnosis start instruction being used as the trigger.

[bit28 to bit15] Reserved

Be sure to write "0" to these bits always.

[bit14 to bit0] D14 to D0: Error generation address bits

During RAM diagnosis for XBS RAM, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.
(Absolute address) = (0001_0000_H) + (Offset address set with TEAR0X + 2'b00)

40.4.2 TEST Error Address Register 1 XBS RAM: TEAR1X

This section explains the bit structure of TEST Error Address Register 1 XBS RAM.

Only when an error occurs in the address different from that held in TEAR0X during RAM diagnosis for XBS RAM, TEST error address register 1 (TEAR1X) holds that address.

TEAR1X: Address 3010_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	TER2	TER1	TER0	Reserved				
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit29] TER31 to TER29: Diagnosis error factor specification bits

During RAM diagnosis for XBS RAM, these bits hold a diagnosis pattern for which the error occurred. D14 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D14 to D0 are invalid with no error generated.
-	-	1	An error occurs during March diagnosis.
-	1	-	An error occurs during checker diagnosis.
1	-	-	An error occurs during unique diagnosis.

These bits are initialized (cleared to "000") by hardware, the RAM diagnosis start instruction being used as the trigger.

[bit28 to bit15] Reserved

Be sure to write "0" to these bits always.

[bit14 to bit0] D14 to D0: Error generation address bits

During RAM diagnosis, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.
(Absolute address) = (0001_0000_H) + (Offset address set with TEAR1X + 2'b00)

40.4.3 TEST Error Address Register 2 XBS RAM: TEAR2X

This section explains the bit structure of TEST Error Address Register 2 XBS RAM.

Only when an error occurs in the address different from that held in TEAR0X and TEAR1X during RAM diagnosis for XBS RAM, TEST error address register 2 (TEAR2X) holds that address.

TEAR2X: Address 3014_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	TER2	TER1	TER0	Reserved				
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	D14	D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit29] TER2 to TER0: Diagnosis error factor specification bits

During RAM diagnosis for XBS RAM, these bits hold a diagnosis pattern for which the error occurred. D14 to D0 are valid only when one of the bits is set to "1".

TER2	TER1	TER0	Function
0	0	0	D14 to D0 are invalid with no error generated.
-	-	1	An error occurs during March diagnosis.
-	1	-	An error occurs during checker diagnosis.
1	-	-	An error occurs during unique diagnosis.

These bits are initialized (cleared to "000") by hardware, the RAM diagnosis start instruction being used as the trigger.

[bit28 to bit15] Reserved

Be sure to write "0" to these bits always.

[bit14 to bit0] D14 to D0: Error generation address bits

During RAM diagnosis, these bits hold the address in which the error occurred. A valid value is indicated only when {TER2 to TER0} are not "000".

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.
(Absolute address) = (0001_0000_H) + (Offset address set with TEAR2X + 2'b00)

40.4.4 TEST Start Address Register XBS RAM: TASARX

This section explains the bit structure of TEST Start Address Register XBS RAM.

TEST start address register (TASARX) specifies the start address of RAM diagnosis and initialization for XBS RAM.

TASARX: Address 301A$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	ST14	ST13	ST12	ST11	ST10	ST9	ST8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] Reserved

Be sure to write "0" to this bit always.

[bit14 to bit0] ST14 to ST0: RAM diagnosis start address bits

These bits are used to specify the address from which the RAM diagnosis and initialization start for XBS RAM.

Note:

Setting of a value outside the XBS RAM area and a value that sets TASARX.ST14 to ST0 > TAEARX.ED14 to ED0 is disabled.

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.
(Absolute address) = (0001_0000$_H$) + (Offset address set with TASARX + 2'b00)

40.4.5 TEST End Address Register XBS RAM: TAEARX

This section explains the bit structure of TEST End Address Register XBS RAM.

TEST end address register (TAEARX) specifies the end address of RAM diagnosis and initialization for XBS RAM.

TAEARX: Address 3018_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	ED14	ED13	ED12	ED11	ED10	ED9	ED8
Initial value	0	1	1	1	1	1	1	1
Attribute	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
Initial value	1	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15] Reserved

Be sure to write "0" to this bit always.

[bit14 to bit0] ED14 to ED0: RAM diagnosis end address bits

These bits are used to specify the address with which the RAM diagnosis and initialization end for XBS RAM.

Note:

Setting of a value outside the XBS RAM area and a value that sets TASARX.ST14 to ST0 > TAEARX.ED14 to ED0 is disabled.

Note:

The above-mentioned address is an offset of the word length.
The absolute address is calculated by adding the base address to the offset address where lower two bits were added.
(Absolute address) = (0001_0000_H) + (Offset address set with TAEARX + 2'b11)

40.4.6 TEST Diagnosis Function Register XBS RAM: TTCRX

This section explains the bit structure of TEST Diagnosis Function Register XBS RAM.

The TEST diagnosis function register (TTCRX) specifies the RAM diagnosis content for XBS RAM, and holds the diagnosis result and its status.

TTCRX: Address $301E_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						TSTAT	OVFLW
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TEIE	TEI	TCIE	TCI	TTYP2	TTYP1	TTYP0	TRUN
Initial value	0	0	0	0	1	1	0	0
Attribute	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R/W	R/W	R,WX

[bit15 to bit10] Reserved

Be sure to write "0" to these bits always.

[bit9] TSTAT: RAM diagnosis error detection bit

TSTAT	Function
0	No error is detected with the RAM diagnosis.
1	An error is detected with the RAM diagnosis.

If an error occurs during RAM diagnosis for XBS RAM, this bit is set to "1".
This bit is initialized (cleared to "0") by hardware, the RAM diagnosis start instruction being used as the trigger.

[bit8] OVFLW: RAM diagnosis error overflow bit

OVFLW	Function
0	During the RAM diagnosis, an error occurs in three or less addresses.
1	During the RAM diagnosis, an error occurs in four or more addresses.

If a RAM diagnosis error for XBS RAM occurs in four or more addresses, this bit is set to "1".
This bit is initialized (cleared to "0") by hardware, the RAM diagnosis start instruction being used as the trigger.

[bit7] TEIE: Interrupt enable bit resulting from a diagnosis error

TEIE	Function
0	Prohibition of an interrupt resulting from a diagnosis error
1	Enabling of an interrupt resulting from a diagnosis error

This bit is used to enable an interrupt resulting from a RAM diagnosis error for XBS RAM.

"0": Prohibits an interrupt resulting from a RAM diagnosis error.

"1": Enables an interrupt resulting from a RAM diagnosis error. If TTCRX.TEI=1 is set and the RAM diagnosis ends, the interrupt signal (RAM diagnosis error interrupt) is output.

[bit6] TEI: Diagnosis error generation bit

TEI	Function
0	Read: No error occurrence during the RAM diagnosis Write: Flag clearing
1	Read: Error occurred during the RAM diagnosis Write: No influence on the operation

If TTCRX.TSTAT=1 is set when RAM diagnosis end for XBS RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set TTCRX.TSTAT=1 when RAM diagnosis is ended.
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit5] TCIE: Interrupt enable bit for a diagnosis end factor

TCIE	Function
0	Prohibition of an interrupt for the diagnosis end factor
1	Enabling of an interrupt for the diagnosis end factor

This bit is used to enable an interrupt for the RAM diagnosis end factor for XBS RAM.

"0": Prohibits an interrupt resulting from a RAM diagnosis end.

"1": Enables an interrupt resulting from a RAM diagnosis end. The interrupt signal (RAM diagnosis end interrupt) is output with TTCRX.TCI= 1.

[bit4] TCI: Diagnosis end bit

TCI	Function
0	Read: The RAM diagnosis has not completed yet. Write: Flag clearing
1	Read: The RAM diagnosis ended. Write: No influence on the operation.

If RAM diagnosis end for XBS RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set when a RAM diagnosis is ended. (It will not be set for forced termination by a key code)
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit3 to bit1] TTYP2 to TTYP0: RAM diagnosis content indication bit

These bits are used to set the RAM diagnosis type for XBS RAM to be executed.
The RAM diagnosis types are executed in the following order.

1. Unique (unique data is {Address [3:0],{6{Address [7:0]}}})
2. Checker
3. March (all "0" -> all "1" are executed in that order.)

These bits are used to determine whether or not each type is executed.

TTYP2	TTYP1	TTYP0	Function
1	1	0	Execution of unique and checker
-	-	1	Execution of March
-	1	-	Execution of checker
1	-	-	Execution of unique

By default, the unique and checker diagnoses are executed (110_B). However, to change the RAM diagnosis content, be sure to specify this change before the RAM diagnosis operation start instruction.
If March is executed last, the RAM content is all "1".

[bit0] TRUN: RAM diagnosis operation status bit

TRUN	Function
0	The RAM diagnosis is stopping.
1	The RAM diagnosis is in progress.

This bit is used to set or hold the RAM diagnosis status for XBS RAM.

"1": Set when a RAM diagnosis is started by the key code setting.
"0": Set when all diagnoses are complete or forcibly terminated by the key code.

40.4.7 TEST Initialization Function Register XBS RAM: TICRX

This section explains the bit structure of TEST Initialization Function Register XBS RAM.

The TEST initialization function register (TICRX) specifies the RAM initialization content, and holds the initialization result and its status for XBS RAM.

TICRX: Address 301D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				ICIE	ICI	ITYP	IRUN
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R(RM1),W	R/W	R,WX

[bit7 to bit4] Reserved

Be sure to write "0" to these bits always.

[bit3] ICIE: Interrupt enable bit for a RAM initialization end factor

ICIE	Function
0	Prohibition of an interrupt for a RAM initialization end factor
1	Enabling of an interrupt for a RAM initialization end factor

This bit is used to enable an interrupt for the RAM initialization end factor for XBS RAM.

"0": Prohibits an interrupt resulting from a RAM initialization end.

"1": Enables an interrupt resulting from a RAM initialization end. The interrupt signal (RAM initialization complete interrupt) is output with TICRX.ICI= 1.

[bit2] ICI: RAM initialization end bit

ICI	Function
0	Read: The RAM initialization does not end. Write: Flag clearing.
1	Read: The RAM initialization ended. Write: No influence on the operation.

If RAM initialization end for XBS RAM is detected, this bit is set to "1".
When "0" is written in this bit, it is cleared to "0". However, writing "1" to this bit is invalid and this bit holds the previous value.

"1": Set when a RAM initialization is ended. (It will not be set for forced termination by a key code)
"0": Set when "0" is written.

Note:

At read access of the read-modify-write instruction, "1" is always read.

[bit1] ITYP: RAM initialization content indication bit

ITYP	Function
0	Initialization to All "0"
1	Initialization to All "1"

This bit is used to set the type to be executed during RAM initialization for XBS RAM.

"0": Initializes to all "0".
"1": Initializes to all "1".

[bit0] IRUN: RAM initialization operation status bit

IRUN	Function
0	RAM Initialization is stopping.
1	RAM Initialization is in progress.

This bit is used to set or hold the RAM initialization status for XBS RAM.

"1": Set when RAM initialization is started by the key code setting.
"0": Set when all initialization is completed or forcibly terminated by the key code.

40.4.8 TEST Software Reset Generation Control Register XBS RAM: TSRCRX

This section explains the bit structure of TEST Soft Reset Generation Control Register XBS RAM.

The TEST software reset generation control register (TSRCRX) specifies the generation of the software reset for initializing internal circuits for XBS RAM's RAM diagnosis.

TSRCRX: Address 3020_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SRST	Reserved						
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

[bit7] SRST: Software reset enabling bit

SRST	Function
0	Prohibition of a software reset
1	Enabling of a software reset

This bit is used to enable a software reset for the internal circuit for RAM diagnosis for XBS RAM.

This bit reads out "0".

"1": Reset pulses occur for 4τ only and the internal circuit for RAM diagnosis except this register is reset.

τ: Peripheral clock

[bit6 to bit0] Reserved

Be sure to write "0" to these bits always.

40.4.9 TEST Fake Error Generation Control Register XBS RAM: TFECRX

This section explains the bit structure of TEST Fake Error Generation Control Register XBS RAM.

TEST fake error generation control register (TFECRX) generates a fake error in RAM diagnosis operation for XBS RAM. You can specify RAM diagnosis operations for which you want to generate an error.

TFECRX: Address 301C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				FERR	ETYP2	ETYP1	ETYP0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved

Be sure to write "0" to these bits always.

[bit3] FERR: Fake error enable bit for RAM diagnosis

FERR	Function
0	Prohibition of a fake error (normal operation)
1	Enabling of a fake error

This bit is used to enable a fake error for RAM diagnosis for XBS RAM.

"0": Prohibits a fake error. (normal operation)

"1": Enables a fake error. Data write including intentional error is enabled following ETYP2 to ETYP0.

[bit2 to bit0] ETYP2 to ETYP0: Fake error process specification bits

These bits are used to specify a process to generate a fake error.

ETYP2	ETYP1	ETYP0	Process to generate a fake error
-	-	1	March diagnosis
-	1	-	Checker diagnosis
1	-	-	Unique diagnosis

40.4.10 TEST Key Code Control Register XBS RAM: TKCCRX

This section explains the bit structure of TEST Key Code Control Register XBS RAM.

The TEST key code control register (TKCCRX) is used to start or forcibly terminate the RAM diagnosis or initialization for XBS RAM.

TKCCRX: Address 3023_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	KEY1	KEY0	Reserved				CODE1	CODE0
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W

[bit7, bit6] KEY1, KEY0: Key code control bits

Key code control bits. Set the operation instruction content to CODE[1:0] (no change during operation) and perform the operation.
The procedure is:

1. 00 -> 01 -> 10 -> 11: Write in this order.
2. Same values in CODE[1:0]
3. Different operations (access or read other registers for RAM diagnosis, or continuous write in the different order other than the above) within the procedure will be invalid.

Note:

The key code process will be continued even if any access to the registers in the RAMECC is made in the procedure.

[bit5 to bit2] Reserved

Be sure to write "0" to these bits always.

[bit1, bit0] CODE1, CODE0: RAM diagnosis/initialization control bits

These bits specify operational direction for the key code procedure above.

CODE1, CODE0	Function
00	Forced termination
01	Initialization start
10	Diagnosis start
11	Setting prohibited

If this value is changed or set to "11" during operating the key code above, the key code procedure itself will be invalid.

40.5 Operation

This section explains the operation of RAM diagnosis.

40.5.1 RAM Diagnosis
40.5.2 RAM Initialization
40.5.3 Interrupt-Related Register
40.5.4 RAM Diagnosis Fake Error Generation Procedure
40.5.5 Number of Required Cycles
40.5.6 Note

40.5.1 RAM Diagnosis

This section explains the RAM diagnosis.

XBS RAM diagnosis is performed only in the following order.

1. Unique (unique data is {Address [3:0],{6{Address [7:0]}}})
2. Checker
3. March (all "0" -> all "1" are executed in that order.)

The RAM diagnosis is performed following the settings in the TTYP[2:0] bits of the TEST diagnosis function register (TTCRX). By default, unique and checker are executed.

The coverage of the RAM diagnosis for XBS RAM is specified by the TEST start address register (TASARX) and TEST end address register (TAEARX).

Following procedure is required for RAM diagnosis for XBS RAM.

1. Before start diagnosing, read TRUN of the TEST diagnosis function register (TTCRX) and IRUN of the TEST initialization function register (TICRX), and check that they are "0".

In the case where TTCRX.TRUN or TICRX,IRUN is not "0":

- ☐ Wait for TTCRX.TRUN="0", then clear TTCRX.TCI.
- ☐ Wait for TICRX.IRUN="0", then clear TICRX.ICI.

2. Write continuously "02_H" -> "42_H" -> "82_H" -> "$C2_H$" to the TEST key code control register (TKCCRX), then start diagnosing.

When all RAM diagnosis types for XBS RAM are completed, TRUN bit of the TEST diagnosis function register (TTCRX) becomes "0" to finish the RAM diagnosis. The results of the diagnosis is retained in the TEST error address register 0 to 2 (TEAR0X to TEAR2X) and TEST diagnosis function register (TTCRX). The RAM holds the diagnosis data.
In addition, write continuously "00_H" -> "40_H" -> "80_H" -> "$C0_H$" to the TEST key code control register (TKCCRX) to terminate the RAM diagnosis for XBS RAM forcibly. RAM diagnosis ends even if it is in progress. In this case, the diagnosis result is not reliable.

40.5.2 RAM Initialization

This section explains the RAM initialization.

Only either of the following RAM initialization operation types for XBS RAM is specified with the ITYP bit of the TEST initialization function register (TICRX):

- Write all "0" (default)
- Write all "1"

ECC area has the values depending on the written values.
The coverage of the RAM initialization for XBS RAM is specified by the TEST start address register (TASARX) and TEST end address register (TAEARX).

Following procedure is required for XBS RAM's RAM diagnosis.

1. Before start diagnosing, read TRUN of the TEST diagnosis function register (TTCRX) and IRUN of the TEST initialization function register (TICRX), and check that they are "0".

In the case where TTCRX.TRUN or TICRX,IRUN is not "0":

- ☐ Wait for TTCRX.TRUN="0", then clear TTCRX.TCI.
- ☐ Wait for TICRX.IRUN="0", then clear TICRX.ICI.

2. Write continuously "01_H" -> "41_H" -> "81_H" -> "$C1_H$" to the TEST key code control register (TKCCRX), then start diagnosing.

When RAM initialization is completed, IRUN bit of the TEST initialization function register (TICRX) becomes "0" to finish the RAM initialization.

In addition, write continuously "00_H" -> "40_H" -> "80_H" -> "$C0_H$" to the TEST key code control register (TKCCRX) to terminate the RAM initialization forcibly. RAM initialization ends even if it is in progress. In this case, the initialization results is not guaranteed.

40.5.3 Interrupt-Related Register

This section explains the interrupt-related register.

To generate an interrupt, write "1" to the interrupt generation enabling bits (TEIE, TCIE, and ICIE) according to the purposes, and set the RAM diagnosis interrupt vector and RAM diagnosis interrupt level.

Interrupt factor	Interrupt vector	Interrupt level
TTCRX.TEI (RAM diagnosis error interrupt)	#35($000FFF70_H$)	ICR19(0453_H)
TTCRX.TCI (RAM diagnosis end interrupt)	#35($000FFF70_H$)	ICR19(0453_H)
TICRX.ICI (RAM initialization complete interrupt)	#35($000FFF70_H$)	ICR19(0453_H)

For details of the interrupt levels and interrupt vectors, see "Chapter: Interrupt Control (Interrupt Controller)."
The interrupt request flags (TEI, TCI, ICI) are not automatically cleared. So, to clear them, use software before return from interrupt processing. (Write "0" in the TEI, TCI, and ICI bits.)

40.5.4 RAM Diagnosis Fake Error Generation Procedure

This section explains the RAM diagnosis fake error generation procedure.

This function intentionally generates fake errors for software debugging.
Set the RAM diagnosis fake error generation for XBS RAM as following procedure:

1. Specify the error type with the TEST fake error generation control register (TFECRX).

   (1) Set a diagnosis pattern to the TFECRX.ETYP[2:0] to generate a fake error.
   (2) Specify a diagnosis pattern to generate a fake error by writing TFECRX.FERR="1".

2. Set the diagnosis start with the TEST diagnosis function register (TTCRX).

   (1) Set a diagnosis pattern to operate with the TTCRX.TTYP[2:0].

   (2) Write continuously four times, "02_H" -> "42_H" -> "82_H" -> "$C2_H$", to the TEST key code control register (TKCCRX), then start diagnosis pattern (see "40.5.1 RAM Diagnosis").

40.5.5 Number of Required Cycles

This section explains the number of required cycles.

The following shows the estimation of the cycle count required for various RAM diagnosis and initialization for XBS RAM.

XBS RAM: 24 kByte = 6k word address (MB91F552)

(1) "RAM diagnosis (unique)"

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

The processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses, and the entire number of cycles is as follows:

(	1	+	1	+	1	) ×	6144 (6k)	+ 1	=	18433
	Write		Read 1		Read 2		Word			Total

(2) "RAM diagnosis (checker)"

- Write 1 (1 cycle): W1
- Read 1 (1 cycle): R1

The processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses. To perform the partial write function diagnosis, five write processes and four read processes are provided for each word address. So, the following is obtained.

- Write 2 (1 × 5 cycles): W2
- Read 2 (2 × 4 cycles): R2

Moreover, the same processing is repeated with data different from above data. The entire number of cycles is as follows:

((	1	+	1	) ×	6144 (6k)	+1	+	5	+	8	) ×	2	=	24604
	W1		R1		Word			W2		R2		Repetition		Total

(3) "RAM diagnosis (March)"

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

This diagnosis has 3 writes and 2 reads per a word address so that the processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses. Moreover, the same processing is repeated with data different from above data. So, the entire number of cycles is as follows:

(	3	+	4	) ×	6144 (6k)	×	2	=	86016
	Write		Read		Word		Repetition		Total

(4) "RAM initialization"

- Write (1 cycle)

The processes above exist for each word address, and a set of these processes exists for a portion equivalent to all word addresses. The entire number of cycles is as follows:

1	×	6144 (6k)	=	6144
Write		Word		Total

The time required for 24 kByte RAM diagnosis of 2 MHz and 80 MHz operations is obtained as follows:

Table 40-2. Time Required for RAM Diagnosis and Initialization for 24 kByte

	Unique	Checker	March	Initialization	Total
Number of cycles	18433	24604	86016	6144	135197
2 [MHz] (=500 [ns])	9216.5 [µs]	12302 [µs]	43008 [µs]	3072 [µs]	68 [ms]
80 MHz (=12.5 [ns])	230.4 [µs]	307.6 [µs]	1075.2 [µs]	76.8 [µs]	1.7 [ms]

Moreover, the time required for the diagnosis with initial register values is obtained as follows:

Table 40-3. Time Required of Diagnosis (Initial Setting) after Power-on Reset Is Released

	Unique	Checker	Total
Number of cycles	18433	24604	43037
2 [MHz] (=500 [ns])	9216.5 [µs]	12302 [µs]	21.5 [ms]

40.5.6 Note

This section explains the note.

Accessing to RAM is prohibited during RAM diagnosis or during initialization.
RAM diagnosis and initialization cannot be used during debugging with the on-chip debugger (OCD).

While performing diagnostic or initialization process, start setting is ignored and the currently running process will continue.
To start anything, perform steps below to make sure that no operation is being performed.

1. Make sure that all of TTCR:TRUN, TICR and IRUN are "0".
2. Start the key code operation with TKCCR (TKCCRX) for diagnosis or initialization.

Forced termination can be performed by key code operation. Perform as shown below.

Input "00_H" -> "40_H" -> "80_H" -> "$C0_H$" to TKCCR continuously.

41. Timing Protection Unit

This chapter explains the Timing Protection Unit.

41.1 Overview

41.2 Features

41.3 Configuration

41.4 Registers

41.5 Operation

41.1 Overview

This section explains the overview of the Timing Protection Unit.

Timing Protection Unit (TPU) is a timer that OS uses for the ensuring safety of the system by the watch of the time of Task/ISR. The control target that OS supervises is as follows.

- Resource lock time
- Global interrupt lock time
- Task/ISR dead line
- Task/ISR runtime
- Inter-arrival time (Interrupt Frequency)

The control registers can be accessed only in a privileged mode because the built-in timers of TPU are controlled by OS, and the controls are all programmable.

41.2 Features

This section explains features of the Timing Protection Unit.

- Count with system clock (HCLK)
- Built-in timers: Max 8 timers
- 24-bit up-counter
- Two operational modes of Normal/Overflow
  - ☐ Normal Mode: When a count value is exceeded to the setting value, interrupt is generated.
  - ☐ Overflow Mode: The interrupt is generated by the counter overflow.
- Automatic restart function
- Global prescaler (division factor 1/1 to 1/64)
- Prescaler by timer (1, 1/2, 1/4, 1/16)
- Reading function of counter value
- Software control of Start/ Stop/ Continue
- Status display of each timer (stopped/active)
- Debug mode support
- Access protection function of TPU control register

41.3 Configuration

This section explains the configuration of the Timing Protection Unit.

There is no block diagram.

41.4 Registers

This section explains registers of the Timing Protection Unit.

For all registers, writing is permitted only for privileged mode/debugging access.
The area of 0x00000900-0x00009FF is TPU register area. The area not shown in the following is all reserved.

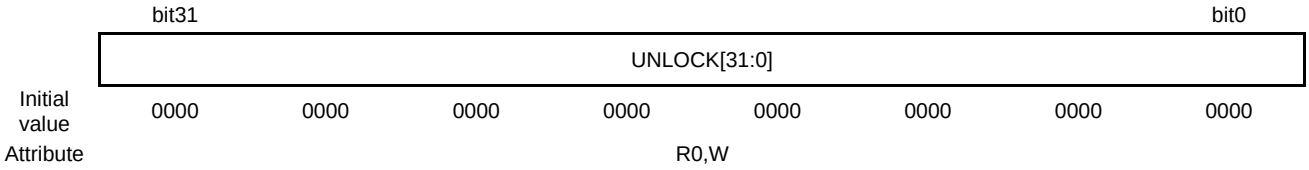
Table 41-1. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0900	TPUUNLOCK				TPU unlock register
0x0904	TPULST	Reserved	TPUVST	Reserved	TPU lock status register
0x0908	TPUCFG				TPU control register
0x090C	TPUTIR	Reserved	Reserved	Reserved	TPU timer interrupt status register
0x0910	TPUTST	Reserved	Reserved	Reserved	TPU timer status register
0x0914	TPUTIE	Reserved	Reserved	Reserved	TPU timer interrupt enable register
0x0918	TPUTMID				TPU module ID register
0x0930	TPUTCN00				TPU timer control register 0 ch.0
0x0934	TPUTCN01				TPU timer control register 0 ch.1
0x0938	TPUTCN02				TPU timer control register 0 ch.2
0x093C	TPUTCN03				TPU timer control register 0 ch.3
0x0940	TPUTCN04				TPU timer control register 0 ch.4
0x0944	TPUTCN05				TPU timer control register 0 ch.5
0x0948	TPUTCN06				TPU timer control register 0 ch.6
0x094C	TPUTCN07				TPU timer control register 0 ch.7
0x0950	TPUTCN10	Reserved	Reserved	Reserved	TPU timer control register 1 ch.0
0x0954	TPUTCN11	Reserved	Reserved	Reserved	TPU timer control register 1 ch.1
0x0958	TPUTCN12	Reserved	Reserved	Reserved	TPU timer control register 1 ch.2
0x095C	TPUTCN13	Reserved	Reserved	Reserved	TPU timer control register 1 ch.3
0x0960	TPUTCN14	Reserved	Reserved	Reserved	TPU timer control register 1 ch.4
0x0964	TPUTCN15	Reserved	Reserved	Reserved	TPU timer control register 1 ch.5
0x0968	TPUTCN16	Reserved	Reserved	Reserved	TPU timer control register 1 ch.6
0x096C	TPUTCN17	Reserved	Reserved	Reserved	TPU timer control register 1 ch.7
0x0970	TPUTCC0				TPU timer current count register ch.0
0x0974	TPUTCC1				TPU timer current count register ch.1
0x0978	TPUTCC2				TPU timer current count register ch.2
0x097C	TPUTCC3				TPU timer current count register ch.3
0x0980	TPUTCC4				TPU timer current count register ch.4
0x0984	TPUTCC5				TPU timer current count register ch.5
0x0988	TPUTCC6				TPU timer current count register ch.6
0x098C	TPUTCC7				TPU timer current count register ch.7

41.4.1 TPU Unlock Register: TPUUNLOCK (TPU Unlock Register)

The bit configuration of TPU unlock register is shown below.

TPUUNLOCK: Address 0900_H (Access: Word)



This register is used to specify access prohibition/permission to the TPU control register (TPUCFG, TPUTCN1n (n: timer channel number)).
It is required to prevent the illegal update of TPU control registers due to the malfunction of system.
Writing to this register is permitted only at the privileged mode. The readout value is always "0".
Be sure to keep access within 32-bit width (word) because Lock/Unlock is judged with 32-bit.

[bit31 to bit0] UNLOCK[31:0]: LOCK/UNLOCK value

If present value of UNLOCK is written to the register, access to the TPU control register is permitted.
To prohibit accessing, write the values other than a present value of UNLOCK.

41.4.2 TPU Lock Status Register: TPULST (TPU Lock Status Register)

The bit configuration of TPU lock status register is shown below.

TPULST: Address 0904_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							LST
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX

This register is used to indicate the lock status of TPU.
This register is read only, and writing to the register has no influence in operation.

[bit7to bit1] (Reserved): (Reserved bit)

When writing to these bits, "0" must be set. The readout value is always "0".

[bit0] LST (Lock Status): Lock status display

This bit indicates whether access to the TPU control register is locked.

LST	Lock Status
0	Access permission
1	Access prohibition

41.4.3 TPU Access Violation Detection Register: TPUVST (TPU Access Violation Status Register)

The bit configuration of TPU access violation detection register is shown below.

TPUVST: Address 0906_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					IULST	ULVST	AVST
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R,W0

An illegal access to the TPU register is detected, and the factor is maintained. When an illegal access to the register is detected, the corresponding bit of the detected violation factor is set, and it is processed as an illegal instruction exception. Writing to this register is permitted only in the privileged mode.

[bit7to bit3] (Reserved): (Reserved bit)

When writing to these bits, "0" must be set. The readout value is always "0".

[bit2] IULST (Illegal Unlock Access Status): Illegal unlock operation detection

When an illegal unlock access is detected, this bit becomes "1". Writing to this bit is effective only if the value is "0".
When a value other than the value set for UNLOCK is written in the TPUUNLOCK register in privileged mode when TPU control register access is prohibited (TPULST.LST=1)(including cases other than word access), an illegal unlock operation is detected.

[bit1] ULVST (Unlock Access Violation Status): Control register access violation detection while access prohibiting

When writing in TPU control register (TPUCFG, TPUTCN1n) is detected while prohibiting the TPU control register access, this bit becomes "1". Writing to this bit is effective only if the value is "0".
When there is a write operation to TPUCFG, TPUTCN1n in privileged mode when TPU control register access is prohibited (TPULST.LST=1), an illegal access is detected.

[bit0] AVST (Access Violation Status): Access violation detection

When the access violations other than IULST and ULVST are detected, this bit becomes "1". Only when "0" is written, it becomes effective. It concretely becomes a register access by the instruction fetch.

41.4.4 TPU Control Register: TPUCFG (TPU Configuration Register)

The bit configuration of TPU control register is shown below.

TPUCFG: Address 0908_H (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							DBGE
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	GLBPSE	Reserved	GLBPS[5:0]					
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							INTE
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W

It is a register that controls the entire TPU.

[bit31 to bit25] (Reserved): (Reserved bit)

These bits are reserved bits. Be sure to write "0". The readout value is always "0".

[bit24] DBGE (Debug Mode Enable): Debug mode transition

This bit is used to control transition to debug mode.
When debug mode is permitted, all timers stop operating. Each timer restarts operation when coming off debug mode.

DBGE	Debug Mode
0	All timer operation permission (normal mode)
1	All timer operation suppression (debug mode)

[bit23] GLBPSE (Global Prescaler Enable): Global prescaler operation enable

The operation of global prescaler is controlled. When the operation is prohibited, all timers do not perform count operation.

GLBPSE	Global Prescaler
0	Operation prohibition
1	Operation permission

[bit22] (Reserved): (Reserved bit)

This is a reserved bit. Be sure to write "0". The readout value is always "0".

[bit21to bit16] GLBPS[5:0] (Global Prescaler Bits): Global prescaler frequency dividing setting

These bits are used to specify the value of dividing frequency of the clock that supplied to all timers in common. Update of the bits has to be done when TPUCFG.GLBPSE=0 (timer operation disabled).
In TPU, the system clock (HCLK) is divided with global prescaler and the clock is supplied to each timer.
GLBPS[5:0] indicates the value of dividing frequency as it is.

GLBPS[5:0]	Global Prescaler Output
000000	HCLK/1
000001	HCLK/2
000010	HCLK/3
...	
111111	HCLK/64

[bit15 to bit1] (Reserved): (Reserved bit)

These bits are reserved bits. Be sure to write "0". The readout value is always "0".

[bit0] INTE (TPU Interrupt Enable): TPU interrupt enable

This bit is used to enable the interrupt request from TPU.

INTE	TPU Interrupt
0	Interrupt disable
1	Interrupt enable

41.4.5 TPU Timer Interrupt Request Register: TPUTIR (TPU Timer Interrupt Request Register)

The bit configuration of TPU timer interrupt request register is shown below.

TPUTIR: Address 0090C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IR[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

This register indicates interrupt request status from each timer in TPU. This register is read only.
Writing to the register causes no influence in operation.

[bit7 to bit0] IR[7:0] (Interrupt Request): Interrupt request

These bits indicate presence of the interrupt request for each channel. These bits show that there is an interrupt request factor regardless of timer interrupt enable register (TPUTIE). The requests are actually used as interrupt requests only when they are from channels where TPUTIE is effective.

Bit 0 to 7 corresponds to channel 0 to 7 respectively.

IRn	Interrupt Request
0	Ch.n No interrupt request
1	Ch.n Interrupt request

(n = 0 to 7)

41.4.6 TPU Timer Status Register: TPUTST (TPU Timer Status Register)

The bit configuration of TPU timer status register is shown below.

TPUTST: Address 00910_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TS[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

This register indicates the operation status of each timer in TPU. This register is read only. Writing to the register causes no influence in operation.

[bit7 to bit0] TS[7:0] (Timer Status): Timer operation status

These bits indicate timer operation status of each channel.
Bit 0 to 7 corresponds to channel 0 to 7 respectively.

TSn	Operation Status
0	Ch.n Stopped
1	Ch.n Operating

(n = 0 to 7)

41.4.7 TPU Timer Interrupt Enable Register: TPUTIE (TPU Timer Interrupt Register)

The bit configuration of TPU timer interrupt enable register is shown below.

TPUTIE: Address 00914$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IE[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used to enable interrupt of each timer in TPU.

[bit7to bit0] IE[7:0] (Interrupt Enable): Timer interrupt enable

These bits are used to enable timer interrupt request for each channel.
Bit 0 to 7 corresponds to channel 0 to 7 respectively.

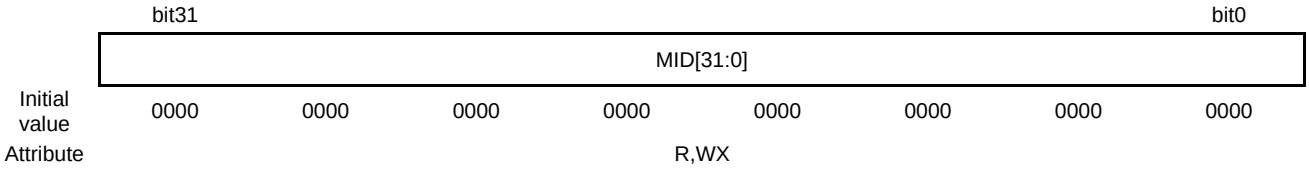
IEn	Interrupt Request
0	Ch.n Interrupt disable
1	Ch.n Interrupt enable

(n = 0 to 7)

41.4.8 TPU Module ID Register: TPUTMID (TPU Module ID Register)

The bit configuration of TPU module ID register is shown below.

TPUTMID: Address 00918_H (Access: Byte, Half-word, Word)



This register is used to indicate the TPU module ID. This register is read only.
Writing to the register causes no influence in operation.
It is used to identify the function of built-in TPU. In the OS, it is used to distinguish the type of TPU.

41.4.9 TPU Timer Control Register 00 to 07: TPUTCN00-07 (TPU Timer Control Register 00-07)

The bit configuration of TPU timer control register 00 to 07 is shown below.

TPUTCN00 toTPUTCN07: Address 00930_H to $0094C_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	START	STOP	CONT	IES	IEC	IRC	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W	R0,W0	R0,W0

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	ECPL[23:16]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ECPL[15:8]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ECPL[7:0]							
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

It is a control register of each timer.

[bit31] START (Start): Timer operation start

This bit is used to instruct to start the timer operation. The timer operation is started to write "1" to this bit. The readout value is always "0". When operation is started by this bit in the normal mode, the timer starts counting from "0".
When operation is started by this bit in the overflow mode, the timer starts counting from "0" or ECPL[23:0](TPUTCN1n.PL=1).
Writing "0" to the bit causes no influence in operation.

[bit30] STOP (Stop): Timer operation stop

This bit is used to instruct to stop the timer operation. The timer operation is stopped by writing "1" to this bit. The readout value is always "0".
Writing "0" to the bit causes no influence in operation.

[bit29] CONT (Continue): Timer operation restart

This bit is used to instruct to restart the timer operation. The timer operation is restarted by writing "1" to this bit. The readout value is always "0".
When the operation is restarted by this bit, operation is restarted from the count value that has stopped.
When START, STOP, and CONT bit are set at the same time, priority is judged in order of START > CONT > STOP.
Writing "0" to the bit causes no influence in operation.

[bit28] IES (Interrupt Enable Set): Interrupt enable bit set

This bit is used to instruct to set the timer interrupt enable bit. The interrupt enable bit (TPUTIE.IE[n]) is set by writing "1" in this bit. The readout value is always "0". Writing "0" to the bit causes no influence in operation.

[bit27] IEC (Interrupt Enable Clear): Interrupt enable bit clear

This bit is used to instruct to clear the timer interrupt enable bit. The interrupt enable bit (TPUTIE.IE[n]) is cleared by writing "1" in this bit. The readout value is always "0". Writing "0" to the bit causes no influence in operation.

[bit26] IRC (Interrupt Request Clear): Interrupt request clear

This bit is used to instruct the timer interrupt request clear. The interrupt request bit (TPUIR.IR[n]) is cleared by writing "1" to this bit.
The readout value is always "0". Writing "0" to the bit causes no influence in operation.

[bit25, bit24] (Reserved): (Reserved bit)

These bits are reserved bits. Be sure to write "0". The readout value is always "0".

[bit23 to bit0] ECPL[23:0] (End Count or Pre Load): Counter end value or pre-load value

The value used as the end value or pre-load value of the counter is set.
ECPL[23:0] is used as the end value of the counter in the normal mode.
ECPL[23:0] is used as pre-load value in the overflow mode.

41.4.10 TPU Timer Control Register 10 to 17: TPUTCN10-17 (TPU Timer Control Register 10-17)

The bit configuration of TPU timer control register 10 to 17 is shown below.

TPUTCN10 to TPUTCN17: Address 00950 to 0096C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			PL	FRT	TMOD	PS[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W	R/W

It is a control register of each timer.

[bit7 to bit5] (Reserved): (Reserved bit)

These bits are reserved bits. Be sure to write "0". The readout value is always "0".

[bit4] PL (Pre-Load): Pre-load instructions

This bit is used to specify pre-load of ECPL[23:0] when the timer operation is started. This bit is effective only when the timer is in the overflow mode.

PL	Pre-load
0	Pre-load invalid
1	Pre-load valid

[bit3] FRT (Free Running Timer): Free-run instructions

This bit is used to instruct free-run operation. It is effective in both normal mode/overflow modes.
After the interrupt is generated by the end value of the counter, the count is restarted from "0" automatically when this bit is made effective in the normal mode.
After the interrupt is generated by the counter overflow, the count is restarted from "0" (TPUTCN1n.PL=0) or ECPL[23:0] (TPUTCN1n.PL=1)automatically when this bit is made effective in the overflow mode.

FRT	Free-run
0	Free-run invalid
1	Free-run valid

[bit2] TMOD (TPU Mode): TPU operation mode

This bit is used to specify TPU operation mode. In the operation mode of the timer, there is the normal mode in which the count is incremented from "0" to ECPL[23:0], or the overflow mode in which the count is started from "0" (TPUTCN1n.PL=0)or ECPL[23:0] (TPUTCN1n.PL=1) and the counter overflow is detected.

TMOD	Timer Operation Mode
0	Normal mode
1	Overflow mode

[bit1, bit0] PS[1:0] (Individual Prescaler): Timer prescaler setting

The prescaler value of each timer is set. The output of global prescaler is input to each timer, and this input is divided and used as the operating frequency of each timer.

PS[1:0]	Prescaler
00	1/1
01	1/2
10	1/4
11	1/16

41.4.11 TPU Counter Value Register 0 to 7: TPUTCC0-7 (TPU Timer Current Count Register 0-7)

The bit configuration of TPU timer counter value register0 to 7is shown below.

TPUTCC0 to TPUTCC7: Address 00970 to 0098C$_H$ (Access: Byte, Half-word, Word)

	bit31	bit24	bit23					bit0
	Reserved		TCC[23:0]					
Initial value	0000	0000	0000	0000	0000	0000	0000	0000
Attribute	R0,W0	R0,W0				R,WX		

This register indicates the present counter value of the timer. This register is read only.

[bit31 to bit24] (Reserved): (Reserved bit)

These bits are reserved bits. Be sure to write "0". The readout value is always "0".

[bit23to bit0] TCC[23:0] (Timer Current Count): Timer count value

These bits indicate the present counter value.

41.5 Operation

This section explains the operations.

41.5.1 TPU Control Register Access Protection
41.5.2 Global Prescaler
41.5.3 Interrupt Control
41.5.4 Timer Operation
41.5.5 Free-run Function
41.5.6 Individual Prescaler Function
41.5.7 Debug Support Function
41.5.8 Operation Flow

41.5.1 TPU Control Register Access Protection

This section explains the TPU control register access protection.

The TPU register is permitted to be accessed only in the privileged mode because all TPU registers are the system registers. The illegal instruction exception (data access error) is generated if accessing it in the user mode.

The TPU register not only has a function for access protection as the system register, it also has a function for register access protection with the Lock code to prevent writing of the TPU control register as a result of malfunctioning.

The target registers of the access protection are the following two registers.

- TPU control register (TPUCFG)
- TPU timer control register 10 to 17 (TPUTCN10-17)

To make the TPU control register access protection function effective, write the values other than a present set value of UNLOCK[31:0] in the TPU unlock register (TPUUNLOCK). When the access protection function becomes effective, the LST bit of the TPU lock status register is set to indicate the lock state.

When writing it in the control register of the protection target, write the value set last time in UNLOCK[31:0]. It becomes TPULST.LST=0 and the unlock state when the lock is released.

After generating reset, the register access protection function is in the invalid state (TPULST.LST=0).

When the TPU control register access protection function is effective (TPULST.LST=1), and when the values other than UNLOCK[31:0] are written in TPU unlock register (TPUUNLOCK), the error reply is returned to AHB and the data access error is generated in CPU as an illegal access. Then, the violation factor is set in the TPU access violation detection register (TPUVST.IULST=1).

When the TPU control register access protection function is effective (TPULST.LST=1), and when there is a write request to the TPU control register (TPUCFG) and TPU timer control register (TPUTCFG1n), the data access error is generated in CPU as an illegal access.

Then, the violation factor is set in the TPU access violation detection register (TPUVST.ULVST =1).

Moreover, it is judged that the access by the instruction fetch is a malfunction and generates the illegal instruction exception. Then, it becomes TPUVST.AVST=1.

41.5.2 Global Prescaler

This section explains the global prescaler.

The global prescaler is a common prescaler used with all timers of TPU. The global prescaler divides HCLK (input clock of TPU) according to a set value of TPUCFG.GLBPS[5:0]. The value of division can be set by 1 to 64.

The global prescaler function controls operation by the TPUCFG.GLBPSE bit. The global prescaler function is enabled by writing "1" in TPUCFG.GLBPSE and it is disabled by writing "0" in TPUCFG.GLBPSE.

When TPUCFG.GLBPSE=0, the prescaler function is disabled and the clock of all timers doesn't become valid.

Please, update TPUCFG.GLBPS[5:0] after setting the global prescaler function disabled (TPUCFG.GLBPSE=0).

41.5.3 Interrupt Control

This section explains the interrupt control.

The generation of the interrupt request is controlled by the TPUCFG.INTE bit that controls the interrupt request by TPU and the TPUTIE.IE[n] bits that controls the interrupt of each timer. When an effective interrupt request by each timer exists in TPUCFG.INTE=1 ((TPUTIE.IE[n]=1)&(TPUTIR.IR[n]=1)), NMI is generated in TPU.

The interrupt factor of each channel can be confirmed with TPUTIR.IR[n].

Interrupt enable/disable of each channel is controlled with TPUTCN0n.IES/TPUTCN0n.IEC. If "1" is written in TPUTCN0n.IES, the interrupt is permitted and it becomes TPUTIE.IE[n]=1. If "1" is written in TPUTCN0n.IEC, the interrupt is prohibited and it becomes TPUTIE.IE[n]=0. If "1" is written to TPUTCN0n.IES and TPUTCN0n.IEC at the same time, it gives higher priority to "clear" than "set."

Please write "1" in TPUTCN0n.IRC when you clear the interrupt request of each channel.

41.5.4 Timer Operation

This section explains the timer operation.

Each timer is configured by 24-bit up-counter.

The timer has two operation modes of normal mode/overflow mode. The operation mode is controlled by the TPUTCN1n.TMOD bit. The operation mode becomes the normal mode when TPUTCN1n.TMOD=0, and becomes the overflow mode when TPUTCN1n.TMOD=1.

Normal Mode

The timer operates as the up-counter in the normal mode. When the counter value is equal or larger than TPUTCN0n.ECPL[23:0], the interrupt flag (TPUTIR.IR[n] (n: the timer channel)) is set. An actual interrupt request is generated when TPUTIE.IE[n]=1.

The timer starts counting from "0" by writing "1" to the TPUTCN0n.START bit.

During timer count, timer operation is indicated with TPUTST.TS[n]=1. If the interrupt flag is set (TPUTIR.IR[n]=1), the count is stopped and TPUTST.TS[n]=0 is indicated.

If "1" is written in the TPUTCN0n.STOP bit, the counter stops operating, and TPUTST.TS[n]=0 is indicated. The counter value at this time (When stopping) is maintained, and does not become "0".

If "1" is written in the TPUTCN0n.CONT bit, the counter operation is restarted, and becomes TPUTST.TS[n]=1.

Overflow Mode

When the overflow of the timer is detected in the overflow mode, the interrupt request flag (TPUTIR.IR[n]) is set. An actual interrupt request is generated when TPUTIE.IE[n]=1.

Pre-load to the counter is possible in the overflow mode. The value of TPUTCN0n.ECPL[23:0] is pre-loaded and the count is started after TPUTCN1n.PL=1 is set and operation starts. The timer starts counting from "0" if TPUTCN1n.PL=0.

41.5.5 Free-run Function

This section explains the free-run function.

Each timer can set free-run operation. The free-run operation is a function to restart the count automatically after the timer counts to the interrupt generation factor. In this case, because the counter operation doesn't stop, it keeps operating as timer operation status TPUTST.TS[n]=1.

The free-run function becomes effective if you set TPUTCN1n.FRT=1. The free-run function can use both normal mode/overflow modes.

At the normal mode, the count is restarted from "0".In the overflow mode, the count is restarted from "0" when TPUTCN1n.PL=0, and the count is restarted after the value of TPUTCN0n.ECPL[23:0] is loaded when TPUTCN1n.PL=1.

41.5.6 Individual Prescaler Function

This section explains the individual prescaler function.

TPU has the individual prescaler for each timer, and it can divide by 1, 2, 4, or 16 the global prescaler output. Individual prescaler is set with TPUTCN1n.PS[1:0].

41.5.7 Debug Support Function

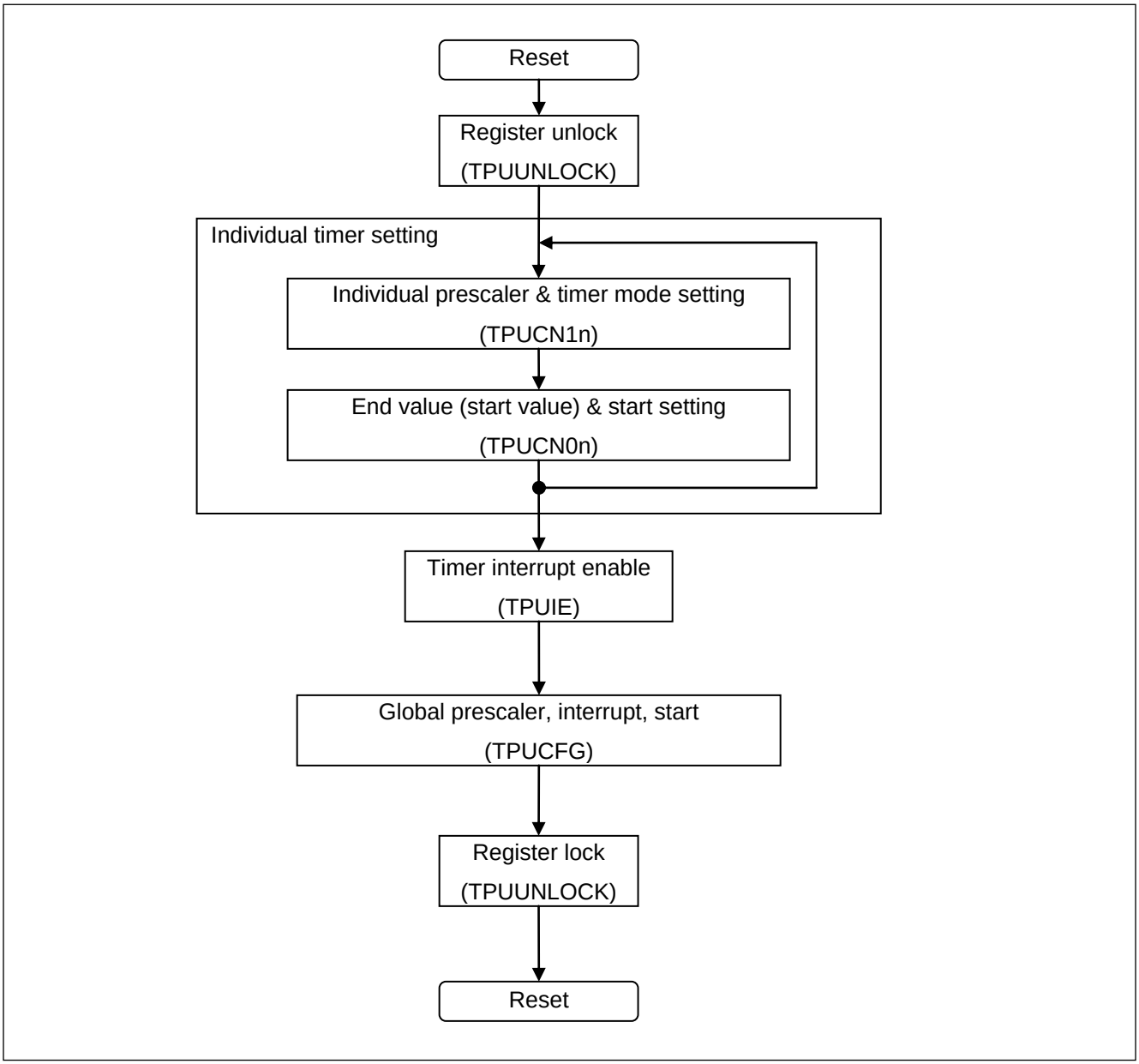
This section explains the debug support function.

TPU can be stopped by writing "1" in the debug mode control bit of the TPU control register (TPUCFG.DBGE=1) with software. The debug mode of TPU is released when "0" is written in TPUCFG.DBGE and operation is restarted.

41.5.8 Operation Flow

This section explains the operation flow.

Initialization Flow



42. Clock Monitor

This chapter explains the clock monitor.

42.1 Overview
42.2 Features
42.3 Configuration
42.4 Registers
42.5 Operation
42.6 Setting
42.7 Q&A
42.8 Notes

42.1 Overview

This section explains the overview of clock monitor.

The clock monitor is a macro that outputs internal clock signals to external pins. The clock monitor has a function for dividing the frequency of a clock signal before output to the pin, allowing clock signals to be used for synchronization of external circuits with MCU functions.

42.2 Features

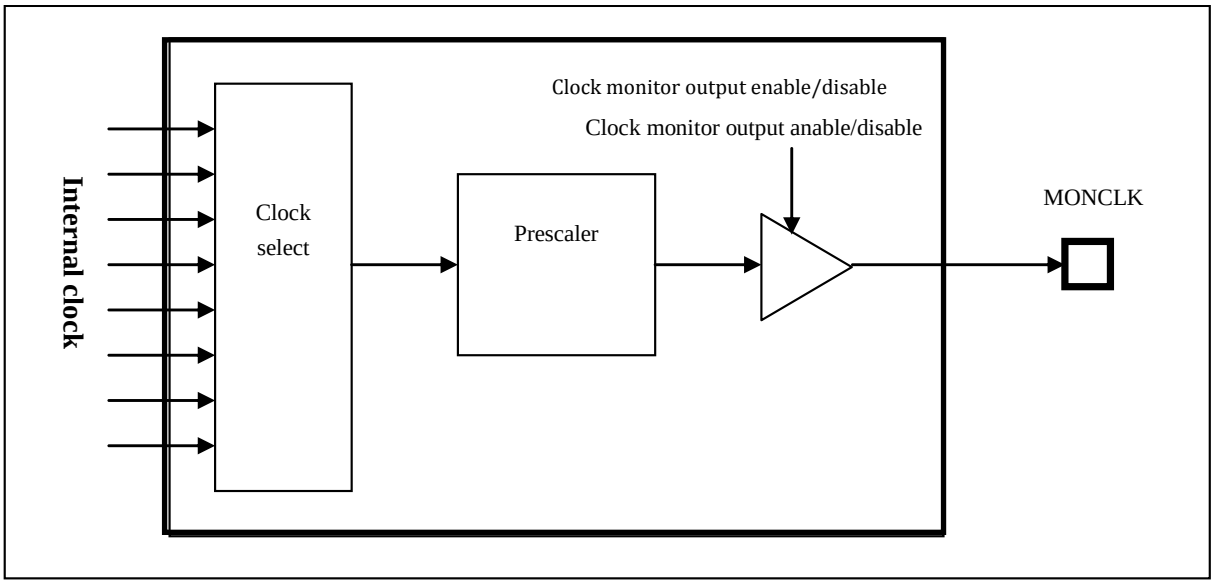
This section explains the features of clock monitor.

- Format: Divide the internal clock signal and output to a pin (MONCLK)
- Channels: 1
- Division ratio: CLK/1, CLK/2, CLK/3 to CLK/16
- Allows for glitch-less output
- Programmable mark level (outputs "L" or "H" before the clock output is enabled)
- Interrupts: None
- Stops clock output in stop mode and becomes high impedance.

42.3 Configuration

This section explains the configuration of clock monitor.

Figure 42-1. Configuration Diagram of Clock Monitor



MB No. (Number of Pin)	Pin Number of MONCLK
MB91F552(64pin)	11

42.4 Registers

This section explains the registers of clock monitor.

Table 42-1. Register Map

Address	Register				Register Function
	+0	+1	+2	+3	
0x04A8	Reserved	Reserved	CSCFG	CMCFG	Clock Monitor Configuration Registers

42.4.1 Clock Monitor Configuration Registers: CMCFG

This section shows the bit configuration of the clock monitor configuration registers.

CMCFG: Address 04AB$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CMPRE3	CMPRE2	CMPRE1	CMPRE0	CMSEL3	CMSEL2	CMSEL1	CMSEL0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit4] CMPRE3 to CMPRE0: Output frequency prescaler bits

Division ratio setting of selected source clock by CMSEL bits.

CMPRE3	CMPRE2	CMPRE1	CMPRE0	Clock Frequency Output to the MONCLK Pin
0	0	0	0	Source clock divided by 1 (Initial value)
0	0	0	1	Source clock divided by 2
0	0	1	0	Source clock divided by 3
0	0	1	1	Source clock divided by 4
0	1	0	0	Source clock divided by 5
0	1	0	1	Source clock divided by 6
0	1	1	0	Source clock divided by 7
0	1	1	1	Source clock divided by 8
1	0	0	0	Source clock divided by 9
1	0	0	1	Source clock divided by 10
1	0	1	0	Source clock divided by 11
1	0	1	1	Source clock divided by 12
1	1	0	0	Source clock divided by 13
1	1	0	1	Source clock divided by 14
1	1	1	0	Source clock divided by 15
1	1	1	1	Source clock divided by 16

[bit3 to bit0] CMSEL3 to CMSEL0: Output source clock selection bits

Selected source clock for output signal of MONCLK pin.

CMSEL3	CMSEL2	CMSEL1	CMSEL0	Clock Source Output to MONCLK pin
0	0	0	0	MONCLK output disabled (high impedance state) (initial value)
0	0	0	1	Main oscillation before CSV
0	0	1	0	CR oscillation
0	0	1	1	Main oscillation after CSV

CMSEL3	CMSEL2	CMSEL1	CMSEL0	Clock Source Output to MONCLK pin
0	1	0	0	High-speed comparator clock
0	1	0	1	Clock for PWM (PWMCLK)
0	1	1	0	PLL output for PWM
0	1	1	1	Reserved (Setting prohibited)
1	0	0	0	PLL output
1	0	0	1	SSCG output
1	0	1	0	PLL output after CAN prescaler (CAN system clock)
1	0	1	1	CCLK
1	1	0	0	HCLK
1	1	0	1	PCLK1 (Spread peripheral clock)
1	1	1	0	PCLK2 (Peripheral clock after spread/non spread selection)
1	1	1	1	Reserved (Setting prohibited)

CSCFG: Address $04AA_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	MONCKI	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W0	R,WX	R,WX	R/W	R/W0	R/W0	R/W0	R/W0

[bit7] Reserved

This bit is reserved. Always set this bit to "0" when writing.

[bit6, bi5] Reserved

These bits are reserved. Writing to these bits has no influence on operation.

[bit4] MONCKI: Clock monitor MONCLK inverter

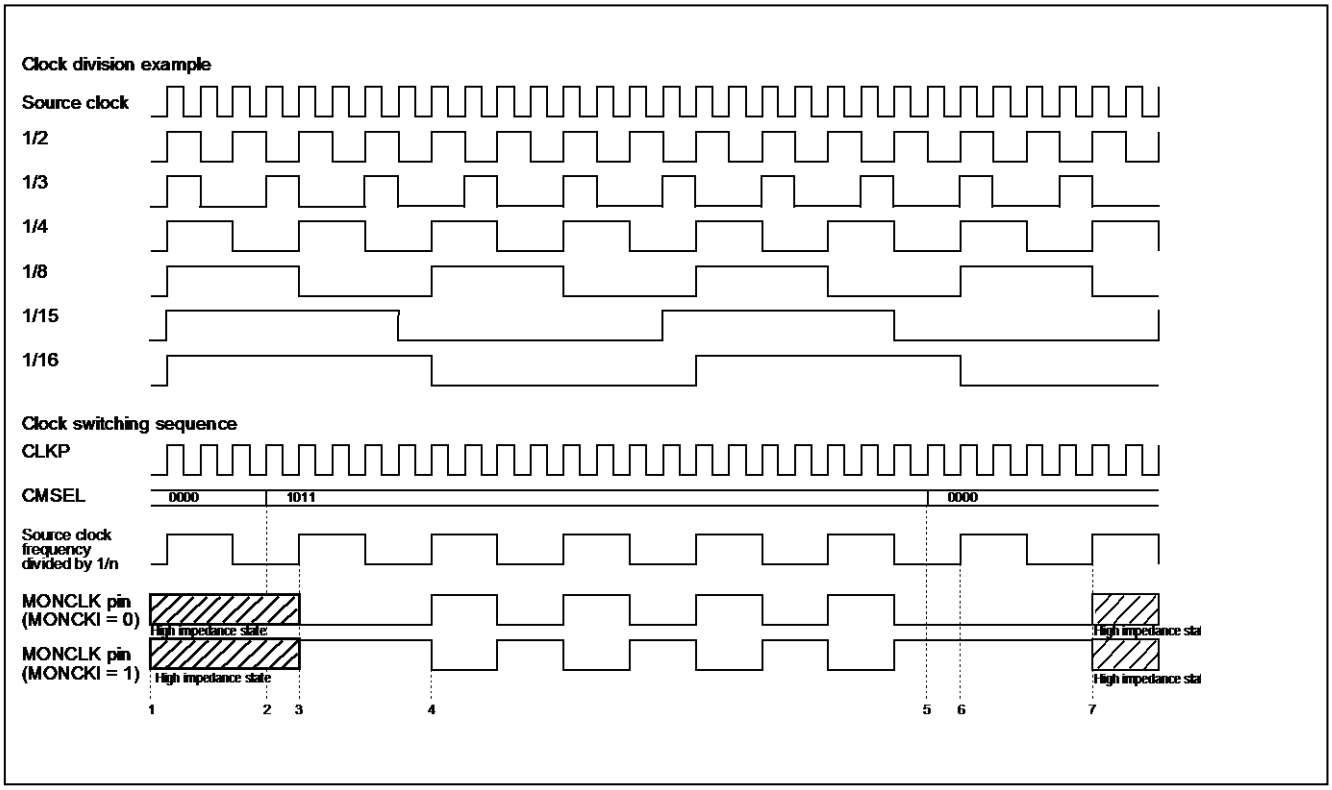
MONCKI	Function
0	MONCLK mark level is low level (initial value)
1	MONCLK mark level is high level

[bit3 to bit0] Reserved

This bit is reserved. Always set this bit to "0" when writing.

42.5 Operation

This section explains operation description of clock monitor.



1. The MONCLK pin is in the high impedance state.
2. CMSEL is set to the selected clock (prescaler) from 0000_B (no clock selected).
3. The MONCLK pin is set to the output "L" status (or output "H" if MONCKI is set to "1") for the duration of 1 internal (prescaled) clock.
4. After 1 period of the selected (prescaler) internal clock, MONCLK outputs the selected (prescaler) internal clock.
5. CMSEL is set to 0000_B (no clock selected) from the selected clock (prescaler).
6. The MONCLK pin is set to the output "L" status (or output "H" if MONCKI is set to "1") for the duration of 1 internal (prescaled) clock.
7. The MONCLK pin switches to the high impedance state.

42.6 Setting

This section explains the setting of clock monitor.

Setting	Setting Register	Setting Procedure
Setting of the prescaler value	Output frequency prescaler (CMCFG.CMPRE3 to CMPRE0)	Section 42.7.2
Setting of the source clock	Output source clock select (CMCFG.CMSEL3 to CMSEL0)	Section 42.7.1
Setting of the mark level	Clock monitor inverter (CSCFG.MONCKI)	Section 42.7.4
Enabling of clock monitor output	Output source clock select (CMCFG.CMSEL3 to CMSEL0)	Section 42.7.3

42.7 Q&A

This section explains the Q&A of clock monitor.

42.7.1 How to Configure the Output Pin (MONCLK)

42.7.2 How to Select the Output Frequency

42.7.3 How to Enable or Disable Clock Monitor Output

42.7.4 How to Set the Clock Output Mark Level

42.7.1 How to Configure the Output Pin (MONCLK)

Setting of the output pin (MONCLK) is shown.

Use the output source clock selection bits (CMCFG.CMSEL3 to CMSEL0).

42.7.2 How to Select the Output Frequency

Selection of the output frequency is shown.

Use the output frequency prescaler bits (CMCFG.CMPRE3 to CMPRE0).

Clock Division Ratio	When Output Frequency (Example) HCLK Is Selected		Output Frequency Prescaler (CMCFG.CMPRE3 to CMPRE0)
	HCLK=32 MHz	HCLK=40 MHz	
1/2	16 MHz	20 MHz	Set to 0001_B
1/3	10.7 MHz	13.3 MHz	Set to 0010_B
1/4	8 MHz	10 MHz	Set to 0011_B
1/8	4 MHz	5 MHz	Set to 0111_B
1/15	2.1 MHz	2.7 MHz	Set to 1110_B
1/16	2 MHz	2.5 MHz	Set to 1111_B

42.7.3 How to Enable or Disable Clock Monitor Output

Enabling or disabling clock monitor output is shown.

Use the output source clock selection bits (CMCFG.CMSEL3 to CMSEL0).

Operation Description	Output Source Clock Select Bits (CMCFG.CMSEL3 to CMSEL0)
Disable clock monitor output (Set the pin to the high impedance state)	Set to 0000_B
Enable clock monitor output	Set to 0001_B to 1111_B (However, the 0111_B and 1111_B settings are prohibited.)

42.7.4 How to Set the Clock Output Mark Level

Setting of the clock output mark level is shown.

Use the clock monitor MONCLK inverter bit (CSCFG.MONCKI).

42.8 Notes

This section explains the notes on clock monitor.

In order to achieve glitch-free switching, use the following procedure when changing the source clock (CMSEL3 to CMSEL0) or prescaler ratio (CMPRE3 to CMPRE0).

- The CMPRE3 to CMPRE0 registers can be written only when the CMSEL3 to CMSEL0 registers are "0_H".
- The CMPRE3 to CMPRE0 registers can be written only when "0_H" is written to the CMSEL3 to CMSEL0 registers during the same write access.
- At least 2 cycles of the monitor clock division are required during the 2 write accesses to CMPRE and CMCFG.
- When selecting another effective clock while a clock is already selected as the clock source (CMSEL is not "0_H"), first set CMSEL to "0_H" and check that CMSEL returns "0_H" on read before writing the target clock setting value to CMSEL.
- If the clock selected as the monitor clock is stopped during monitoring, rewriting to any registers have no effect until the selected clock is started again or the unit is reset.

(Access example)

1. Access

   CNCFG.CMSEL = 0

   CMCFG.CMPRE = Prescaler

2. Access

   CMCFG.CMSEL = Clock

The CSCFG.MONCKI flag can also be written in the same procedure as above only when CMSEL3 to CMSEL0 are "0_H".

43. PWM

This chapter explains the PWM.

43.1 Overview
43.2 Features
43.3 Configuration
43.4 Registers
43.5 Explanation of Operation
43.6 Notes

43.1 Overview

This section explains the overview of the PWM module.

The PWM module is used to obtain pulse width modulation (PWM) output. By setting the cycle, duty, and phase in a software program, the PWM module can easily be adapted to a wide range of applications.

43.2 Features

This section explains the features of the PWM.

PWM division clock

Select one of the following 8 types.

The PWM clock divided by 1, 2, 4, 8, 16, 32, 64, or 128 is output.

PWM cycle

- Setting range = duty value to 65535 (specified by a 16-bit register)
- Cycle = PWM division clock × (PWMCCB register value + 1): When the normal waveform is selected
  PWM division clock × (PWMCCB register value × 2): When the center aligned waveform is selected

(Example) PWM division clock = 200 MHz (5 ns), PWMCCB value = 63999

Cycle = 5 ns × (63999 + 1) = 320 us

PWM duty

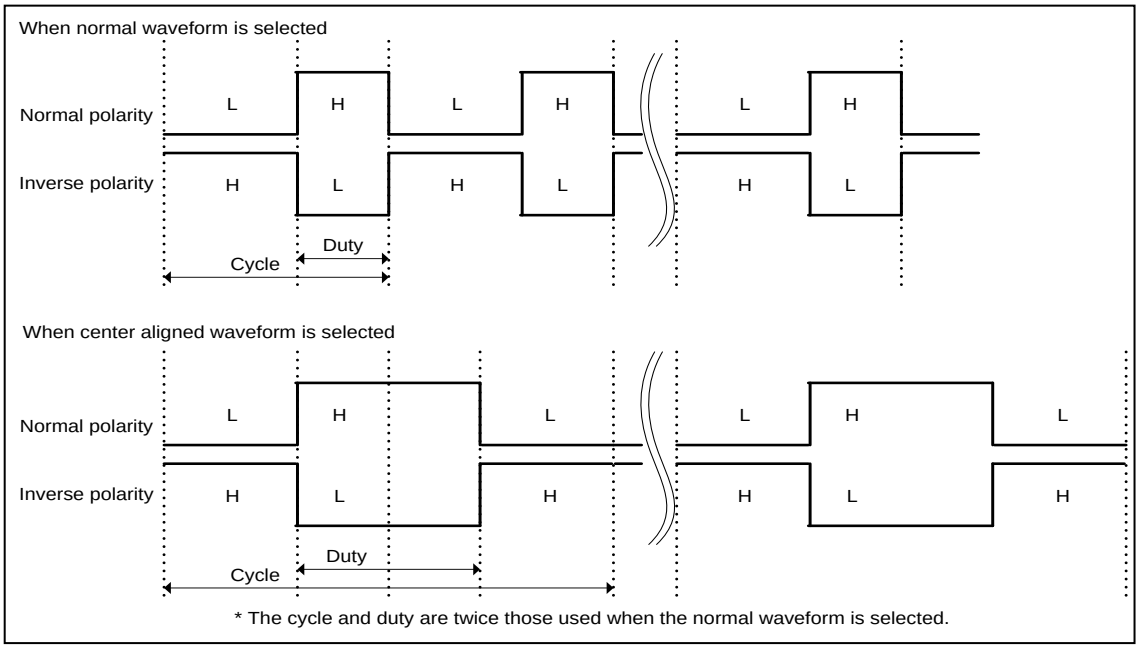
- Setting range = 0 to cycle value (specified with a 16-bit register)
- Duty = PWM division clock × (PWMCDB register value + 1): When the normal waveform is selected
  PWM division clock × (PWMCDB register value × 2): When the center aligned waveform is selected

PWM phase

- Setting range = 0 to cycle value (specified with a 16-bit register)
- Phase = PWM division clock × (PWMCPB register value)

PWM output waveform

- 3 PWM generators capable of 2-channel output (a total of 6 PWM output channels) are supported.
- Each PWM generator is capable of complementary output or independent output.
- Either normal waveform or center aligned waveform can be selected.



Interrupts

- The following 7 types of interrupts are used.
  - ☐ Special event interrupt 0/1
  - ☐ 0 detection interrupt 0/1
  - ☐ Compare clear interrupt 0/1
  - ☐ Soft overwrite (SOW) interrupt 0 to 2
  - ☐ Fault interrupt 0 to 5
  - ☐ Capture interrupt 0 to 2
  - ☐ Trigger interrupt 0 to 3

Master clock generation

- Cycle
  - ☐ Setting range = duty value to 65535 (specified with a 16-bit register)
- Duty
  - ☐ Setting range = 0 to cycle value (specified with a 16-bit register)
- Parallel operation is possible using the SYNCIN and SYNCOUT pins.
- One of 8 filter widths can be selected for the SYNCIN pin.

Fault functions

- One of the following 2 operations to be performed if a fault occurs can be set.
  - ☐ Fault operation: The PWM output state is selected from high, low, inverted, and unchanged.
  - ☐ Fault operation (preferred): The PWM output state is selected from high, low, and unchanged.
- Each fault function can be released from the base point of the PWM edge at an arbitrary timing.
- The time from the base point of the PWM edge to the occurrence of a fault can be measured.

Soft overwrite function

- The PWM output state can be selected from high, low, and unchanged, according to register settings (PWMSOWCONn.OSL1, 0). (n=0, 1, 2)
- The PWM output changes to the state set in a register (PWMSOWCONn.OSL1 and 0) at an arbitrary timing from the base point of the PWM edge. (n=0, 1, 2)

Blanking function

The result (fault) from the comparator can be blanked (masked). The blanking time can be set with a register.

A/D converter trigger generation

12 types of A/D converter triggers can be generated: 4 types of triggers invoked at an arbitrary timing from the base point of the PWM edge, 6 types of triggers invoked upon occurrence of a fault, and 2 types of special event triggers.

Dead time function

One of the following 2 types of dead time can be added to the PWM output.

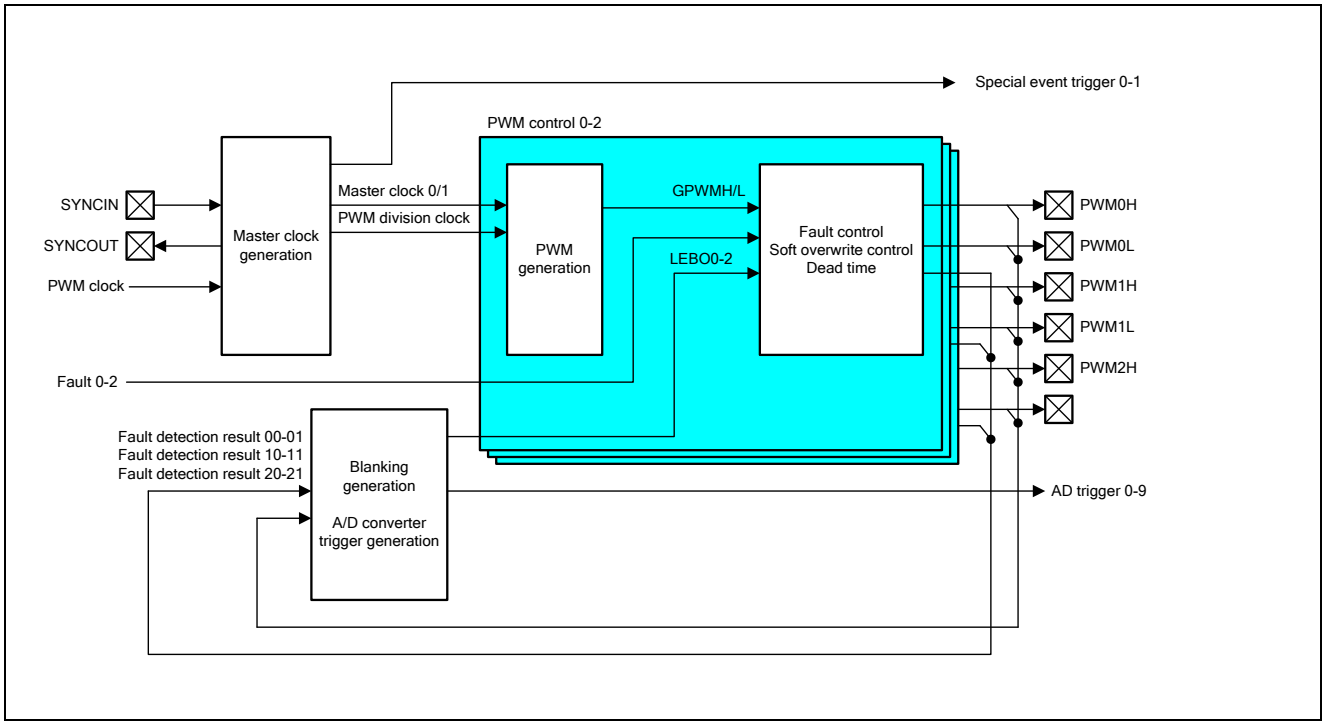
- Normal mode: The dead time is added to the rising edge of the PWM output.
- Extended mode 1: The dead time is added to the rising and falling edges of the PWM output.

43.3 Configuration

This section explains the configuration of the PWM.

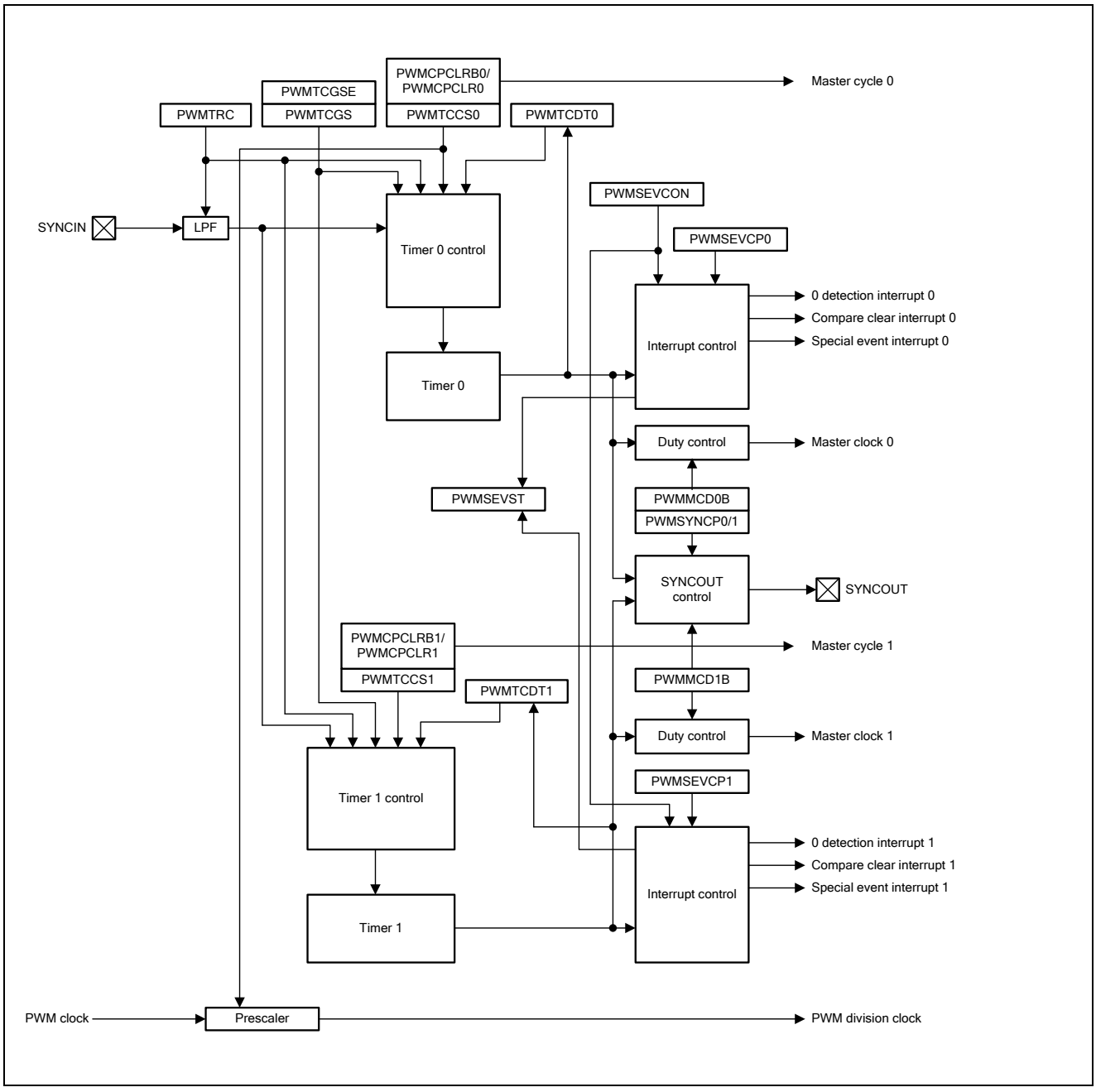
Overall configuration of the PWM

Figure 43-1. Overall Configuration of the PWM



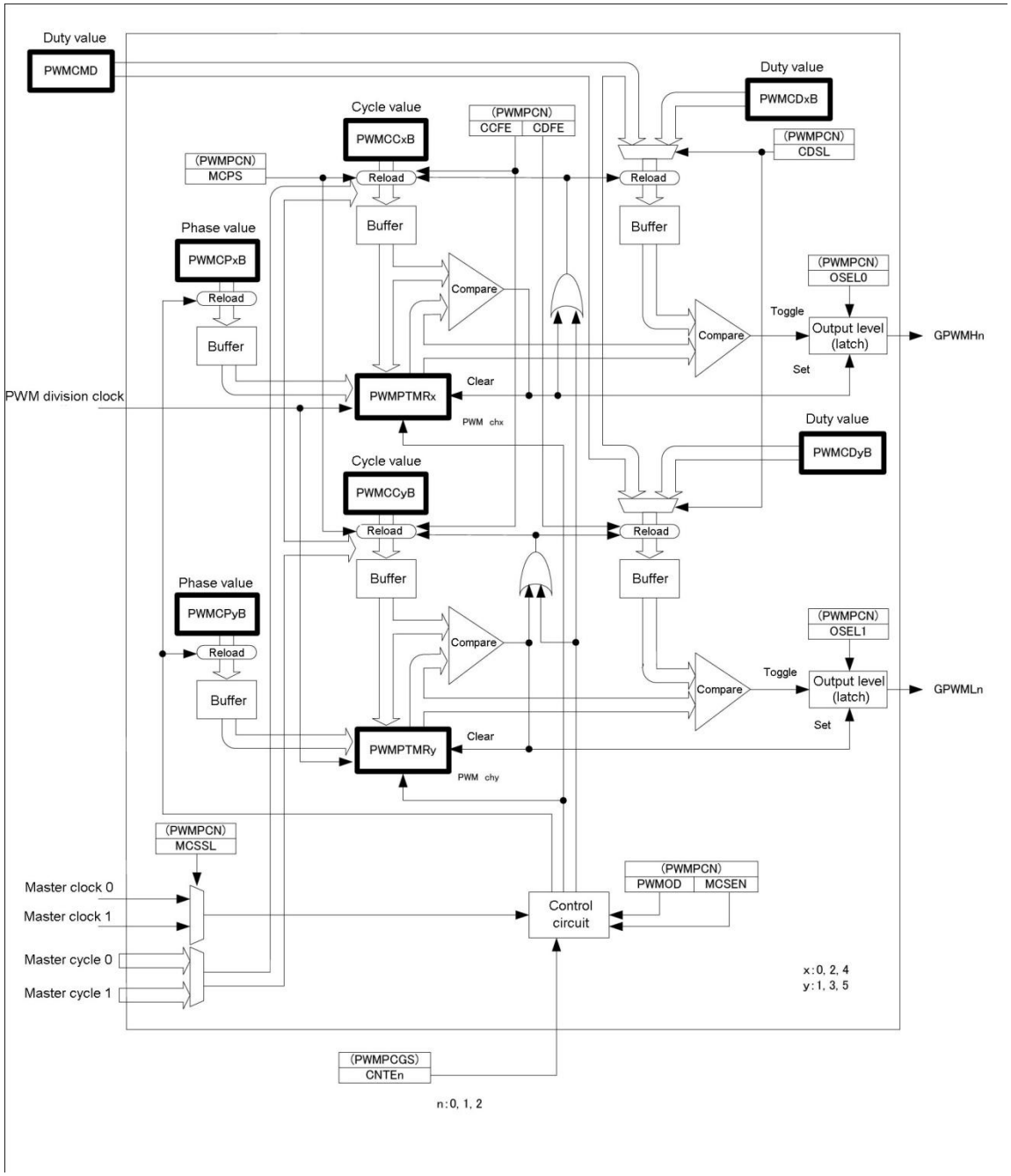
Configuration for master clock generation

Figure 43-2. Configuration for Master Clock Generation



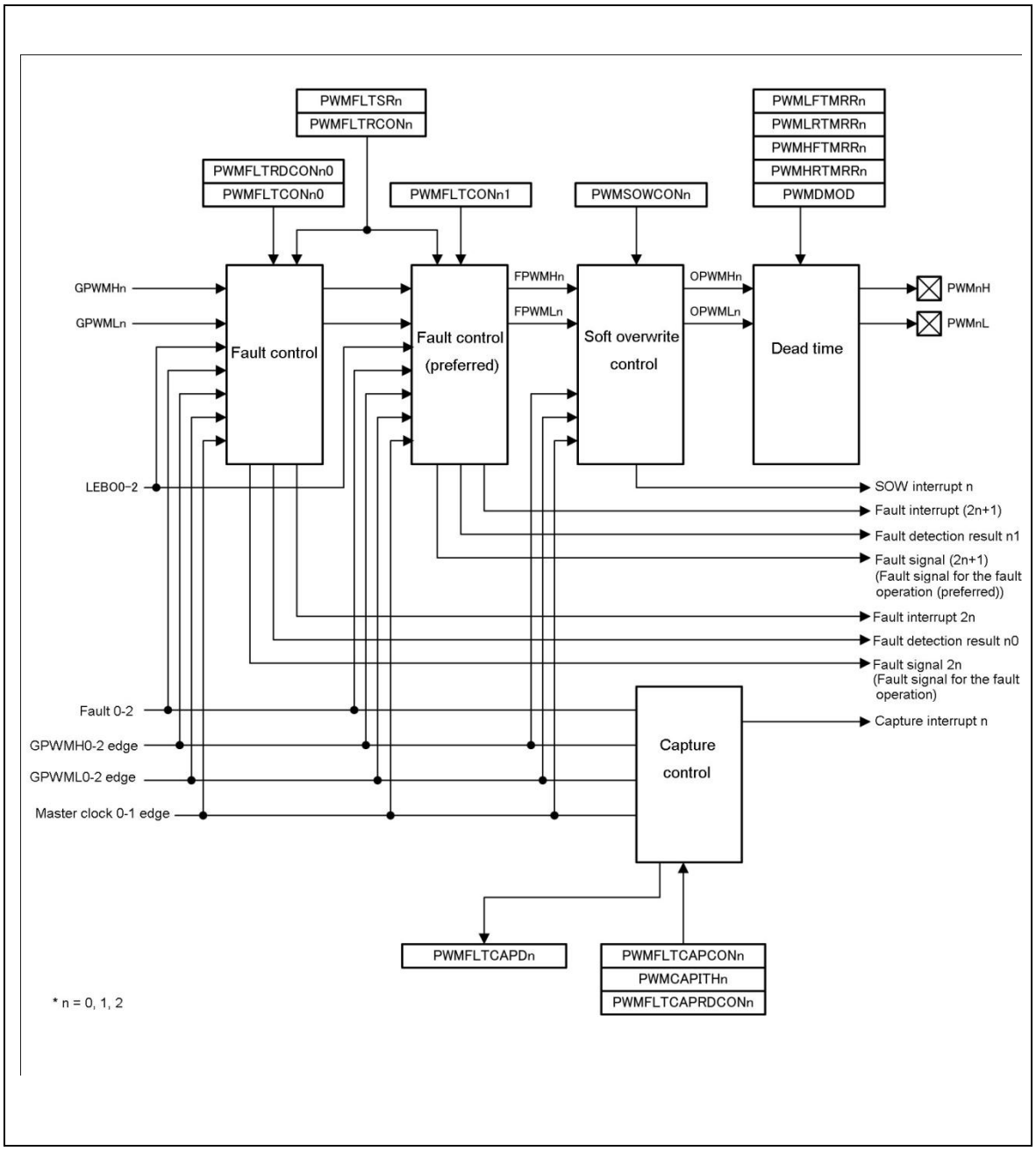
Configuration for PWM generation

Figure 43-3. Configuration for PWM Generation



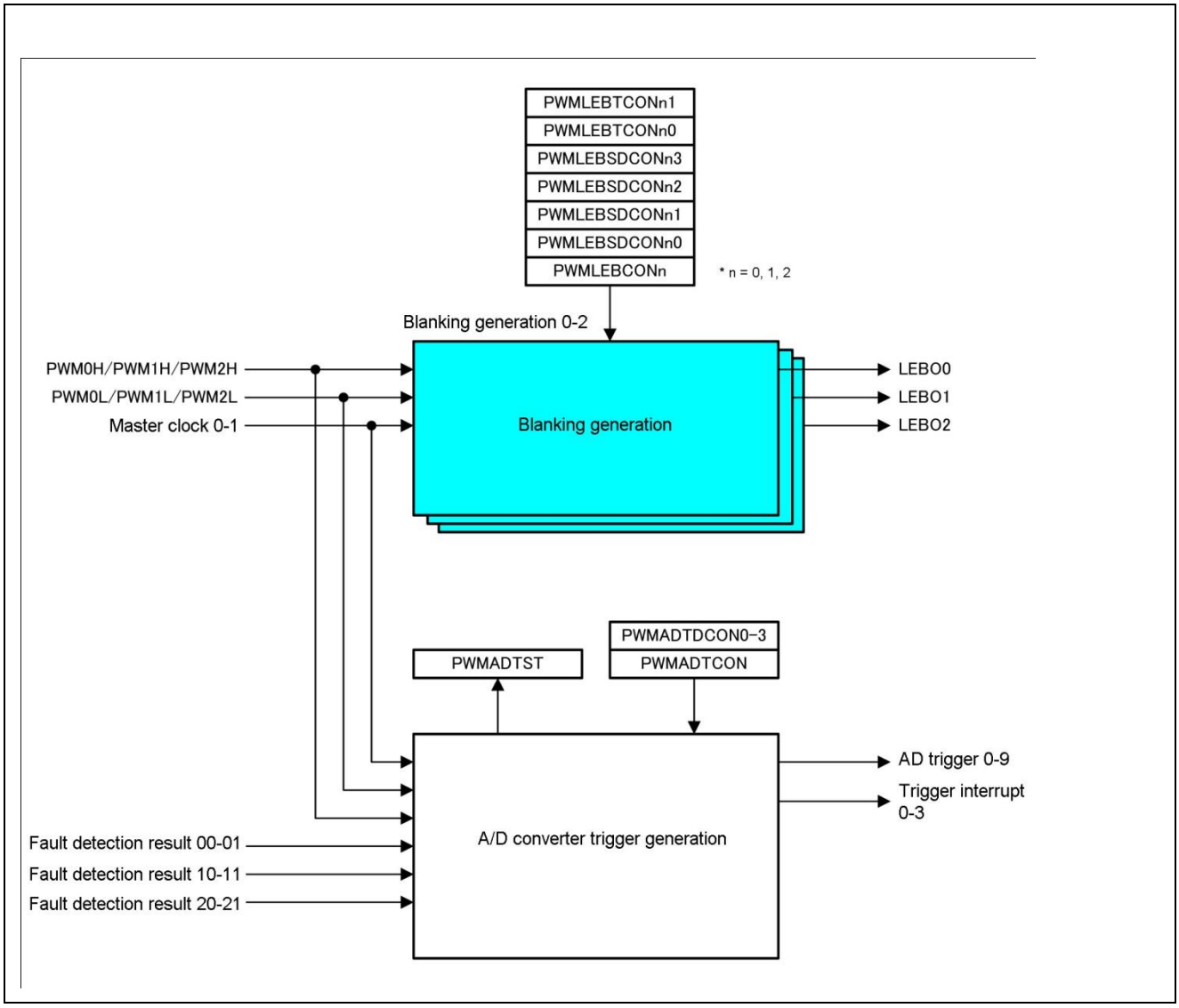
Configuration for fault control, soft overwrite control, and dead time

Figure 43-4. Configuration for Fault Control, Soft Overwrite Control, and Dead Time



Configuration for blanking generation and A/D converter trigger generation

Figure 43-5. Configuration for Blanking Generation and A/D Converter Trigger Generation



43.4 Registers

This section explains the registers of the PWM.

Table 43-1. Master Clock-related Register List

Address	+0	+1	+2	+3
0x31C0	Timer simultaneous activation register (PWMTCGS)	Reserved	Reserved	Timer simultaneous activation enable register (PWMTCGSE)
0x31C4	Compare clear buffer register 0 (PWMCPCLRB0) Compare clear register 0 (PWMCPCLR0)		Timer data register (PWMTCDT0)	
0x31C8	Timer state control register 0 (PWMTCCS0)			Reserved
0x31CC	Compare clear buffer register 1 (PWMCPCLRB1) Compare clear register 1 (PWMCPCLR1)		Timer data register (PWMTCDT1)	
0x31D0	Timer state control register 1 (PWMTCCS1)			Reserved
0x31D4	Timer reset control register (PWMTRC)			Reserved
0x31D8	SYNC compare register 0 (PWMSYNCP0)		SYNC compare register 1 (PWMSYNCP1)	
0x31DC	Special event control register (PWMSEVCON)		Reserved	
0x31E0	Special event status register (PWMSEVST)	Reserved	Reserved	
0x31E4	Special event compare register 0 (PWMSEVCP0)		Special event compare register 1 (PWMSEVCP1)	
0x31E8	Master duty setting register 0 (PWMMCD0B)		Master duty setting register 1 (PWMMCD1B)	

Table 43-2. PWM Common Status Register List

Address	+0	+1	+2	+3
0x31EC	PWM status register 0 (PWMST0)	PWM status register 1 (PWMST1)	PWM status register 2 (PWMST2)	Fault status register (PWMFLTST)

Table 43-3. PWM Generation-related Register List

Address	+0	+1	+2	+3
0x31F0	Common duty setting register (PWMCMD)		PWM simultaneous activation register (PWMPCGS)	Reserved
0x31F4	PWM control register 01 (PWMPCN01)		Reserved	
0x31F8	PWM cycle setting register 0 (PWMCC0B)		PWM phase setting register 0 (PWMCP0B)	
0x31FC	PWM duty setting register 0 (PWMCD0B)		PWM timer register 0 (PWMPTMR0)	

Address	+0	+1	+2	+3
0x3200	PWM cycle setting register 1 (PWMCC1B)		PWM phase setting register 1 (PWMCP1B)	
0x3204	PWM duty setting register 1 (PWMCD1B)		PWM timer register 1 (PWMPTMR1)	
0x3208	PWM control register 23 (PWMPCN23)		Reserved	
0x320C	PWM cycle setting register 2 (PWMCC2B)		PWM phase setting register 2 (PWMCP2B)	
0x3210	PWM duty setting register 2 (PWMCD2B)		PWM timer register 2 (PWMPTMR2)	
0x3214	PWM cycle setting register 3 (PWMCC3B)		PWM phase setting register 3 (PWMCP3B)	
0x3218	PWM duty setting register 3 (PWMCD3B)		PWM timer register 3 (PWMPTMR3)	
0x321C	PWM control register 45 (PWMPCN45)		Reserved	
0x3220	PWM cycle setting register 4 (PWMCC4B)		PWM phase setting register 4 (PWMCP4B)	
0x3224	PWM duty setting register 4 (PWMCD4B)		PWM timer register 4 (PWMPTMR4)	
0x3228	PWM cycle setting register 5 (PWMCC5B)		PWM phase setting register 5 (PWMCP5B)	
0x322C	PWM duty setting register 5 (PWMCD5B)		PWM timer register 5 (PWMPTMR54	

Table 43-4. Fault Function-related Register List

Address	+0	+1	+2	+3
0x3230	Fault control register 00 (PWMFLTCON00)		Fault control register 01 (PWMFLTCON01)	
0x3234	Fault reset control register 0 (PWMFLTRCON0)		Fault capture control register 0 (PWMFLTCAPCON0)	
0x3238	Fault soft reset register 0 (PWMFLTSR0)		Reserved	
0x323C	Capture interrupt threshold setting register 0 (PWMCAPITH0)		Reserved	
0x3240	Fault reset delay control register 00 (PWMFLTRDCON00)			
0x3244	Fault reset delay control register 01 (PWMFLTRDCON01)			
0x3248	Fault capture counter reset delay control register 0 (PWMFLTCAPRDCON0)			
0x324C	Fault capture data register 0 (PWMFLTCAPD0)			
0x3250	Fault control register 10 (PWMFLTCON10)		Fault control register 11 (PWMFLTCON11)	
0x3254	Fault reset control register 1 (PWMFLTRCON1)		Fault capture control register 1 (PWMFLTCAPCON1)	
0x3258	Fault soft reset register 1 (PWMFLTSR1)		Reserved	

Address	+0	+1	+2	+3
0x325C	Capture interrupt threshold setting register 1 (PWMCAPITH1)		Reserved	
0x3260	Fault reset delay control register 10 (PWMFLTRDCON10)			
0x3264	Fault reset delay control register 11 (PWMFLTRDCON11)			
0x3268	Fault capture counter reset delay control register 1 (PWMFLTCAPRDCON1)			
0x326C	Fault capture data register 1 (PWMFLTCAPD1)			
0x3270	Fault control register 20 (PWMFLTCON20)		Fault control register 21 (PWMFLTCON21)	
0x3274	Fault reset control register 2 (PWMFLTRCON2)		Fault capture control register 2 (PWMFLTCAPCON2)	
0x3278	Fault soft reset register 2 (PWMFLTSR2)		Reserved	
0x327C	Capture interrupt threshold setting register 2 (PWMCAPITH2)		Reserved	
0x3280	Fault reset delay control register 20 (PWMFLTRDCON20)			
0x3284	Fault reset delay control register 21 (PWMFLTRDCON21)			
0x3288	Fault capture counter reset delay control register 2 (PWMFLTCAPRDCON2)			
0x328C	Fault capture data register 2 (PWMFLTCAPD2)			

Table 43-5. Soft Overwrite Register List

Address	+0	+1	+2	+3
0x3290	Soft overwrite control register 0 (PWMSOWCON0)			
0x3294	Soft overwrite control register 1 (PWMSOWCON1)			
0x3298	Soft overwrite control register 2 (PWMSOWCON2)			

Table 43-6. Dead Time Register List

Address	+0	+1	+2	+3
0x329C	Dead time mode register (PWMDMOD)	Reserved	Reserved	
0x32A0	High-side rising dead time setting register 0 (PWMHRTMRR0)		High-side falling dead time setting register 0 (PWMHFTMRR0)	
0x32A4	Low-side rising dead time setting register 0 (PWMLRTMRR0)		Low-side falling dead time setting register 0 (PWMLFTMRR0)	

Address	+0	+1	+2	+3
0x32A8	High-side rising dead time setting register 1 (PWMHRTMRR1)		High-side falling dead time setting register 1 (PWMHFTMRR1)	
0x32AC	Low-side rising dead time setting register 1 (PWMLRTMRR1)		Low-side falling dead time setting register 1 (PWMLFTMRR1)	
0x32B0	High-side rising dead time setting register 2 (PWMHRTMRR2)		High-side falling dead time setting register 2 (PWMHFTMRR2)	
0x32B4	Low-side rising dead time setting register 2 (PWMLRTMRR2)		Low-side falling dead time setting register 2 (PWMLFTMRR2)	

Table 43-7. Blanking-related Register List

Address	+0	+1	+2	+3
0x32B8	Blanking control register 0 (PWMLEBCON0)			Reserved
0x32BC	Blanking start delay control register 00 (PWMLEBSDCON00)		Blanking start delay control register 01 (PWMLEBSDCON01)	
0x32C0	Blanking start delay control register 02 (PWMLEBSDCON02)		Blanking start delay control register 03 (PWMLEBSDCON03)	
0x32C4	Blanking time control register 00 (PWMLEBTCON00)			
0x32C8	Blanking time control register 01 (PWMLEBTCON01)			
0x32CC	Blanking control register 1 (PWMLEBCON1)			Reserved
0x32D0	Blanking start delay control register 10 (PWMLEBSDCON10)		Blanking start delay control register 11 (PWMLEBSDCON11)	
0x32D4	Blanking start delay control register 12 (PWMLEBSDCON12)		Blanking start delay control register 13 (PWMLEBSDCON13)	
0x32D8	Blanking time control register 10 (PWMLEBTCON10)			
0x32DC	Blanking time control register 11 (PWMLEBTCON11)			
0x32E0	Blanking control register 2 (PWMLEBCON2)			Reserved
0x32E4	Blanking start delay control register 20 (PWMLEBSDCON20)		Blanking start delay control register 21 (PWMLEBSDCON21)	
0x32E8	Blanking start delay control register 22 (PWMLEBSDCON22)		Blanking start delay control register 23 (PWMLEBSDCON23)	
0x32EC	Blanking time control register 20 (PWMLEBTCON20)			
0x32F0	Blanking time control register 21 (PWMLEBTCON21)			

Table 43-8.A/D Converter Trigger Generation Register List

Address	+0	+1	+2	+3
0x32F4	A/DC trigger control register (PWMADTCON)			
0x32F8	A/DC trigger status register (PWMADTST)	Reserved	Reserved	
0x32FC	A/DC trigger delay control register 0 (PWMADTDCON0)		A/DC trigger delay control register 1 (PWMADTDCON1)	
0x3300	A/DC trigger delay control register 2 (PWMADTDCON2)		A/DC trigger delay control register 3 (PWMADTDCON3)	

43.4.1 Master Clock-related Registers

This section explains the master clock-related registers.

The master clock-related registers are the timer simultaneous activation register, timer simultaneous activation enable register, compare clear buffer register, compare clear register, timer data register, timer state control register, timer reset control register, SYNC compare register, special event control register, special event status register, special event compare register, and master duty setting register.

43.4.1.1 Timer Simultaneous Activation Register: PWMTCGS

This section shows the bit configuration of the timer simultaneous activation register.

The timer simultaneous activation register (PWMTCGS) is used to control simultaneous timer enable and simultaneous timer clear for the timer. The time for which to execute simultaneous timer enable and simultaneous timer clear is specified with the timer simultaneous activation enable register (PWMTCGSE).

PWMTCGS: Address 31C0H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						GSTOP	GSCLR
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W	R0,W

[bit7 to bit2] (Reserved)

[bit1] GSTOP: Simultaneous timer enable bit

GSTOP	Function	
	During Read Operation	During Write Operation
0	"0" is always read.	Enable counting simultaneously. (Start counting.)
1		Disable counting simultaneously. (Stop counting.)

- This bit is used to simultaneously stop or start the counting of the timer specified by the timer simultaneous activation enable register (PWMTCGSE).
- When this bit is set to "0":

  The counting of the timer specified by the timer simultaneous activation enable register (PWMTCGSE) is started. Also, the STOP bit in the timer state control register (PWMTCCS) for the timer specified by the timer simultaneous activation enable register (PWMTCGSE) is set to "0".
- When this bit is set to "1":

  The counting of the timer specified by the timer simultaneous activation enable register (PWMTCGSE) is stopped. Also, the STOP bit in the timer state control register (PWMTCCS) for the timer specified by the timer simultaneous activation enable register (PWMTCGSE) is set to "1".
- The read value is always "0".

[bit0] GSCLR: Simultaneous timer clear bit

GSCLR	Function	
	During Read Operation	During Write Operation
0	"0" is always read.	Do not initialize the counter.
1		Initialize the counter to "0000_H" simultaneously.

- This bit is used to initialize the timer specified by the timer simultaneous activation enable register (PWMTCGSE) to "0000_H".
- When this bit is set to "1":

The timer specified by the timer simultaneous activation enable register (PWMTCGSE) is initialized. Also, the SCLR bit in the timer state control register (PWMTCCS) for the timer specified by the timer simultaneous activation enable register (PWMTCGSE) is set to "1".

- When this bit is set to "0":

  The instruction for initializing the timer specified by the timer simultaneous activation enable register (PWMTCGSE) is canceled. Also, the SCLR bit in the timer state control register (PWMTCCS) for the timer specified by the timer simultaneous activation enable register (PWMTCGSE) is set to "0".

- The read value is always "0".

43.4.1.2 Timer Simultaneous Activation Enable Register: PWMTCGSE

This section shows the bit configuration of the timer simultaneous activation enable register.

The timer simultaneous activation enable register (PWMTCGSE) sets the timer to be activated or cleared simultaneously.

PWMTCGSE: Address 31C3H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						FRT1	FRT0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit7 to bit2] (Reserved)

[bit1, bit0] FRT1, FRT0: Simultaneous activation/clear setting bits

FRT1, FRT0	Function
0	Do not activate or clear the timer simultaneously.
1	Activate or clear the timer simultaneously.

- Each of these bits sets the timer to be activated or cleared simultaneously.
- When these bits are set to "0":

  The timer is not activated or cleared when the timer synchronous activation register (PWMTCGS) is configured.
- When these bits are set to "1":

  The corresponding timer is activated or cleared when the timer synchronous activation register (PWMTCGS) is configured.
- FRT0 corresponds to the timer of master clock 0, and FRT1 corresponds to the timer of master clock 1.

43.4.1.3 Compare Clear Buffer Register: PWMCPCLRB0, 1 / Compare Clear Register: PWMCPCLR0, 1

This section shows the bit configurations of the compare clear buffer register and compare clear register.

The compare clear buffer register (PWMCPCLRB) is a 16-bit buffer register contained in the compare clear register (PWMCPCLR).
The PWMCPCLRB and PWMCPCLR registers are located at the same address.

PWMCPCLRB0, 1: Address 31C4H, 31CCH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial Value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial Value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

[bit15 to bit0] CL15 to CL00: Compare clear value buffer bits

CL15 to CL00	Function
	Compare clear value buffer

- The compare clear buffer register is a buffer register located at the same address as the compare clear register (PWMCPCLR).
- If the buffer function is disabled (BFE:bit23 in the timer state control register (PWMTCCS) is set to 0), or if the timer stops, the value in the compare clear buffer register is immediately transferred to the compare clear register.
- If the buffer function is enabled, the value is transferred to the compare clear register when "0" is detected as the timer count value.

Note:

Do not set "0000_H" in the compare clear buffer register.
When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

PWMCPCLR0, 1: Address 31C4H, 31CCH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial Value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial Value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

[bit15 to bit0] CL15 to CL00: Compare clear value bits

CL15 to CL00	Function
	Compare clear value

- The compare clear register is used for comparison with the count value of the timer.
- In the up-count mode, the timer is reset to "0000_H" if this register matches the timer count value.
- In the up/down count mode, the timer switches from counting up to counting down if this register matches the timer count value or from counting down to counting up if "0" is detected.

Note:

When accessing this register, use a half-word or word access instruction.
Do not use a read-modify-write instruction when accessing this register.

43.4.1.4 Timer Data Register: PWMTCDT0 to PWMTCDT1

This section shows the bit configuration of the timer data register.

The timer data register (PWMTCDT) is used to read the count value of the timer. It can also be used to set the timer count value.

PWMTCDT0: Address 31C6H (access: half word and word)

PWMTCDT1: Address 31CEH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	T15	T14	T13	T12	T11	T10	T09	T08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	T07	T06	T05	T04	T03	T02	T01	T00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

[bit15 to bit0] T15 to T00: Count value bits

T15 to T00	Function
	Count value

- The timer data register is used to read the count value of the timer.
- The count value is cleared to "0000_H" immediately when reset occurs.
- The timer value can be set by writing it to this register. Note that the value needs to be written while the timer is inactive (STOP:bit22 in the timer state control register (PWMTCCS) is set to 1). A value rewritten to this register before the next PWM division clock is ignored.
- The time is initialized immediately if any of the following occurs.
  - ☐ Reset
  - ☐ While the timer is active (STOP:bit22 in the timer state control register (PWMTCCS) is set to 0), the clear bit (SCLR:bit20) in the timer state control register (PWMTCCS) is set to 1.
  - ☐ In the up-count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 0), the compare clear register matches the timer count value.
  - ☐ When SYNCIN is enabled (SYNCEN1:bit26 and SYNCEN0:bit24 in the timer reset control register (PWMTRC) are both set to 1), a rising edge is input from the SYNCIN pin.

Note:

When inactive (STOP:bit22 in the timer state control register (PWMTCCS) is set to 1), the timer is not initialized even if the clear bit (SCLR:bit20) in the timer state control register (PWMTCCS) is set to 1.
When accessing the timer data register, use a half-word or word access instruction.
If a count value is written in the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), an unintended count may be made.
To write a count value is written in the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), perform the following steps.

1. Stop the counting of the timer. (Write "1" to STOP:bit21 in the timer state control register (PWMTCCS).)
2. Set a count value in the timer data register.
3. Perform software clear. (Write "1" to SCLR:bit20 in the timer state control register (PWMTCCS).)
4. Start the counting of the timer.

43.4.1.5 Timer State Control Register: PWMTCCS0, PWMTCCS1

This section shows the bit configuration of the timer state control register.

The timer state control register (PWMTCCS) is used to control the timer operation.

PWMTCCS0: Address 31C8H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R(RM1),W	R/W	R,W	R,W	R,W	R(RM1),W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	BFE	STOP	MODE	SCLR	Reserved	PRS2	PRS1	PRS0
Initial Value	0	1	0	0	0	0	0	0
Attribute	R/W	R,W	R/W	R0,W	R0,W0	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved				MODE2	MSI5	MSI4	MSI3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

PWMTCCS1: Address 31D0H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R(RM1),W	R/W	R,W	R,W	R,W	R(RM1),W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	BFE	STOP	MODE	SCLR	Reserved			
Initial Value	0	1	0	0	0	0	0	0
Attribute	R/W	R,W	R/W	R0,W	R0,W0	R0,W0	R0,W0	R0,W0

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved				MODE2	MSI5	MSI4	MSI3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

[bit31] (Reserved)

Be sure to write "0".

[bit30] IRQZF: 0 detection interrupt flag bit

IRQZF	Function	
	During Read Operation	During Write Operation
0	0 has not been detected.	This bit is cleared.
1	0 has been detected.	This bit remains unaffected.

- When the timer count value is "0000_H", this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read. This bit is not set if software clear is performed ("1" is written to SCLR:bit20 in the timer state control register (PWMTCCS)) when the timer is active (STOP:bit22 in the timer state control register (PWMTCCS) is set to 0).
In the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), this bit is set to "1" if any of the interrupts set in the interrupt mask selection bits (MSI2 to MSI0:bit28 to bit26 in the timer state control register (PWMTCCS) are other than "000_B") occurs. This bit is not set to "1" unless an interrupt occurs.
In the up-count mode (MODE:bit21 is set to 0), this bit is set each time 0 is detected, regardless of the value of MSI2 to MSI0:bit28 to bit26.
If software clear ("0" is written) and hardware set occur simultaneously, hardware set takes precedence.

[bit29] IRQZE: 0 detection interrupt request enable bit

IRQZE	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

If this bit and the interrupt flag bit (IRQZF:bit30) are set to "1", an interrupt request for the CPU is generated.

[bit28 to bit26] MSI2 to MSI0: Interrupt mask selection bits

MSI2	MSI1	MSI0	Function
0	0	0	Generate an interrupt when the first match occurs.
0	0	1	Generate an interrupt when the second match occurs.
0	1	0	Generate an interrupt when the third match occurs.
0	1	1	Generate an interrupt when the fourth match occurs.
1	0	0	Generate an interrupt when the fifth match occurs.
1	0	1	Generate an interrupt when the sixth match occurs.
1	1	0	Generate an interrupt when the seventh match occurs.
1	1	1	Generate an interrupt when the eighth match occurs.

- When MODE2:bit11 in the timer state control register (PWMTCCS) is set to 0
  - ☐ In the up-count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 0), these bits are used to set the mask count of the compare clear interrupt. In the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), these bits are used to set the mask count of the 0 detection interrupt.

- ☐ When these bits are set to "0", the interrupt factor is not masked.

- When MODE2:bit11 in the timer state control register (PWMTCCS) is set to 1
  - ☐ In the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), these bits are used to set the mask count of the 0 detection interrupt.
  - ☐ Setting the up-count mode (setting MODE:bit21 in the timer state control register (PWMTCCS) to 0) is prohibited.

Note:

The read value is the mask counter value.
If a read-modify-write instruction is executed, the read value is the mask register value.
During the write operation, data is written to the mask register. The value written to the mask register while the timer is active (STOP:bit22 in the timer state control register (PWMTCCS) is set to 0) is reloaded to the mask counter only when the counter is set to "0". The value written to the mask register while the timer is inactive (STOP:bit22 in the timer state control register (PWMTCCS) is set to 1) is reloaded to the counter immediately.

[bit25] ICLR: Compare clear interrupt flag bit

ICLR	Function	
	During Read Operation	During Write Operation
0	The compare clear value does not match.	This bit is cleared.
1	The compare clear value matches.	This bit remains unaffected.

- If the compare clear value matches the timer value, this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.
In the up-count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 0), this bit is set to "1" if any of the interrupts set in the interrupt mask selection bits occurs.
This bit is not set to "1" unless an interrupt occurs.
In the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), this bit is set each time compare clear occurs, regardless of the value of the MSI2 to MSI0 bits.
If software clear ("0" is written) and hardware set occur simultaneously, hardware set takes precedence.

[bit24] ICRE: Compare clear interrupt request enable bit

ICRE	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

If this bit and the compare clear interrupt flag bit (ICLR:bit25) are set to "1", an interrupt request for the CPU is generated.

[bit23] BFE: Compare clear buffer enable bit

BFE	Function
0	Disable the compare clear buffer.
1	Enable the compare clear buffer.

- This bit is used to enable the compare clear buffer register (PWMCPCLRB).
- When this bit is set to "0":

The compare clear buffer register (PWMCPCLRB) is disabled. Therefore, data can be written directly to the compare clear register (PWMCPCLR).

- When this bit is set to "1":

  The compare clear buffer register (PWMCPCLRB) is enabled. The data written to and retained in the compare clear buffer register (PWMCPCLRB) is transferred to the compare clear register if the count value "0" is detected from the timer.

[bit22] STOP: Timer enable bit

STOP	Function
0	Enable counting (start counting).
1	Disable counting (stop counting).

- This bit is used to stop and start the counting of the timer.
- When this bit is set to "0":

  The counting of the timer is started.
- When this bit is set to "1":

  The counting of the timer is stopped.
- If SCLR:bit20 in the timer state control register (PWMTCCS) is set to 1 while the timer is inactive (this bit is set to 1), the timer is not initialized.
- This bit reflects the value specified in the GSTOP bit in the timer synchronous activation register (PWMTCGS) when the FRT bit in the timer synchronous activation enable register (PWMTCGSE) is set to 1.

[bit21] MODE: Timer count mode bit

MODE	Function
0	Up-count mode
1	Up/down count mode

- This bit is used to select the count mode of the timer.
- When this bit is set to "0":

  The up-count mode is selected. The timer continues to count up until the count value matches the compare clear register and is reset to "0000_H". Then, it starts counting up again.
- When this bit is set to "1":

  The up/down count mode is selected. The timer continues to count up until the count value matches the compare clear register. Then, it starts counting down. After that, when the count value reaches "0000_H", the counter starts counting up again.
- Data can be written to this bit even when the timer is inactive.

[bit20] SCLR: Timer clear bit

SCLR	Function	
	During Read Operation	During Write Operation
0	"0" is always read.	Do not initialize the counter.
1		Initialize the counter to "0000H".

- This bit is used to initialize the timer to "0000_H".

- Initialization of the timer:

  If this bit is set to "1" while the timer is active (STOP:bit22 in the timer state control register (PWMTCCS) is set to 0), the timer is initialized to "0000_H" in the next count clock that is set. If this bit is set to "1" while the timer is inactive (STOP:bit22 in the timer state control register (PWMTCCS) is set to 1), the timer is not initialized.

- The read value is always "0".
- This bit reflects the value specified in the GSCLR bit in the timer synchronous activation register (PWMTCGS) when the FRT bit in the timer synchronous activation enable register (TCGSE) is set to 1.

Note:

If the counter is initialized by writing "1" to this bit, the 0 detection interrupt and master clock are not generated.
The timer is not cleared if "0" is written before the next count clock after "1" is set.

[bit19] (Reserved)

Be sure to write "0".

[bit18 to bit16] PRS2 to PRS0: Clock frequency selection bits

PRS2	PRS1	PRS0	Function					
			Count Clock	φ=200MHz	φ=100MHz	φ=50MHz	φ=25MHz	φ=12.5MHz
0	0	0	Φ	5ns	10ns	20ns	40ns	80ns
0	0	1	Φ /2	10ns	20ns	40ns	80ns	160ns
0	1	0	Φ /4	20ns	40ns	80ns	160ns	320ns
0	1	1	Φ /8	40ns	80ns	160ns	320ns	640ns
1	0	0	Φ /16	80ns	160ns	320ns	640ns	1.28µs
1	0	1	Φ /32	160ns	320ns	640ns	1.28µs	2.56µs
1	1	0	Φ /64	320ns	640ns	1.28µs	2.56µs	5.12µs
1	1	1	Φ /128	640ns	1.28µs	2.56µs	5.12µs	10.24µs

- These bits are used to select the count clock frequency of the timer.
- Only the PWMTCCS0 register has these bits. The bits in the PWMTCCS1 register are reserved. Be sure to write "0" to bit18 to bit16 in the PWMTCCS1 register.

Note:

Before setting the PRS2 to PRS0 bits, check that the PWM module is stopped.

[bit15 to bit12] (Reserved)

[bit11] MODE2: Interrupt mask mode bit 2

MODE2	MODE	Function
0	0	The value set in MSI5 to MSI3 is invalid.
0	1	The value set in MSI5 to MSI3 is invalid.
1	0	Setting prohibited (normal operation not guaranteed)
1	1	The value set in MSI5 to MSI3 is valid.

- When the timer is in the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), this bit is used to mask the 0 detection interrupt and compare clear interrupt independently.

- If this bit is set to "1" when MODE:bit21 in the timer state control register (PWMTCCS) is set to "1", the value set in MSI5 to MSI3:bit10 to bit8 becomes valid and the compare clear interrupt is masked the specified number of times. The value set in MSI2 to MSI0:bit28 to bit26 in the timer state control register (PWMTCCS) is valid as the number of times the 0 detection interrupt is to be masked.

Note:

If this bit is set to "1" when MODE:bit21 in the timer state control register (PWMTCCS) is set to "0", normal operation is not guaranteed.

[bit10 to bit8] MSI5 to MSI3: Compare clear interrupt mask selection bits

MSI5	MSI4	MSI3	Function
0	0	0	Generate an interrupt when the first match occurs.
0	0	1	Generate an interrupt when the second match occurs.
0	1	0	Generate an interrupt when the third match occurs.
0	1	1	Generate an interrupt when the fourth match occurs.
1	0	0	Generate an interrupt when the fifth match occurs.
1	0	1	Generate an interrupt when the sixth match occurs.
1	1	0	Generate an interrupt when the seventh match occurs.
1	1	1	Generate an interrupt when the eighth match occurs.

- These bits are valid only when MODE2:bit11 in this register is set to 1 and MODE:bit21 in the timer state control register (PWMTCCS) is set to 1. They are used to set the mask count of the compare clear interrupt. The mask count of the 0 detection interrupt is set in MSI2 to MSI0:bit28 to bit26 in the timer state control register (PWMTCCS).
- When these bits are set to"000_B", the compare clear interrupt factor is not masked.

Note:

The read value is the mask counter value.
If a read-modify-write instruction is executed, the read value is the mask register value.
During the write operation, data is written to the mask register. The value written to the mask register while the timer is active (STOP:bit22 in the timer state control register (PWMTCCS) is set to 0) is reloaded to the mask counter only when the counter is set to "0". The value written to the mask register while the timer is inactive (STOP:bit22 in the timer state control register (PWMTCCS) is set to 1) is reloaded to the counter immediately.

[bit7 to bit0] (Reserved)

43.4.1.6 Timer Reset Control Register: PWMTRC

This section shows the bit configuration of the timer reset control register.

The timer reset control register (PWMTRC) is used to control the timer reset operation.

PWMTRC: Address 31D4H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved	LPF2	LPF1	LPF0	Reserved	SYNCEN1	Reserved	SYNCEN0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R/W0	R/W	R/W0	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W0	R/W0	R/W0	R0,WX	R/W0	R/W0	R/W0

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved					SYNOS2	SYNOS1	SYNOS0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

[bit31] (Reserved)

[bit30 to bit28] LPF2 to LPF0: LPF selection bits

LPF2	LPF1	LPF0	Function
0	0	0	Do not filter.
0	0	1	Filter an input of less than 2 cycles.
0	1	0	Filter an input of less than 4 cycles.
0	1	1	Filter an input of less than 8 cycles.
1	0	0	Filter an input of less than 16 cycles.
1	0	1	Filter an input of less than 32 cycles.
1	1	0	Filter an input of less than 64 cycles.
1	1	1	Filter an input of less than 128 cycles.

- These bits are used to set a digital LPF for the input signal from the SYNCIN pin.
- The LPF removal width is represented by the number of PWM division clock cycles.

Note:

Before setting the LPF2 to LPF0 bits, check that the timer is stopped.

If LPF2 to LPF0 is set to other than 3'b000, the SYNCIN input is delayed by the number of cycles equivalent to the set removal width.

[bit27] (Reserved)

Be sure to write "0".

[bit26] SYNCEN1: SYNCIN synchronization enable bit 1

SYNCEN1	Function
0	Disable SYNCIN synchronization.
1	Enable SYNCIN synchronization.

- This bit is used to enable timer synchronization on the master clock 1 side using the SYNCIN pin.
- If this bit is set to "0", the timer on the master clock 1 side is not synchronized with the SYNCIN pin.
- If this bit is set to "1", the timer on the master clock 1 side is initialized to "0000_H" at the rising edge of the SYNCIN pin.

[bit25] (Reserved)

Be sure to write "0".

[bit24] SYNCEN0: SYNCIN synchronization enable bit 0

SYNCEN0	Function
0	Disable SYNCIN synchronization.
1	Enable SYNCIN synchronization.

- This bit is used to enable timer synchronization on the master clock 0 side using the SYNCIN pin.
- If this bit is set to "0", the timer on the master clock 0 side is not synchronized with the SYNCIN pin.
- If this bit is set to "1", the timer on the master clock 0 side is initialized to "0000_H" at the rising edge of the SYNCIN pin.

[bit23] (Reserved)

[bit22 to bit20] (Reserved)

Be sure to write "0".

[bit19] (Reserved)

[bit18 to bit16] (Reserved)

Be sure to write "0".

[bit15 to bit11] (Reserved)

[bit10 to bit8] SYNOS2 to SYNOS0: SYNCOUT output selection bits

SYNOS2	SYNOS1	SYNOS0	Function
0	0	0	Do not output SYNCOUT (output "L").
0	0	1	
0	1	0	
0	1	1	
1	0	0	Output 0 detection for master clock 0.
1	0	1	Output compare match detection for master clock 0.
1	1	0	Output 0 detection for master clock 1.
1	1	1	Output compare match detection for master clock 1.

These bits are used to select the output signal to the SYNCOUT pin.

Note:

Before setting the SYNOS2 to SYNOS0 bits, check that the timer is stopped.

[bit7 to bit0] (Reserved)

43.4.1.7 SYNC Compare Register: PWMSYNCP0, PWMSYNCP1

This section shows the bit configuration of the SYNC compare register.

The SYNC compare register (PWMSYNCP) is used to control the position of the rising edge of the SYNCOUT output.

PWMSYNCP0: Address 31D8H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	S0CP15	S0CP14	S0CP13	S0CP12	S0CP11	S0CP10	S0CP09	S0CP08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	S0CP07	S0CP06	S0CP05	S0CP04	S0CP03	S0CP02	S0CP01	S0CP00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] S0CP15 to S0CP00: Compare value bits

S0CP15 to S0CP00	Function
	Set the timer compare value for the SYNCOUT output.

- These bits are used to set the value to be compared with the timer on the master clock 0 side and determine the position of the rising edge of the signal to be output to the SYNCOUT pin when SYNOS2 to SYNOS0:bit10 to bit8 in the timer reset control register (PWMTRC) are set to "101".
- The pulse generated through comparison with the timer on the master clock 0 side is output from the SYNCOUT pin, with its width extended by the value set in master duty setting register 0 (PWMMCD0B).
- Set a value that is smaller than the cycle of the master clock.

PWMSYNCP1: Address 31DAH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	S1CP15	S1CP14	S1CP13	S1CP12	S1CP11	S1CP10	S1CP09	S1CP08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	S1CP07	S1CP06	S1CP05	S1CP04	S1CP03	S1CP02	S1CP01	S1CP00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] S1CP15 to S1CP00: Compare value bits

S1CP15 to S1CP00	Function
	Set the timer compare value for the SYNCOUT output.

- These bits are used to set the value to be compared with the timer on the master clock 1 side and determine the position of the rising edge of the signal to be output to the SYNCOUT pin when SYNOS2 to SYNOS0:bit10 to bit8 in the timer reset control register (PWMTRC) are set to "111".

- The pulse generated through comparison with the timer on the master clock 1 side is output from the SYNCOUT pin, with its width extended by the value set in master duty setting register 1 (PWMMCD1B).
- Set a value that is smaller than the cycle of the master clock.

Note:

When accessing the SYNC compare register, use a half-word or word access instruction.

43.4.1.8 Special Event Control Register: PWMSEVCON

This section shows the bit configuration of the special event control register.

The special event control register (PWMSEVCON) is used to control the special event trigger and special event interrupt request.

PWMSEVCON: Address 31DCH (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved			ISEVPS03	ISEVPS02	ISEVPS01	ISEVPS00	ISEVE0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			ISEVPS13	ISEVPS12	ISEVPS11	ISEVPS10	ISEVE1
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W

[bit15 to bit13] (Reserved)

[bit12 to bit9] ISEVPS03 to ISEVPS00: Interrupt frequency selection bits 0

ISEVPS03	ISEVPS02	ISEVPS01	ISEVPS00	Function
0	0	0	0	Generate an interrupt when the first match occurs.
0	0	0	1	Generate an interrupt when the second match occurs.
0	0	1	0	Generate an interrupt when the third match occurs.
:				:
1	1	0	1	Generate an interrupt when the 14th match occurs.
1	1	1	0	Generate an interrupt when the 15th match occurs.
1	1	1	1	Generate an interrupt when the 16th match occurs.

- These bits are used to set the interrupt frequency of special event trigger 0 and special event interrupt 0.
- When these bits are set to "0000_B", the special event trigger and special event interrupt factor are not masked.

Note:

While the timer is active (STOP:bit22 in the timer state control register (PWMTCCS0) is set to 0), the value written to the interrupt frequency selection bits is reloaded to the counter only when the frequency counter is set to "0". While the timer is inactive (STOP:bit22 in the timer state control register (PWMTCCS0) is set to 1), the value written to the interrupt frequency selection bits is reloaded to the counter immediately.

[bit8] ISEVE0: Interrupt request enable bit 0

ISEVE0	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for special event interrupt 0.
- If this bit and interrupt flag bit 0 (ISEVF0:bit0) in the special event status register (PWMSEVST) are set to "1", an interrupt request for the CPU is generated.

[bit7 to bit5] (Reserved)

[bit4 to bit1] ISEVPS13 to ISEVPS10: Interrupt frequency selection bits 1

ISEVPS13	ISEVPS12	ISEVPS11	ISEVPS10	Function
0	0	0	0	Generate an interrupt when the first match occurs.
0	0	0	1	Generate an interrupt when the second match occurs.
0	0	1	0	Generate an interrupt when the third match occurs.
:				:
1	1	0	1	Generate an interrupt when the 14th match occurs.
1	1	1	0	Generate an interrupt when the 15th match occurs.
1	1	1	1	Generate an interrupt when the 16th match occurs.

- These bits are used to set the interrupt frequency of special event trigger 1 and special event interrupt 1.
- When these bits are set to "0000_B", the special event trigger and special event interrupt factor are not masked.

Note:

While the timer is active (STOP:bit22 in the timer state control register (PWMTCCS1) is set to 0), the value written to the interrupt frequency selection bits is reloaded to the counter only when the frequency counter is set to "0". While the timer is inactive (STOP:bit22 in the timer state control register (PWMTCCS1) is set to 1), the value written to the interrupt frequency selection bits is reloaded to the counter immediately.

[bit0] ISEVE1: Interrupt request enable bit 1

ISEVE1	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for special event interrupt 1.
- If this bit and interrupt flag bit 1 (ISEVF1:bit1) in the special event status register (PWMSEVST) are set to "1", an interrupt request for the CPU is generated.

43.4.1.9 Special Event Status Register: PWMSEVST

This section shows the bit configuration of the special event status register.

The special event status register (PWMSEVST) is used to control the special event interrupt request.

PWMSEVST: Address 31E0H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						ISEVF1	ISEVF0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1)/W	R(RM1)/W

[bit7 to bit2] (Reserved)

[bit1] ISEVF1: Special event interrupt flag bit 1

ISEVF1	Function	
	During Read Operation	During Write Operation
0	No match is found.	This bit is cleared.
1	A match is found.	This bit remains unaffected.

- If the value of special event compare register 1 matches the timer value, this bit is set to "1"
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

[bit0] ISEVF0: Special event interrupt flag bit 0

ISEVF0	Function	
	During Read Operation	During Write Operation
0	No match is found.	This bit is cleared.
1	A match is found.	This bit remains unaffected.

- If the value of special event compare register 0 matches the timer value, this bit is set to "1"
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.

If software clear ("0" is written) and hardware set occur simultaneously, hardware set takes precedence.

43.4.1.10 Special Event Compare Register: PWMSEVCP0, PWMSEVCP1

This section shows the bit configuration of the special event compare register.

The special event compare register (PWMSEVCP) is used to control the special event trigger and special event interrupt request.

PWMSEVCP0: Address 31E4H (access: half word and word)

PWMSEVCP1: Address 31E6H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SECP15	SECP14	SECP13	SECP12	SECP11	SECP10	SECP09	SECP08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SECP07	SECP06	SECP05	SECP04	SECP03	SECP02	SECP01	SECP00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SECP15 to SECP00: Compare value bits

SECP15 to ECP00	Function
	Set the compare value of the special event trigger.

- These bits are used for comparison with the counter value of the timer.
- If the counter value of the timer matches, the special event trigger is generated and the special event interrupt flags is set to "1".
- The frequency of special event trigger generation and special event interrupt flag setting can be set using the interrupt frequency selection bits (ISEVPS03 to ISEVPS00:bit12 to bit9 and ISEVPS13 to ISEVPS10:bit4 to bit1) in the special event control register (PWMSEVCON).
- Set a value that is smaller than the cycle of the master clock.
- When generating the special event trigger and special event interrupt, make sure that the master clock cycle is equal to or greater than (3 × PWM division clock cycle + 3 × peripheral clock (PCLK) cycle). (Make sure that the special event trigger is generated at intervals equal to or greater than (3 × PWM division clock cycle + 3 × peripheral clock (PCLK) cycle.)

Note:

When accessing the special event compare register, use a half-word or word access instruction.

43.4.1.11 Master Duty Setting Register: PWMCD0B, PWMCD1B

This section shows the bit configuration of the master duty setting register.

The master duty setting register (PWMMCDB) is used to control the duty of the master clock.

PWMMCD0B: Address 31E8H (access: half word and word)

PWMMCD1B: Address 31EAH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D09	D08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D07	D06	D05	D04	D03	D02	D01	D00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D00: Master duty value bits

D15 to D00	Function
	Set the master duty value.

- These bits are used to set the duty of the master clock.
- PWMMCD0B is the setting register for master clock 0, and PWMMCD1B is the setting register for master clock 1.
- Set a value that is smaller than the cycle of the master clock. If a value equal to or greater than the master clock cycle is set, "H" is always output from the SYNCOUT pin.
- If "0000_H" is set, "L" is always output from the SYNCOUT pin.

Note:

Before setting the master duty setting register, check that the timer is stopped.

When accessing the master duty setting register, use a half-word or word access instruction.

The value set in the master duty setting register is reflected on the timer as follows.

- Master clock 0: Reflected on the timer upon 0 detection 0.
- Master clock 1: Reflected on the timer upon 0 detection 1.
- SYNCOUT pin: Reflected on the timer at the signal output timing selected in the SYNCOUT output selection bits (SYNCOS2 to SYNCOS0:bit10 to bit8) in the timer reset control register (PWMTRC).

43.4.2 PWM Common Status Registers

This section explains the PWM common status registers.

The PWM common status registers are the PWM common status register and fault status register.

43.4.2.1 PWM Common Status Register: PWMST0 to PWMST2

This section shows the bit configuration of the PWM common status register.

The PWM common status register (PWMST) is used to control fault interrupt requests, soft overwrite interrupt requests, and capture interrupt requests.

PWMST0: Address 31ECH (access: byte, half word, and word)

PWMST1: Address 31EDH (access: byte, half word, and word)

PWMST2: Address 31EEH (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				CAPIF	SOWIF	IFLTF1	IFLTF0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W

[bit7 to bit4] (Reserved)

[bit3] CAPIF: Capture interrupt flag bit

CAPIF	Function	
	During Read Operation	During Write Operation
0	The threshold is not exceeded.	This bit is cleared.
1	The threshold is exceeded.	This bit remains unaffected.

- If the difference between the high-side capture data bits (HCAP15 to HCAP00:bit31 to bit16) in the fault capture data register (PWMFLTCAPD) and the low-side capture data bits (LCAP15 to LCAP00:bit15 to bit0) exceeds the value set in the capture interrupt threshold setting register (PWMCAPITH), this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

[bit2] SOWIF: SOW interrupt flag bit

SOWIF	Function	
	During Read Operation	During Write Operation
0	Not updated	This bit is cleared.
1	Updated	This bit remains unaffected.

- If the OSL buffer enable bit in the soft overwrite control register (PWMSOWCON) is enabled (OSLBFE:bit27 is set to "1"), this bit is set to "1" at the update timing selected by overwrite selection bits 1 (OSL11, OSL10:bit31, bit30) and overwrite selection bits 0 (OSL01, OSL00:bit29, bit28) in the soft overwrite control register (PWMSOWCON).
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

[bit1] IFLTF1: Fault interrupt flag bit 1

IFLTF1	Function	
	During Read Operation	During Write Operation
0	No fault operation (preferred)	This bit is cleared.
1	Fault operation (preferred) in progress	This bit remains unaffected.

- If the fault operation (preferred) occurs, this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

[bit0] IFLTF0: Fault interrupt flag bit 0

IFLTF0	Function	
	During Read Operation	During Write Operation
0	No fault operation	This bit is cleared.
1	Fault operation in progress	This bit remains unaffected.

- If the fault operation occurs, this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.
- If the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function from the cycle before the one in which the GPWMH change point is detected until the fault latch is reset. Therefore, this bit is not set to "1".

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.

If software clear ("0" is written) and hardware set occur simultaneously, hardware set takes precedence.

43.4.2.2 Fault Status Register: PWMFLTST

This section shows the bit configuration of the fault status register.

The fault status register (PWMFLTST) is used to read the fault operation status.

PWMFLTST: Address 31EFH (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		FLTST5	FLTST4	FLTST3	FLTST2	FLTST1	FLTST0
Initial Value	0	0	X	X	X	X	X	X
Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit7, bit6] (Reserved)

[bit5] FLTST5: Fault operation status bit 5

FLTST5	Function
0	No fault operation
1	Fault operation in progress

- This bit indicates the status of the fault operation (preferred) for PWM control 2.
- The fault operation is in progress when there is PWM output based on the values set in the FPWMH fault operation (preferred) selection bits (FLT1H1, FLT1H0:bit11, bit10) and FPWML fault operation (preferred) selection bits (FLT1L1, FLT1L0:bit9, bit8) in fault control register 21 (PWMFLTCON21).
- The value that is set in this bit after a reset depends on the state of the reset pin.

[bit4] FLTST4: Fault operation status bit 4

FLTST4	Function
0	No fault operation
1	Fault operation in progress

- This bit indicates the status of the fault operation for PWM control 2.
- The fault operation is in progress when there is PWM output based on the values set in the FPWMH fault operation selection bits (FLT0H1, FLT0H0:bit11, bit10) and FPWML fault operation selection bits (FLT0L1, FLT0L0:bit9, bit8) in fault control register 20 (PWMFLTCON20).
- The value that is set in this bit after a reset depends on the state of the reset pin.

[bit3] FLTST3: Fault operation status bit 3

FLTST3	Function
0	No fault operation
1	Fault operation in progress

- This bit indicates the status of the fault operation (preferred) for PWM control 1.
- The fault operation is in progress when there is PWM output based on the values set in the FPWMH fault operation (preferred) selection bits (FLT1H1, FLT1H0:bit11, bit10) and FPWML fault operation (preferred) selection bits (FLT1L1, FLT1L0:bit9, bit8) in fault control register 11 (PWMFLTCON11).
- The value that is set in this bit after a reset depends on the state of the reset pin.

[bit2] FLTST2: Fault operation status bit 2

FLTST2	Function
0	No fault operation
1	Fault operation in progress

- This bit indicates the status of the fault operation for PWM control 1.
- The fault operation is in progress when there is PWM output based on the values set in the FPWMH fault operation selection bits (FLT0H1, FLT0H0:bit11, bit10) and FPWML fault operation selection bits (FLT0L1, FLT0L0:bit9, bit8) in fault control register 10 (PWMFLTCON10).
- The value that is set in this bit after a reset depends on the state of the reset pin.

[bit1] FLTST1: Fault operation status bit 1

FLTST1	Function
0	No fault operation
1	Fault operation in progress

- This bit indicates the status of the fault operation (preferred) for PWM control 0.
- The fault operation is in progress when there is PWM output based on the values set in the FPWMH fault operation (preferred) selection bits (FLT1H1, FLT1H0:bit11, bit10) and FPWML fault operation (preferred) selection bits (FLT1L1, FLT1L0:bit9, bit8) in fault control register 01 (PWMFLTCON01).
- The value that is set in this bit after a reset depends on the state of the reset pin.

[bit0] FLTST0: Fault operation status bit 0

FLTST0	Function
0	No fault operation
1	Fault operation in progress

- This bit indicates the status of the fault operation for PWM control 0.
- The fault operation is in progress when there is PWM output based on the values set in the FPWMH fault operation selection bits (FLT0H1, FLT0H0:bit11, bit10) and FPWML fault operation selection bits (FLT0L1, FLT0L0:bit9, bit8) in fault control register 00 (PWMFLTCON00).
- The value that is set in this bit after a reset depends on the state of the reset pin.

43.4.3 PWM Generation-related Registers

This section explains the PWM generation-related registers.

The PWM generation-related registers are the common duty setting register, PWM synchronous activation register, PWM control register, PWM cycle setting register, PWM phase setting register, PWM duty setting register, and PWM timer register.

43.4.3.1 Common Duty Setting Register: PWMCMD

This section shows the bit configuration of the common duty setting register.

The common duty setting register (PWMCMD) is used to control the common PWM duty.

PWMCMD: Address 31F0H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	MCD15	MCD14	MCD13	MCD12	MCD11	MCD10	MCD09	MCD08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MCD07	MCD06	MCD05	MCD04	MCD03	MCD02	MCD01	MCD00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] MCD15 to MCD00: Duty value bits

MCD15 to MCD00	Function
	Set the common duty values.

- These bits are used to set the PWM duty when the PWM duty selection bit in the PWM control register (PWMPCN) selects the common duty (CDSL:bit14 is set to "0").
- In the common duty setting register, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).

Note:

When accessing the common duty setting register, use a half-word or word access instruction.

For details on how the value of the common duty setting register is reflected on the timer, see the explanation of the duty buffer enable bit (CDFE:bit15) in the PWM control register (PWMPCN).

43.4.3.2 PWM Simultaneous Activation Register: PWMPCGS

This section shows the bit configuration of the PWM simultaneous activation register.

The PWM simultaneous activation register (PWMPCGS) is used to control the simultaneous activation of the PWM timer.

PWMPCGS: Address 31F2H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					CNTE2	CNTE1	CNTE0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

[bit7 to bit3] (Reserved)

[bit2] CNTE2: Timer 2 operation enable bit

CNTE2	Description
0	Disable timer 2 operation.
1	Enable timer 2 operation.

- This bit is used to enable the operation of timer 2.
- When this bit is set to "0": The operation of timer 2 is disabled.
- When this bit is set to "1": The operation of timer 2 is enabled.

[bit1] CNTE1: Timer 1 operation enable bit

CNTE1	Description
0	Disable timer 1 operation.
1	Enable timer 1 operation.

- This bit is used to enable the operation of timer 1.
- When this bit is set to "0": The operation of timer 1 is disabled.
- When this bit is set to "1": The operation of timer 1 is enabled.

[bit0] CNTE0: Timer 0 operation enable bit

CNTE0	Description
0	Disable timer 0 operation.
1	Enable timer 0 operation.

- This bit is used to enable the operation of timer 0.
- When this bit is set to "0": The operation of timer 0 is disabled.
- When this bit is set to "1": The operation of timer 0 is enabled.

43.4.3.3 PWM Status Control Register: PWMPCN01 to PWMPCN45

This section shows the bit configuration of the PWM control register.

The PWM control register (PWMPCN) is used to control the operation of the PWM.

PWMPCN01: Address 31F4H (access: byte, half word, and word)

PWMPCN23: Address 3208H (access: byte, half word, and word)

PWMPCN45: Address 321CH (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CDFE	CDSL	CCFE	MCPS	MCSSL	PWMOD	MCSEN	OWFS
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						OSEL1	OSEL0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit15] CDFE: Duty buffer enable bit

CDFE	Function
0	Disable the duty buffer.
1	Enable the duty buffer.

- This bit is used to enable the duty buffer.
- When this bit is set to "0":

  The duty buffer is disabled. The data written to the common duty setting register (PWMCMD) (when CDSL:bit14 is set to "0") or to the PWM duty setting register (PWMCDB) (when CDSL:bit14 is set to "1") is transferred immediately to the duty buffer.
- When this bit is set to "1":

  The duty buffer is enabled. The data written to the common duty setting register (PWMCMD) (when CDSL:bit14 is set to "0") or to the PWM duty setting register (PWMCDB) (when CDSL:bit14 is set to "1") is transferred to the duty buffer in one of the following ways based on the PWM output waveform selection bit.

  - ☐ If the normal waveform is selected (OWFS:bit8 is set to "0"), the data is transferred to the duty buffer when the count value of the timer matches the cycle value.
  - ☐ If the center aligned waveform is selected (OWFS:bit8 is set to "1"), the data is transferred to the duty buffer when the count value of the timer is found to be "0".

[bit14] CDSL: PWM duty selection bit

CDSL	Function
0	Select the common duty value.
1	Select the PWM duty value.

This bit is used to select whether to use the common duty setting register (PWMCMD) or PWM duty setting register (PWMCDB) as the PWM duty.

[bit13] CCFE: Cycle buffer enable bit

CCFE	Function
0	Disable the cycle buffer.
1	Enable the cycle buffer.

- This bit is used to enable the cycle buffer.
- When this bit is set to "0":

  The cycle buffer is disabled. The data written to the compare clear buffer register (PWMCPCLRB) (when MCPS:bit12 is set to "0") or to the PWM cycle setting register (PWMCCB) (when MCPS:bit12 is set to "1") is transferred immediately to the cycle buffer.
- When this bit is set to "1":

  The cycle buffer is enabled. The data written to the compare clear buffer register (PWMCPCLRB) (when MCPS:bit12 is set to "0") or to the PWM cycle setting register (PWMCCB) (when MCPS:bit12 is set to "1") is transferred to the cycle buffer in one of the following ways based on the PWM output waveform selection bit.

  - ☐ If the normal waveform is selected (OWFS:bit8 is set to "0"), the data is transferred to the cycle buffer when the count value of the timer matches the cycle value.
  - ☐ If the center aligned waveform is selected (OWFS:bit8 is set to "1"), the data is transferred to the cycle buffer when the count value of the timer is found to be "0".

[bit12] MCPS: Cycle mode selection bit

MCPS	Function
0	Master clock cycle mode
1	Independent cycle mode

- This bit is used to set the cycle mode of the PWM.
- When this bit is set to "0":

  The master clock cycle mode is selected. Therefore, the PWM cycle is the value set in the compare clear buffer register 0 (PWMCPCLRB0) when MCSSL:bit11 is set to "0" or the value set in the compare clear buffer register 1 (PWMCPCLRB1) when MCSSL:bit11 is set to "1".
- When this bit is set to "1":

  The independent cycle mode is selected. Therefore, the PWM cycle is the value set in the PWM cycle setting register (PWMCCB).

[bit11] MCSSL: Master clock selection bit

MCSSL	Function
0	Select master clock 0.
1	Select master clock 1.

- This bit is used to select the master clock to be used when the master clock cycle mode is selected (MCPS:bit12 is set to "0") and when master clock synchronization is enabled (MCSEN:bit9 is set to "1").

[bit10] PWMOD: PWM mode bit

PWMOD	Function
0	Complementary output mode
1	Independent output mode

- This bit is used to set the PWM mode.
- When this bit is set to "0":

  The complementary output mode is selected. Only the PWMPTMRx timer is put into operation. While the PWM waveform generated by PWMPTMRx is output to the high side of the PWM as is, the inverted PWM waveform is output to the low side. (x=0, 2, 4)
- When this bit is set to "1":

  The independent output mode is selected. Both the PWMPTMRx and PWMPTMRy timers are put into operation. The PWM waveform generated by PWMPTMRx is output to the high side of the PWM as is, and the PWM waveform generated by PWMPTMRy is output to the low side as is. (x=0, 2, 4 _y=1, 3, 5)

[bit9] MCSEN: Master clock synchronization enable bit

MCSEN	Function
0	Disable master clock synchronization.
1	Enable master clock synchronization.

- This bit is used to enable master clock synchronization.
- When this bit is set to "0":

  The timer is not synchronized with the master clock.
- When this bit is set to "1":

  The timer is synchronized with the master clock. The timer makes a phase shift to the value set in the PWM phase setting register (PWMCPB) at the rising edge of the master clock.

[bit8] OWFS: PWM output waveform selection bit

OWFS	Function
0	Output the normal waveform.
1	Output the center aligned waveform.

- This bit is used to select the PWM output waveform.
- When this bit is set to "0": The normal waveform is output.
- When this bit is set to "1": The center aligned waveform is output.

[bit7 to bit2] (Reserved)

[bit1] OSEL1: PWM output polarity selection bit 1

OSEL1	Description
0	Normal polarity
1	Inverse polarity

- This bit is used to select the PWM output polarity.
- When this bit is set to "0": The GPWMLn signal has normal polarity. (n=0, 1, 2)
- When this bit is set to "1": The GPWMLn signal has inverted polarity. (n=0, 1, 2)

[bit0] OSEL0: PWM output polarity selection bit 0

OSEL0	Description
0	Normal polarity
1	Inverse polarity

- This bit is used to select the PWM output polarity.
- When this bit is set to "0": The GPWMHn signal has normal polarity. (n=0, 1, 2)
- When this bit is set to "1": The GPWMHn signal has inverted polarity. (n=0, 1, 2)

Note:

Before setting the PWM control register, check that the timer is stopped.

43.4.3.4 PWM Cycle Setting Register: PWMCC0B to PWMCC5B

This section shows the bit configuration of the PWM cycle setting register.

The PWM cycle setting register (PWMCCB) sets the cycle of the PWM output waveform.

PWMCC0B: Address 31F8H (access: half word and word)

PWMCC1B: Address 3200H (access: half word and word)

PWMCC2B: Address 320CH (access: half word and word)

PWMCC3B: Address 3214H (access: half word and word)

PWMCC4B: Address 3220H (access: half word and word)

PWMCC5B: Address 3228H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D09	D08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D07	D06	D05	D04	D03	D02	D01	D00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D00: PWM cycle setting bits

D15 to D00	Function
	Cycle of the PWM output waveform

- The PWM cycle setting register has a buffer.
- When the cycle buffer is enabled (CCFE:bit13 in the PWM control register (PWMPCN) is set to "1"), data is transferred from the PWM cycle setting register to the buffer if the timer value matches the value set in the PWM cycle setting register (PWMCCB). When the cycle buffer is disabled (CCFE:bit13 in the PWM control register (PWMPCN) is set to "0"), data is transferred from the PWM cycle setting register to the buffer immediately after the data is written to the register.
- If data is written to the PWM cycle setting register, be sure to set the PWM duty setting register (PWMCDB) or common duty setting register (PWMCMD) before enabling the operation (setting CNTEn in the PWM simultaneous activation register (PWMPCGS) to "1"). (n=0, 1, 2)

Note:

If the PWM output waveform selection bit (PWMPCN.OWFS) is set to "0" (normal waveform), the waveform is output at the set PWM cycle.

If the PWM output waveform selection bit (PWMPCN.OWFS) is set to "1" (center aligned waveform), the waveform is output at twice the set PWM cycle.

When accessing the PWM cycle setting register, use a half-word or word access instruction.

43.4.3.5 PWM Phase Setting Register: PWMCP0B to PWMCP5B

This section shows the bit configuration of the PWM phase setting register.

The PWM phase setting register (PWMCPB) sets the phase of the PWM output waveform.

PWMCP0B: Address 31FAH (access: half word and word)

PWMCP1B: Address 3202H (access: half word and word)

PWMCP2B: Address 320EH (access: half word and word)

PWMCP3B: Address 3216H (access: half word and word)

PWMCP4B: Address 3222H (access: half word and word)

PWMCP5B: Address 322AH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D09	D08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D07	D06	D05	D04	D03	D02	D01	D00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D00: PWM phase setting bits

D15 to D00	Function
	Phase of the PWM output waveform

- The PWM phase setting register has a buffer.
- When master clock synchronization is enabled (MCSEN:bit9 in the PWM control register (PWMPCN) is set to "1"), data is transferred from the PWM phase setting register to the buffer if the timer value matches the value set in the PWM cycle setting register (PWMCCB). If the timer value of the PWM becomes equal to the "value of the phase setting buffer - 1", data is loaded to the timer. When master clock synchronization is disabled (MCSEN:bit9 in the PWM control register (PWMPCN) is set to "0"), data is transferred from the PWM phase setting register to the buffer immediately after the data is written to register. Data is loaded to the PWM timer at the rising edge of the master clock.
- In the PWM phase setting register, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- When master clock synchronization is enabled (MCSEN:bit9 in the PWM control register (PWMPCN) is set to "1"), a phase shift occurs starting from the rising edge of the master clock. When master clock synchronization is disabled (MCSEN:bit9 in the PWM control register (PWMPCN) is set to "0"), a phase shift occurs if the timer operation enable bit in the PWM simultaneous activation register (PWMPCGS) is asserted.

Note:

If the PWM output waveform selection bit (PWMPCN.OWFS) is set to "0" (normal waveform), the waveform is output at the set PWM phase.

If the PWM output waveform selection bit (PWMPCN.OWFS) is set to "1" (center aligned waveform), the waveform is output at twice the set PWM phase.

When accessing the PWM phase setting register, use a half-word or word access instruction.

43.4.3.6 PWM Duty Setting Register: PWMCD0B to PWMCD5B

This section shows the bit configuration of the PWM duty setting register.

The PWM duty setting register (PWMCDB) sets the duty of the PWM output waveform.

PWMCD0B: Address 31FCH (access: half word and word)

PWMCD1B: Address 3204H (access: half word and word)

PWMCD2B: Address 3210H (access: half word and word)

PWMCD3B: Address 3218H (access: half word and word)

PWMCD4B: Address 3224H (access: half word and word)

PWMCD5B: Address 322CH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D09	D08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D07	D06	D05	D04	D03	D02	D01	D00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D00: PWM duty setting bits

D15 to D00	Function
	Duty of the PWM output waveform

- The PWM duty setting register has a buffer.
- When the duty buffer is enabled (CDFE:bit15 in the PWM control register (PWMPCN) is set to "1"), data is transferred from the PWM duty setting register to the buffer if the timer value matches the value set in the PWM cycle setting register (PWMCCB). When the duty buffer is disabled (CDFE:bit15 in the PWM control register (PWMPCN) is set to "0"), data is transferred from the PWM duty setting register to the buffer immediately after the data is written to the register.
- In the PWM duty setting register, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).

Note:

If the PWM output waveform selection bit (PWMPCN.OWFS) is set to "0" (normal waveform), the waveform is output at the set PWM duty.

If the PWM output waveform selection bit (PWMPCN.OWFS) is set to "1" (center aligned waveform), the waveform is output at twice the set PWM duty.

When accessing the PWM duty setting register, use a half-word or word access instruction.

43.4.3.7 PWM Timer Register: PWMPTMR0 to PWMPTMR5

This section shows the bit configuration of the PWM timer register.

The PWM timer register (PWMPTMR) reads the count value of the PWM timer.

PWMPTMR0: Address 31FEH (access: half word and word)

PWMPTMR1: Address 3206H (access: half word and word)

PWMPTMR2: Address 3212H (access: half word and word)

PWMPTMR3: Address 321AH (access: half word and word)

PWMPTMR4: Address 3226H (access: half word and word)

PWMPTMR5: Address 322EH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D09	D08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D07	D06	D05	D04	D03	D02	D01	D00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15 to bit0] D15 to D00: PWM timer value bits

D15 to D00	Function
	Count value of the timer

- The count value of the PWM timer can be read.
- This is a read only register.

Note:

When accessing the PWM timer register, use a half-word or word access instruction.

43.4.4 Fault Function-related Registers

This section explains the fault function-related registers.

The fault function-related registers are the fault control register, fault reset control register, fault capture control register, fault soft reset register, capture interrupt threshold setting register, fault reset delay control register, fault capture counter reset delay control register, and fault capture data register.

43.4.4.1 Fault Control Register: PWMFLTCON00 to PWMFLTCON21

This section shows the bit configuration of the fault control register.

The fault control register (PWMFLTCON) is used to control the fault operation.

PWMFLTCON00: Address 3230H (access: byte, half word, and word)

PWMFLTCON10: Address 3250H (access: byte, half word, and word)

PWMFLTCON20: Address 3270H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CFHSL3	CFHSL2	CFHSL1	CFHSL0	FLT0H1	FLT0H0	FLT0L1	FLT0L0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	IFLTE0	LAT0	BLS01	BLS00	Reserved	FLS01	FLS00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R/W	R0,WX	R/W	R/W

[bit15 to bit12] CFHSL3 to CFHSL0: Capture counter reset factor selection bits

CFHSL3	CFHSL2	CFHSL1	CFHSL0	Function
0	0	0	0	No factor selected (counter not reset)
:				
0	1	1	1	
1	0	0	0	Rising edge of PWM0H
1	0	0	1	Rising edge of PWM0L
1	0	1	0	Rising edge of PWM1H
1	0	1	1	Rising edge of PWM1L
1	1	0	0	Rising edge of PWM2H
1	1	0	1	Rising edge of PWM2L
1	1	1	0	Rising edge of master clock 0
1	1	1	1	Rising edge of master clock 1

These bits are used to set the reset factor for the fault capture counter.

[bit11, bit10] FLT0H1, FLT0H0: FPWMH fault operation selection bits

FLT0H1	FLT0H0	Function
0	0	Do not change the output level.
0	1	Inverse the output level.
1	0	Fix the output level to "L".
1	1	Fix the output level to "H".

- These bits are used to set how to change the output level of the high-side PWM (FPWMH) in the event of a fault.

- Make sure that both the high-side and low-side PWMs do not output the "H" level.
- If these bits are set to "01", the fault input is masked by the re-inversion prevention function from the cycle before the one in which the GPWMH change point is detected until the fault latch is reset.
- The re-inversion prevention function is enabled only when the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0").
- The re-inversion prevention function is also enabled when the fault latch is not used. If these bits are set to "01", make sure that the fault latch is reset.

[bit9, bit8] FLT0L1, FLT0L0: FPWML fault operation selection bits

FLT0L1	FLT0L0	Function
0	0	Do not change the output level.
0	1	Inverse the output level.
1	0	Fix the output level to "L".
1	1	Fix the output level to "H".

- These bits are used to set how to change the output level of the low-side PWM (FPWML) in the event of a fault.
- Make sure that both the high-side and low-side PWMs do not output the "H" level.
- If these bits are set to "01", the fault input is masked by the re-inversion prevention function from the cycle before the one in which the GPWMH change point is detected until the fault latch is reset.
- The re-inversion prevention function is enabled only when the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0").
- The re-inversion prevention function is also enabled when the fault latch is not used. If these bits are set to "01", make sure that the fault latch is reset.

[bit7] (Reserved)

[bit6] IFLTE0: Interrupt request enable bit 0

IFLTE0	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for fault interrupt 0.
- If this bit and interrupt flag bit 0 (IFLTF0:bit0) in the PWM common status register (PWMST) are set to "1", an interrupt request for the CPU is generated.

[bit5] LAT0: Fault latch bit 0

LAT0	Function
0	Do not use the fault latch.
1	Use the fault latch.

- This bit is used to enable the use of the fault latch.
- When this bit is set to "0":

  The fault input selected by the fault selection bits (FLS01, FLS00:bit1, bit0) is not latched. Therefore, the fault operation ends when the fault input is negated.
- When this bit is set to "1":

The fault input selected by the fault selection bits (FLS01, FLS00:bit1, bit0) is latched. Therefore, the fault operation continues until the fault latch is reset.

Note:

Before setting this bit to "1", reset fault latch 0 using the fault soft reset register (PWMFLTSR).

[bit4, bit3] BLS01, BLS00: Blanking selection bits 0

BLS01	BLS00	Function
0	0	Select blanking 0 (LEB0).
0	1	Select blanking 1 (LEB1).
1	0	Select blanking 2 (LEB2).
1	1	Disable blanking.

These bits are used to select how long the fault input selected by the fault selection bits (FLS01, FLS00:bit1, bit0) are to be masked.

[bit2] (Reserved)

[bit1, bit0] FLS01, FLS00: Fault selection bits 0

FLS01	FLS00	Function
0	0	Do not perform the fault operation.
0	1	Input CMP0.
1	0	Input CMP1.
1	1	Input CMP2.

These bits are used to set the fault input of the fault operation.

Note:

Before setting any bit other than the interrupt request enable bit 0 (IFLTE0:bit6) in the fault control register, check that the master clock timer and PWM timer are stopped.

PWMFLTCON01: Address 3232H (access: byte, half word, and word)

PWMFLTCON11: Address 3252H(access: byte, half word, and word)

PWMFLTCON21: Address 3272H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CFLSL3	CFLSL2	CFLSL1	CFLSL0	FLT1H1	FLT1H0	FLT1L1	FLT1L0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	IFLTE1	LAT1	BLS11	BLS10	Reserved	FLS11	FLS10
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R/W	R0,WX	R/W	R/W

[bit15 to bit12] CFLSL3 to CFLSL0: Capture counter reset factor selection bits

CFLSL3	CFLSL2	CFLSL1	CFLSL0	Function
0	0	0	0	No factor selected (counter not reset)
:				
0	1	1	1	
1	0	0	0	Falling edge of PWM0H
1	0	0	1	Falling edge of PWM0L
1	0	1	0	Falling edge of PWM1H
1	0	1	1	Falling edge of PWM1L
1	1	0	0	Falling edge of PWM2H
1	1	0	1	Falling edge of PWM2L
1	1	1	0	Falling edge of master clock 0
1	1	1	1	Falling edge of master clock 1

These bits are used to set the reset factor for the fault capture counter.

[bit11, bit10] FLT1H1, FLT1H0: FPWMH fault operation (preferred) selection bits

FLT1H1	FLT1H0	Function
0	0	Do not change the output level (through).
0	1	
1	0	Fix the output level to "L".
1	1	Fix the output level to "H".

- These bits are used to set how to change the output level of the high-side PWM (FPWMH) in the event of a fault.
- This fault operation (preferred) takes precedence over the fault operation set by the FPWMH fault operation selection bits (FLT0H1, FLT0H0:bit11, bit10) in PWMFLTCON0x (where x = 0 to 2).
- Make sure that both the high-side and low-side PWMs do not output the "H" level.

[bit9, bit8] FLT1L1, FLT1L0: FPWML fault operation (preferred) selection bits

FLT1L1	FLT1L0	Function
0	0	Do not change the output level (through).
0	1	
1	0	Fix the output level to "L".
1	1	Fix the output level to "H".

- These bits are used to set how to change the output level of the low-side PWM (FPWML) in the event of a fault.
- This fault operation (preferred) takes precedence over the fault operation set by the FPWML fault operation selection bits (FLT0L1, FLT0L0:bit9, bit8) in PWMFLTCON0x (where x = 0 to 2).
- Make sure that both the high-side and low-side PWMs do not output the "H" level.

[bit7] (Reserved)

[bit6] IFLTE1: Interrupt request enable bit 1

IFLTE1	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for fault interrupt 1.
- If this bit and interrupt flag bit 1 (IFLTF1:bit1) in the PWM common status register (PWMST) are set to "1", an interrupt request for the CPU is generated.

[bit5] LAT1: Fault latch bit 1

LAT1	Function
0	Do not use the fault latch.
1	Use the fault latch.

- This bit is used to enable the use of the fault latch.
- When this bit is set to "0":

  The fault input selected by the fault selection bits (FLS11, FLS10:bit1, bit0) is not latched. Therefore, the fault operation (preferred) ends when the fault input is negated.
- When this bit is set to "1":

  The fault input selected by the fault selection bits (FLS11, FLS10:bit1, bit0) is latched. Therefore, the fault operation (preferred) continues until the fault latch is reset.

Note:

Before setting this bit to "1", reset fault latch 1 using the fault soft reset register (PWMFLTSR).

[bit4, bit3] BLS11, BLS10: Blanking selection bits 1

BLS11	BLS10	Function
0	0	Select blanking 0 (LEB0).
0	1	Select blanking 1 (LEB1).
1	0	Select blanking 2 (LEB2).
1	1	Disable blanking.

These bits are used to select how long the fault input selected by the fault selection bits (FLS11, FLS10:bit1, bit0) are to be masked.

[bit2] (Reserved)

[bit1, bit0] FLS11, FLS10: Fault selection bits 1

FLS11	FLS10	Function
0	0	Do not perform the fault operation.
0	1	Input CMP0.
1	0	Input CMP1.
1	1	Input CMP2.

These bits are used to set the fault input of the fault operation (preferred).

Note:

Before setting any bit other than the interrupt request enable bit 1 (IFLTE1:bit6) in the fault control register, check that the master clock timer and PWM timer are stopped.

43.4.4.2 Fault Reset Control Register: PWMFLTRCON0 to PWMFLTRCON2

This section shows the bit configuration of the fault reset control register.

The fault reset control register (PWMFLTRCON) is used to control the reset of the fault latch.

PWMFLTRCON0: Address 3234H (access: byte, half word, and word)

PWMFLTRCON1: Address 3254H (access: byte, half word, and word)

PWMFLTRCON2: Address 3274H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	FHSL02	FHSL01	FHSL00	Reserved	FLSL02	FLSL01	FLSL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	FHSL12	FHSL11	FHSL10	Reserved	FLSL12	FLSL11	FLSL10
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W

[bit15] (Reserved)

[bit14 to bit12] FHSL02 to FHSL00: Fault reset factor selection bits

FHSL02	FHSL01	FHSL00	Function
0	0	0	Rising edge of PWM channel 0 (GPWMH0)
0	0	1	Rising edge of PWM channel 1 (GPWML0)
0	1	0	Rising edge of PWM channel 2 (GPWMH1)
0	1	1	Rising edge of PWM channel 3 (GPWML1)
1	0	0	Rising edge of PWM channel 4 (GPWMH2)
1	0	1	Rising edge of PWM channel 5 (GPWML2)
1	1	0	Rising edge of master clock 0
1	1	1	Rising edge of master clock 1

These bits are used to set the reset factor for the fault latch on the fault operation side.

[bit11] (Reserved)

[bit10 to bit8] FLSL02 to FLSL00: Fault reset factor selection bits

FLSL02	FLSL01	FLSL00	Function
0	0	0	Falling edge of PWM channel 0 (GPWMH0)
0	0	1	Falling edge of PWM channel 1 (GPWML0)
0	1	0	Falling edge of PWM channel 2 (GPWMH1)
0	1	1	Falling edge of PWM channel 3 (GPWML1)
1	0	0	Falling edge of PWM channel 4 (GPWMH2)
1	0	1	Falling edge of PWM channel 5 (GPWML2)

FLSL02	FLSL01	FLSL00	Function
1	1	0	Falling edge of master clock 0
1	1	1	Falling edge of master clock 1

These bits are used to set the reset factor for the fault latch on the fault operation side.

[bit7] (Reserved)

[bit6 to bit4] FHSL12 to FHSL10: Fault reset (preferred) factor selection bits

FHSL12	FHSL11	FHSL10	Function
0	0	0	Rising edge of PWM channel 0 (GPWMH0)
0	0	1	Rising edge of PWM channel 1 (GPWML0)
0	1	0	Rising edge of PWM channel 2 (GPWMH1)
0	1	1	Rising edge of PWM channel 3 (GPWML1)
1	0	0	Rising edge of PWM channel 4 (GPWMH2)
1	0	1	Rising edge of PWM channel 5 (GPWML2)
1	1	0	Rising edge of master clock 0
1	1	1	Rising edge of master clock 1

These bits are used to set the reset factor for the fault latch on the fault operation (preferred) side.

[bit3] (Reserved)

[bit2 to bit0] FLSL12 to FLSL10: Fault reset (preferred) factor selection bits

FLSL12	FLSL11	FLSL10	Function
0	0	0	Falling edge of PWM channel 0 (GPWMH0)
0	0	1	Falling edge of PWM channel 1 (GPWML0)
0	1	0	Falling edge of PWM channel 2 (GPWMH1)
0	1	1	Falling edge of PWM channel 3 (GPWML1)
1	0	0	Falling edge of PWM channel 4 (GPWMH2)
1	0	1	Falling edge of PWM channel 5 (GPWML2)
1	1	0	Falling edge of master clock 0
1	1	1	Falling edge of master clock 1

These bits are used to set the reset factor for the fault latch on the fault operation (preferred) side.

Note:

Before setting the fault reset control register, check that the master clock timer and PWM timer are stopped.

43.4.4.3 Fault Capture Control Register: PWMFLTCAPCON0 to PWMFLTCAPCON2

This section shows the bit configuration of the fault capture control register.

The fault capture control register (PWMFLTCAPCON) is used to control the reset of the fault latch as well as fault capture.

PWMFLTCAPCON0: Address 3236H (access: byte, half word, and word)

PWMFLTCAPCON1: Address 3256H (access: byte, half word, and word)

PWMFLTCAPCON2: Address 3276H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved		LRST1L	LRST1H	Reserved		LRST0L	LRST0H
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	CEHSL2	CEHSL1	CEHSL0	CAPIE	CELSL2	CELSL1	CELSL0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15, bit14] (Reserved)

[bit13] LRST1L: Falling edge starting point fault reset (preferred) enable bit

LRST1L	Function
0	Disable the falling edge starting point fault reset (preferred).
1	Enable the falling edge starting point fault reset (preferred).

- This bit is used to enable the fault latch to be reset by the fault reset (preferred) factor selected by the fault reset (preferred) factor selection bits (FLSL12 to FLSL10:bit2 to bit0) in the fault reset control register (PWMFLTRCON).
- If fault reset (preferred) and fault set occur simultaneously, fault set takes precedence.

[bit12] LRST1H: Rising edge starting point fault reset (preferred) enable bit

LRST1H	Function
0	Disable the rising edge starting point fault reset (preferred).
1	Enable the rising edge starting point fault reset (preferred).

- This bit is used to enable the fault latch to be reset by the fault reset (preferred) factor selected by the fault reset (preferred) factor selection bits (FHSL12 to FHSL10:bit6 to bit4) in the fault reset control register (PWMFLTRCON).
- If fault reset (preferred) and fault set occur simultaneously, fault set takes precedence.

[bit11, bit10] (Reserved)

[bit9] LRST0L: Falling edge starting point fault reset enable bit

LRST0L	Function
0	Disable the falling edge starting point fault reset.
1	Enable the falling edge starting point fault reset.

- This bit is used to enable the fault latch to be reset by the fault reset factor selected by the fault reset factor selection bits (FLSL02 to FLSL00:bit10 to bit8) in the fault reset control register (PWMFLTRCON). If the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is

set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), this bit is used to cancel the re-inversion prevention function.

- If fault reset and fault set occur simultaneously, fault set takes precedence.

[bit8] LRST0H: Rising edge starting point fault reset enable bit

LRST0H	Function
0	Disable the rising edge starting point fault reset.
1	Enable the rising edge starting point fault reset.

- This bit is used to enable the fault latch to be reset by the fault reset factor selected by the fault reset factor selection bits (FHSL02 to FHSL00:bit14 to bit12) in the fault reset control register (PWMFLTRCON).
- If fault reset and fault set occur simultaneously, fault set takes precedence. If the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), this bit is used to cancel the re-inversion prevention function.

[bit7] (Reserved)

[bit6 to bit4] CEHSL2 to CEHSL0: High-side PWM capture edge selection bits

CEHSL2	CEHSL1	CEHSL0	Function
0	0	0	Do not capture.
0	0	1	PWM rising edge (FPWMH) based on the fault operation
0	1	0	PWM falling edge (FPWMH) based on the fault operation
0	1	1	PWM rising edge (FPWMH) based on the fault operation (preferred)
1	0	0	PWM falling edge (FPWMH) based on the fault operation (preferred)
1	0	1	Setting prohibited
1	1	0	
1	1	1	

- These bits are used to set the timing of capturing the capture counter value into the high-side PWM fault capture data (HCAP15 to HCAP00:bit31 to bit16 in PWMFLTCAPD).
- If the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function from the cycle before the one in which the GPWMH change point is detected until the fault latch is reset. Therefore, if CEHSL2 to CEHSL0 are set to "001" or "010", the capture counter value is not captured.

Note:

If the selected fault operation or fault operation (preferred) is "Do not change the output level" (in the fault control register (PWMFLTCON), FLT0H1, FLT0H0:bit11, bit10 or FLT0L1, FLT0L0:bit9, bit8 are set to "00" or FLT1H1:bit11 and FLT1L1:bit9 are set to "0"), the edge selected by CEHSL2 to CEHSL0 is not generated.

[bit3] CAPIE: Interrupt request enable bit

CAPIE	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for the capture interrupt.
- If this bit and the interrupt flag bit (CAPIF:bit3) in the PWM common status register (PWMST) are set to "1", an interrupt request for the CPU is generated.

[bit2 to bit0] CELSL2 to CELSL0: Low-side PWM capture edge selection bits

CELSL2	CELSL1	CELSL0	Function
0	0	0	Do not capture.
0	0	1	PWM rising edge (FPWML) based on the fault operation
0	1	0	PWM falling edge (FPWML) based on the fault operation
0	1	1	PWM rising edge (FPWML) based on the fault operation (preferred)
1	0	0	PWM falling edge (FPWML) based on the fault operation (preferred)
1	0	1	Setting prohibited
1	1	0	
1	1	1	

- These bits are used to set the timing of capturing the capture counter value into the low-side PWM fault capture data (LCAP15 to LCAP00:bit15 to bit0 in PWMFLTCAPD).
- If the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function. Therefore, if CELSL2 to CELSL0 are set to "001" or "010", the capture counter value is not captured.

Note:

If the selected fault operation or fault operation (preferred) is "Do not change the output level" (in the fault control register (PWMFLTCON), FLT0H1, FLT0H0:bit11, bit10 or FLT0L1, FLT0L0:bit9, bit8 are set to "00" or FLT1H1:bit11 and FLT1L1:bit9 are set to "0"), the edge selected by CELSL2 to CELSL0 is not generated.

43.4.4.4 Fault Soft Reset Register: PWMFLTSR0 to PWMFLTSR2

This section shows the bit configuration of the fault soft reset register.

The fault soft reset register (PWMFLTSR) is used to control the reset of the fault latch.

PWMFLTSR0: Address 3238H (access: byte, half word, and word)

PWMFLTSR1: Address 3258H (access: byte, half word, and word)

PWMFLTSR2: Address 3278H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							FSRST1
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							FSRST0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W

[bit15 to bit9] (Reserved)

[bit8] FSRST1: Soft reset bit 1

FSRST1	Function
0	There is no effect on operation.
1	Reset the fault latch.

- This bit is used to reset the fault latch (preferred).
- This is a write only bit.
- The read value is "0".
- If soft reset and fault set occur simultaneously, fault set takes precedence.

[bit7 to bit1] (Reserved)

[bit0] FSRST0: Soft reset bit 0

FSRST0	Function
0	There is no effect on operation.
1	Reset the fault latch.

- This bit is used to reset the fault latch.
- This is a write only bit.
- The read value is "0".
- If soft reset and fault set occur simultaneously, fault set takes precedence.
- If the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), this bit is used to cancel the re-inversion prevention function.

Note:

Before restarting the PWM, reset the fault latch by using the soft reset (setting FSRST0 and FSTST1 to "1").

If soft reset instructions are written consecutively (by setting FSRST0 and FSTST1 to "1"), the soft reset instruction written before the first fault latch reset ends is ignored.

43.4.4.5 Capture Interrupt Threshold Setting Register: PWMCAPITH0 to PWMCAPITH2

This section shows the bit configuration of the capture interrupt threshold setting register.

The capture interrupt threshold setting register (PWMCAPITH) is used to control the capture interrupt.

PWMCAPITH0: Address 323CH (access: half word and word)

PWMCAPITH1: Address 325CH (access: half word and word)

PWMCAPITH2: Address 327CH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D09	D08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D07	D06	D05	D04	D03	D02	D01	D00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] D15 to D00: Threshold bits

D15 to D00	Function
	Set the capture interrupt threshold.

- These bits are used to set the threshold to control the capture interrupt.
- If the difference between the high-side capture data bits (HCAP15 to HCAP00:bit31 to bit16) in the fault capture data register (PWMFLTCAPD) and the low-side capture data bits (LCAP15 to LCAP00:bit15 to bit0) exceeds the value set in the capture interrupt threshold setting register (PWMCAPITH), the capture interrupt flag is set to "1".
- The difference between the high-side capture data and low-side capture data is calculated and compared with the threshold at the timing selected by high-side PWM capture edge selection bits (CEHSL2 to CEHSL0:bit6 to bit4) and low-side PWM capture edge selection bits (CELSL2 to CELSL0:bit2 to bit0) in the fault capture control register (PWMFLTCAPCON).

Note:

When accessing the capture interrupt threshold setting register, use a half-word or word access instruction.

43.4.4.6 Fault Reset Delay Control Register: PWMFLTRDCON00 to PWMFLTRDCON21

This section shows the bit configuration of the fault reset delay control register.

The fault reset delay control register (PWMFLTRDCON) is used to control the reset of the fault latch.

PWMFLTRDCON00: Address 3240H (access: half word and word)

PWMFLTRDCON10: Address 3260H (access: half word and word)

PWMFLTRDCON20: Address 3280H (access: half word and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	FHDLY15	FHDLY14	FHDLY13	FHDLY12	FHDLY11	FHDLY10	FHDLY09	FHDLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	FHDLY07	FHDLY06	FHDLY05	FHDLY04	FHDLY03	FHDLY02	FHDLY01	FHDLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	FLDLY15	FLDLY14	FLDLY13	FLDLY12	FLDLY11	FLDLY10	FLDLY09	FLDLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FLDLY07	FLDLY06	FLDLY05	FLDLY04	FLDLY03	FLDLY02	FLDLY01	FLDLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit16] FHDLY15 to FHDLY00: Fault reset delay bits

FHDLY15 to FHDLY00	Function
	Set the amount of delay for the fault reset.

- These bits are used to set the amount of delay from the starting point of the fault reset factor selected by the fault reset factor selection bits (FHSL02 to FHSL00:bit14 to bit12) in the fault reset control register (PWMFLTRCON) to the reset of the fault latch.

  Delay time = (set value) × PWM division clock cycle

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- The set value is reflected on the timer based on the timing of the fault reset factor selected by the fault reset factor selection bits (FHSL02 to FHSL00:bit14 to bit12) in the fault reset control register (PWMFLTRCON).

[bit15 to bit0] FLDLY15 to FLDLY00: Fault reset delay bits

FLDLY15 to FLDLY00	Function
	Set the amount of delay for the fault reset.

- These bits are used to set the amount of delay from the starting point of the fault reset factor selected by the fault reset factor selection bits (FLSL02 to FLSL00:bit10 to bit8) in the fault reset control register (PWMFLTRCON) to the reset of the fault latch.

Delay time = (set value) × PWM division clock cycle

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- The set value is reflected on the timer based on the timing of the fault reset factor selected by the fault reset factor selection bits (FLSL02 to FLSL00:bit10 to bit8) in the fault reset control register (PWMFLTRCON).

Note:

When accessing the fault reset delay control register, use a half-word or word access instruction.

Before making a change to the fault reset delay control register, check that the PWM timer is stopped.

The phase of the master clock that can be selected by the fault reset factor selection bits (FLSL02 to FLSL00 in the fault reset control register (PWMFLTRCON)) or the fault reset (preferred) factor selection bits (FLSL12 to FLSL10 in the fault reset control register (PWMFLTRCON)) corresponds to that applied when the phase of the PWM output is set to "0000_H". Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0000_H" corresponds to the start of the PWM output at the rising edge of the master clock.

PWMFLTRDCON01: Address 3244H (access: half word and word)

PWMFLTRDCON11: Address 3264H (access: half word and word)

PWMFLTRDCON21: Address 3284H (access: half word and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	FHDLY15	FHDLY14	FHDLY13	FHDLY12	FHDLY11	FHDLY10	FHDLY09	FHDLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	FHDLY07	FHDLY06	FHDLY05	FHDLY04	FHDLY03	FHDLY02	FHDLY01	FHDLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	FLDLY15	FLDLY14	FLDLY13	FLDLY12	FLDLY11	FLDLY10	FLDLY09	FLDLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FLDLY07	FLDLY06	FLDLY05	FLDLY04	FLDLY03	FLDLY02	FLDLY01	FLDLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit16] FHDLY15 to FHDLY00: Fault reset (preferred) delay bits

FHDLY15 to FHDLY00	Function
	Set the amount of delay for the fault reset.

- These bits are used to set the amount of delay from the starting point of the fault reset factor selected by the fault reset (preferred) factor selection bits (FHSL12 to FHSL10:bit6 to bit4) in the fault reset control register (PWMFLTRCON) to the reset of the fault latch.

  Delay time = (set value) × PWM division clock cycle

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).

- The set value is reflected on the timer based on the timing of the fault reset factor selected by the fault reset (preferred) factor selection bits (FHSL12 to FHSL10:bit6 to bit4) in the fault reset control register (PWMFLTRCON).

[bit15 to bit0] FLDLY15 to FLDLY00: Fault reset (preferred) delay bits

FLDLY15 to FLDLY00	Function
	Set the amount of delay for the fault reset.

- These bits are used to set the amount of delay from the starting point of the fault reset factor selected by the fault reset (preferred) factor selection bits (FLSL12 to FLSL10:bit2 to bit0) in the fault reset control register (PWMFLTRCON) to the reset of the fault latch.

  Delay time = (set value) × PWM division clock cycle

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- The set value is reflected on the timer based on the timing of the fault reset factor selected by the fault reset (preferred) factor selection bits (FLSL12 to FLSL10:bit2 to bit0) in the fault reset control register (PWMFLTRCON).

Note:

When accessing the fault reset delay control register, use a half-word or word access instruction.

Before making a change to the fault reset delay control register, check that the PWM timer is stopped.

The phase of the master clock that can be selected by the fault reset (preferred) factor selection bits (FLSL12 to FLSL10 in the fault reset control register (PWMFLTRCON)) corresponds to that applied when the phase of the PWM output is set to "0000_H". Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0000_H" corresponds to the start of the PWM output at the rising edge of the master clock.

43.4.4.7 Fault Capture Counter Reset Delay Control Register: PWMFLTCAPRDCON0 to PWMFLTCAPRDCON2

This section shows the bit configuration of the fault capture counter reset delay control register.

The fault capture counter reset delay control register (PWMFLTCAPRDCON) is used to control the reset of the fault capture counter.

PWMFLTCAPRDCON0: Address 3248H (access: half word and word)

PWMFLTCAPRDCON1: Address 3268H (access: half word and word)

PWMFLTCAPRDCON2: Address 3288H (access: half word and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	FHDLY15	FHDLY14	FHDLY13	FHDLY12	FHDLY11	FHDLY10	FHDLY09	FHDLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	FHDLY07	FHDLY06	FHDLY05	FHDLY04	FHDLY03	FHDLY02	FHDLY01	FHDLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	FLDLY15	FLDLY14	FLDLY13	FLDLY12	FLDLY11	FLDLY10	FLDLY09	FLDLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FLDLY07	FLDLY06	FLDLY05	FLDLY04	FLDLY03	FLDLY02	FLDLY01	FLDLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit16] FHDLY15 to FHDLY00: Counter reset delay bits

FHDLY15 to FHDLY00	Function
	Set the amount of delay for the capture counter reset.

- These bits are used to set the amount of delay from the starting point of the capture counter reset factor selected by the capture counter reset factor selection bits (CFHSL3 to CFHSL0:bit15 to bit12) in the fault control register (PWMFLTCON) to the reset of the capture counter.

  Delay time = (set value) × PWM division clock cycle

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- The set value is reflected on the timer based on the timing of the capture counter reset factor selected by the capture counter reset factor selection bits (CFHSL3 to CFHSL0:bit15 to bit12) in the fault reset control register (PWMFLTRCON).

[bit15 to bit0] FLDLY15 to FLDLY00: Counter reset delay bits

FLDLY15 to FLDLY00	Function
	Set the amount of delay for the capture counter reset.

- These bits are used to set the amount of delay from the starting point of the capture counter reset factor selected by the capture counter reset factor selection bits (CFLSL3 to CFLSL0:bit15 to bit12) in the fault control register (PWMFLTCON) to the reset of the capture counter.
  - Delay time = (set value) × PWM division clock cycle
- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- The set value is reflected on the timer based on the timing of the capture counter reset factor selected by the capture counter reset factor selection bits (CFLSL3 to CFLSL0:bit15 to bit12) in the fault reset control register (PWMFLTRCON).

Note:

When accessing the fault capture counter reset delay control register, use a half-word or word access instruction.

Before making a change to the fault capture counter reset delay control register, check that the PWM timer is stopped.

When the phase of the PWM output is set to "0000_H", the phase of the master clock that can be selected by the capture counter reset factor selection bits (CFHSL3 to CFHSL0 and CFLSL3 to CFLSL0 in the fault control register (PWMFLTCON)) is displaced forward by 1 PWM division clock cycle. Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0001_H" corresponds to the start of the PWM output at the rising edge of the master clock.

43.4.4.8 Fault Capture Data Register: PWMFLTCAPD0 to PWMFLTCAPD2

This section shows the bit configuration of the fault capture data register.

The fault capture data register (PWMFLTCAPD) reads the capture data of the capture counter in the event of a fault.

PWMFLTCAPD0: Address 324CH (access: half word and word)

PWMFLTCAPD1: Address 326CH (access: half word and word)

PWMFLTCAPD2: Address 328CH (access: half word and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	HCAP15	HCAP14	HCAP13	HCAP12	HCAP11	HCAP10	HCAP09	HCAP08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	HCAP07	HCAP06	HCAP05	HCAP04	HCAP03	HCAP02	HCAP01	HCAP00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	LCAP15	LCAP14	LCAP13	LCAP12	LCAP11	LCAP10	LCAP09	LCAP08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LCAP07	LCAP06	LCAP05	LCAP04	LCAP03	LCAP02	LCAP01	LCAP00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit16] HCAP15 to HCAP00: High-side capture data bits

HCAP15 to HCAP00	Function
	Capture value of the capture counter

- These bits are used to read the capture counter value captured at the timing selected by the high-side PWM capture edge selection bits (CEHSL2 to CEHSL0:bit6 to bit4) in the fault capture control register (PWMFLTCAPCON).
- The time from the reset of the capture counter to the high-side fault operation or fault operation (preferred) can be measured.
- This is a read only register.

[bit15 to bit0] LCAP15 to LCAP00: Low-side capture data bits

LCAP15 to LCAP00	Function
	Capture value of the capture counter

- These bits are used to read the capture counter value captured at the timing selected by the low-side PWM capture edge selection bits (CELSL2 to CELSL0:bit2 to bit0) in the fault capture control register (PWMFLTCAPCON).
- The time from the reset of the capture counter to the low-side fault operation or fault operation (preferred) can be measured.
- This is a read only register.

Note:

When accessing the fault capture data register, use a half-word or word access instruction.

43.4.5 Soft Overwrite Register

This section explains the soft overwrite register.

The soft overwrite register is the soft overwrite control register.

43.4.5.1 Soft Overwrite Control Register: PWMSOWCON0 to PWMSOWCON2

This section shows the bit configuration of the soft overwrite control register.

The soft overwrite control register (PWMSOWCON) is used to control soft overwrite.

PWMSOWCON0: Address 3290H (access: byte, half word, and word)

PWMSOWCON1: Address 3294H (access: byte, half word, and word)

PWMSOWCON2: Address 3298H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	OSL11	OSL10	OSL01	OSL00	OSLBFE	STSL2	STSL1	STSL0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							SOWIE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	STDLY15	STDLY14	STDLY13	STDLY12	STDLY11	STDLY10	STDLY09	STDLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STDLY07	STDLY06	STDLY05	STDLY04	STDLY03	STDLY02	STDLY01	STDLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31, bit30] OSL11, OSL10: Overwrite selection bits 1

OSL11	OSL10	Function
0	0	Do not change the output level.
0	1	
1	0	Fix the output level to "L".
1	1	Fix the output level to "H".

- These bits are used to fix the level of the low-side PWM (OPWML) output.
- If the OSL buffer is disabled (OSLBFE:bit27 is set to "0"), the set value is reflected on the PWM output immediately after it is written. If the OSL buffer is enabled (OSLBFE:bit27 is set to "1"), the set value is reflected on the PWM output according to the timing set by the overwrite trigger selection bits (STSL2 to STSL0:bit26 to bit24) and trigger delay bits (STDLY15 to STDLY00:bit15 to bit0).
- Make sure that both the high-side and low-side PWMs do not output the "H" level.

[bit29, bit28] OSL01, OSL00: Overwrite selection bits 0

OSL01	OSL00	Function
0	0	Do not change the output level.
0	1	
1	0	Fix the output level to "L".
1	1	Fix the output level to "H".

- These bits are used to fix the level of the high-side PWM (OPWMH) output.
- If the OSL buffer is disabled (OSLBFE:bit27 is set to "0"), the set value is reflected on the PWM output immediately after it is written. If the OSL buffer is enabled (OSLBFE:bit27 is set to "1"), the set value is reflected on the PWM output according to the timing set by the overwrite trigger selection bits (STSL2 to STSL0:bit26 to bit24) and trigger delay bits (STDLY15 to STDLY00:bit15 to bit0).
- Make sure that both the high-side and low-side PWMs do not output the "H" level.

[bit27] OSLBFE: OSL buffer enable bit

OSLBFE	Function
0	Disable the OSL buffer.
1	Enable the OSL buffer.

- This bit is used to enable the OSL buffer.
- When this bit is set to "0":

  The OSL buffer is disabled. Therefore, the data written in the overwrite selection bits (OSL1 to OSL0:bit31 to bit28) is transferred to the OSL buffer immediately.
- When this bit is set to "1":

  The OSL buffer is enabled. The data written in the overwrite selection bits (OSL1 to OSL0:bit31 to bit28) is transferred to the OSL buffer according to the timing set by the overwrite trigger selection bits (STSL2 to STSL0:bit26 to bit24) and trigger delay bits (STDLY15 to STDLY00:bit15 to bit0).

[bit26 to bit24] STSL2 to STSL0: Overwrite trigger selection bits

STSL2	STSL1	STSL0	Function
0	0	0	Rising edge of the high-side PWM (GPWMH)
0	0	1	Falling edge of the high-side PWM (GPWMH)
0	1	0	Rising edge of the low-side PWM (GPWML)
0	1	1	Falling edge of the low-side PWM (GPWML)
1	0	0	Rising edge of master clock 0
1	0	1	Falling edge of master clock 0
1	1	0	Rising edge of master clock 1
1	1	1	Falling edge of master clock 1

- These bits are used to set the trigger for updating the overwrite selection bits (OSL1 to OSL0:bit31 to bit28).

[bit23 to bit17] (Reserved)

[bit16] SOWIE: Interrupt request enable bit

SOWIE	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for the SOW interrupt.
- If this bit and the interrupt flag bit (SOWIF:bit2) in the PWM common status register (PWMST) are set to "1", an interrupt request for the CPU is generated.

[bit15 to bit0] STDLY15 to STDLY00: Trigger delay bits

STDLY15 to STDLY00	Function
	Set the amount of delay for the OSL update trigger.

- These bits are used to set the amount of delay from the starting point of the trigger selected by the overwrite trigger selection bits (STSL2 to STSL0:bit26 to bit24) to the update of the overwrite selection bits (OSL1 to OSL0:bit31 to bit28).

  Delay time = (set value) × PWM division clock cycle

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- The set value is reflected on the timer according to the timing of the trigger selected by the overwrite trigger selection bits (STSL2 to STSL0:bit26 to bit24).

Note:

When accessing these bits, use a half-word or word access instruction.

The phase of the master clock that can be selected by the overwrite trigger selection bits (STSL2 to STSL0) corresponds to that applied when the phase of the PWM output is set to "0000_H". Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0000_H" corresponds to the start of the PWM output at the rising edge of the master clock.

43.4.6 Dead Time Registers

This section explains the dead time registers.

The dead time registers are the dead time mode register, high-side rising edge dead time setting register, high-side falling edge dead time setting register, low-side rising edge dead time setting register, and low-side falling edge dead time setting register.

43.4.6.1 Dead Time Mode Register: PWMDMOD

This section shows the bit configuration of the dead time mode register.

The dead time mode register (PWMDMOD) is used to control the dead time.

PWMDMOD: Address 329CH (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		DMODE21	DMODE20	DMODE11	DMODE10	DMODE01	DMODE00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit7, bit6] (Reserved)

[bit5, bit4] DMODE21, DMODE20: Dead time mode bits 2

DMODE21	DMODE20	Function
0	0	Do not add the dead time.
0	1	
1	0	Normal dead time mode
1	1	Extended dead time mode 1

- These bits are used to set the PWM2H/PWM2L dead time mode.
- The normal dead time mode adds the dead time only to the rising edge of the PWM output.
- The extended dead time mode 1 adds the dead time to both the rising edge and falling edge of the PWM output.
- Set the normal dead time mode or extended dead time mode 1 only in complementary mode (PWMOD:bit10 in the PWM control register (PWMPCN45) is set to "0").

[bit3, bit2] DMODE11, DMODE10: Dead time mode bits 1

DMODE11	DMODE10	Function
0	0	Do not add the dead time.
0	1	
1	0	Normal dead time mode
1	1	Extended dead time mode 1

- These bits are used to set the PWM1H/PWM1L dead time mode.
- The normal dead time mode adds the dead time only to the rising edge of the PWM output.
- The extended dead time mode 1 adds the dead time to both the rising edge and falling edge of the PWM output.
- Set the normal dead time mode or extended dead time mode 1 only in complementary mode (PWMOD:bit10 in the PWM control register (PWMPCN23) is set to "0").

[bit3, bit2] DMODE01, DMODE00: Dead time mode bits 0

DMODE01	DMODE00	Function
0	0	Do not add the dead time.
0	1	
1	0	Normal dead time mode
1	1	Extended dead time mode 1

- These bits are used to set the PWM0H/PWM0L dead time mode.
- The normal dead time mode adds the dead time only to the rising edge of the PWM output.
- The extended dead time mode 1 adds the dead time to both the rising edge and falling edge of the PWM output.
- Set the normal dead time mode or extended dead time mode 1 only in complementary mode (PWMOD: bit10 in the PWM control register (PWMPCN01) is set to "0").

Note:

Before setting the dead time mode register, check that the PWM timer is stopped.

43.4.6.2 High-side Rising Edge Dead Time Setting Register: PWMHRTMRR0 to PWMHRTMRR2

This section shows the bit configuration of the high-side rising edge dead time setting register.

The high-side rising edge dead time setting register (PWMHRTMRR) is used to control the dead time for the rising edge of the high-side PWM output.

PWMHRTMRR0: Address 32A0H (access: half word and word)

PWMHRTMRR1: Address 32A8H (access: half word and word)

PWMHRTMRR2: Address 32B0H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TM15	TM14	TM13	TM12	TM11	TM10	TM09	TM08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TM07	TM06	TM05	TM04	TM03	TM02	TM01	TM00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] TM15 to TM00: Timer value bits

TM15 to TM00	Function
	Set the amount of dead time.

- These bits are used to set the amount of dead time for the rising edge of the PWMH output.

  Non-overlap time = (set value) × PWM division clock cycle
- These bits are valid only in the normal dead time mode or extended dead time mode 1.
- In these bits, set a value that is equal to or larger than "0001_H".
- The set value is reflected on the timer at the rising edge of the PWMH output.

Note:

When accessing the high-side rising edge dead time setting register, use a half-word or word access instruction.

43.4.6.3 High-side Falling Edge Dead Time Setting Register: PWMHFTMRR0 to PWMHFTMRR2

This section shows the bit configuration of the high-side falling edge dead time setting register.

The high-side falling edge dead time setting register (PWMHFTMRR) is used to control the dead time for the falling edge of the high-side PWM output.

PWMHFTMRR0: Address 32A2H (access: half word and word)

PWMHFTMRR1: Address 32AAH (access: half word and word)

PWMHFTMRR2: Address 32B2H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TM15	TM14	TM13	TM12	TM11	TM10	TM09	TM08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TM07	TM06	TM05	TM04	TM03	TM02	TM01	TM00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] TM15 to TM00: Timer value bits

TM15 to TM00	Function
	Set the amount of dead time.

- These bits are used to set the amount of dead time for the falling edge of the PWMH output.

  Overlap time = (set value) × PWM division clock cycle
- These bits are valid only in the extended dead time mode 1.
- The set value is reflected on the timer at the falling edge of the PWMH output.

Note:

When accessing the high-side falling edge dead time setting register, use a half-word or word access instruction.

43.4.6.4 Low-side Rising Edge Dead Time Setting Register: PWMLRTMRR0 to PWMLRTMRR2

This section shows the bit configuration of the low-side rising edge dead time setting register.

The low-side rising edge dead time setting register (PWMLRTMRR) is used to control the dead time for the rising edge of the low-side PWM output.

PWMLRTMRR0: Address 32A4H (access: half word and word)

PWMLRTMRR1: Address 32ACH (access: half word and word)

PWMLRTMRR2: Address 32B4H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TM15	TM14	TM13	TM12	TM11	TM10	TM09	TM08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TM07	TM06	TM05	TM04	TM03	TM02	TM01	TM00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] TM15 to TM00: Timer value bits

TM15 to TM00	Function
	Set the amount of dead time.

- These bits are used to set the amount of dead time for the rising edge of the PWML output.

  Non-overlap time = (set value) × PWM division clock cycle
- These bits are valid only in the normal dead time mode or extended dead time mode 1.
- In these bits, set a value that is equal to or larger than "0001_H".
- The set value is reflected on the timer at the rising edge of the PWML output.

Note:

When accessing the low-side rising edge dead time setting register, use a half-word or word access instruction.

43.4.6.5 Low-side Falling Edge Dead Time Setting Register: PWMLFTMRR0 to PWMLFTMRR2

This section shows the bit configuration of the low-side falling edge dead time setting register.

The low-side falling edge dead time setting register (PWMLFTMRR) is used to control the dead time for the falling edge of the low-side PWM output.

PWMLFTMRR0: Address 32A6H (access: half word and word)

PWMLFTMRR1: Address 32AEH (access: half word and word)

PWMLFTMRR2: Address 32B6H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TM15	TM14	TM13	TM12	TM11	TM10	TM09	TM08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TM07	TM06	TM05	TM04	TM03	TM02	TM01	TM00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] TM15 to TM00: Timer value bits

TM15 to TM00	Function
	Set the amount of dead time.

- These bits are used to set the amount of dead time for the falling edge of the PWML output.

  Overlap time = (set value) × PWM division clock cycle
- These bits are valid only in the extended dead time mode 1.
- The set value is reflected on the timer at the falling edge of the PWML output.

Note:

When accessing the low-side falling edge dead time setting register, use a half-word or word access instruction.

43.4.7 Blanking-related Registers

This section explains the blanking-related registers.

The blanking-related registers are the blanking control register, blanking start delay control register, and blanking time control register.

43.4.7.1 Blanking Control Register: PWMLEBCON0 to PWMLEBCON2

This section shows the bit configuration of the blanking control register.

The blanking control register (PWMLEBCON) is used to control blanking.

PWMLEBCON0: Address 32B8H (access: byte, half word, and word)

PWMLEBCON1: Address 32CCH (access: byte, half word, and word)

PWMLEBCON2: Address 32E0H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved	BTSL32	BTSL31	BTSL30	Reserved	BTSL22	BTSL21	BTSL20
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved	BTSL12	BTSL11	BTSL10	Reserved	BTSL02	BTSL01	BTSL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved				PHR	PHF	PLR	PLF
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

[bit31] (Reserved)

[bit30 to bit28] BTSL32 to BTSL30: Blanking trigger selection bits 3

BTSL32	BTSL31	BTSL30	Function
0	0	0	Falling edge of PWM0L
0	0	1	Falling edge of PWM1L
0	1	0	Falling edge of PWM2L
0	1	1	Falling edge of master clock 0
1	0	0	Falling edge of master clock 1
1	0	1	Setting prohibited
1	1	0	
1	1	1	

These bits are used to set the blanking trigger.

[bit27] (Reserved)

[bit26 to bit24] BTSL22 to BTSL20: Blanking trigger selection bits 2

BTSL22	BTSL21	BTSL20	Function
0	0	0	Rising edge of PWM0L
0	0	1	Rising edge of PWM1L
0	1	0	Rising edge of PWM2L
0	1	1	Rising edge of master clock 0
1	0	0	Rising edge of master clock 1
1	0	1	Setting prohibited
1	1	0	
1	1	1	

These bits are used to set the blanking trigger.

[bit23] (Reserved)

[bit22 to bit20] BTSL12 to BTSL10: Blanking trigger selection bits 1

BTSL12	BTSL11	BTSL10	Function
0	0	0	Falling edge of PWM0H
0	0	1	Falling edge of PWM1H
0	1	0	Falling edge of PWM2H
0	1	1	Falling edge of master clock 0
1	0	0	Falling edge of master clock 1
1	0	1	Setting prohibited
1	1	0	
1	1	1	

These bits are used to set the blanking trigger.

[bit19] (Reserved)

[bit18 to bit16] BTSL02 to BTSL00: Blanking trigger selection bits 0

BTSL02	BTSL01	BTSL00	Function
0	0	0	Rising edge of PWM0H
0	0	1	Rising edge of PWM1H
0	1	0	Rising edge of PWM2H
0	1	1	Rising edge of master clock 0
1	0	0	Rising edge of master clock 1
1	0	1	Setting prohibited
1	1	0	
1	1	1	

These bits are used to set the blanking trigger.

[bit15 to bit12] (Reserved)

[bit11] PHR: High-side PWM rising edge blanking bit

PHR	Function
0	Do not generate blanking.
1	Generate blanking.

This bit is used to enable blanking to be generated by the trigger selected by the blanking trigger selection bits 0 (BTSL02 to BTSL00:bit18 to bit16).

[bit10] PHF: High-side PWM falling edge blanking bit

PHF	Function
0	Do not generate blanking.
1	Generate blanking.

This bit is used to enable blanking to be generated by the trigger selected by the blanking trigger selection bits 1 (BTSL12 to BTSL10:bit22 to bit20).

[bit9] PLR: Low-side PWM rising edge blanking bit

PLR	Function
0	Do not generate blanking.
1	Generate blanking.

This bit is used to enable blanking to be generated by the trigger selected by the blanking trigger selection bits 2 (BTSL22 to BTSL20:bit26 to bit24).

[bit8] PLF: Low-side PWM falling edge blanking bit

PLF	Function
0	Do not generate blanking.
1	Generate blanking.

This bit is used to enable blanking to be generated by the trigger selected by the blanking trigger selection bits 3 (BTSL32 to BTSL30:bit30 to bit28).

[bit7 to bit0] (Reserved)

Note:

Before setting the blanking control register, check that the PWM timer is stopped.

43.4.7.2 Blanking Start Delay Control Register: PWMLEBSDCON00 to PWMLEBSDCON23

This section shows the bit configuration of the blanking start delay control register.

The blanking start delay control register (PWMLEBSDCON) is used to control blanking.

PWMLEBSDCON00: Address 32BCH (access: half word and word)

PWMLEBSDCON10: Address 32D0H (access: half word and word)

PWMLEBSDCON20: Address 32E4H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DLY15	DLY14	DLY13	DLY12	DLY11	DLY10	DLY09	DLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DLY07	DLY06	DLY05	DLY04	DLY03	DLY02	DLY01	DLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] DLY15 to DLY00: Delay value bits

DLY15 to DLY00	Function
	Set the amount of blanking start delay.

- These bits are used to set the amount of delay from the starting point of the trigger selected by the blanking trigger selection bits 0 (BTSL02 to BTSL00:bit18 to bit16) in the blanking control register (PWMLEBCON) to the blanking start position.

  Delay amount = (set value) × PWM division clock cycle

- The set value is reflected on the delay timer according to the timing of the trigger selected by the blanking trigger selection bits 0 (BTSL02 to BTSL00:bit18 to bit16) in the blanking control register (PWMLEBCON).
- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- If the blanking start position is set in the period from the cycle before the one in which the GPWMH change point is detected to the fault latch reset when the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function before the blanking start position. Therefore, this setting becomes invalid.

Note:

When accessing the blanking start delay control register, use a half-word or word access instruction.

Before making a change to the blanking start delay control register, check that the PWM timer is stopped.

When the phase of the PWM output is set to "0000_H", the phase of the master clock that can be selected by the blanking trigger selection bits (BTSLn2 to BTSLn0 (where n = 0, 1, 2, or 3)) is displaced forward by 1 PWM division clock cycle. Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0001_H" corresponds to the start of the PWM output at the rising edge of the master clock.

PWMLEBSDCON01: Address 32BEH (access: half word and word)

PWMLEBSDCON11: Address 32D2H (access: half word and word)

PWMLEBSDCON21: Address 32E6H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DLY15	DLY14	DLY13	DLY12	DLY11	DLY10	DLY09	DLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DLY07	DLY06	DLY05	DLY04	DLY03	DLY02	DLY01	DLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] DLY15 to DLY00: Delay value bits

DLY15 to DLY00	Function
	Set the amount of blanking start delay.

- These bits are used to set the amount of delay from the starting point of the trigger selected by the blanking trigger selection bits 1 (BTSL12 to BTSL10:bit22 to bit20) in the blanking control register (PWMLEBCON) to the blanking start position.
- Delay amount = (set value) × PWM division clock cycle
- The set value is reflected on the delay timer according to the timing of the trigger selected by the blanking trigger selection bits 1 (BTSL12 to BTSL10:bit22 to bit20) in the blanking control register (PWMLEBCON).
- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- If the blanking start position is set in the period from the cycle before the one in which the GPWMH change point is detected to the fault latch reset when the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function before the blanking start position. Therefore, this setting becomes invalid.

Note:

When accessing the blanking start delay control register, use a half-word or word access instruction.

Before making a change to the blanking start delay control register, check that the PWM timer is stopped.

When the phase of the PWM output is set to "0000_H", the phase of the master clock that can be selected by the blanking trigger selection bits (BTSLn2 to BTSLn0 (where n = 0, 1, 2, or 3)) is displaced forward by 1 PWM division clock cycle. Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0001_H" corresponds to the start of the PWM output at the rising edge of the master clock.

PWMLEBSDCON02: Address 32C0H (access: half word and word)

PWMLEBSDCON12: Address 32D4H (access: half word and word)

PWMLEBSDCON22: Address 32E8H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DLY15	DLY14	DLY13	DLY12	DLY11	DLY10	DLY09	DLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DLY07	DLY06	DLY05	DLY04	DLY03	DLY02	DLY01	DLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] DLY15 to DLY00: Delay value bits

DLY15 to DLY00	Function
	Set the amount of blanking start delay.

- These bits are used to set the amount of delay from the starting point of the trigger selected by the blanking trigger selection bits 2 (BTSL22 to BTSL20:bit26 to bit24) in the blanking control register (PWMLEBCON) to the blanking start position.

  Delay amount = (set value) × PWM division clock cycle

- The set value is reflected on the delay timer according to the timing of the trigger selected by the blanking trigger selection bits 2 (BTSL22 to BTSL20:bit26 to bit24) in the blanking control register (PWMLEBCON).
- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- If the blanking start position is set in the period from the cycle before the one in which the GPWMH change point is detected to the fault latch reset when the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function before the blanking start position. Therefore, this setting becomes invalid.

Note:

When accessing the blanking start delay control register, use a half-word or word access instruction.

Before making a change to the blanking start delay control register, check that the PWM timer is stopped.

When the phase of the PWM output is set to "0000_H", the phase of the master clock that can be selected by the blanking trigger selection bits (BTSLn2 to BTSLn0 (where n = 0, 1, 2, or 3)) is displaced forward by 1 PWM division clock cycle. Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0001_H" corresponds to the start of the PWM output at the rising edge of the master clock.

PWMLEBSDCON03: Address 32C2H (access: half word and word)

PWMLEBSDCON13: Address 32D6H (access: half word and word)

PWMLEBSDCON23: Address 32EAH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DLY15	DLY14	DLY13	DLY12	DLY11	DLY10	DLY09	DLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DLY07	DLY06	DLY05	DLY04	DLY03	DLY02	DLY01	DLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] DLY15 to DLY00: Delay value bits

DLY15 to DLY00	Function
	Set the amount of blanking start delay.

- These bits are used to set the amount of delay from the starting point of the trigger selected by the blanking trigger selection bits 3 (BTSL32 to BTSL30:bit30 to bit28) in the blanking control register (PWMLEBCON) to the blanking start position.
- Delay amount = (set value) × PWM division clock cycle
- The set value is reflected on the delay timer according to the timing of the trigger selected by the blanking trigger selection bits 3 (BTSL32 to BTSL30:bit30 to bit28) in the blanking control register (PWMLEBCON).
- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).
- If the blanking start position is set in the period from the cycle before the one in which the GPWMH change point is detected to the fault latch reset when the PWM mode is set to complementary output (PWMOD:bit10 in the PWM control register (PWMPCN) is set to "0") and the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function before the blanking start position. Therefore, this setting becomes invalid.

Note:

When accessing the blanking start delay control register, use a half-word or word access instruction.

Before making a change to the blanking start delay control register, check that the PWM timer is stopped.

When the phase of the PWM output is set to "0000_H", the phase of the master clock that can be selected by the blanking trigger selection bits (BTSLn2 to BTSLn0 (where n = 0, 1, 2, or 3)) is displaced forward by 1 PWM division clock cycle. Therefore, when the phase of the PWM output is set to "0000_H", the delay value "0001_H" corresponds to the start of the PWM output at the rising edge of the master clock.

43.4.7.3 Blanking Time Control Register: PWMLEBTCON00 to PWMLEBTCON21

This section shows the bit configuration of the blanking time control register.

The blanking time control register (PWMLEBTCON) is used to control blanking.

PWMLEBTCON00: Address 32C4H (access: half word and word)

PWMLEBTCON10: Address 32D8H (access: half word and word)

PWMLEBTCON20: Address 32ECH (access: half word and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	LEBH15	LEBH14	LEBH13	LEBH12	LEBH11	LEBH10	LEBH09	LEBH08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	LEBH07	LEBH06	LEBH05	LEBH04	LEBH03	LEBH02	LEBH01	LEBH00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	LEBL15	LEBL14	LEBL13	LEBL12	LEBL11	LEBL10	LEBL09	LEBL08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LEBL07	LEBL06	LEBL05	LEBL04	LEBL03	LEBL02	LEBL01	LEBL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit16] LEBH15 to LEBH00: LEB width bits

LEBH15 to LEBH00	Function
	Set the blanking width.

- These bits are used to set the blanking width for the starting point of the trigger selected by the blanking trigger selection bits 0 (BTSL02 to BTSL00:bit18 to bit16) in the blanking control register (PWMLEBCON).
- Blanking width = (set value) × PWM division clock cycle
- The set value is reflected on the timer at the start of blanking.
- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).

[bit15 to bit0] LEBL15 to LEBL00: LEB width bits

LEBL15 to LEBL00	Function
	Set the blanking width.

- These bits are used to set the blanking width for the starting point of the trigger selected by the blanking trigger selection bits 1 (BTSL12 to BTSL10:bit22 to bit20) in the blanking control register (PWMLEBCON).

  Blanking width = (set value) × PWM division clock cycle
- The set value is reflected on the timer at the start of blanking.

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).

Note:

When accessing the blanking time control register, use a half-word or word access instruction.

Before making a change to the blanking time control register, check that the PWM timer is stopped.

PWMLEBTCON01: Address 32C8H (access: half word and word)

PWMLEBTCON11: Address 32DCH (access: half word and word)

PWMLEBTCON21: Address 32F0H (access: half word and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	LEBH15	LEBH14	LEBH13	LEBH12	LEBH11	LEBH10	LEBH09	LEBH08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	LEBH07	LEBH06	LEBH05	LEBH04	LEBH03	LEBH02	LEBH01	LEBH00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	LEBL15	LEBL14	LEBL13	LEBL12	LEBL11	LEBL10	LEBL09	LEBL08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LEBL07	LEBL06	LEBL05	LEBL04	LEBL03	LEBL02	LEBL01	LEBL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit16] LEBH15 to LEBH00: LEB width bits

LEBH15 to LEBH00	Function
	Set the blanking width.

- These bits are used to set the blanking width for the starting point of the trigger selected by the blanking trigger selection bits 2 (BTSL22 to BTSL20:bit26 to bit24) in the blanking control register (PWMLEBCON).

  Blanking width = (set value) × PWM division clock cycle

- The set value is reflected on the timer at the start of blanking.

[bit15 to bit0] LEBL15 to LEBL00: LEB width bits

LEBL15 to LEBL00	Function
	Set the blanking width.

- These bits are used to set the blanking width for the starting point of the trigger selected by the blanking trigger selection bits 3 (BTSL32 to BTSL30:bit30 to bit28) in the blanking control register (PWMLEBCON).

  Blanking width = (set value) × PWM division clock cycle

- The set value is reflected on the timer at the start of blanking.

- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).

Note:

When accessing the blanking time control register, use a half-word or word access instruction.

Before making a change to the blanking time control register, check that the PWM timer is stopped.

43.4.8 A/D Converter Trigger Generation Registers

This section explains the A/D converter trigger generation registers.

The A/D converter trigger generation registers are the A/DC trigger control register, A/DC trigger status register, and A/DC trigger delay control register.

43.4.8.1 A/DC Trigger Control Register: PWMADTCON

This section shows the bit configuration of the A/DC trigger control register.

The A/DC trigger control register (PWMADTCON) is used to control A/D converter trigger generation.

PWMADTCON: Address 32F4H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	ATSL33	ATSL32	ATSL31	ATSL30	ATSL23	ATSL22	ATSL21	ATSL20
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	ATSL13	ATSL12	ATSL11	ATSL10	ATSL03	ATSL02	ATSL01	ATSL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved		TRGEN5	TRGEN4	TRGEN3	TRGEN2	TRGEN1	TRGEN0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				ADTIE3	ADTIE2	ADTIE1	ADTIE0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W

[bit31 to bit28] ATSL33 to ATSL30: A/D trigger 3 factor selection bits

ATSL33	ATSL32	ATSL31	ATSL30	Function
0	0	0	0	No trigger
0	0	0	1	Rising edge of PWM0H
0	0	1	0	Falling edge of PWM0H
0	0	1	1	Rising edge of PWM0L
0	1	0	0	Falling edge of PWM0L
0	1	0	1	Rising edge of PWM1H
0	1	1	0	Falling edge of PWM1H
0	1	1	1	Rising edge of PWM1L
1	0	0	0	Falling edge of PWM1L
1	0	0	1	Rising edge of PWM2H
1	0	1	0	Falling edge of PWM2H
1	0	1	1	Rising edge of PWM2L
1	1	0	0	Falling edge of PWM2L
1	1	0	1	Setting prohibited
1	1	1	0	
1	1	1	1	

These bits are used to set the timing trigger for A/D trigger 3.

Note:

Before setting these bits, check that the PWM timer is stopped.

[bit27 to bit24] ATSL23 to ATSL20: A/D trigger 2 factor selection bits

ATSL23	ATSL22	ATSL21	ATSL20	Function
0	0	0	0	No trigger
0	0	0	1	Rising edge of PWM0H
0	0	1	0	Falling edge of PWM0H
0	0	1	1	Rising edge of PWM0L
0	1	0	0	Falling edge of PWM0L
0	1	0	1	Rising edge of PWM1H
0	1	1	0	Falling edge of PWM1H
0	1	1	1	Rising edge of PWM1L
1	0	0	0	Falling edge of PWM1L
1	0	0	1	Rising edge of PWM2H
1	0	1	0	Falling edge of PWM2H
1	0	1	1	Rising edge of PWM2L
1	1	0	0	Falling edge of PWM2L
1	1	0	1	Setting prohibited
1	1	1	0	
1	1	1	1	

These bits are used to set the timing trigger for A/D trigger 2.

Note:

Before setting these bits, check that the PWM timer is stopped.

[bit23 to bit20] ATSL13 to ATSL10: A/D trigger 1 factor selection bits

ATSL13	ATSL12	ATSL11	ATSL10	Function
0	0	0	0	No trigger
0	0	0	1	Rising edge of PWM0H
0	0	1	0	Falling edge of PWM0H
0	0	1	1	Rising edge of PWM0L
0	1	0	0	Falling edge of PWM0L
0	1	0	1	Rising edge of PWM1H
0	1	1	0	Falling edge of PWM1H
0	1	1	1	Rising edge of PWM1L
1	0	0	0	Falling edge of PWM1L
1	0	0	1	Rising edge of PWM2H

ATSL13	ATSL12	ATSL11	ATSL10	Function
1	0	1	0	Falling edge of PWM2H
1	0	1	1	Rising edge of PWM2L
1	1	0	0	Falling edge of PWM2L
1	1	0	1	Setting prohibited
1	1	1	0	
1	1	1	1	

These bits are used to set the timing trigger for A/D trigger 1.

Note:

Before setting these bits, check that the PWM timer is stopped.

[bit19 to bit16] ATSL03 to ATSL00: A/D trigger 0 factor selection bits

ATSL03	ATSL02	ATSL01	ATSL00	Function
0	0	0	0	No trigger
0	0	0	1	Rising edge of PWM0H
0	0	1	0	Falling edge of PWM0H
0	0	1	1	Rising edge of PWM0L
0	1	0	0	Falling edge of PWM0L
0	1	0	1	Rising edge of PWM1H
0	1	1	0	Falling edge of PWM1H
0	1	1	1	Rising edge of PWM1L
1	0	0	0	Falling edge of PWM1L
1	0	0	1	Rising edge of PWM2H
1	0	1	0	Falling edge of PWM2H
1	0	1	1	Rising edge of PWM2L
1	1	0	0	Falling edge of PWM2L
1	1	0	1	Setting prohibited
1	1	1	0	
1	1	1	1	

These bits are used to set the timing trigger for A/D trigger 0.

Note:

Before setting these bits, check that the PWM timer is stopped.

[bit15, bit14] (Reserved)

[bit13] TRGEN5: Trigger enable bit 5

TRGEN5	Function
0	Disable trigger generation.
1	Enable trigger generation.

This bit is used to enable fault detection result 21 to be generated as A/D trigger 9.

[bit12] TRGEN4: Trigger enable bit 4

TRGEN4	Function
0	Disable trigger generation.
1	Enable trigger generation.

- This bit is used to enable fault detection result 20 to be generated as A/D trigger 8.
- If the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function from the cycle before the one in which the GPWMH change point is detected until the fault latch is reset. Therefore, no trigger is generated during this period.

[bit11] TRGEN3: Trigger enable bit 3

TRGEN3	Function
0	Disable trigger generation.
1	Enable trigger generation.

This bit is used to enable fault detection result 11 to be generated as A/D trigger 7.

[bit10] TRGEN2: Trigger enable bit 2

TRGEN2	Function
0	Disable trigger generation.
1	Enable trigger generation.

- This bit is used to enable fault detection result 10 to be generated as A/D trigger 6.
- If the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function from the cycle before the one in which the GPWMH change point is detected until the fault latch is reset. Therefore, no trigger is generated during this period.

[bit9] TRGEN1: Trigger enable bit 1

TRGEN1	Function
0	Disable trigger generation.
1	Enable trigger generation.

This bit is used to enable fault detection result 01 to be generated as A/D trigger 5.

[bit8] TRGEN0: Trigger enable bit 0

TRGEN0	Function
0	Disable trigger generation.
1	Enable trigger generation.

- This bit is used to enable fault detection result 00 to be generated as A/D trigger 4.
- If the fault operation is set to inverted (bit11, bit10 (FLT0H1, FLT0H0) or bit9, bit8 (FLT0L1, FLT0L0) in the fault control register (PWMFLTCON) are set to "01"), the fault input is masked by the re-inversion prevention function from the cycle before the one in which the GPWMH change point is detected until the fault latch is reset. Therefore, no trigger is generated during this period.

[bit7 to bit4] (Reserved)

[bit3] ADTIE3: Interrupt request enable bit 3

ADTIE3	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for trigger interrupt 3.
- If this bit and the interrupt flag bit (ADTIF3:bit3) in the A/D trigger status register (PWMADTST) are set to "1", an interrupt request for the CPU is generated.

[bit2] ADTIE2: Interrupt request enable bit 2

ADTIE2	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for trigger interrupt 2.
- If this bit and the interrupt flag bit (ADTIF2:bit2) in the A/D trigger status register (PWMADTST) are set to "1", an interrupt request for the CPU is generated.

[bit1] ADTIE1: Interrupt request enable bit 1

ADTIE1	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for trigger interrupt 1.
- If this bit and the interrupt flag bit (ADTIF1:bit1) in the A/D trigger status register (PWMADTST) are set to "1", an interrupt request for the CPU is generated.

[bit0] ADTIE0: Interrupt request enable bit 0

ADTIE0	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- This bit is used to enable interrupt request generation for trigger interrupt 0.
- If this bit and the interrupt flag bit (ADTIF0:bit0) in the A/D trigger status register (PWMADTST) are set to "1", an interrupt request for the CPU is generated.

43.4.8.2 A/DC Trigger Status Register: PWMADTST

This section shows the bit configuration of the A/DC trigger status register.

The A/DC trigger status register (PWMADTST) is used to control the A/D trigger interrupt request.

PWMADTST: Address 32F8H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				ADTIF3	ADTIF2	ADTIF1	ADTIF0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W

[bit7 to bit4] (Reserved)

[bit3] ADTIF3: Interrupt flag bit 3

ADTIF3	Function	
	During Read Operation	During Write Operation
0	No trigger	This bit is cleared.
1	Trigger generated	This bit remains unaffected.

- If A/D trigger 3 is generated, this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

[bit2] ADTIF2: Interrupt flag bit 2

ADTIF2	Function	
	During Read Operation	During Write Operation
0	No trigger	This bit is cleared.
1	Trigger generated	This bit remains unaffected.

- If A/D trigger 2 is generated, this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

[bit1] ADTIF1: Interrupt flag bit 1

ADTIF1	Function	
	During Read Operation	During Write Operation
0	No trigger	This bit is cleared.
1	Trigger generated	This bit remains unaffected.

- If A/D trigger 1 is generated, this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

[bit0] ADTIF0: Interrupt flag bit 0

ADTIF0	Function	
	During Read Operation	During Write Operation
0	No trigger	This bit is cleared.
1	Trigger generated	This bit remains unaffected.

- If A/D trigger 0 is generated, this bit is set to "1".
- When this bit is set to "0": This bit is cleared.
- When this bit is set to "1": This bit remains unaffected.

Note:

If a read-modify-write (RMW) instruction is executed, "1" is always read.

If software clear ("0" is written) and hardware set occur simultaneously, hardware set takes precedence.

43.4.8.3 A/DC Trigger Delay Control Register: PWMADTDCON0 to PWMADTDCON3

This section shows the bit configuration of the A/DC trigger delay control register.

The A/DC trigger delay control register (PWMADTDCON) is used to control A/D converter trigger generation.

PWMADTDCON0: Address 32FCH (access: half word and word)

PWMADTDCON1: Address 32FEH (access: half word and word)

PWMADTDCON2: Address 3300H (access: half word and word)

PWMADTDCON3: Address 3302H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	DLY15	DLY14	DLY13	DLY12	DLY11	DLY10	DLY09	DLY08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DLY07	DLY06	DLY05	DLY04	DLY03	DLY02	DLY01	DLY00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] DLY15 to DLY00: Delay value bits

DLY15 to DLY00	Function
	Set the amount of delay for the A/D trigger.

- These bits are used to set the amount of delay from the starting point of the trigger selected by the A/D trigger n factor selection bits (ATSLn3 to ATSLn0) in the A/DC trigger control register (PWMADTCON) to trigger generation. (n=3 to 0)
- Delay amount = (set value) × PWM division clock cycle
- The set value is reflected on the delay timer according to the timing of the trigger selected by the A/D trigger n factor selection bits (ATSLn3 to ATSLn0) in the A/DC trigger control register (PWMADTCON). (n=3 to 0)
- In these bits, set a value that is smaller than the PWM cycle. For details on the setting of the PWM cycle, see the explanation of the cycle mode selection bit (MCPS:bit12) in the PWM control register (PWMPCN).

Note:

When accessing the A/DC trigger delay control register, use a half-word or word access instruction.

Before making a change to the A/DC trigger delay control register, check that the PWM timer is stopped.

43.5 Explanation of Operation

This section explains the operation.

Simultaneous activation of the master clock timer

Of the 2 channels of timers, the specified timer is activated and cleared simultaneously.

Master clock timer

When the counting operation is enabled, the timer starts to count up from the value set in the timer data register (PWMTCDT). The count value is used as the base time of the master clock.

PWM generation timer

When the counting operation is enabled, the timer starts to count up from "0000_H". The count value is used as the base time of the PWM output.

43.5.1 PWM Interrupts

This section explains the PWM interrupts.

Table 43-9 and Table 43-10 list the interrupt control bits and interrupt factors for master clock generation.

Table 43-9. Interrupt Control Bits and Interrupt Factors for Master Clock Generation (Master Clock 0)

	Master Clock Generation		
	Compare Clear 0	**0 Detection 0**	**Special Event 0**
Interrupt Request Flag Bit	ICLR:bit25 in the timer state control register 0 (PWMTCCS0)	IRQZF:bit30 in the timer state control register 0 (PWMTCCS0)	ISEVF0:bit0 in the special event status register (PWMSEVST)
Interrupt Request Enable Bit	ICRE:bit24 in the timer state control register 0 (PWMTCCS0)	IRQZE:bit29 in the timer state control register 0 (PWMTCCS0)	ISEVE0:bit8 in the special event control register (PWMSEVCON)
Interrupt Factor	The value of timer 0 matches compare clear register 0 (PWMCPCLR0).	The value of timer 0 becomes "0000H".	The value of timer 0 matches special event compare register 0 (PWMSEVCP0).

Table 43-10. Interrupt Control Bits and Interrupt Factors for Master Clock Generation (Master Clock 1)

	Master Clock Generation		
	Compare Clear 1	**0 Detection 1**	**Special Event 1**
Interrupt Request Flag Bit	ICLR:bit25 in the timer state control register 1 (PWMTCCS1)	IRQZF:bit30 in the timer state control register 1 (PWMTCCS1)	ISEVF1:bit1 in the special event status register (PWMSEVST)
Interrupt Request Enable Bit	ICRE:bit24 in the timer state control register 1 (PWMTCCS1)	IRQZE:bit29 in the timer state control register 1 (PWMTCCS1)	ISEVE1:bit0 in the special event control register (PWMSEVCON)
Interrupt Factor	The value of timer 1 matches compare clear register 1 (PWMCPCLR1).	The value of timer 1 becomes "0000H".	The value of timer 1 matches special event compare register 1 (PWMSEVCP1).

If the timer value matches the compare clear register (PWMCPCLR), ICLR:bit25 in the timer state control register (PWMTCCS) is set. If the interrupt request is enabled (ICRE:bit24 in PWMTCCS is set to 1) when this bit is set, the interrupt request is output to the interrupt controller.

If the timer value becomes "0000_H", IRQZF:bit30 in the timer state control register (PMWTCCS) is set. If the interrupt request is enabled (IRQZE:bit29 in PWMTCCS is set to 1) when this bit is set, the interrupt request is output to the interrupt controller.

If the timer value matches the special event compare register (PWMSEVCP), ISEVF0:bit0 or ISEVF1:bit1 in the special event status register (PWMSEVST) is set. If the interrupt request is enabled (ISEVE0:bit8 or ISEVE1:bit0 in PWMSEVCON is set to 1) when this bit is set, the interrupt request is output to the interrupt controller.

Table 43-11,Table 43-12 and Table 43-13 list the interrupt control bits and interrupt factors of PMW control.

Table 43-11. Interrupt Control Bits and Interrupt Factors of PWM Control 0

	PWM control 0			
	Fault 0	**Fault 1**	**SOW0**	**CAPIF0**
Interrupt Request Flag Bit	IFLTF0:bit0 in PWM common status register 0 (PWMST0)	IFLTF1:bit1 in PWM common status register 0 (PWMST0)	SOWIF:bit2 in PWM common status register 0 (PWMST0)	CAPIF:bit3 in PWM common status register 0 (PWMST0)
Interrupt Request Enable Bit	IFLTE0:bit6 in fault control register 00 (PWMFLTCON00)	IFLTE1:bit6 in fault control register 01 (PWMFLTCON01)	SOWIE:bit16 in soft overwrite control register 0 (PWMSOWCON0)	CAPIE:bit3 in fault capture control register 0 (PWMFLTCAPCON0)
Interrupt Factor	Fault operation triggered by the fault input from the comparator	Fault operation (preferred) triggered by the fault input from the comparator	When OSLBFE:bit27 in soft overwrite control register 0 (PWMSOWCON0) is set to "1", OSL11, OSL10:bit31, bit30 and OSL01, OSL00:bit29, bit28 are updated.	The difference between HCAP15 to HCAP00:bit31 to bit16 and LCAP15 to LCAP00:bit15 to bit0 in fault capture data register 0 (PWMFLTCAPD0) exceeds the value set in capture interrupt threshold setting register 0 (PWMCAPITH0).

Table 43-12. Interrupt Control Bits and Interrupt Factors of PWM Control 1

	PWM control 1			
	Fault 2	**Fault 3**	**SOW1**	**CAPIF1**
Interrupt Request Flag Bit	IFLTF0:bit0 in PWM common status register 1 (PWMST1)	IFLTF1:bit1 in PWM common status register 1 (PWMST1)	SOWIF:bit2 in PWM common status register 1 (PWMST1)	CAPIF:bit3 in PWM common status register 1 (PWMST1)
Interrupt Request Enable Bit	IFLTE0:bit6 in fault control register 10 (PWMFLTCON10)	IFLTE1:bit6 in fault control register 11 (PWMFLTCON11)	SOWIE:bit16 in soft overwrite control register 1 (PWMSOWCON1)	CAPIE:bit3 in fault capture control register 1 (PWMFLTCAPCON1)
Interrupt Factor	Fault operation triggered by the fault input from the comparator	Fault operation (preferred) triggered by the fault input from the comparator	When OSLBFE:bit27 in soft overwrite control register 1 (PWMSOWCON1) is set to "1", OSL11, OSL10:bit31, bit30 and OSL01, OSL00:bit29, bit28 are updated.	The difference between HCAP15 to HCAP00:bit31 to bit16 and LCAP15 to LCAP00:bit15 to bit0 in fault capture data register 1 (PWMFLTCAPD1) exceeds the value set in capture interrupt threshold setting register 1 (PWMCAPITH1).

Table 43-13. Interrupt Control Bits and Interrupt Factors of PWM Control 2

	PWM control 2			
	Fault 4	**Fault 5**	**SOW2**	**CAPIF2**
Interrupt Request Flag Bit	IFLTF0:bit0 in PWM common status register 2 (PWMST2)	IFLTF1:bit1 in PWM common status register 2 (PWMST2)	SOWIF:bit2 in PWM common status register 2 (PWMST2)	CAPIF:bit3 in PWM common status register 2 (PWMST2)
Interrupt Request Enable Bit	IFLTE0:bit6 in fault control register 20 (PWMFLTCON20)	IFLTE1:bit6 in fault control register 21 (PWMFLTCON21)	SOWIE:bit16 in soft overwrite control register 2 (PWMSOWCON2)	CAPIE:bit3 in fault capture control register 2 (PWMFLTCAPCON2)
Interrupt Factor	Fault operation triggered by the fault input from the comparator	Fault operation (preferred) triggered by the fault input from the comparator	When OSLBFE:bit27 in soft overwrite control register 2 (PWMSOWCON2) is set to "1", OSL11, OSL10:bit31, bit30 and OSL01, OSL00:bit29, bit28 are updated.	The difference between HCAP15 to HCAP00:bit31 to bit16 and LCAP15 to LCAP00:bit15 to bit0 in fault capture data register 2 (PWMFLTCAPD2) exceeds the value set in capture interrupt threshold setting register 2 (PWMCAPITH2).

If the fault operation is triggered by the fault input from the comparator, IFLTF0:bit0 in the PWM common status register (PWMST) is set. If the interrupt request is enabled (IFLTE0:bit6 in PWMFLTCONx0 is set to 1) when this bit is set, the interrupt request is output to the interrupt controller.

If the fault operation (preferred) is triggered by the fault input from the comparator, IFLTF1:bit1 in the PWM common status register (PWMST) is set. If the interrupt request is enabled (IFLTE1:bit6 in PWMFLTCONx1 is set to 1) when this bit is set, the interrupt request is output to the interrupt controller.

If the value set in the overwrite selection bits (OSL11, OSL10:bit31, bit30 and OSL01, OSL00:bit29, bit28) in the soft overwrite control register (PWMSOWCON) is updated and transferred to the OSL buffer when the OSL buffer enable bit (OSLBFE:bit27) in the soft overwrite control register (PWMSOWCON) is set to "1", SOWIF:bit2 in the PWM common status register (PWMST) is set. If the interrupt request is enabled (SOWIE:bit16 in PWMSOWCON is set to 1) when this bit is set, the interrupt request is output to the interrupt controller.

If the difference between the high-side capture data bits (HCAP15 to HCAP00:bit31 to bit16) in the fault capture data register (PWMFLTCAPD) and the low-side capture data bits (LCAP15 to LCAP00:bit15 to bit0) exceeds the value set in the capture interrupt threshold setting register (PWMCAPITH), CAPIF:bit3 in the PWM common status register (PWMST) is set. If the interrupt request is enabled (CAPIE:bit3 in PWMFLTCAPCON is set to 1) when this bit is set, the interrupt request is output to the interrupt controller.

Table 43-14 lists the interrupt control bits and interrupt factors for A/D converter trigger generation.

Table 43-14. Interrupt Control Bits and Interrupt Factors for A/D Converter Triggers

	A/D Converter Trigger Generation			
	Trigger 0	**Trigger 1**	**Trigger 2**	**Trigger 3**
Interrupt Request Flag Bit	ADTIF0:bit0 in the A/DC trigger status register (PWMADTST)	ADTIF1:bit1 in the A/DC trigger status register (PWMADTST)	ADTIF2:bit2 in the A/DC trigger status register (PWMADTST)	ADTIF3:bit3 in the A/DC trigger status register (PWMADTST)
Interrupt Request Enable Bit	ADTIE0:bit0 in the A/DC trigger control register (PWMADTCON)	ADTIE1:bit1 in the A/DC trigger control register (PWMADTCON)	ADTIE2:bit2 in the A/DC trigger control register (PWMADTCON)	ADTIE3:bit3 in the A/DC trigger control register (PWMADTCON)
Interrupt Factor	Occurrence of A/D trigger 0	Occurrence of A/D trigger 1	Occurrence of A/D trigger 2	Occurrence of A/D trigger 3

If A/D trigger n occurs, ADTIFn:bitn in the A/DC trigger status register (PWMADTST) is set. If the interrupt request is enabled (ADTIEn:bitn in PWMADTCON is set to 1) when this bit is set, the interrupt request is output to the interrupt controller. (n=0 to 3)

43.5.2 Timer Operation for Master Clock Generation

This section explains the timer operation for master clock generation.

After the reset is complete, the timer starts to count up from the value set in the timer data register (PWMTCDT). The count value is used as the base time of the master clock.

43.5.2.1 Timer Clear

This section explains timer clear.

The count value of the timer is cleared in one of the following cases.

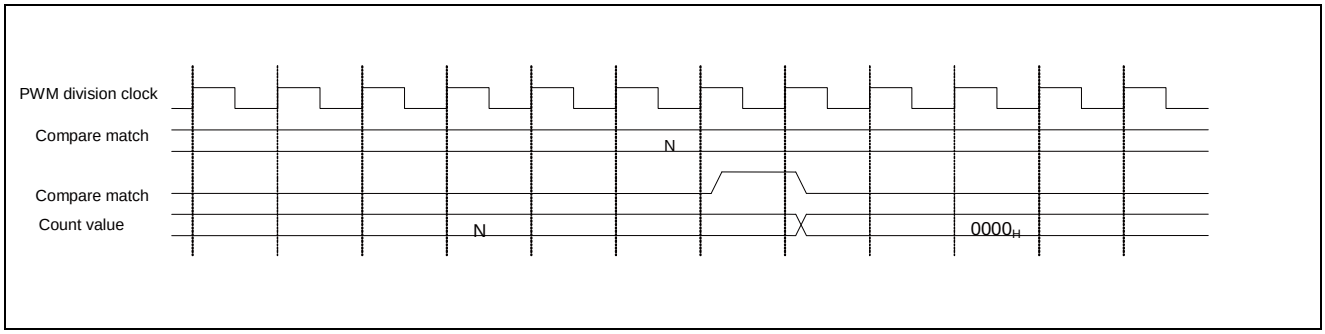
- In the up-count mode (MODE:bit21 in the PWMTCCS register is set to 0), the count value is found to match the compare clear register.
- SCLR:bit20 in the PWMTCCS register is set to "1" while the timer is active.
- "0000_H" is written in the PWMTCDT register while the timer is inactive.
- The timer is reset.
- When SYNCIN is enabled (SYNCEN1:bit26 and SYNCEN0:bit24 in the PWMTRC register are both set to 1), a rising edge is input from the SYNCIN pin.

When reset, the counter is cleared immediately. In the case of synchronization using SYNCIN, the counter is cleared in sync with the count timing when software clear occurs or the counter value is found to match the compare clear register.

Note:

Even if SCLR:bit20 in the PWMTCCS register is set to "1" while the timer is inactive, the count value of the timer is not cleared.
If "0000_H" is written in the PWMTCDT register in the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), an unintended count may be made.
For details on how to set the PWMTCDT register in the up/down count mode (MODE:bit21 in the timer state control register (PWMTCCS) is set to 1), see "43.4.1.4 Timer Data Register: PWMTCDT0 to PWMTCDT1."

Figure 43-6. Timer Clear Timing



43.5.2.2 Timer Mode

This chapter explains the timer mode.

One of the following modes can be selected for the timer.

- Up-count mode (MODE:bit21 in the PWMTCCS register is set to 0)
- Up/down count mode (MODE:bit21 in the PWMTCCS register is set to 1)

In the up-count mode, the counter starts to count up from the value set in advance in the timer data register (PWMTCDT) and continues counting up until the count value matches the value of the compare clear register (PWMCPCLR). When cleared to "0000_H", the counter starts to count up again.

In the up/down count mode, the counter starts to count up from the value set in advance in the timer data register (PWMTCDT) and continues counting up until the count value matches the value of the compare clear register (PWMCPCLR). It then starts to count down and continues counting down when the counter value reaches "0000_H", after which the counter starts to count up again.

43.5.2.3 Compare Clear Buffer

This section explains the compare clear buffer.

The compare clear register (PWMCPCLR) has a buffer function that can be enabled or disabled. When the buffer function is enabled (BFE:bit23 in the PWMTCCS register is set to 1), the data written in the compare clear buffer register (PWMCPCLRB) is transferred to the PWMCPCLR register if the time value "0" is detected. When the buffer function is disabled (BFE:bit23 in the PWMTCCS register is set to 0), data can be written directly to the PWMCPCLR register.

Figure 43-7. Operation in Up-count Mode When the Compare Clear Buffer Is Disabled (BFE:bit23 in the PWMTCCS Register Is Set to 0)

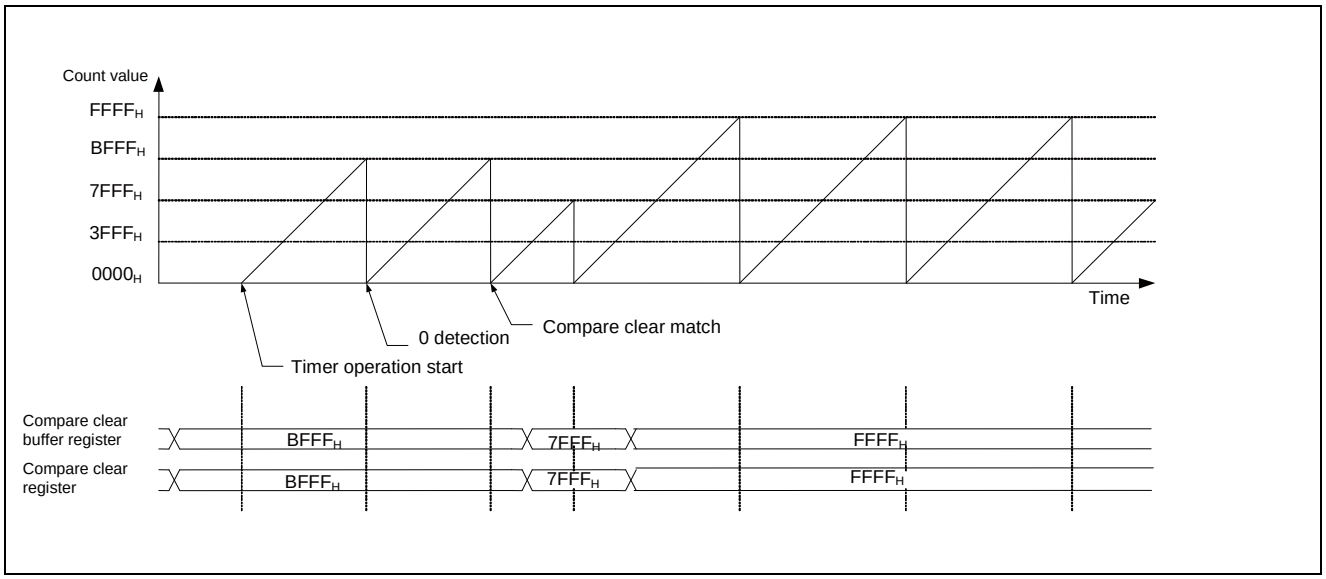


Figure 43-8. Operation in Up-count Mode When the Compare Clear Buffer Is Enabled (BFE:bit23 in the PWMTCCS Register Is Set to 1)

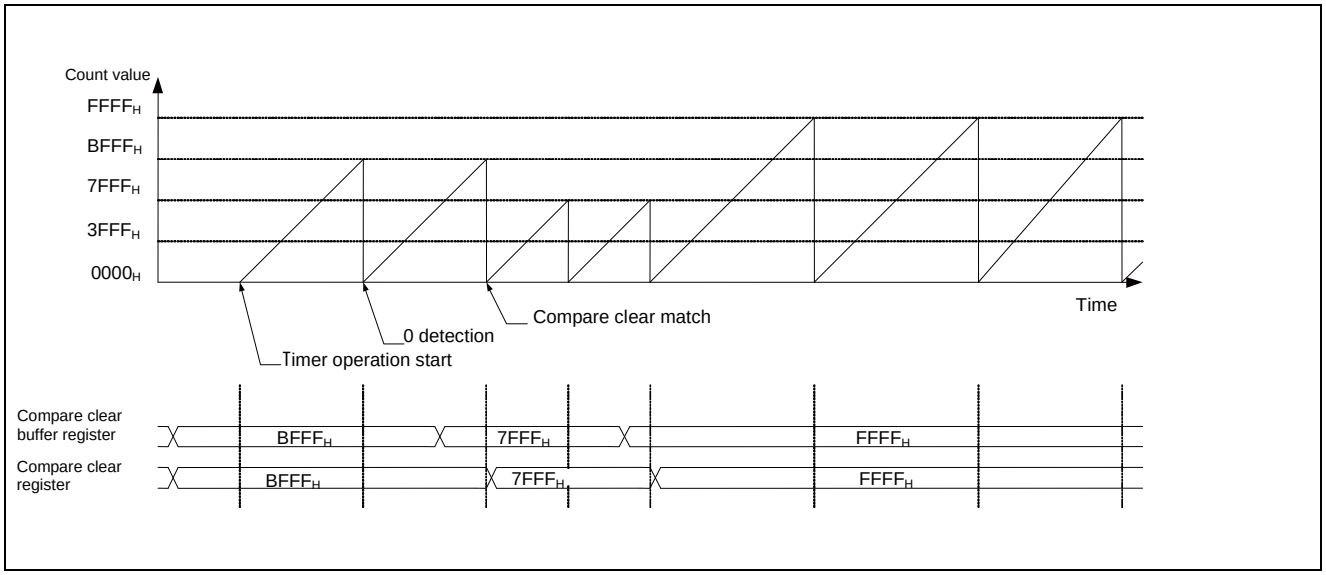
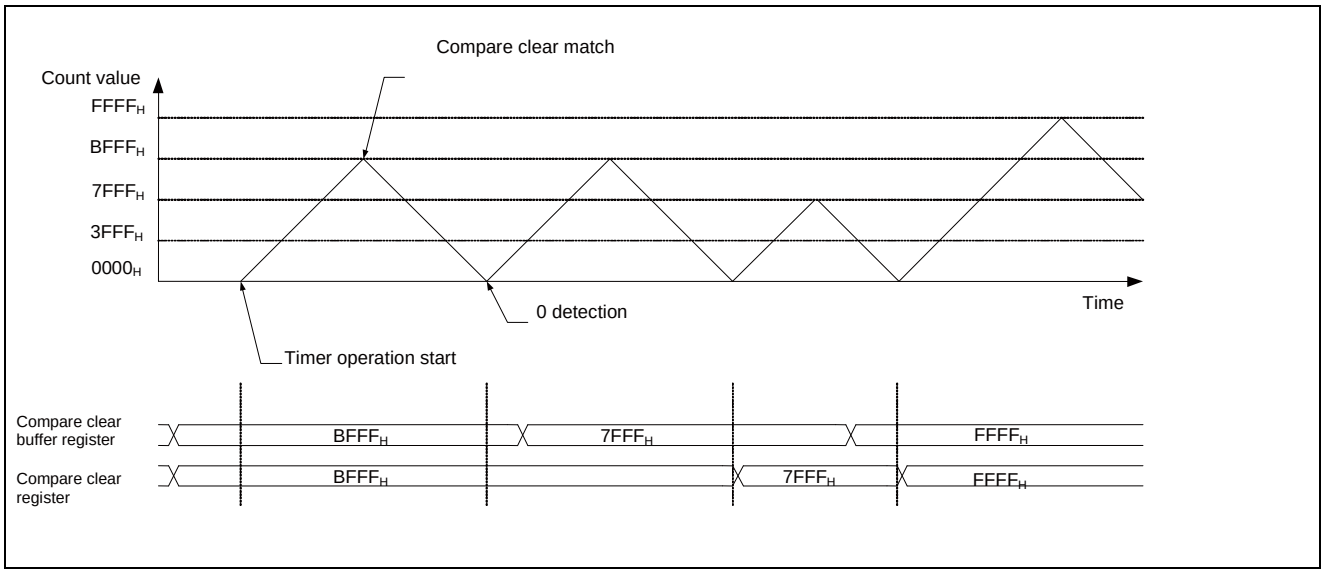


Figure 43-9. Operation in Up/down Count Mode When the Compare Clear Buffer Is Enabled (BFE:bit23 in the PWMTCCS Register Is Set to 1)



43.5.2.4 Timer Interrupts

This section explains the timer interrupts.

The timer can generate the following 2 types of interrupts.

- Compare clear interrupt
- 0 detection interrupt

The compare clear interrupt is generated when the timer value matches the value of the compare clear register.

The 0 detection interrupt is generated when the timer value reaches "0000_H".

Note:

Software clear (SCLR:bit20 in the PWMTCCS register is set to 1) does not generate the 0 detection interrupt.

Figure 43-10. Interrupts Generated in Up-count Mode (MODE:bit21 in the PWMTCCS Register Is Set to 0)

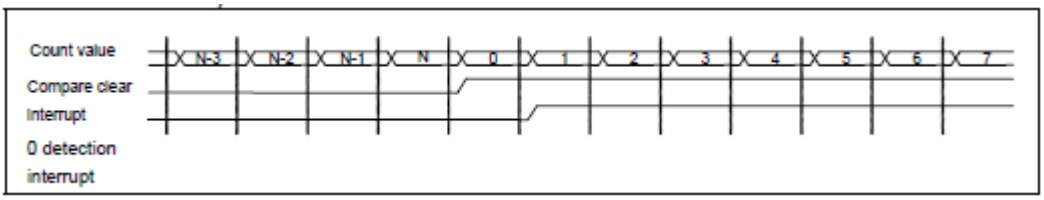
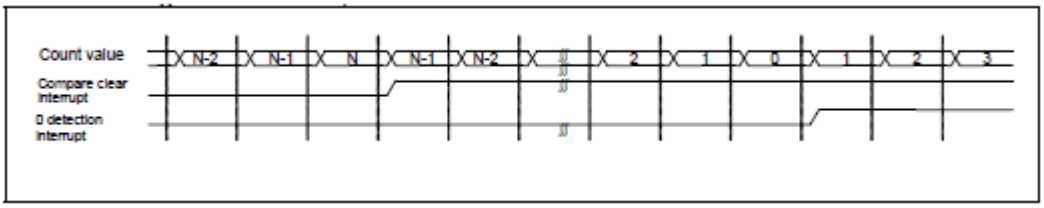


Figure 43-11. Interrupts Generated in Up/down Count Mode (MODE:bit21 in the PWMTCCS Register Is Set to 1)



43.5.2.5 Interrupt Mask Function

This section explains the interrupt mask function.

This function can mask the 0 detection interrupt or compare clear interrupt or both. The method of masking one of these interrupts is explained below.

- Interrupt requests can be masked by setting MSI2 to MSI0:bit28 to bit26 in the PWMTCCS register. The MSI2 to MSI0 bits are a 3-bit reload down register that reloads the count value when it reaches "000_B". The count value can also be loaded by writing it directly in the MSI2 to MSI0 bits. The mask count is the value set in MSI2 to MSI0.

  When the MSI2 to MSI0 bits are set to "000_B", interrupt requests are not masked.

- The type of interrupt request differs depending on the count mode (MODE:bit21 in the PWMTCCS register). In the up-count mode, only the compare clear interrupt can be masked. The 0 detection interrupt is generated each time "0" is detected. In the up/down count mode, only the 0 detection interrupt can be masked.

The method of masking both of the interrupt requests is explained below.

- Only when the timer is in the up/down count mode, both of the interrupts can be masked by setting MODE2 in the PWMTCCS register and MODE in the TCCS register to 1.

To mask the 0 detection interrupt, use the MSI2 to MSI0 bits in the PWMTCCS register. To mask the compare clear interrupt, use the MSI5 to MSI3 bits in the TCCS register.

Note:

Software clear (SCLR:bit20 in the PWMTCCS register is set to 1) does not generate the 0 detection interrupt.

Figure 43-12. Compare Clear Interrupt Masked in the Up-count Mode

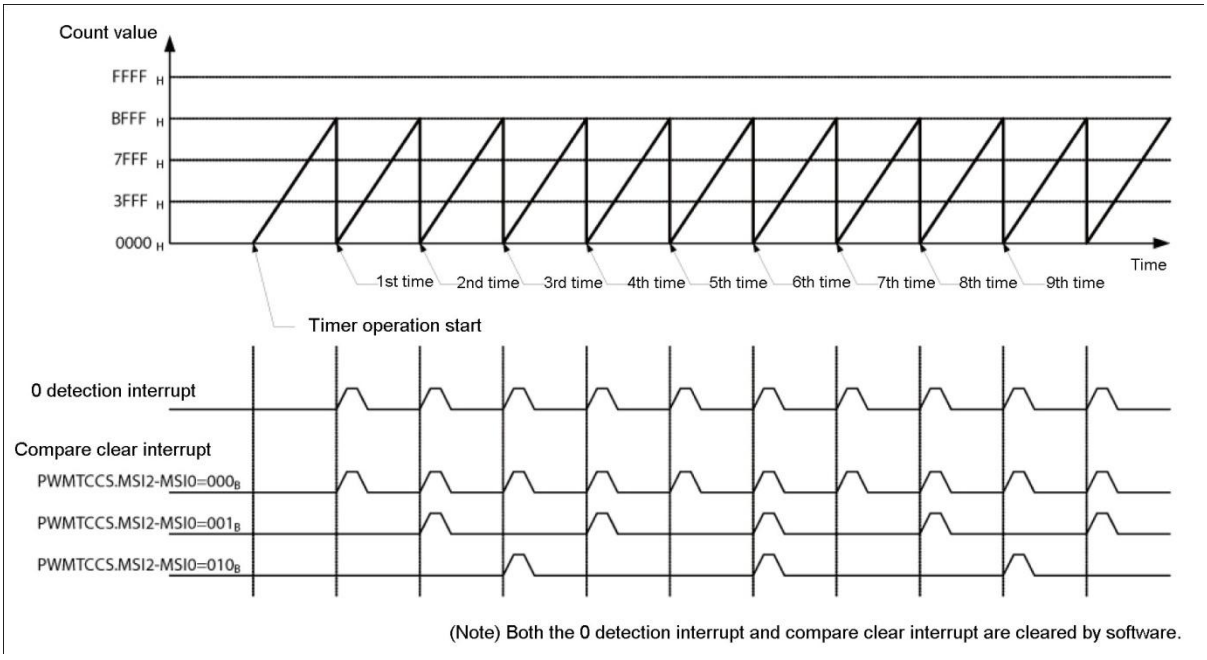


Figure 43-13.Detection Interrupt Masked in Up/down Count Mode

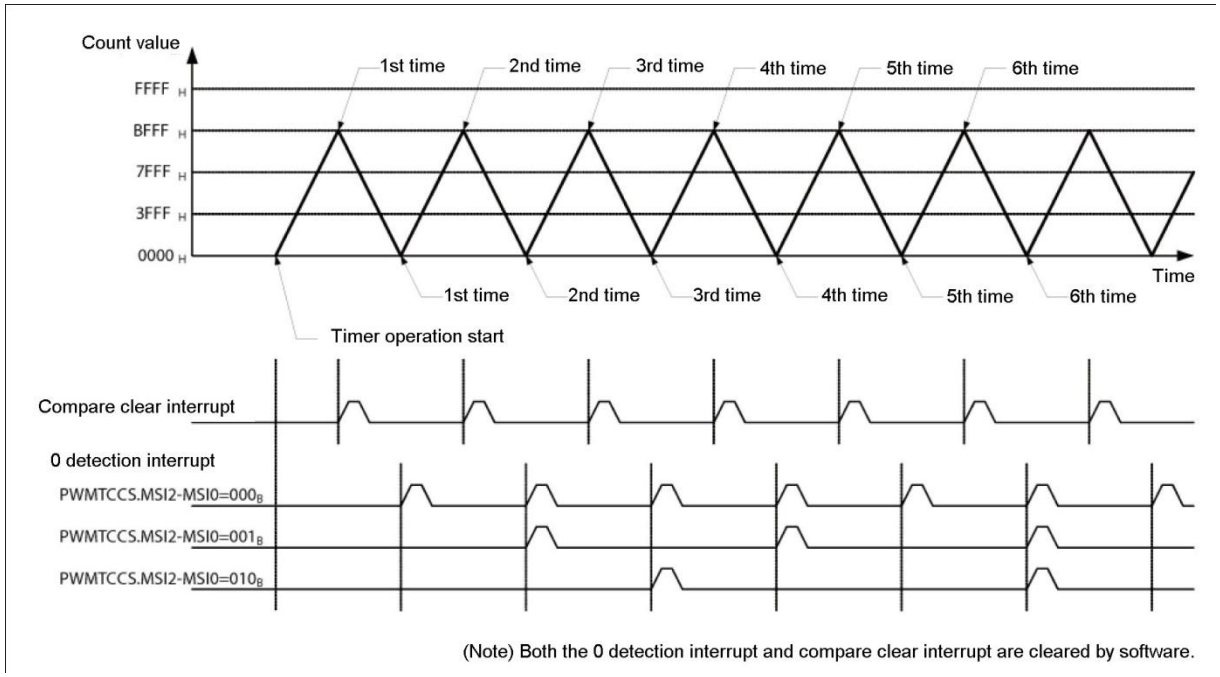
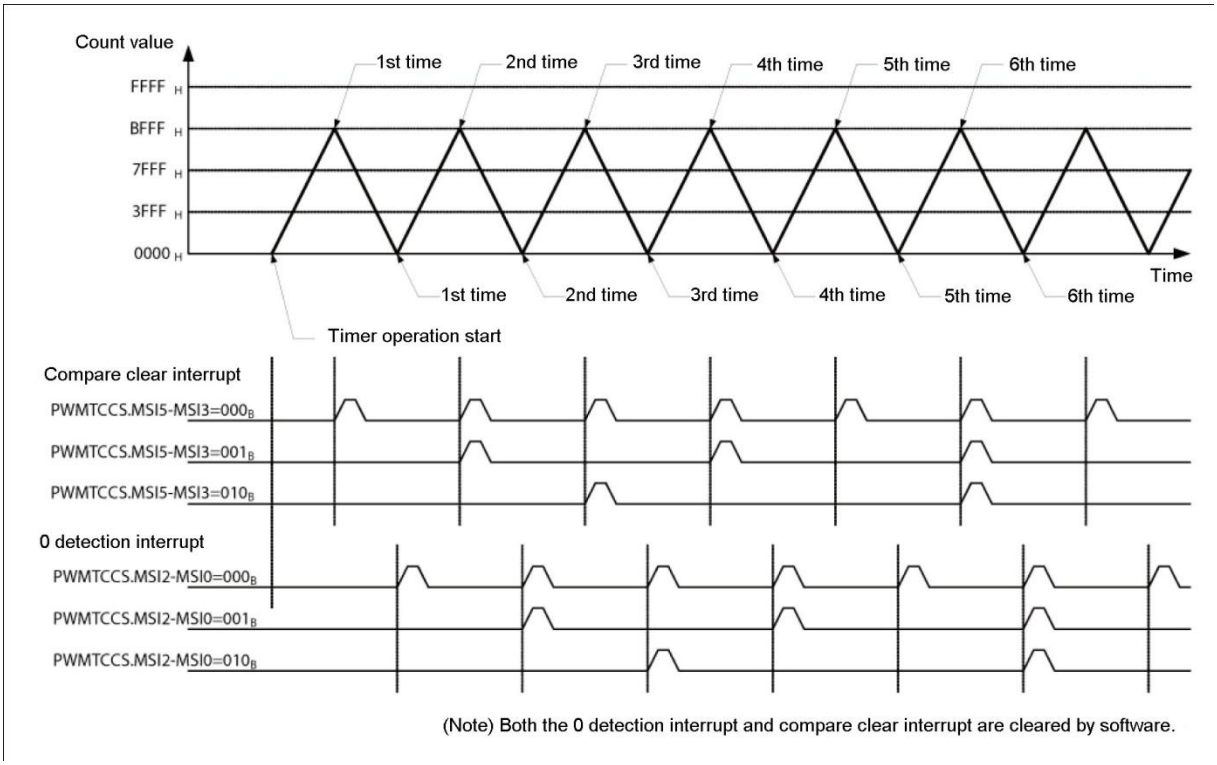


Figure 43-14. Detection Interrupt and Compare Clear Interrupt Masked in Up/down Count Mode



43.5.2.6 SYNCIN and SYNCOUT Functions

This section explains the SYNCIN and SYNCOUT functions.

The SYNCIN and SYNCOUT functions are used when two or more units of this module are operated in parallel. Figure 43-15 shows an example of the master operation applicable when the output to the SYNCOUT pin is 0 detection (SYNCOUT output selection bits (SYNOS2 to SYNOS0:bit10 to bit8) in the PWMTRC register). The "H" width of SYNCOUT (master clock) is the value set in the PWMMCDB register. Figure 43-16 shows an example of the slave operation. During the slave operation, the counter value is cleared to "0000_H" at the rising edge of SYNCIN.

Figure 43-15. SYNCOUT Output (Master Operation)

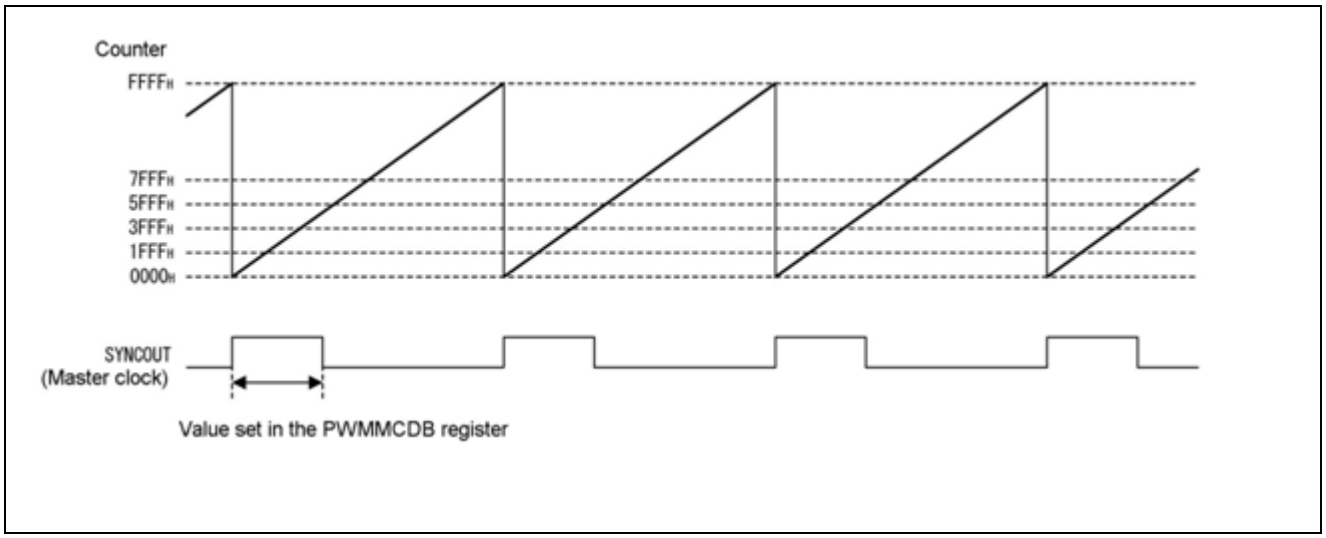
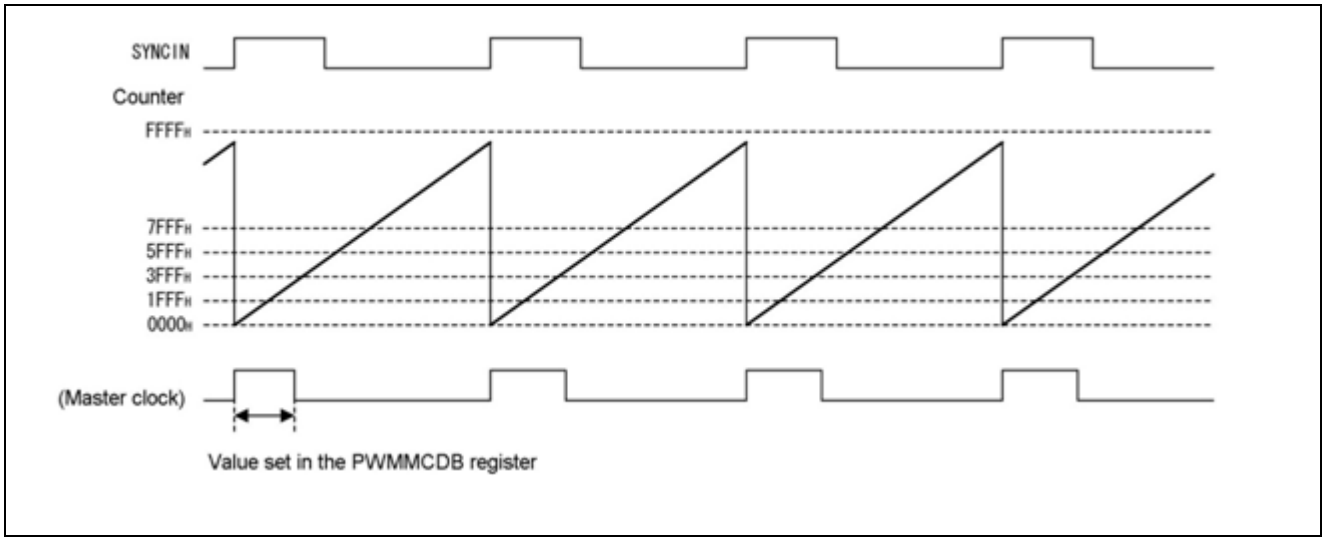


Figure 43-16. SYNCIN Input (Slave Operation)



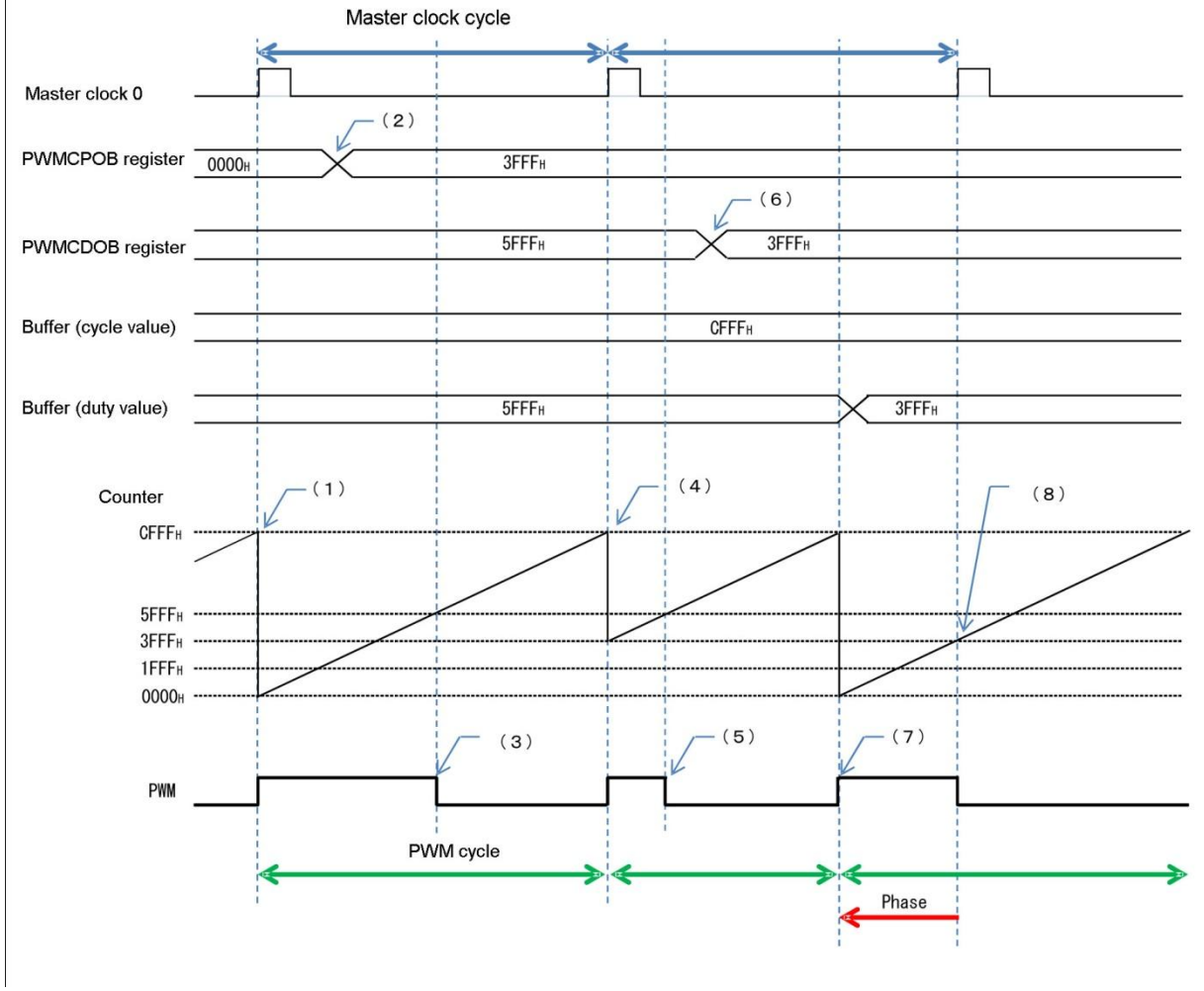
43.5.3 PWM Generation Operation

This section explains the PWM generation operation.

43.5.3.1 PWM Operation (Normal Waveform, Master Clock Synchronous, and Master Clock Cycle)

The PWM operation (normal waveform, master clock synchronous, and master clock cycle) is shown below.

Figure 43-17. PWM Operation (Normal Waveform, Master Clock Synchronous, and Master Clock Cycle)



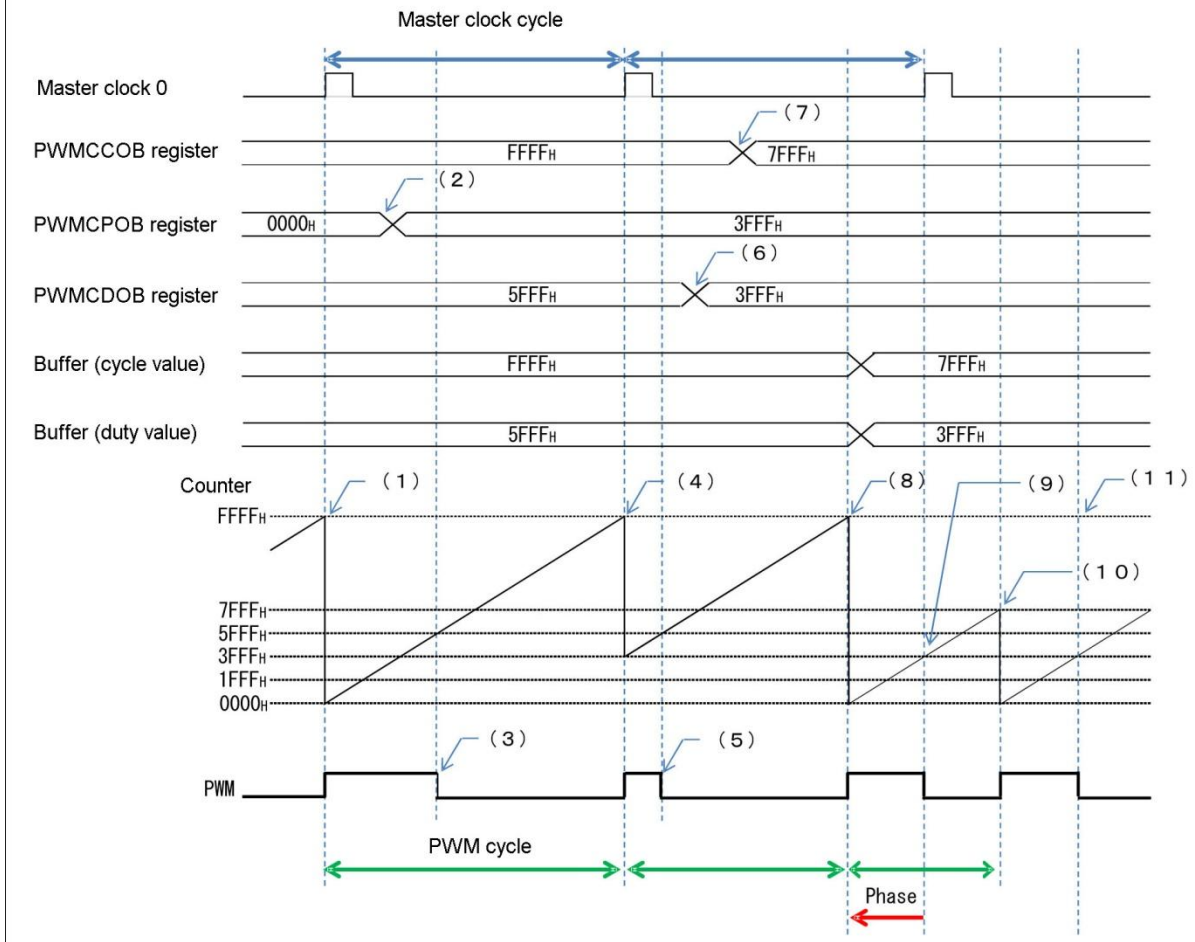
Explanation of operation

1. Master clock 0 loads PWMCP0B ("0000_H") to the counter. The PWM signal is changed to the "H" level.
2. Data is written to PWMCP0B (phase value).
3. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
4. Master clock 0 loads PWMCP0B ("$3FFF_H$") to the counter. The PWM signal is changed to the "H" level.
5. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
6. Data is written to PWMCD0B (duty value).
7. When the buffer (cycle value) and counter value match, the counter is cleared to "0". The PWM signal is changed to the "H" level. The duty value is transferred to the buffer (duty value).
8. Master clock 0 loads PWMCP0B ("$3FFF_H$") to the counter. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.

43.5.3.2 PWM Operation (Normal Waveform, Master Clock Synchronous, and Independent Cycle)

The PWM operation (normal waveform, master clock synchronous, and independent cycle) is shown below.

Figure 43-18. PWM Operation (Normal Waveform, Master Clock Synchronous, and Independent Cycle)



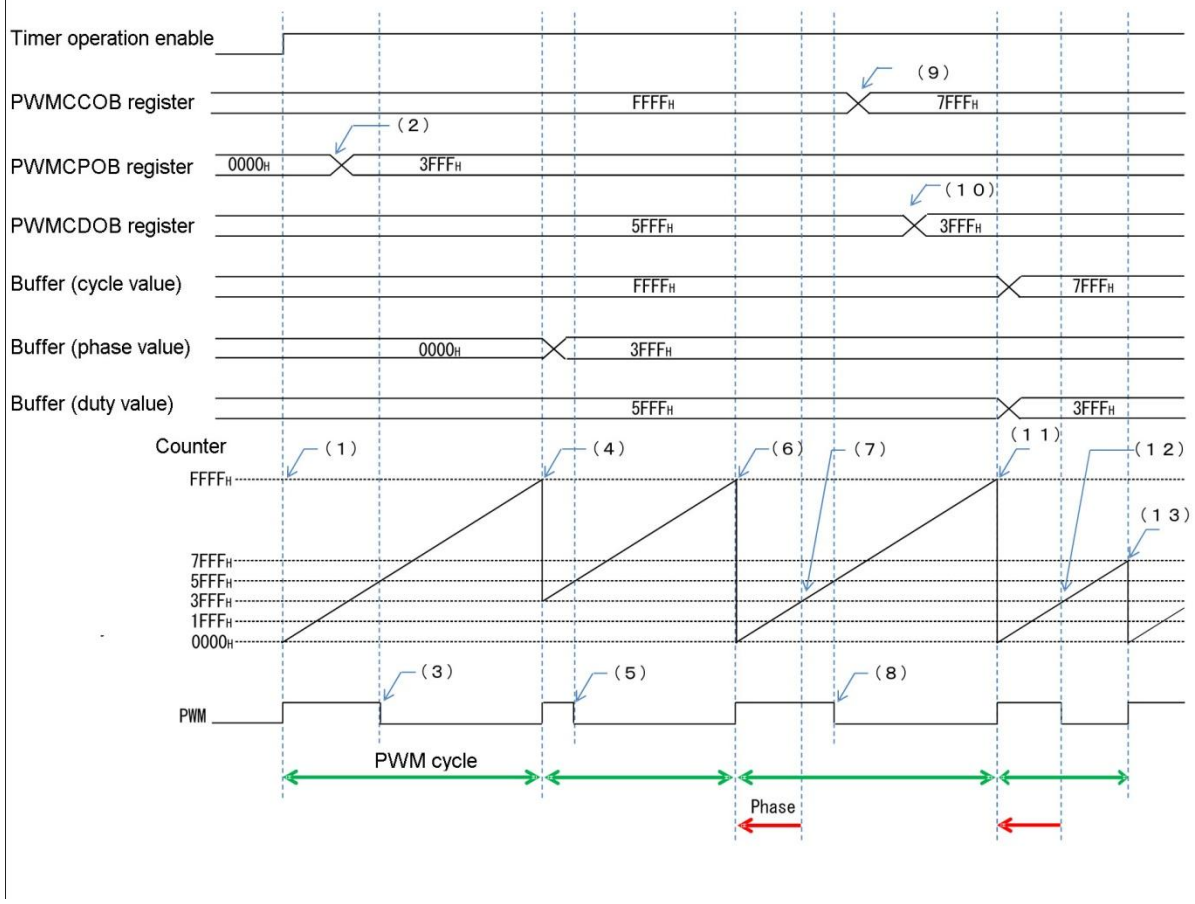
Explanation of operation

1. Master clock 0 loads PWMCP0B ("0000_H") to the counter. The PWM signal is changed to the "H" level.
2. Data is written to PWMCP0B (phase value).
3. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
4. Master clock 0 loads PWMCP0B ("$3FFF_H$") to the counter. The PWM signal is changed to the "H" level.
5. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
6. Data is written to PWMCD0B (duty value).
7. Data is written to PWMCC0B (cycle value).
8. When the buffer (cycle value) and counter value match, the counter is cleared to "0000_H". The PWM signal is changed to the "H" level. The cycle value is transferred to the buffer (cycle value). The duty value is transferred to the buffer (duty value).
9. Master clock 0 loads PWMCP0B ("$3FFF_H$") to the counter. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
10. When the buffer (cycle value) and counter value match, the counter is cleared to "0000_H". The PWM signal is changed to the "H" level.
11. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.

43.5.3.3 PWM Operation (Normal Waveform, Master Clock Asynchronous, and Independent Cycle)

The PWM operation (normal waveform, master clock asynchronous, and independent cycle) is shown below.

Figure 43-19. PWM Operation (Normal Waveform, Master Clock Asynchronous, and Independent Cycle)



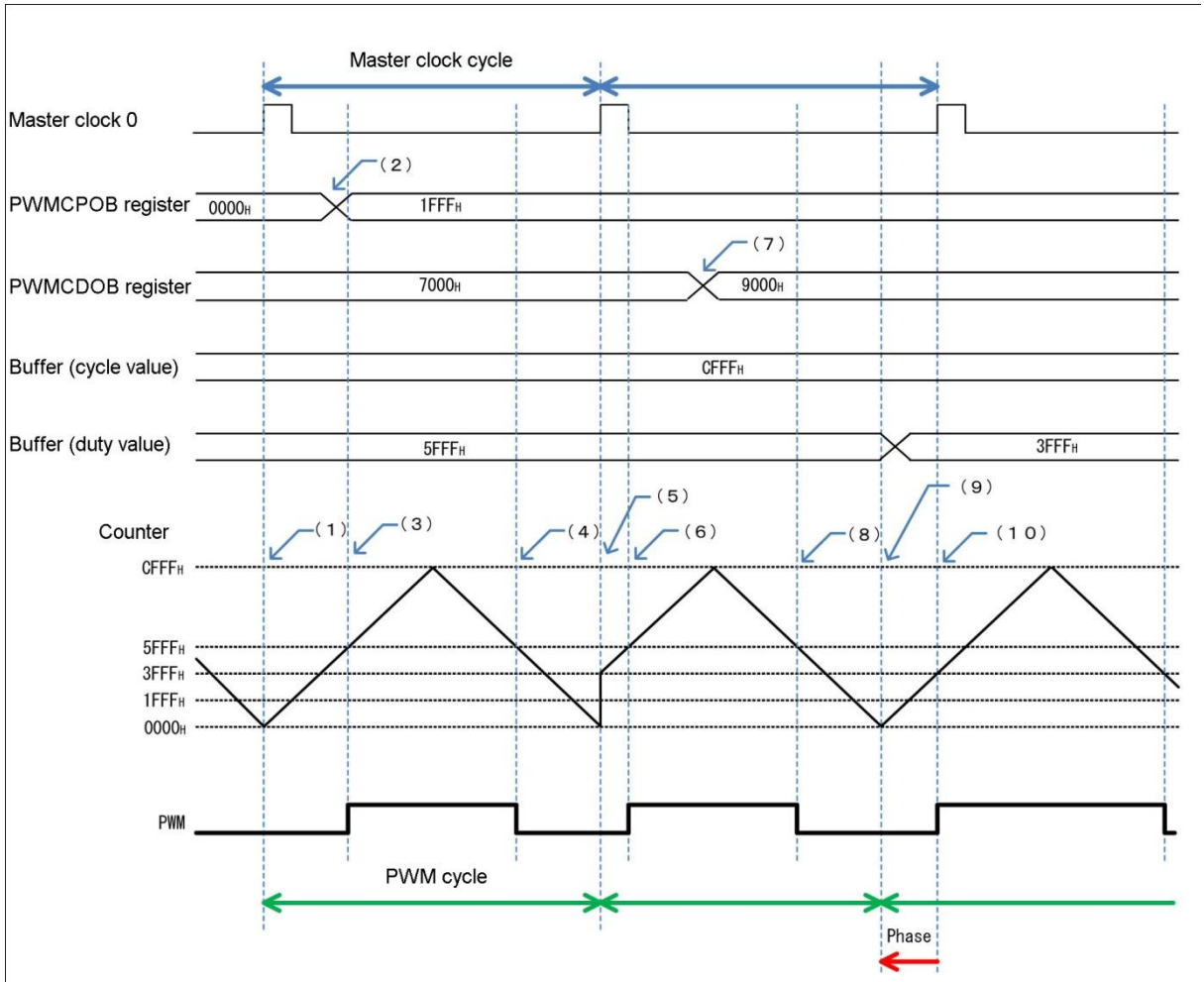
Explanation of operation

1. When timer operation enable is set to "H", the counter starts to count up. The PWM signal is changed to the "H" level.
2. Data is written to PWMCP0B (phase value).
3. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
4. When the buffer (phase value) is "0000_H" immediately after the timer operation is enabled, PWMCP0B ("$3FFF_H$") is loaded to the counter if the buffer (cycle value) and counter value match. The PWM signal is changed to the "H" level. The phase value is transferred to the buffer (phase value).
5. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
6. When the buffer (cycle value) and counter value match, the counter is cleared to "0000_H". The PWM signal is changed to the "H" level.
7. When the buffer (phase value) - 1 matches the counter value, PWMCP0B ("$3FFF_H$") is loaded to the counter.
8. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
9. Data is written to PWMCC0B (cycle value).
10. Data is written to PWMCD0B (duty value).
11. When the buffer (cycle value) and counter value match, the counter is cleared to "0000_H". The PWM signal is changed to the "H" level. The cycle value is transferred to the buffer (cycle value). The duty value is transferred to the buffer (duty value).
12. When the buffer (phase value) - 1 matches the counter value, PWMCP0B ("$3FFF_H$") is loaded to the counter. When the counter value becomes greater than the buffer (duty value), the PWM signal is inverted to the "L" level.
13. When the buffer (cycle value) and counter value match, the counter is cleared to "0000_H". The PWM signal is changed to the "H" level.

43.5.3.4 PWM Operation (Center Aligned Waveform, Master Clock Synchronous, and Master Clock Cycle)

The PWM operation (center aligned waveform, master clock synchronous, and master clock cycle) is shown below.

Figure 43-20. PWM Operation (Center Aligned Waveform, Master Clock Synchronous, and Master Clock Cycle)



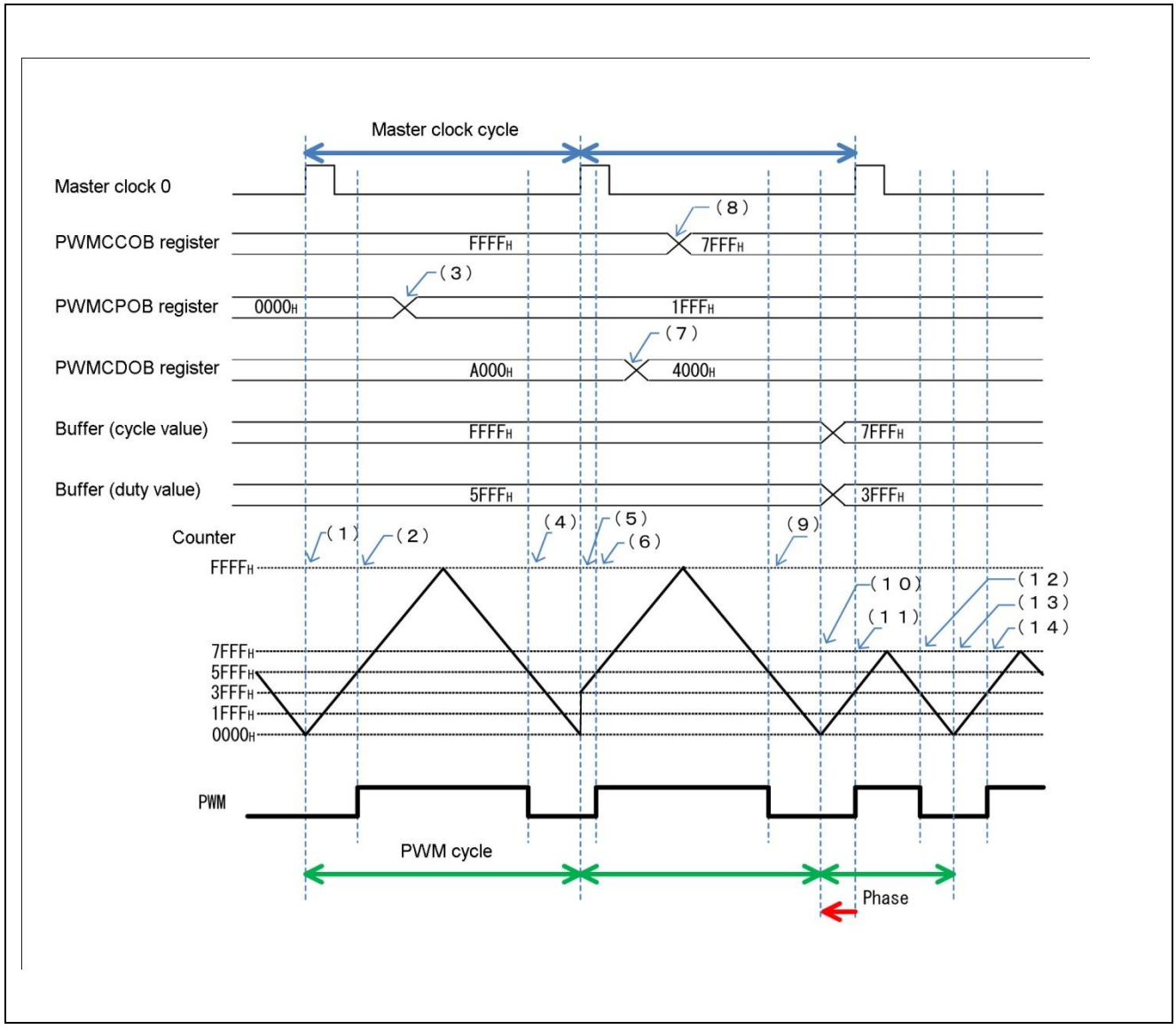
Explanation of operation

1. Master clock 0 loads PWMCP0B ("0000_H") to the counter.
2. Data is written to PWMCP0B (phase value).
3. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
4. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
5. Master clock 0 loads PWMCP0B ("$3FFE_H$") to the counter.
6. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
7. Data is written to PWMCD0B (duty value).
8. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
9. The counter value is "0000_H". The duty value is transferred to the buffer (duty value).
10. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.

43.5.3.5 PWM Operation (Center Aligned Waveform, Master Clock Synchronous, and Independent Cycle)

The PWM operation (center aligned waveform, master clock synchronous, and independent cycle) is shown below.

Figure 43-21. PWM Operation (Center Aligned Waveform, Master Clock Synchronous, and Independent Cycle)



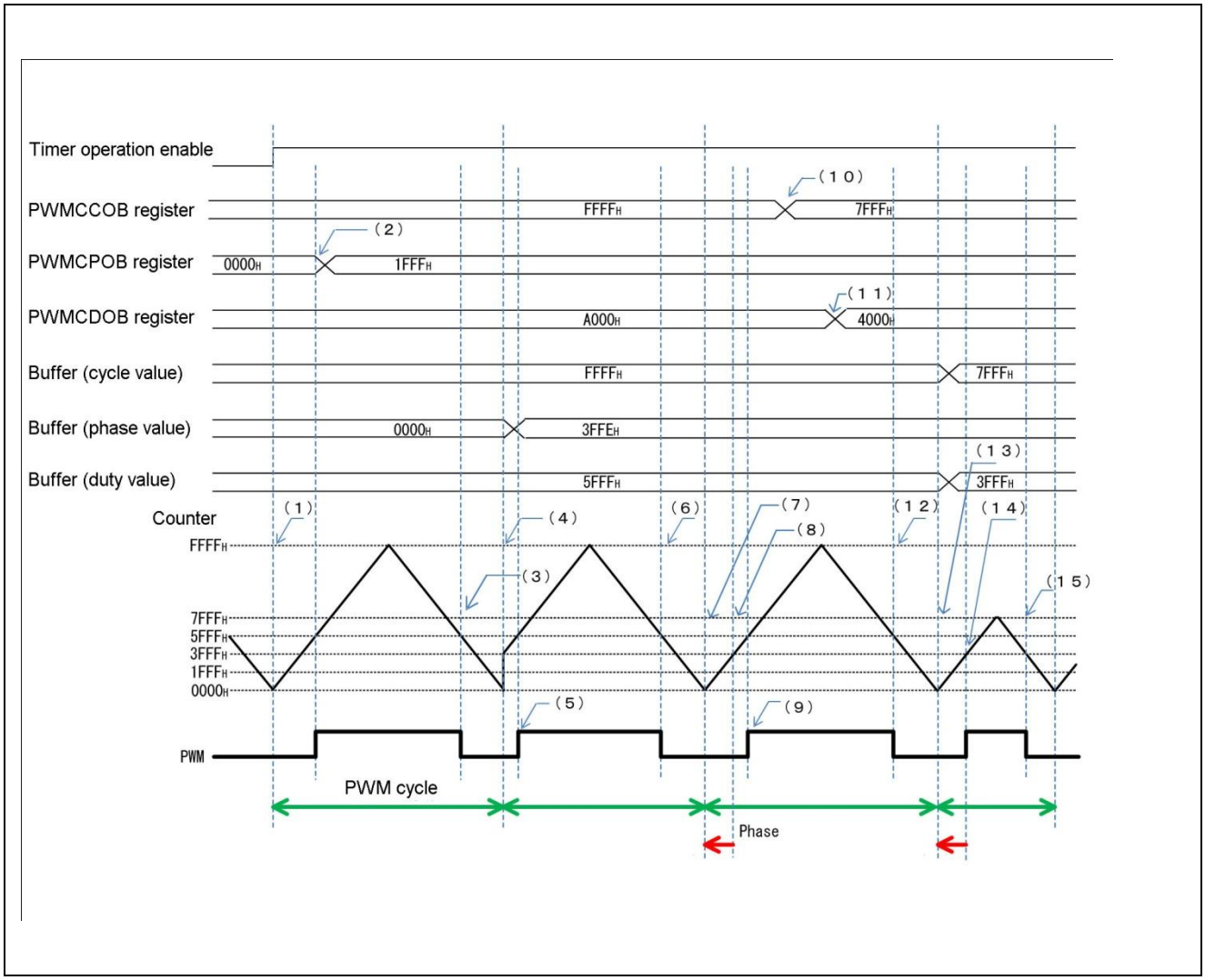
Explanation of operation

1. Master clock 0 loads PWMCP0B ("0000H") to the counter.
2. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
3. Data is written to PWMCP0B (phase value).
4. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
5. Master clock 0 loads PWMCP0B ("3FFEH") to the counter.
6. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
7. Data is written to PWMCD0B (duty value).
8. Data is written to PWMCC0B (cycle value).
9. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
10. The counter value is "0000H". The cycle value is transferred to the buffer (cycle value). The duty value is transferred to the buffer (duty value).
11. Master clock 0 loads PWMCP0B ("3FFEH") to the counter. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
12. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
13. The counter value is "0000H". The counter counts up.
14. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.

43.5.3.6 PWM Operation (Center Aligned Waveform, Master Clock Asynchronous, and Independent Cycle)

The PWM operation (center aligned waveform, master clock asynchronous, and independent cycle) is shown below.

Figure 43-22. PWM Operation (Center Aligned Waveform, Master Clock Asynchronous, and Independent Cycle)



Explanation of operation

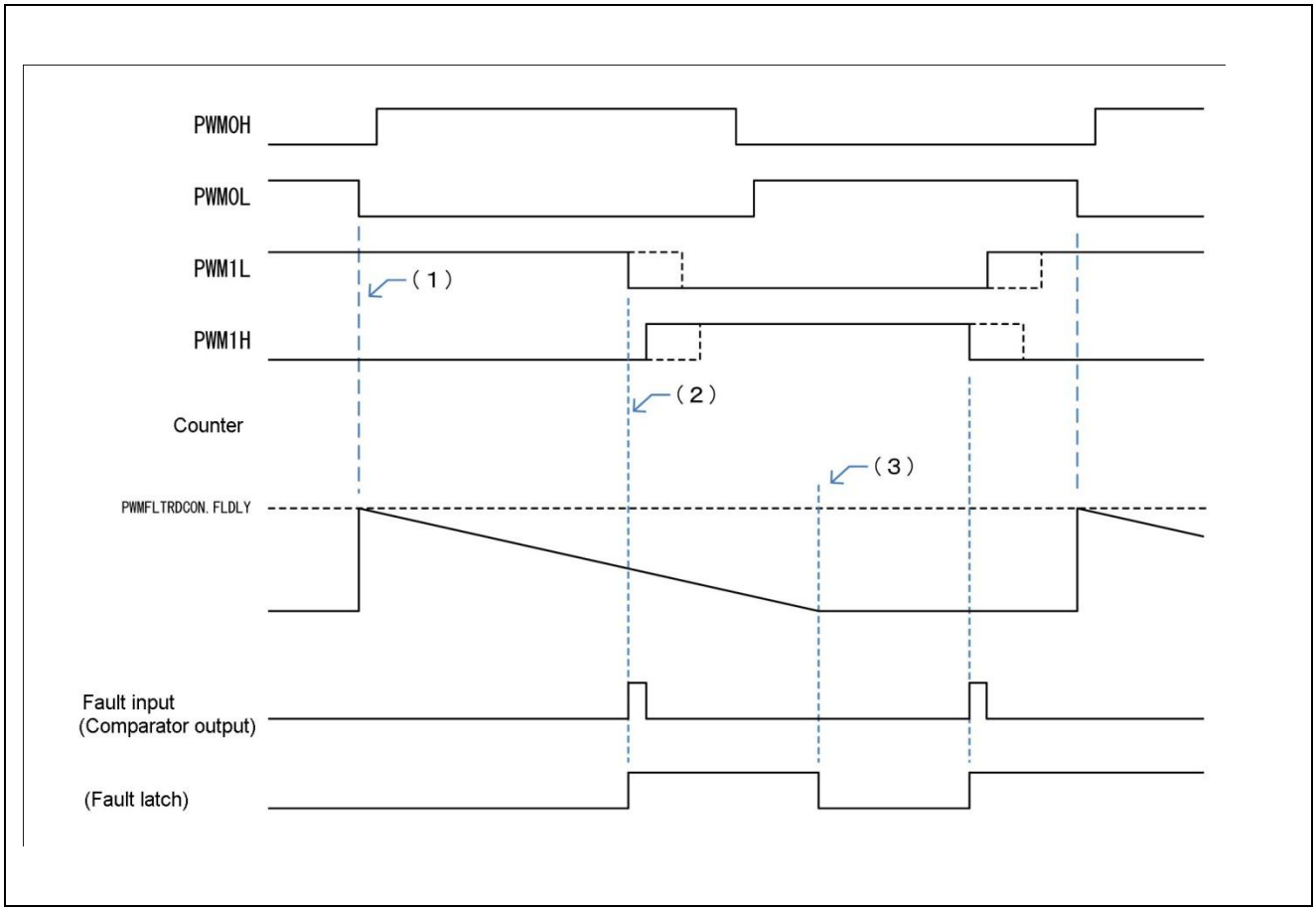
1. When timer operation enable is set to "H", the counter starts to count up.
2. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level. Data is written to PWMCP0B (phase value).
3. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
4. When the buffer (phase value) is "0000H" immediately after the timer operation is enabled, PWMCP0B ("3FFEH") is loaded to the counter if the counter value is "0000H". The phase value is transferred to the buffer (phase value).
5. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
6. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
7. The counter value is "0000_H", and the counter counts up.
8. When the buffer (phase value × 2) - 1 matches the counter value, PWMCP0B ("$3FFF_H$") is loaded to the counter.
9. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
10. Data is written to PWMCC0B (cycle value).
11. Data is written to PWMCD0B (duty value).
12. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.
13. The counter value is "0000_H". The cycle value is transferred to the buffer (cycle value). The duty value is transferred to the buffer (duty value).
14. When the buffer (phase value × 2) - 1 matches the counter value, PWMCP0B ("$3FFE_H$") is loaded to the counter. When the counter value becomes equal to or greater than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "H" level.
15. When the counter value becomes smaller than the "buffer (cycle value) - buffer (duty value)", the PWM signal is inverted to the "L" level.

43.5.4 Fault Control Operation

This section explains the fault control operation.

Figure 43-23 shows an example of the fault operation. This operation example assumes that the fault operation selection bits (FLT0H1, FLT0H0:bit11, bit10 and FLT0L1, FLT0L0:bit9, bit8) in the fault control register (PWMFLTCON10) are set to "01" and that the fault reset factor selection bits (FLSL02 to FLSL00:bit10 to bit8) in the fault reset control register (PWMFLTRCON1) are set to "001".

Figure 43-23. Fault Operation Example

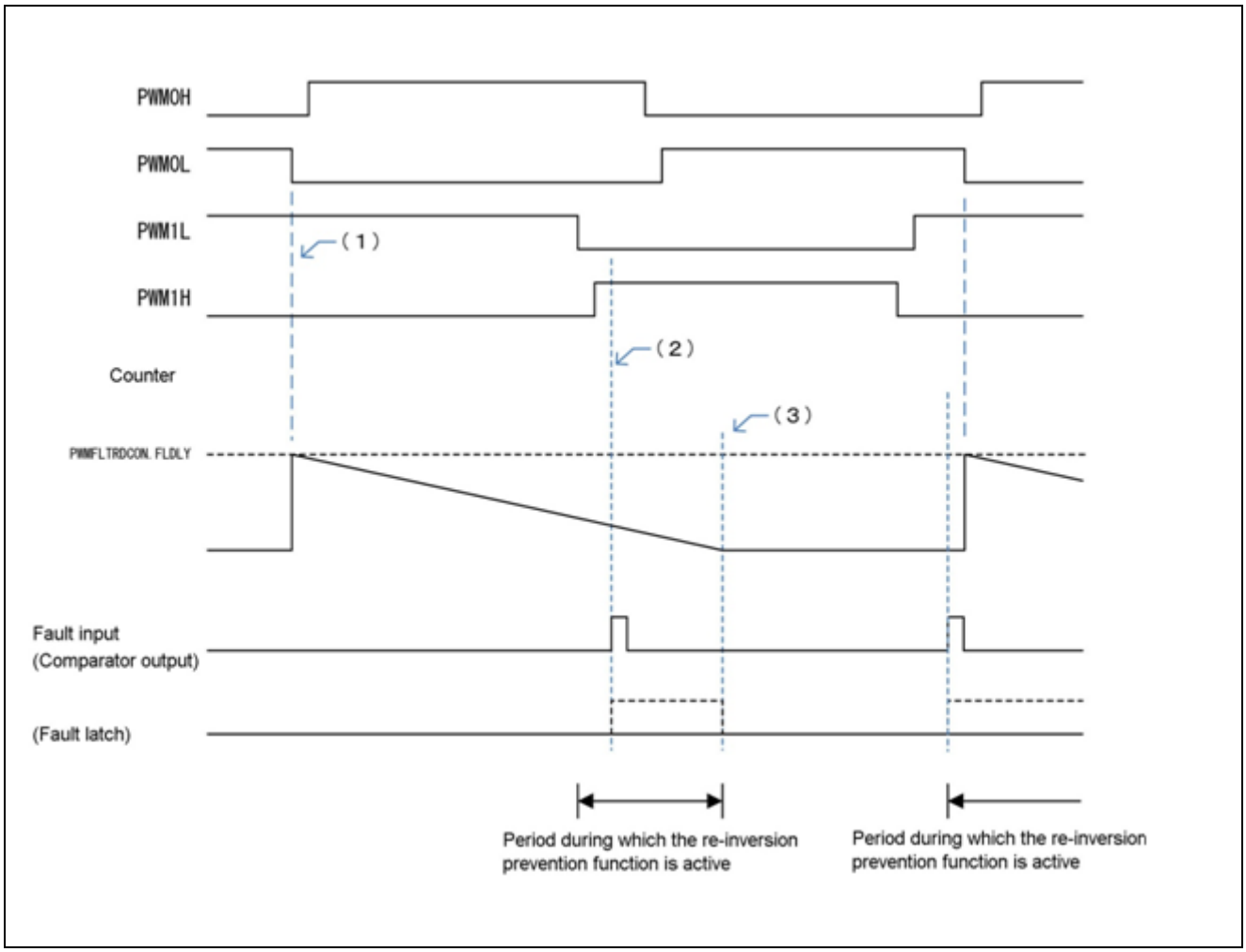


Explanation of operation

1. The value set in FLDLY15 to FLDLY0 in the PWMFLTRDCON register is loaded to the counter at the falling edge of PWM0L.
2. The fault operation inverts PWH1H and PWH1L. PWM1H is output with the dead time added to it.
3. When the counter value becomes "0000_H", the fault latch is reset.

Figure 43-24 shows an example of the fault operation in which the re-inversion prevention function prevents the PWM signal from being re-inverted. This operation example assumes that the fault operation selection bits (FLT0H1, FLT0H0:bit11, bit10 and FLT0L1, FLT0L0:bit9, bit8) in the fault control register (PWMFLTCON10) are set to "01" and that the fault reset factor selection bits (FLSL02 to FLSL00:bit10 to bit8) in the fault reset control register (PWMFLTRCON1) are set to "001".

Figure 43-24. Example of the Fault Operation in Which the Re-inversion Prevention Function Prevents the PWM Signal from Being Re-inverted



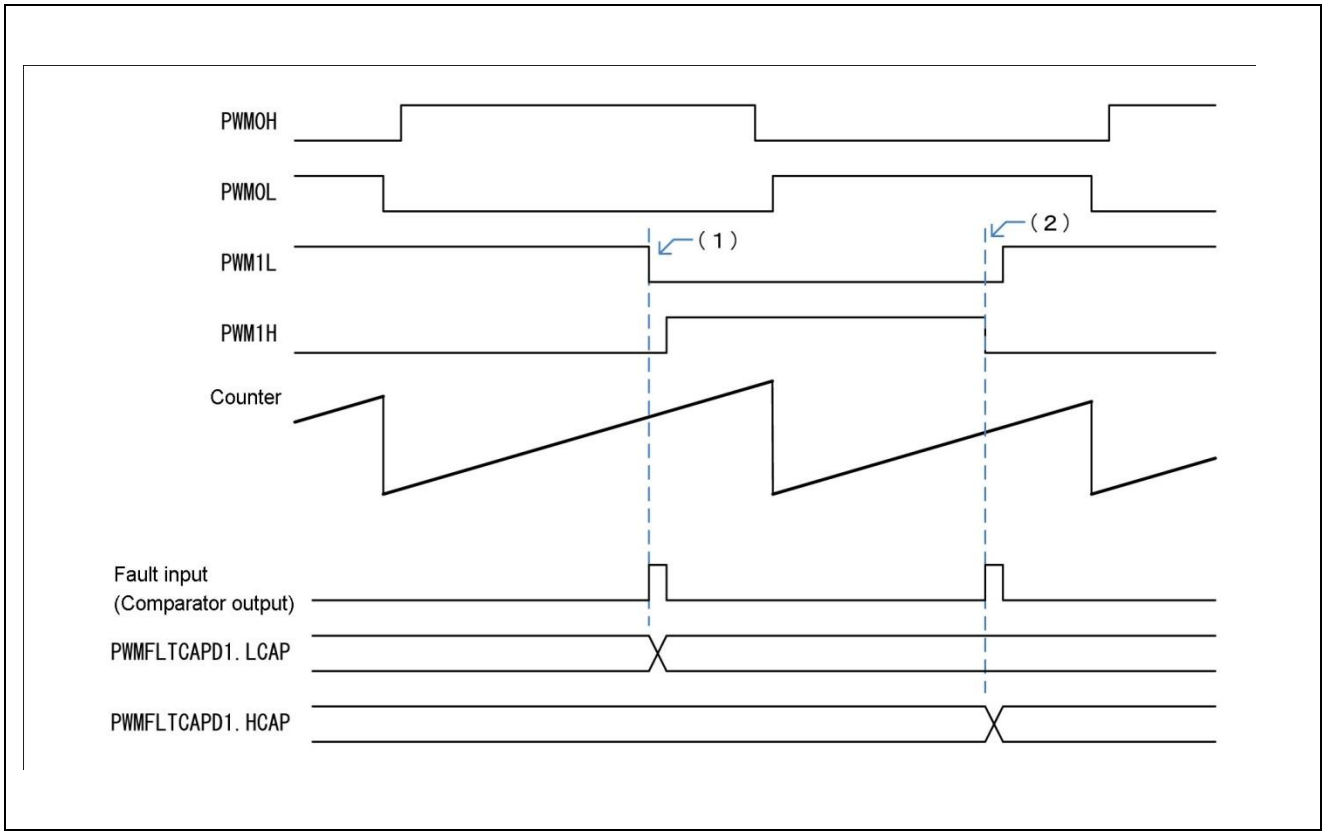
Explanation of operation

1. The value set in FLDLY15 to FLDLY0 in the PWMFLTRDCON register is loaded to the counter at the falling edge of PWM0L.
2. The re-inversion prevention function prevents PWM1H and PWM1L from being inverted by the fault input. PWM1H is output with the dead time added to it. Also, the re-inversion prevention function prevents the fault latch from becoming active.
3. The re-inversion prevention function is canceled when the fault latch is reset (counter value becomes "0000_H").

Figure 43-25 shows an example of the fault capture operation. This operation example assumes the following settings.

- The rising edge of PWM0L is selected in the capture counter reset factor selection bits in the fault control register (PWMFLTCON10) (CFHSL3 to CFHSL0:bit15 to bit12 are set to "1001").
- The falling edge of PWM0L is selected in the capture counter reset factor selection bits in the fault control register (PWMFLTCON11) (CFLSL3 to CFLSL0:bit15 to bit12 are set to "1001").
- The falling edge of PWM is selected in the high-side PWM capture edge selection bits in the fault capture control register (PWMFLTCAPCON1) (CEHSL2 to CEHSL0:bit6 to bit4 are set to "010").
- The falling edge of PWM is selected in the low-side PWM capture edge selection bits in the fault capture control register (PWMFLTCAPCON1) (CELSL2 to CELSL0:bit2 to bit0 are set to "010").

Figure 43-25. Fault Capture Operation Example



Explanation of operation

1. When the fault operation causes the falling edge of PWM1L, the counter value is captured in the low-side capture data bits (LCAP15 to LCAP00:bit15 to bit0) in the fault capture data register (PWMFLTCAPD1).
2. When the fault operation causes the falling edge of PWM1H, the counter value is captured in the high-side capture data bits (HCAP15 to HCAP00:bit31 to bit16) in the fault capture data register (PWMFLTCAPD1).

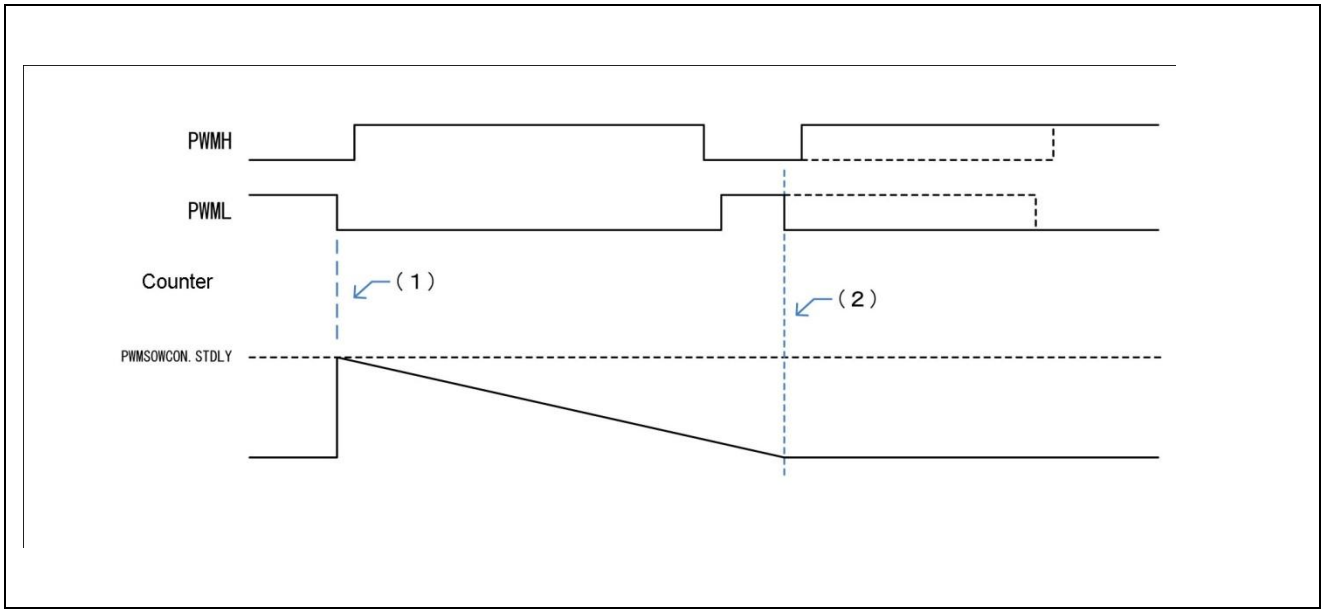
43.5.5 Soft Overwrite Control Operation

This section explains the soft overwrite control operation.

Figure 43-26. shows an example of the soft overwrite operation. This operation example assumes that the soft overwrite control register (PWMSOWCON) is set as follows.

- The overwrite selection bits 1 fix the output level to "L" (OSL11, OSL10:bit31, bit30 are set to "10").
- The overwrite selection bits 0 fix the output level to "H" (OSL01, OSL00:bit29, bit28 are set to "11").
- The OSL buffer enable bit is enabled (OSLBFE:bit27 is set to "1").
- The overwrite trigger is the falling edge of PWM (STSL2 to STSL0:bit26 to bit24 are set to "011").

Figure 43-26. Example of the Soft Overwrite Operation



Explanation of operation

1. At the falling edge of PWML, the value set in the trigger delay bits (STDLY15 to STDLY0:bit15 to bit0) in the soft overwrite control register (PWMSOWCON) is loaded to the counter.
2. When the counter value becomes "0000_H", the output level of PWMH is fixed to "H" and the output level of PWML is fixed to "L". PWMH is output with the dead time added to it.

43.5.6 Dead Time Operation

This section explains the dead time operation.

Figure 43-27. Dead Time in Normal Mode

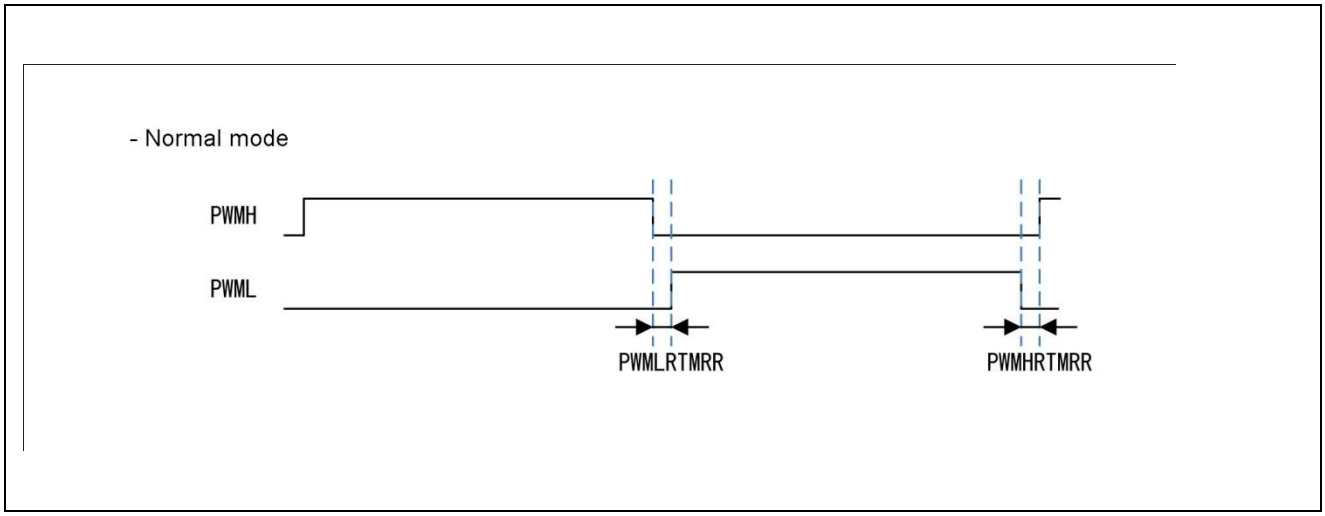
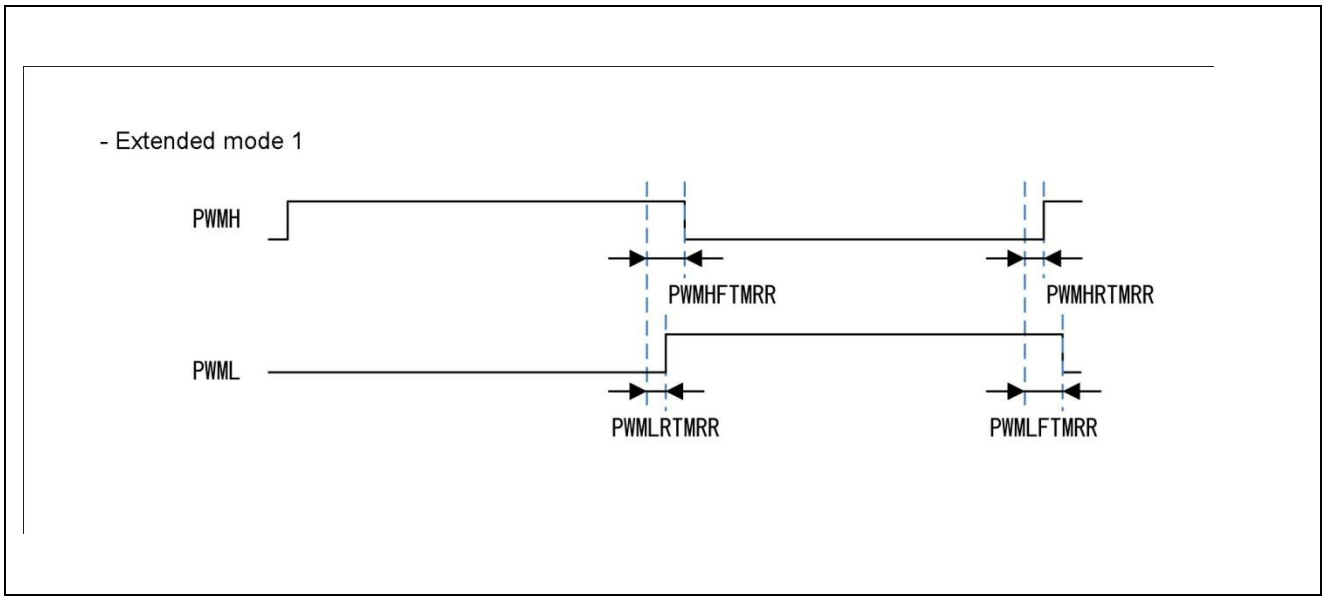


Figure 43-28. Dead Time in Extended Mode 1

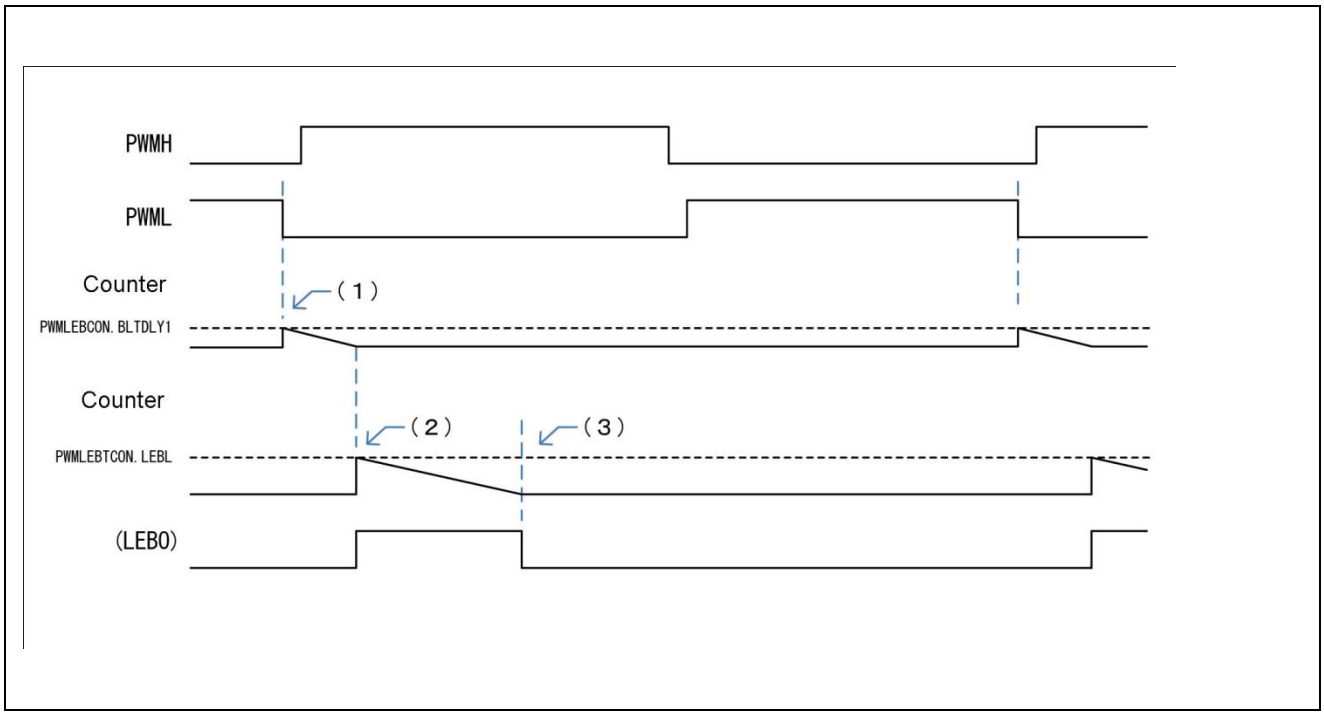


43.5.7 Blanking Generation Operation

This section explains the blanking generation operation.

Figure 43-29 shows an example of the blanking operation. This operation example assumes that the blanking trigger set in the blanking control register (PWMLEBCON) is the falling edge of PWML.

Figure 43-29. Example of the Blanking Operation



Explanation of operation

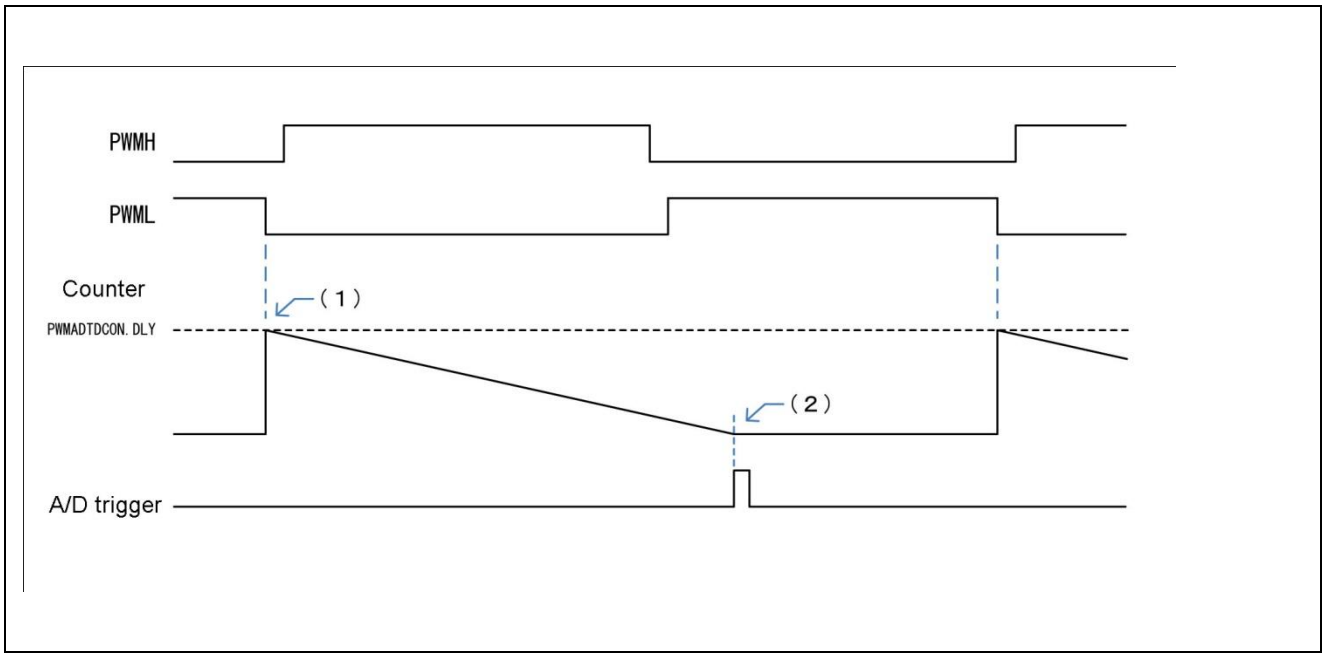
1. At the falling edge of PWML, the value set in the delay value bits (DLY15 to DLY00:bit15 to bit0) in the blanking start delay control register (PWMLEBSDCON) is loaded to the counter.
2. When the counter value becomes "0000_H", the output level of LEBO is changed to "H" and the value set in the LWB width bits (LEBL15 to LEBL00:bit15 to bit0) in the blanking time control register (PWMLEBTCON) is loaded as the counter value.
3. When the counter value becomes "0000_H", the output level of LEBO is changed to "L".

43.5.8 A/D Converter Trigger Generation Operation

This section explains the A/D converter trigger generation operation.

Figure 43-30, shows an example of the A/D converter trigger generation operation. This operation example assumes that the trigger selected in the A/DC trigger control register (PWMADTCON) is the falling edge of PWML.

Figure 43-30. Example of the A/D Converter Trigger Generation Operation



Explanation of operation

1. At the falling edge of PWML, the value set in the delay value bits (DLY15 to DLY00:bit15 to bit0) in the A/DC trigger delay control register (PWMADTDCON) is loaded to the counter.
2. When the counter value becomes "0000_H", the A/D trigger is output.

43.6 Notes

This chapter explains the notes about the PWM.

The notes about the use of the PWM are explained below.

Notes on accessing the buffer register

- The PWMCPCLR register in the timer has a buffer function. Do not use a read-modify-write instruction when accessing this register.

Notes on program settings

- While the timer value becomes "0000_H" when the timer is reset, the 0 detection interrupt flag is not set.
- While software clear (setting SCLR in the PWMTCCS register to 1) initializes the timer, the 0 detection interrupt and master clock are not generated.
- If counting starts when the compare value matches the count value, the compare clear flag is not set.
- Make sure that the compare value is other than "0000_H". Setting the compare value to "0000_H" will result in the following operations.
  - ☐ When the timer mode bit (MODE in the PWMTCCS register) is the up-count mode, the timer value is updated to and then fixed to "0000_H". The 0 detection interrupt flag and compare clear flag continue to be set for each count clock.
  - ☐ When the timer mode bit (MODE in the PWMTCCS register) is the up/down count mode, the timer counts up from "0000_H" to "$FFFF_H$" repeatedly. The 0 detection interrupt flag and compare clear flag are set when the timer value becomes "0000_H".

Notes on interrupts

- Be sure to clear the interrupt flag (IRQZF) before setting the interrupt request enable bit (IRQZE) in the timer state control register (PWMTCCS) to "1".
- Be sure to clear the interrupt flag (ICLR) before setting the interrupt request enable bit (ICRE) in the timer state control register (PWMTCCS) to "1".
- Be sure to clear the interrupt flag (ISEVF) in the special event status register (PWMSEVST) before setting the interrupt request enable bit (ISEVE) in the special event control register (PWMSEVCON) to "1".
- Be sure to clear the interrupt flag (IFLTF) in the PWM common status register (PWMST) before setting the interrupt request enable bit (IFLTE) in the fault control register (PWMFLTCON) to "1".
- Be sure to clear the interrupt flag (SOWIF) in the PWM common status register (PWMST) before setting the interrupt request enable bit (SOWIE) in the soft overwrite control register (PWMSOWCON) to "1".
- Be sure to clear the interrupt flag (CAPIF) in the PWM common status register (PWMST) before setting the interrupt request enable bit (CAPIE) in the fault capture control register (PWMFLTCAPCON) to "1".
- Be sure to clear the interrupt flag (ADTIF) in the A/DC trigger status register (PWMADTST) before setting the interrupt request enable bit (ADTIE) in the A/DC trigger control register (PWMADTCON) to "1".

Notes on accessing the PWMTCCS register

- In the case of a read-modify-write instruction, the set value is read from MSI2 to MSI0/MSI5 to MSI3.
- In the normal read operation, the counter value is read from MSI2 to MSI0/MSI5 to MSI3.

Notes on the parallel operation (master/slave operation)

- Make sure that the master clock cycle on the slave side is larger than the master clock cycle on the master side.
- The PWM output on the slave side is delayed from that on the master side because of the delay from the master-side SYNCOUT to the slave-side SYNCIN (5 to 30 ns + wiring delay in the board) and the internal delay from SYNCIN to the reset of the master clock timer (PWMCLK × 2 to 3 + PWM division clock × 2).

Notes on A/D converter trigger generation

- When generating the A/D converter trigger and trigger interrupt, make sure that the PWM cycle is equal to or greater than (3 × PWM division clock cycle + 3 × peripheral clock (PCLK) cycle). (Make sure that the A/D converter trigger is generated at intervals equal to or greater than (3 × PWM division clock cycle + 3 × peripheral clock (PCLK) cycle.)

Notes on PWM generation

- The following restrictions apply to the combination of the independent cycle mode (MCPS:bit12 in the PWM control register (PWMPCN) is set to "1") and the master clock synchronization (MCSEN:bit9 in the PWM control register (PWMPCN) is set to "1").
  - ☐ Make sure that the independent cycle (PWM cycle setting register (PWMCCB)) is 1/N times the master clock cycle. (N is an integer.)
  - ☐ Do not use SYNCIN synchronization for the master clock (set SYNCEN:bit26 or SYNCEN0:bit24 in the timer reset control register (PWMTRC) to "1").
  - ☐ When changing the PWM cycle, change the PWM cycle of the first PWM waveform between two adjacent rising edges of the master clock.
- When using the center aligned waveform (setting OWFS:bit9 in the PWM control register (PWMPCN) to "1"), note the following restrictions.
  - ☐ Do not set 16'h0000 or the cycle value as the duty value.
  - ☐ Do not use master clock synchronization (set MCSEN:bit9 in the PWM control register (PWMPCN) to "1") or SYNCIN synchronization (SYNCEN1:bit26 or SYNCEN0:bit24 in the timer reset control register (PWMTRC) to "1").
- When using the master clock cycle mode (MCPS:bit12 in the PWM control register (PWMPCN) to "0") in combination with master clock synchronization (setting MCSEN:bit9 in the PWM control register (PWMPCN) to "1"), disable the PWM-generated buffer (set CDFE:bit15 and CCFE:bit13 in the PWM control register (PWMPCN) to "0").
- When controlling the PWM by using the phase setting, do not change the cycle setting.
- Do not use master clock asynchronization (set MCSEN:bit9 in the PWM control register (PWMPCN) to "0") in combination with the master clock cycle mode (set MCPS:bit12 in the PWM control register (PWMPCN) to "0").

  When using master clock synchronization (setting MCSEN:bit9 in the PWM control register (PWMPCN) to "1") or master clock SYNCIN synchronization (SYNCEN:bit26 or SYNCEN0:bit24 in the timer reset control register (PWMTRC) to "1"), note the following restrictions.
- When "PWM cycle value - cycle of the master clock input for PWM generation = A", set the phase value in the following range: A ≤ phase value ≤ "PWM cycle value - A".
- When "PWM phase value - A" < PWM duty value, the PWN high width is reduced by "PWM cycle value - cycle of the master clock input for PWM generation" for the PWM output duty.

Notes on soft overwrite control

- When starting the PWM output (setting CNTE2 to CNTE0:bit2 to bit0 in the PWM simultaneous activation register (PWMPCGS) to "1"), fix the output level to "H" or "L" by using soft overwrite control (setting OSL11 to OSL10 and OSL01 to OSL00:bit31 to bit28 in the soft overwrite control register (PWMSOWCON) to "10" or "11"). When canceling soft overwrite control, wait for 1 PWM output cycle so that the output starts at the second PWM cycle.
- Before stopping the PWM output (setting CNTE2 to CNTE0:bit2 to bit0 in the PWM simultaneous activation register (PWMPCGS) to "0"), fix the PWM output level to "H" or "L" using soft overwrite control (set OSL11 to OSL10 and OSL01 to OSL00:bit31 to bit28 in the soft overwrite control register (PWMSOWCON) to "10" or "11").
- When the soft overwrite control buffer is enabled (OSLBFE:bit27 in the soft overwrite control register (PWMSOWCON) is set to "1"), wait for the SOW interrupt to occur before rewriting the soft overwrite control register (PWMSOWCON).

44. PWM Dedicated Clock

This chapter explains the PWM dedicated clock.

44.1 Overview

44.2 Features

44.3 Configuration

44.4 Registers

44.5 Clock Gear Circuit

44.6 Operation

44.7 Frequency and Gear Calculation Examples

44.8 Notes

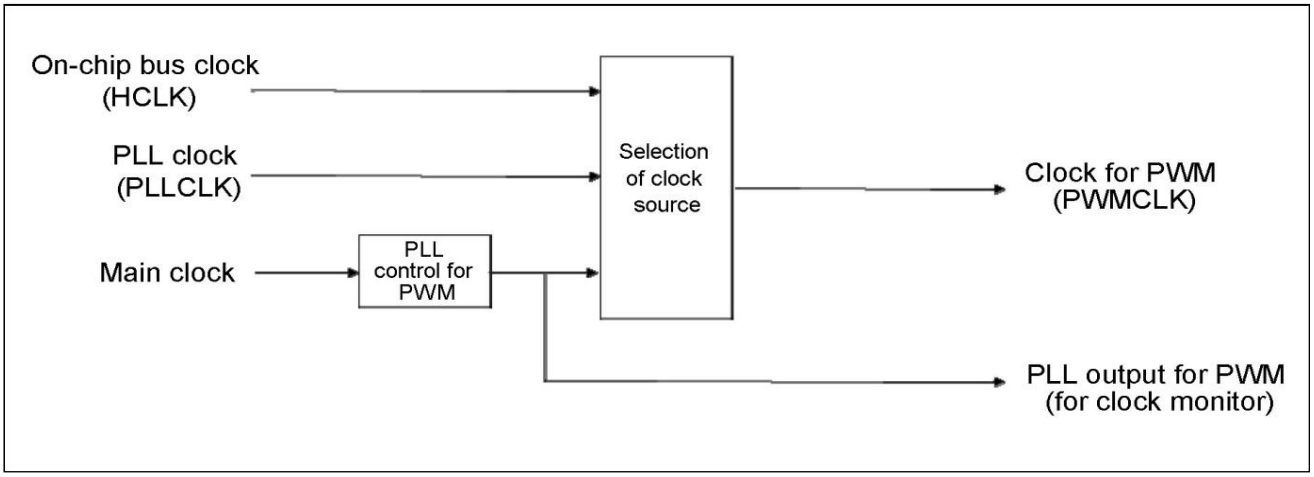
44.1 Overview

This section explains the overview of the PWM dedicated clock.

The PWM PLL installed in this model is separate from the PLL for the CPU core source clock.

This module controls PWM PLL oscillation and controls the clock.

Figure 44-1. Block Diagram



44.2 Features

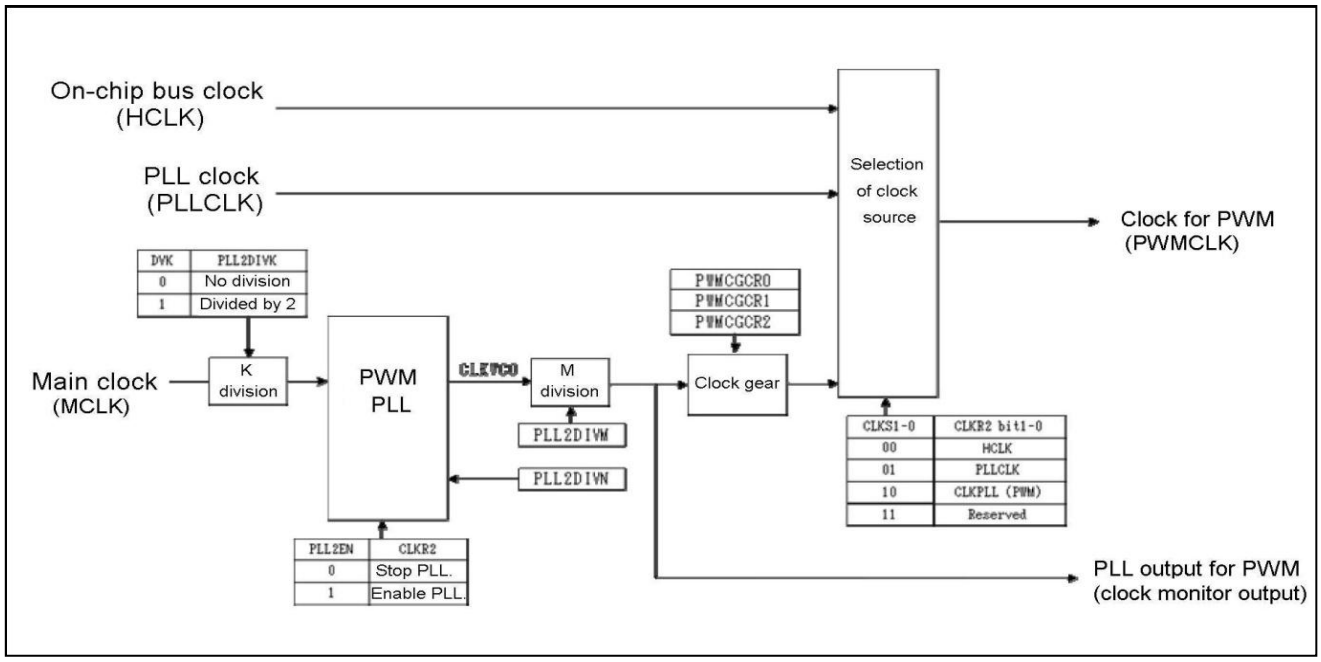
This section explains the features of the PWM dedicated clock.

- Freely programmable PLL multiplication rate
- Clock gear up/down function for preventing voltage drops and voltage surges
- PWM clock (PWMCLK) source clock selection function
- Function that generates an interrupt upon detecting the PWM PLL macro deadlock state

44.3 Configuration

This section explains the configuration of the PWM dedicated clock.

Figure 44-2. Block Diagram (Details)



44.4 Registers

This section explains the PWM dedicated clock registers.

Figure 44-3. Register Map

Address	Registers				Register Functions
	+0	+1	+2	+3	
0x04E8	PLL2DIVM	PLL2DIVN	Reserved	Reserved	PWM PLL multiplication rate (M division) selection register PWM PLL multiplication rate (N division) selection register
0x04EC	Reserved	PLL2DIVK	CLKR2	Reserved	PWM PLL multiplication rate (K division) selection register PWM PLL clock output control register
0x04F0	Reserved	PWMCGRCR0	PWMCGRCR1	PWMCGRCR2	Clock gear configuration setting register 0 Clock gear configuration setting register 1 Clock gear configuration setting register 2

44.4.1 PWM PLL Division (M Division) Selection Register: PLL2DIVM

This section shows the bit configuration of the PWM PLL division (M division) selection register.

This register selects the PWM PLL clock division.

PLL2DIVM: Address $04E8_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-				DVM3	DVM2	DVM1	DVM0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W

[bit7 to bit4] Undefined

"0" is always read. Writing has no influence on an operation.

[bit3 to bit0] DVM3 to DVM0: CLKVCO M division selection

DVM3 to DVM0	CLKVCO M Division (Generation φ: PWMCLK)
0000	CLKVCO: 2 (divided by 2)
0001	
0010	CLKVCO: 3 (divided by 3)
0011	CLKVCO: 4 (divided by 4)
0100	CLKVCO: 5 (divided by 5)
0101	CLKVCO: 6 (divided by 6)
0110	CLKVCO: 7 (divided by 7)
0111	CLKVCO: 8 (divided by 8)
- - - -	- - - -
1111	CLKVCO: 16 (divided by 16)

Note:

- Odd-numbered division ratios (:3, :5, :7, etc.) can be selected for the M division counter, but they are not recommended values.
  The generated output clock has an odd-numbered duty ratio.
  Always select an even-numbered division ratio (:2, :4, :6, etc.).
- The register value cannot be changed when CLK PLL (PWM) is selected as the clock source. (CLKR2:CLKS[1;0]=10)
- To change the PLL2DIVM or PLL2DIVN register, it is recommended to stop PLL (CLKR2:PLL2EN=0) and then enable PLL (CLKR2:PLL2EN=1).

44.4.2 PWM PLL Multiplication Rate (N Division) Selection Register: PLL2DIVN

This section shows the bit configuration of the PWM PLL multiplication rate (N division) selection register.

This register selects the multiplication rate from the PLL input clock to the PWM PLL clock.

PLL2DIVN: Address $04E9_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	DVN6	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7] Undefined

"0" is always read. Writing has no influence on an operation.

[bit6 to bit0] DVN6 to DVN0: PWM PLL macro FB input division ratio setting

DVN6 to DVN0	Division Ratio Setting
0000000-0001011	Setting disable
0001100	13
0001101	14
0001110	15
- - - -	- - - -
1100010	99
1100011	100
1100100-1111111	Setting disable

Note:

- The register value cannot be changed when CLK PLL (PWM) is selected as the clock source. (CLKR2:CLKS[1;0]=10)
- To change the PLL2DIVM or PLL2DIVN register, it is recommended to stop PWM PLL (CLKR2:PLL2EN=0) and then enable PWM PLL (CLKR2:PLL2EN=1).

44.4.3 PWM PLL Multiplication Rate (K Division) Selection Register: PLL2DIVK

This section explains the bit configuration of the PWM PLL multiplication rate (K division) selection register.

This register selects the PWM PLL input clock division.

PLL2DIVK: Address $04ED_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-							DVK
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit7 to bit1] Undefined

"0" is always read. Writing has no influence on an operation.

[bit0] DVK: MCLK K division selection

This bit selects the main clock (MCLK) division for the PWM PLL input clock as follows.

DVK	MCLK (PLL Input Clock) K Division
0	MCLK/1 (no division)
1	MCLK/2 (divided by 2)

Note:

The register value cannot be changed when CLK PLL (PWM) is selected as the clock source. (CLKR2:CLKS[1;0]=10)

44.4.4 PWM PLL Clock Output Control Register: CLKR2

This section shows the bit configuration of the PWM PLL clock output control register.

This register sets the control of PWM PLL clock operation.

CLKR2: Address $04EE_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FPOVF	FPOVIR	FPOVIE	Reserved		PLL2EN	CLKS1	CLKS0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R(RM1),W	R/W	R/W0	R0,W0	R/W	R/W	R/W

[bit7] FPOVF: PWM PLL alarm flag

This flag indicates the detection of the PWM PLL macro deadlock state.

FPOVF	PWM PLL Deadlock State Detection
0	Normal lock state
1	Deadlock state

[bit6] FPOVIR: PWM PLL alarm interrupt request flag

This flag indicates a request for a PWM PLL macro alarm interrupt.

A PWM PLL alarm interrupt is generated when this bit is "1" and the PWM PLL alarm interrupt request (FPOVIE) is "1". If the read-modify-write instruction is executed, "1" will be read out.

FPOVIR	Read	Write
0	Normal lock state	Clear the flag.
1	Deadlock state	Disabled

[bit5] FPOVIE: PWM PLL alarm interrupt request enable

This bit sets whether to generate a PWM PLL alarm interrupt request when the PWM PLL macro alarm interrupt request flag bit is "1".

FPOVIE	PWM PLL Alarm Interrupt Request Enable
0	Disable interrupt requests.
1	Enable interrupt requests.

[bit4 to bit3] Reserved

Be sure to write "0" to these bits.

[bit2] PLL2EN: PWM PLL selection enable

This bit sets PWM PLL operation as follows.

PLL2EN	Function
0	Stop PWM PLL (initial value).
1	Enable PWM PLL operation.

Changes to the PWM PLL operation enable bit (PLL2EN) are prohibited when CLK PLL (PWM) is selected as the clock source (CLKS[1:0]=10).
Before entering stop mode, set this bit to "0".

[bit1, bit0] CLKS1, CLKS0: PWMCLK output selection

These bits set the selection of PWMCLK output from the PWM PLL-I/F as follows.

CLKS1, CLKS0	Function (PWMCLK Output Selection)
00	HCLK (initial value)
01	PLLCLK
10	CLKPLL (PWM)
11	Reserved

To use PWM, set CLKS[1:0]=10.

44.4.5 Clock Gear Configuration Setting Register 0: PWMCGRCR0

This section shows the bit configuration of clock gear configuration setting register 0.

This register sets various settings of the clock gear.

PWMCGRCR0: Address 04F1$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	GRSTS[1:0]		Reserved				GRSTR	GREN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM0),W1	R/W

[bit7,bit6] GRSTS[1:0] (clock GeaR STatus flags): Clock status flag

These bits display the clock gear status.

GRSTS[1:0]	Status
00	Stopped in clock gear low-speed oscillation status, or not using clock gear (PWMCGRCR0.GREN=0), or PWM reset status (CLKR2.PLL2EN)
01	Gear-up operation in progress
10	Stopped in clock gear high-speed oscillation status
11	Gear-down operation in progress

[bit5 to bit2] Reserved

Be sure to write "0" to these bits.

[bit1] GRSTR (clock GeaR STaRt): Clock gear start

Writing "1" to this bit starts clock gear operation.

Clock gear operation varies depending on the GRSTS bit value. (gear-up or gear-down operation)

When GRSTS=00

GRSTS	Operation
Writing "0"	No effect on operation
Writing "1"	Gear-up operation start

When GRSTS=01/11

GRSTS	Operation
Writing "0"	No effect on operation
Writing "1"	No effect on operation

When GRSTS=10

GRSTS	Operation
Writing "0"	No effect on operation
Writing "1"	Gear-down operation start

Note:

Writing to this bit is enabled only when PWMCGRCR0.GREN=1 (clock gear enable).

This bit is automatically cleared to "0" when the clock gear-up (down) operation ends. Also, this bit is cleared to "0" when CLKR2.PLL2EN=0 (PWM PLL stopped).

If a read-modify-write instruction is executed, "0" is always read from this bit. If writing is executed while this bit is "1", writing for the 2nd and subsequent times is ignored.

[bit0] GREN (clock GeaR ENable): Clock gear enable

This bit enables clock gear operation.

GREN	Operation
0	Do not use clock gear.
1	Use clock gear.

Note:

This bit can be written only when PWM PLL is stopped (CLKR2.PLL2EN="0").

44.4.6 Clock Gear Configuration Setting Register 1: PWMCGRCR1

This section shows the bit configuration of clock gear configuration setting register 1.

This register sets various settings of the clock gear.

This register can be written only when PWM PLL is stopped (CLKR2.PLL2EN="0").

PWMCGRCR1: Address04F2$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	GRSTP[1:0]		GRSTN[5:0]					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7,bit6] GRSTP[1:0] (clock GeaR STePselect): Clock gear step number

These bits select the number of steps (number of increments or decrements) for the clock gear up/down time.

GRSTP[1:0]	Number of Steps
00	1
01	2
10	3
11	4

[bit5 to bit0] GRSTN[5:0] (clock GeaR STart step Number select): Clock gear start step selection

These bits select the steps for the start of clock gear operation. A value between 0 and 63 can be selected.

GRSTN[5:0]	Number of Steps
000000	0
000001	1
000010	2
...	.....
111101	61
111110	62
111111	63

Note:

If the setting is GRSTN=111111 (number of steps: 63), the gear does not operate.

44.4.7 Clock Gear Configuration Setting Register 2: PWMCGRCR2

This section shows the bit configuration of division setting register 2.

This register can be written only when PWM PLL is stopped (CLKR2.PLL2EN="0").

PWMCGRCR2: Address 04F3$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	GRLP[7:0]							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit0] GRLP[7:0] (clock GeaR LooP number select): Clock gear iteration number selection

These bits select the number of iterations in 1 step. The number of iterations that can be set is between 1 and 256. The step is incremented/decremented at the completion of the set number of times in these bits.

GRLP[7:0]	Number of Loops
0000_0000	1
0000_0001	2
0000_0010	3
...	.....
1111_1101	254
1111_1110	255
1111_1111	256

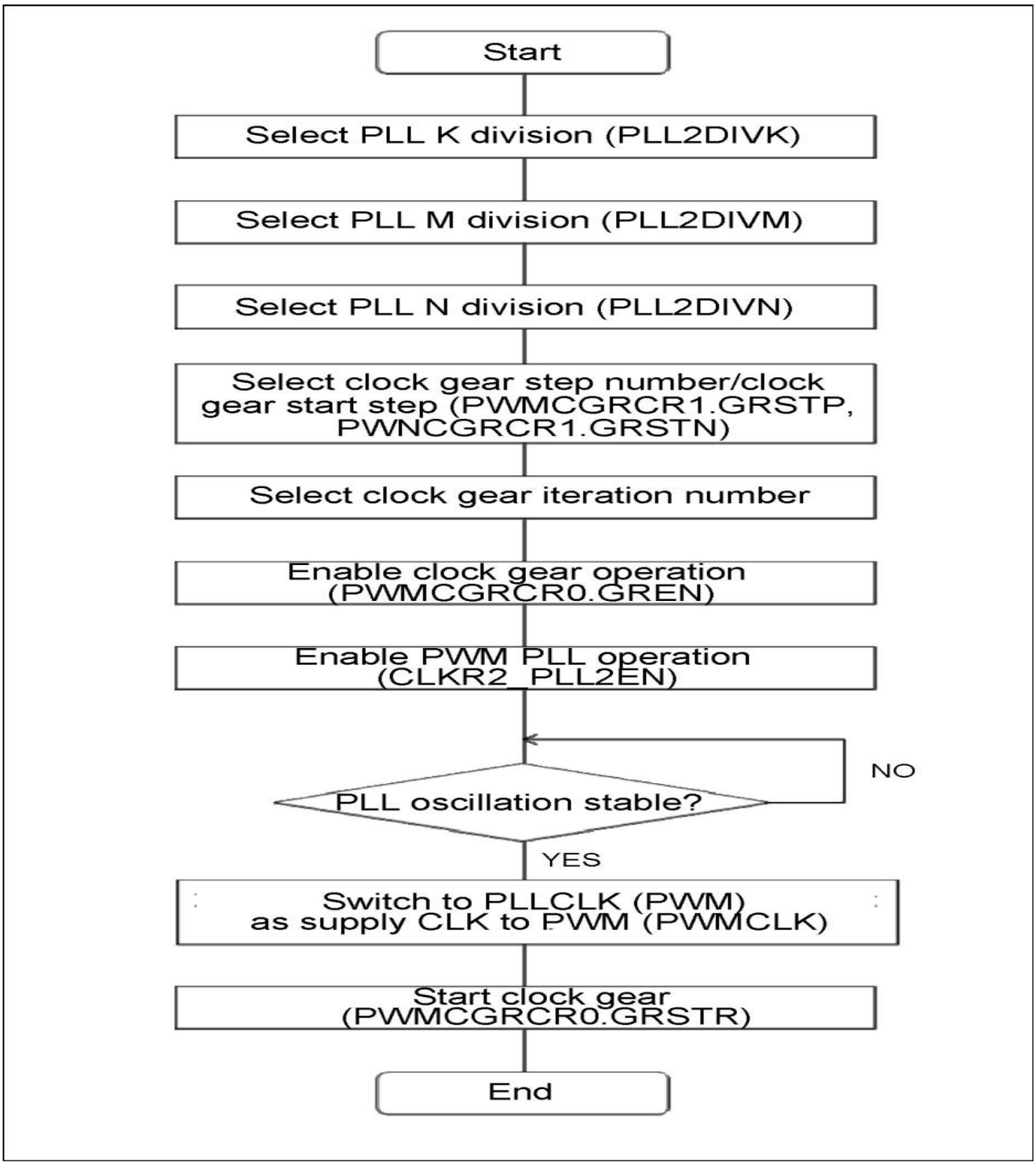
44.5 Clock Gear Circuit

For details on the clock gear circuit, see "Chapter 5: Clock."

44.6 Operation

This section explains the PWM dedicated clock settings.

- Clock setting procedure (example)



The PWM PLL stabilization wait operation uses the main timer and waits for the switching time.
The stabilization wait time is 200 us.

44.7 Frequency and Gear Calculation Examples

This section explains examples of frequency and gear calculations for the PWM dedicated clock.

- PWM clock frequency calculation example

Main Clock (MCLK) [MHz]	Frequency Parameter			PWM PLL Output (CLKVCO) [MHz]	PWM Clock (PWMCLK) [MHz]
	DIVK	DIVM	DIVN		
4	0	0001	110_0011	400	200
16	0	0001	001_1000	400	200

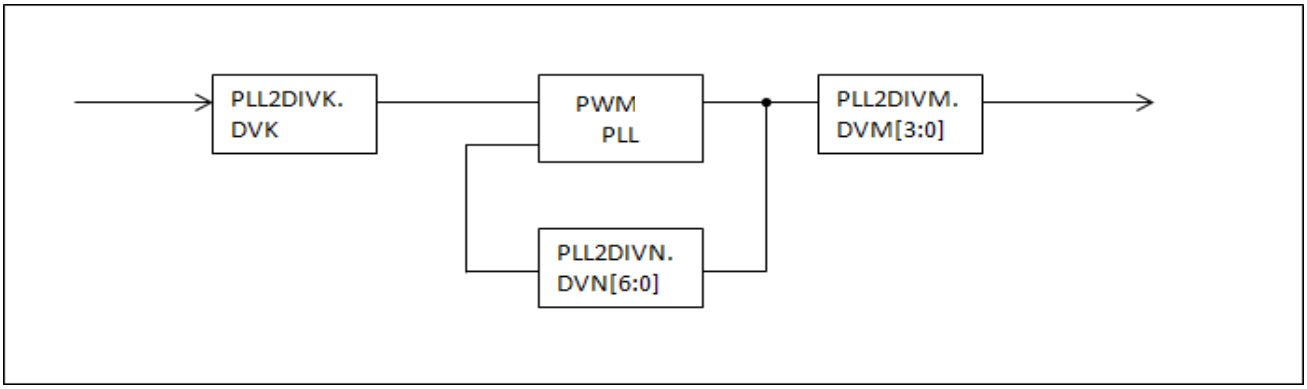
- PWM clock gear-up (down) calculation example

PWM Clock (PWMCLK) [MHz]	Clock Gear Parameter			Clock Gear Up (Down) Time (ms)
	PWMCGRCR1		PWMCGRCR2	
	GRSTP	GRSTN	GRLP	
200	00	00_0000	1111_1111	5.243

Note:

Set the PWM dedicated clock to 200 MHz.

- Frequency calculation
  - ☐ PWM PLL input clock frequency = (Main clock frequency) / (PLL2DIVK.DVK division ratio)
  - ☐ PWM PLL multiplication rate = (PLL2DIVN.DVN[6:0] multiplication rate)
  - ☐ PWM PLL macro oscillation clock frequency = (PWM PLL input clock frequency) x PWM PLL multiplication rate
  - ☐ PWM clock frequency = (PWM PLL macro oscillation clock frequency) / (PLL2DIVM.DVM[3:0] division ratio)



Note:

The PWM PLL macro oscillation clock frequency has an upper limit and a lower limit.

Set the PLL multiplication rate so as not to exceed the following range.

200 MHz ≤ PLL macro oscillation clock frequency ≤ 400 MHz

- Clock gear up (down) time calculation

  Clock gear up (down) time = (PWM clock frequency) x 64 x (64 - PWMCGRCR1.GRSTN[5:0]) x (PWMCGRCR2.GRLP[7:0]) / (PWMCGRCR1.GRSTP[1:0] + 1)

44.8 Notes

This section explains notes on the PWM dedicated clock.

PWM PLL control

After initialization, PWM PLL oscillation stops. While it is stopped, PWM PLL output cannot be selected as the clock source. First, after the program starts, the multiplier of PWM PLL used as the clock source is set. Then, after a wait until PWM PLL is locked, the clock source is changed. For the wait until PWM PLL is locked, use of the main timer interrupt is recommended.
Writing to this register has no effect. If you are going to stop PWM PLL for a purpose such as changing to stop mode, first select the on-chip bus clock (HCLK) as the clock source, and then stop PWM PLL.

PWM PLL multiplier

To change the PWM PLL multiplier to a value other than the initial value, do so after the program starts. Set this value before enabling PWM PLL or at the same time.
After changing the multiplier, wait for the PWM PLL lock time, and then switch the clock source.
For the wait until PWM PLL is locked, use of the main timer interrupt is recommended.
To change the PWM PLL multiplier setting during normal operations, first change the clock source to other than PWM PLL. Change the multiplier setting. After that, in the same way as described above, wait for the PWM PLL lock time and then change the clock source.

Power consumption control

To use a low-power consumption mode (stop mode, watch mode, sleep mode), switch the clock source to the on-chip bus clock (HCLK).

45. PWC with Buffers

This chapter explains the PWC with buffers.

45.1 Overview
45.2 Features
45.3 Configuration
45.4 Registers
45.5 Explanation of Operation
45.6 Notes

45.1 Overview

This section provides an overview of the PWC with buffers.

The PWC with buffers measures the pulse width and cycle of an external input signal. This model is equipped with 2 channels of the PWC with buffers.

45.2 Features

This section explains the features of the PWC with buffers.

Functions of the PWC with buffers

- The PWC with buffers consists of a control register, 2 16-bit timers, and 4 data buffers. A 16-bit timer counts an external input signal. The count value is stored in a data buffer.
- The following 4 types of data from the external input signal are measured. The data is stored in the data buffers by type.
  - ☐ "H" pulse width: Period during which an "H" level signal is input
  - ☐ "L" pulse width: Period during which an "L" level signal is input
  - ☐ Cycle between rising edges: Period from when a rising edge is detected until the next rising edge is detected
  - ☐ Cycle between falling edges: Period from when a falling edge is detected until the next falling edge is detected

Control of the PWC with buffers

- A digital LPF (low-pass filter) can reduce the noise from an external input signal.
- 7 types of digital LPFs are available (2, 4, 8, 16, 32, 64, or 128 cycles). 1 cycle of filtering is the cycle selected by the counter operation clock of a 16-bit timer.
- The external input level can be measured by inverting it.
- The PWC initialization register can execute software clear.

16-bit timer

- A 16-bit timer consists of a 16-bit counter, control register, 16-bit compare clear register (with a buffer register), and prescaler (1 for the 2 16-bit timers).
- 8 types of counter operation clocks are available (φ, φ/4, φ/16, φ/64, φ/256, φ/1024,φ /4096, and φ/16384) (φ: on-chip bus clock).
- One of the 2 timers resets the counter value to "0x0000" when the rising edge of an external input signal is detected. This timer measures the "H" pulse width and cycle between rising edges. The other timer resets the counter value to "0x0000" when the falling edge of an external input signal is detected. This timer measures the "L" pulse width and cycle between falling edges.
- A compare clear interrupt is generated if the compare clear register is compared with a 16-bit timer and they match. A 0 detection interrupt is generated while a 16-bit timer is detecting a count value of "0x0000".
- The compare clear register has a selectable buffer register (data written to this buffer register is transferred to the compare clear register). When a 16-bit timer stops and data is written to the buffer, transfer is performed immediately. When a timer value of "0" is detected during 16-bit timer operation, data is transferred from the buffer.
- If a reset, software clear, or compare match with the compare clear register occurs, the counter value is reset to "0x0000".

Data buffer

- A data buffer consists of a 16-bit FIFO buffer, control register, upper limit compare register, and lower limit compare register.
- The 16-bit FIFO buffer can store data in up to 4 levels (2 bytes are defined as 1 level).
- The level of the stored data can be confirmed.
- The upper limit compare register can set an upper limit value for measurement data. When the measurement data exceeds the upper limit value, an interrupt is generated.
- The lower limit compare register can set a lower limit value for the measurement data. When the measurement data falls below the lower limit value, an interrupt is generated.

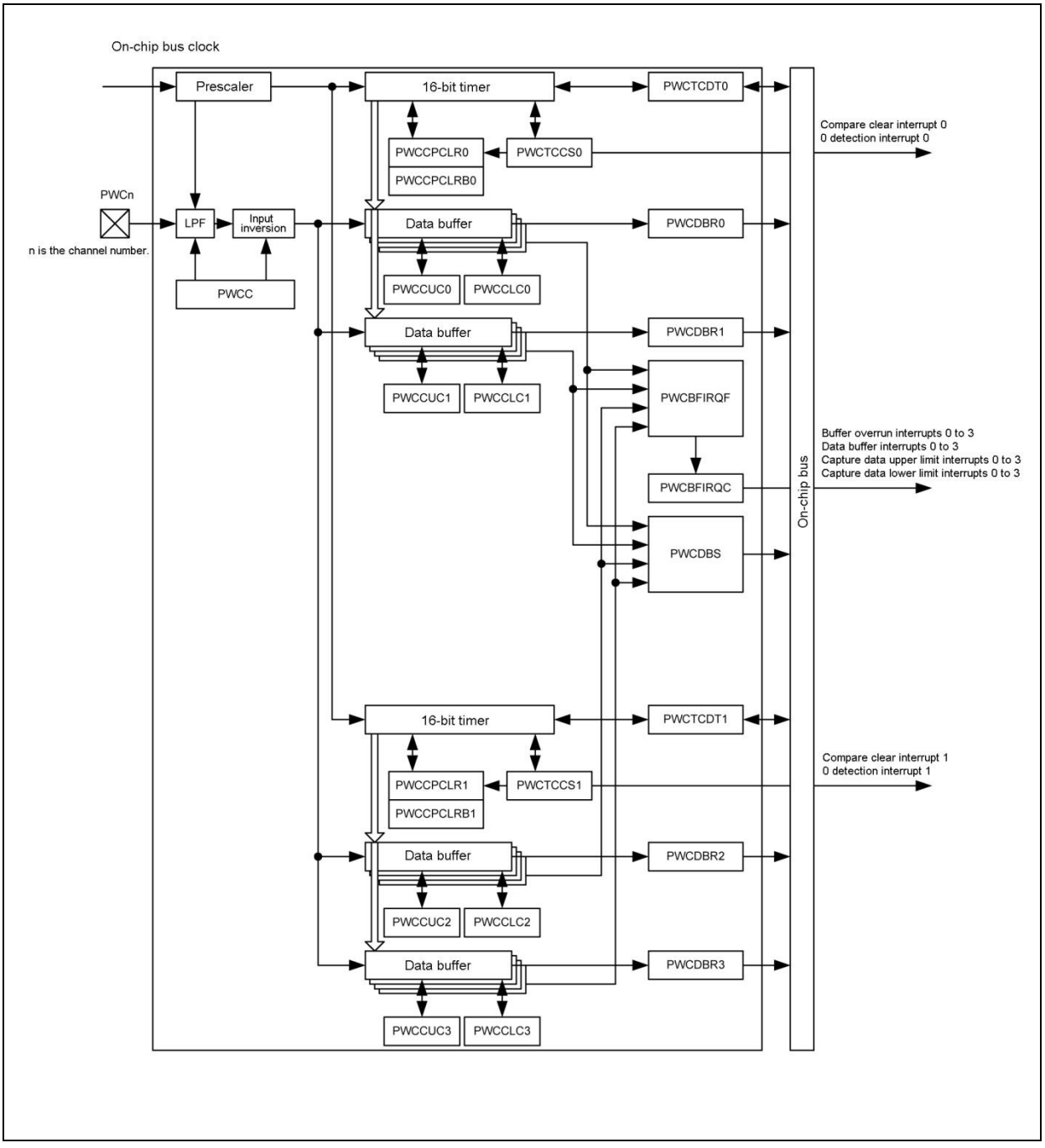
- When data is captured in the buffer, an interrupt is generated. You can select to what level data can be captured in the buffer before an interrupt is generated.
- When new data is captured in a buffer with data stored in the 4 levels (full capacity), the oldest data in the buffer is overwritten with the new data. When data is lost due to overwriting, a buffer overrun interrupt is generated.

45.3 Configuration

This section explains the configuration of the PWC with buffers.

Configuration of the PWC with buffers

Figure 45-1. Configuration of the PWC with Buffers (1 Channel)



45.4 Registers

This section explains the registers of the PWC with buffers.

Table 45-1. Register List

Address	+0	+1	+2	+3
0x3140	PWC initialization register 0 (PWCINIT0)	PWC control register 0 (PWCC0)	Reserved	
0x3144	Compare clear buffer register 00 (PWCCPCLRB00) Compare clear register 00 (PWCCPCLR00)		Timer data register 00 (PWCTCDT00)	
0x3148	Timer state control register 00 (PWCTCCS00)		Reserved	
0x314C	Compare clear buffer register 10 (PWCCPCLRB10) Compare clear register 10 (PWCCPCLR10)		Timer data register 10 (PWCTCDT10)	
0x3150	Timer state control register 10 (PWCTCCS10)		Reserved	
0x3154	Data buffer read register 00 (PWCDBR00)		Data buffer read register 10 (PWCDBR10)	
0x3158	Data buffer read register 20 (PWCDBR20)		Data buffer read register 30 (PWCDBR30)	
0x315C	Buffer state register 0 (PWCDBS0)		Reserved	
0x3160	Buffer interrupt flag register 0 (PWCBFIRQF0)		Reserved	
0x3164	Buffer interrupt control register 0 (PWCBFIRQC0)			Reserved
0x3168	Capture data upper limit compare register 00 (PWCCUC00)		Capture data lower limit compare register 00 (PWCCLC00)	
0x316C	Capture data upper limit compare register 10 (PWCCUC10)		Capture data lower limit compare register 10 (PWCCLC10)	
0x3170	Capture data upper limit compare register 20 (PWCCUC20)		Capture data lower limit compare register 20 (PWCCLC20)	
0x3174	Capture data upper limit compare register 30 (PWCCUC30)		Capture data lower limit compare register 30 (PWCCLC30)	
0x3178	PWC initialization register 1 (PWCINIT1)	PWC control register 1 (PWCC1)	Reserved	

Address	+0	+1	+2	+3
0x317C	Compare clear buffer register 01 (PWCCPCLRB01) Compare clear register 01 (PWCCPCLR01)		Timer data register 01 (PWCTCDT01)	
0x3180	Timer state control register 01 (PWCTCCS01)		Reserved	
0x3184	Compare clear buffer register 11 (PWCCPCLRB11) Compare clear register 11 (PWCCPCLR11)		Timer data register 11 (PWCTCDT11)	
0x3188	Timer state control register 11 (PWCTCCS11)		Reserved	
0x318C	Data buffer read register 01 (PWCDBR01)		Data buffer read register 11 (PWCDBR11)	
0x3190	Data buffer read register 21 (PWCDBR21)		Data buffer read register 31 (PWCDBR31)	
0x3194	Buffer state register 1 (PWCDBS1)		Reserved	
0x3198	Buffer interrupt flag register 1 (PWCBFIRQF1)		Reserved	
0x319C	Buffer interrupt control register 1 (PWCBFIRQC1)			Reserved
0x31A0	Capture data upper limit compare register 01 (PWCCUC01)		Capture data lower limit compare register 01 (PWCCLC01)	
0x31A4	Capture data upper limit compare register 11 (PWCCUC11)		Capture data lower limit compare register 11 (PWCCLC11)	
0x31A8	Capture data upper limit compare register 21 (PWCCUC21)		Capture data lower limit compare register 21 (PWCCLC21)	
0x31AC	Capture data upper limit compare register 31 (PWCCUC31)		Capture data lower limit compare register 31 (PWCCLC31)	

45.4.1 Registers of the PWC with Buffers

This section explains the registers of the PWC with buffers.

The PWC with buffers has the following registers.

- PWC initialization register (PWCINIT)
- PWC control register (PWCC)
- Compare clear buffer register (PWCCPCLRB)/Compare clear register (PWCCPCLR)
- Timer data register (PWCTCDT)
- Timer state control register (PWCTCCS)
- Data buffer read register (PWCDBR)
- Buffer state register (PWCDBS)
- Buffer interrupt flag register (PWCBFIRQF)
- Buffer interrupt control register (PWCBFIRQC)
- Capture data upper limit compare register (PWCCUC)
- Capture data lower limit compare register (PWCCLC)

45.4.1.1 PWC Initialization Register: PWCINIT0, PWCINIT1

This section shows the bit configuration of the PWC initialization registers.

The PWC initialization registers (PWCINIT) are used to initialize the PWC with buffers.

PWCINIT0: Address 3140H (access: byte, half word, and word)

PWCINIT1: Address 3178H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							PWCCLR
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W

[bit7 to bit1] Reserved

The read value is "0". Writing has no effect on operation.

[bit0] PWCCLR: PWC initialization bit

PWCCLR	Function	
	During Read Operation	During Write Operation
0	Always read "0".	Do not initialize the PWC with buffers.
1		Initialize the PWC with buffers.

- This bit is used to initialize the entire PWC with buffers.
- If this bit is set to "1":

  All of the PWC with buffers is initialized.
- If this bit is set to "0":

  The PWC with buffers is not initialized.
- The read value is always "0".

Note:

When initializing the PWC with buffers by accessing this register, do not simultaneously enable start of the PWC with buffers (PWCEN: bit0 in the PWC control register (PWCC) = 1).

45.4.1.2 PWC Control Register: PWCC0, PWCC1

This section shows the bit configuration of the PWC control registers.

The PWC control registers (PWCC) are used to control the operation of the PWC with buffers.

PWCC0: Address 3141H (access: byte, half word, and word)

PWCC1: Address 3179H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			LF2	LF1	LF0	EXINV	PWCEN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W

[bit7 to bit5] Reserved

The read value is "0". Writing has no effect on operation.

[bit4 to bit2] LF2 to LF0: Digital LPF selection bits

LF2	LF1	LF0	Function
0	0	0	No LPF function
0	0	1	Filter input of less than 2 cycles.
0	1	0	Filter input of less than 4 cycles.
0	1	1	Filter input of less than 8 cycles.
1	0	0	Filter input of less than 16 cycles.
1	0	1	Filter input of less than 32 cycles.
1	1	0	Filter input of less than 64 cycles.
1	1	1	Filter input of less than 128 cycles.

- These bits are used to select the function of the LPF that reduces the noise of external input.
- Cycles for which noise is reduced are based on the count clock of a 16-bit timer (set by CLK2 to CLK0: bit2 to 0 in the timer state control register (PWCTCCS)).

[bit1] EXINV: External input level invert bit

EXINV	Function
0	Do not inverse the external input level.
1	Inverse the external input level.

- This bit is used to invert the level of an external input signal.
- If this bit is set to "0":

  The level of the external input signal is not inverted.
- If this bit is set to "1":

  The level of the external input signal is inverted.

Note:

Set this bit while the PWC with buffers is stopped.
If you change the setting for inversion of the external input level while the PWC with buffers is operating, operation is not assured.

[bit0] PWCEN: PWC start enable bit

PWCEN	Function
0	Stop the PWC with buffers.
1	Enable start of the PWC with buffers.

- This bit is used to enable start of the PWC with buffers or stop it.
- If this bit is set to "0":

  The PWC with buffers is stopped.
- If this bit is set to "1":

  Start of the PWC with buffers is enabled. If the PWCEN bit is "1", the PWC with buffers starts operation when an external input edge is detected.

Note:

When enabling the start of the PWC with buffers by accessing this register, do not simultaneously initialize the PWC with buffers (PWCCLR: bit0 in the PWC initialization register (PWCINIT) = 1).

45.4.1.3 Compare Clear Buffer Register: PWCCPCLRB00 to PWCCPCLRB11/Compare Clear Register: PWCCPCLR00 to PWCCPCLR11

This section shows the bit configuration of the compare clear buffer registers/compare clear registers.

The compare clear buffer registers (PWCCPCLRB) are 16-bit buffer registers that exist in the compare clear registers (PWCCPCLR).
Both the PWCCPCLRB and PWCCPCLR registers exist at the same address.

PWCCPCLRB00: Address 3144H (access: half word and word)

PWCCPCLRB10: Address 314CH (access: half word and word)

PWCCPCLRB01: Address 317CH (access: half word and word)

PWCCPCLRB11: Address 3184H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial Value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial Value	1	1	1	1	1	1	1	1
Attribute	W	W	W	W	W	W	W	W

[bit15 to bit0] CL15 to CL00: Compare clear value buffer bits

CL15 to CL00	Function
	Compare clear value buffer

- A compare clear buffer register is a buffer register that exists at the same address as a compare clear register (PWCCPCLR).
- When the buffer function is disabled (BFE: bit7 in the timer state control register (PWCTCCS) = 0) or the timer stops, the value in the compare clear buffer register is immediately transferred to the compare clear register.
- When the buffer function is enabled, a value is transferred to the compare clear register when a count value of "0x0000" is detected from a 16-bit timer.

Note:

Do not set "0x0000" in the compare clear buffer registers.
To access these registers, use a half-word or word access instruction.
Do not access them by using a read-modify-write instruction.

PWCCPCLR00: Address 3144H (access: half word and word)

PWCCPCLR10: Address 314CH (access: half word and word)

PWCCPCLR01: Address 317CH (access: half word and word)

PWCCPCLR11: Address 3184H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08
Initial Value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00
Initial Value	1	1	1	1	1	1	1	1
Attribute	R	R	R	R	R	R	R	R

[bit15 to bit0] CL15 to CL00: Compare clear value bits

CL15 to CL00	Function
	Compare clear value

- A compare clear register is used for comparison with the count value of a 16-bit timer.
- When this register matches the count value of the 16-bit timer, the 16-bit timer is reset to "0x0000".

Note:

To access these registers, use a half-word or word access instruction.
Do not access them by using a read-modify-write instruction.

45.4.1.4 Timer Data Register: PWCTCDT00 to PWCTCDT11

This section shows the bit configuration of the timer data registers.

The timer data registers (PWCTCDT) are used to read the count values of the 16-bit timers.

PWCTCDT00: Address 3146H (access: half word and word)

PWCTCDT10: Address 314EH (access: half word and word)

PWCTCDT01: Address 317EH (access: half word and word)

PWCTCDT11: Address 3186H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	T15	T14	T13	T12	T11	T10	T09	T08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	T07	T06	T05	T04	T03	T02	T01	T00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

[bit15 to bit0] T15 to T00: Count value bits

T15 to T00	Function
	Count value

- The timer data registers are used to read the count values of the 16-bit timers.
- The count value is cleared to "0x0000" immediately when a reset is generated.
- The 16-bit timers are initialized immediately when one of the following factors is generated.
  - ☐ Reset
  - ☐ Detection of a valid edge of an external input
  - ☐ Clear bit (SCLR: bit4) = 1 in the timer state control register (PWCTCCS) during 16-bit timer operation
  - ☐ Match between the compare clear register and the timer count value
  - ☐ Initialization of the PWC with buffers (PWCCLR: bit0 = 1)

Table 45-2. Combinations of the Timer Data Registers and Types of Valid Edges of External Input

Registers	EXINV Bit Value	Valid Edge of External Input
PWCTCDT0n	0	Rising edge
	1	Falling edge
PWCTCDT1n	0	Falling edge
	1	Rising edge

n: Channel number

Note:

Even if the clear bit (SCLR: bit4) in the timer state control register (PWCTCCS) is 1 while a 16-bit timer is stopped, the 16-bit timer is not initialized.
To access the timer data registers, use a half-word or word access instruction.
Writing a count value can cause unintended counting.
If you change the setting of the external input level invert bit (EXINV: bit1 in the PWC control register (PWCC)) while the PWC with buffers is operating, operation is not assured.

45.4.1.5 Timer State Control Register: PWCTCCS00 to PWCTCCS11

This section shows the bit configuration of the timer state control registers.

The timer state control registers (PWCTCCS) are used to control 16-bit timer operation.

PWCTCCS00: Address 3148H (access: byte, half word, and word)

PWCTCCS10: Address 3150H (access: byte, half word, and word)

PWCTCCS01: Address 3180H (access: byte, half word, and word)

PWCTCCS11: Address 3188H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R(RM1),W	R/W	R,W	R,W	R,W	R(RM1),W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BFE	Reserved	Reserved	SCLR	Reserved	CLK2	CLK1	CLK0
Initial Value	0	1	0	0	0	0	0	0
Attribute	R/W	R1,W1	R0,W0	R0,W	R0,W0	R/W	R/W	R/W

Note:

Only timer state control register PWCTCCS0n has bit2 to bit0. Bit2 to bit0 in PWCTCCS1n are a reserved area. The read value is undefined. Be sure to write "0".
n: Channel number

[bit15] Reserved

The read value is "0". Be sure to write "0".

[bit14] IRQZF: 0 detection interrupt flag bit

IRQZF	Function	
	During Read Operation	During Write Operation
0	0 is not detected.	Clear this bit.
1	0 is detected.	Do not affect this bit.

- When the count value of the 16-bit timer is "0x0000", this bit is set to "1".
- If set to "0", this bit is cleared.
- If set to "1", this bit is not affected.
- This bit is cleared when the 0 detection interrupt clear signal is "H".

Note:

The read-modify-write (RMW) instruction always reads "1".
A software clear (writing "1" to SCLR: bit4 in the timer state control register (PWCTCCS)) during 16-bit timer operation does not set this bit.
A timer clear due to an edge detection does not set the 0 detection interrupt flag bit.
If a software clear (writing "0") or a clear due to the interrupt clear signal ("H") occurs simultaneously with a hardware set, the hardware set has priority.

[bit13] IRQZE: 0 detection interrupt request enable bit

IRQZE	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

- When "1" is set in this bit and the interrupt flag bit (IRQZF: bit14), an interrupt request to the CPU is generated.

[bit12 to bit10] MSI2 to MSI0: Interrupt mask selection bits

MSI2	MSI1	MSI0	Function
0	0	0	An interrupt is generated when the 1st match occurs.
0	0	1	An interrupt is generated when the 2nd match occurs.
0	1	0	An interrupt is generated when the 3rd match occurs.
0	1	1	An interrupt is generated when the 4th match occurs.
1	0	0	An interrupt is generated when the 5th match occurs.
1	0	1	An interrupt is generated when the 6th match occurs.
1	1	0	An interrupt is generated when the 7th match occurs.
1	1	1	An interrupt is generated when the 8th match occurs.

- These bits are used to set a mask count for a compare clear interrupt.
- If this bit is set to "0", no interrupt factor is masked.

Note:

The read value is the mask counter value.
When a read-modify-write instruction is performed, the read value is the mask register value.
A write operation writes data to the mask register.
The value written to the mask register while the timer is operating is reloaded to the counter only when the mask counter becomes "0".
The value written to the mask register while the timer is stopped is immediately reloaded to the counter.

[bit9] ICLR: Compare clear interrupt flag bit

ICLR	Function	
	During Read Operation	During Write Operation
0	There is no compare clear match.	Clear this bit.
1	There is a compare clear match.	Do not affect this bit.

- This bit is set to "1" when the compare clear value matches the 16-bit timer value.
- If set to "0", this bit is cleared.
- If set to "1", this bit is not affected.
- This bit is cleared when the compare clear match interrupt clear signal is "H".

Note:

The read-modify-write (RMW) instruction always reads "1".
This bit is set to "1" when the interrupt set by the interrupt mask selection bits is generated.
If the interrupt is not generated, this bit is not set to "1".

If a software clear (writing "0") or a clear due to the interrupt clear signal ("H") occurs simultaneously with a hardware set, the hardware set has priority.

[bit8] ICRE: Compare clear interrupt request enable bit

ICRE	Function
0	Disable interrupt requests.
1	Enable interrupt requests.

When "1" is set in this bit and the compare clear interrupt flag bit (ICLR: bit9), an interrupt request to the CPU is generated.

[bit7] BFE: Compare clear buffer enable bit

BFE	Function
0	Disable the compare clear buffer.
1	Enable the compare clear buffer.

- This bit is used to enable the compare clear buffer register (PWCCPCLRB).
- If this bit is set to "0":

  The compare clear buffer register (PWCCPCLRB) is disabled. Therefore, direct writing to the compare clear register (PWCCPCLR) is possible.
- If this bit is set to "1":

  The compare clear buffer register (PWCCPCLRB) is enabled. The data written and retained in the compare clear buffer register (PWCCPCLRB) is transferred to the compare clear register when a count value of "0" is detected from the 16-bit timer.

[bit6] (Reserved)

The read value is "1". Be sure to write "1".

[bit5] (Reserved)

The read value is "0". Be sure to write "0".

[bit4] SCLR: Timer clear bit

SCLR	Function	
	During Read Operation	During Write Operation
0	Always read "0".	Do not initialize the counter.
1		Initialize the counter to "0x0000".

- This bit is used to initialize the 16-bit timer to "0x0000".
- If this bit is set to "1" while the 16-bit timer is operating, the 16-bit timer is initialized to "0x0000" at the next count clock. If this bit is set to "1" while the 16-bit timer is stopped, the 16-bit timer is not initialized.
- The read value is always "0".

Note:

Writing "1" to this bit does not generate a 0 detection interrupt.
If "1" is set and then "0" is written before the next count clock, the timer is not cleared.

[bit3] Reserved

The read value is "0". Be sure to write "0".

[bit2 to bit0] CLK2 to CLK0: Clock frequency selection bits

CLK2	CLK1	CLK0	Function				
			Count Clock	φ=80 MHz	φ=40MHz	φ=20MHz	φ=2MHz
0	0	0	φ	12.5 ns	25 ns	50 ns	500 ns
0	0	1	φ/4	50 ns	100 ns	200 ns	2 µs
0	1	0	φ/16	200 ns	400 ns	800 ns	8 µs
0	1	1	φ/64	800 ns	1.6 µs	3.2 µs	32 µs
1	0	0	φ/256	3.2 µs	6.4 µs	12.8 µs	128 µs
1	0	1	φ/1024	12.8 µs	25.6 µs	51.2 µs	512 µs
1	1	0	φ/4096	51.2 µs	102.4 µs	204.8 µs	2048 µs
1	1	1	φ/16384	204.8 µs	409.6 µs	819.2 µs	8192 µs

- These bits are used to select the count clock frequency of the 16-bit timer.
- φ is the on-chip bus clock.

Note:

Only PWCTCCS00 and PWCTCCS01 have the clock frequency selection bits. Bit2 to bit0 in PWCTCCS10 and PWCTCCS11 are reserved bits. The read value is undefined. Be sure to write "0".
The 2 16-bit timers that read data from timer data registers 00 and 10 (PWCTCDT00 and PWCTCDT10) count using the count clock selected by PWCTCCS00.
The 2 16-bit timers that read data from timer data registers 01 and 11 (PWCTCDT01 and PWCTCDT11) count using the count clock selected by PWCTCCS01.
When setting the CLK2 to CLK0 bits, be sure to confirm that the PWC with buffers is stopped.

Table 45-3. Presence or Absence of the Clock Frequency Selection Bits

Registers	Clock Frequency Selection Bits
PWCTCCS00	Present
PWCTCCS10	None
PWCTCCS01	Present
PWCTCCS11	None

45.4.1.6 Data Buffer Read Register: PWCDBR00 to PWCDBR31

This section shows the bit configuration of the data buffer read registers.

The data buffer read registers (PWCDBR) are used to read data stored in data buffers.

PWCDBR00: Address 3154H (access: half word and word)

PWCDBR10: Address 3156H (access: half word and word)

PWCDBR20: Address 3158H (access: half word and word)

PWCDBR30: Address 315AH (access: half word and word)

PWCDBR01: Address 318CH (access: half word and word)

PWCDBR11: Address 318EH (access: half word and word)

PWCDBR21: Address 3190H (access: half word and word)

PWCDBR31: Address 3192H (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	BR15	BR14	BR13	BR12	BR11	BR10	BR09	BR08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BR07	BR06	BR05	BR04	BR03	BR02	BR01	BR00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15 to bit0] BR15 to BR00: Data buffer read value bits

BR15 to BR00	Function
	Value read from a data buffer

- The data buffer read registers are used to read the values from data buffers that are measured from external input.
- Once data is read from a data buffer, the same data cannot be read.
- If the buffer has no unread data, an invalid value is read.

Table 45-4. Combinations of Data Buffer Read Registers and Read Data

Registers	EXINV Bit Value	Read Value
PWCDBR0n	0	"H" pulse width
	1	"L" pulse width
PWCDBR1n	0	Cycle between rising edges
	1	Cycle between falling edges
PWCDBR2n	0	"L" pulse width
	1	"H" pulse width
PWCDBR3n	0	Cycle between falling edges
	1	Cycle between rising edges

n: Channel number

Note:

Each data buffer register reads data from a predetermined data buffer. The type of data to be stored is determined by setting the external input level invert bit (EXINV: bit1 in the PWC control register (PWCC)).
If you invert the external input level while the PWC with buffers is operating, operation is not assured.

45.4.1.7 Buffer State Register: PWCDBS0, PWCDBS1

This section shows the bit configuration of the buffer state registers.

The buffer state registers (PWCDBS) are used to read data stored in FIFO buffers.

PWCDBS0: Address 315CH (access: byte, half word, and word)

PWCDBS1: Address 3194H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	BFNE0	BFLV10	BFLV00	Reserved	BFNE1	BFLV11	BFLV01
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	BFNE2	BFLV12	BFLV02	Reserved	BFNE3	BFLV13	BFLV03
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R,WX	R,WX	R0,WX	R,WX	R,WX	R,WX

[bit15] Reserved

The read value is "0". Writing has no effect on operation.

[bit14] BFNE0: Buffer confirmation bit

BFNE0	Function
0	There is no data stored in the data buffer.
1	There is data stored in the data buffer.

- This bit is used to confirm that there is data stored in the data buffer.
- If this bit is "0":

  Data buffer read register 0n (PWCDBR0n) reads invalid data.
- If this bit is "1":

  Data buffer read register 0n (PWCDBR0n) reads valid data.
- n: Channel number

[bit13, bit12] BFLV10, BFLV00: Buffer level confirmation bits

BFLV10	BFLV00	Function
0	0	Data is stored in up to buffer level 1.
0	1	Data is stored in up to buffer level 2.
1	0	Data is stored in up to buffer level 3.
1	1	Data is stored in up to buffer level 4 (to full capacity).

- These bits are used to confirm the level of data stored in the buffer that is read from data buffer read register 0n (PWCDBR0n).
- The read value of these bits is valid when the buffer confirmation bit (BFNE0: bit14) is "1".
- n: Channel number

[bit11] Reserved

The read value is "0". Writing has no effect on operation.

[bit10] BFNE1: Buffer confirmation bit

BFNE1	Function
0	There is no data stored in the data buffer.
1	There is data stored in the data buffer.

- This bit is used to confirm that there is data stored in the data buffer.
- If this bit is "0":

  Data buffer read register 1n (PWCDBR1n) reads invalid data.
- If this bit is "1":

  Data buffer read register 1n (PWCDBR1n) reads valid data.
- n: Channel number

[bit9, bit8] BFLV11, BFLV01: Buffer level confirmation bits

BFLV11	BFLV01	Function
0	0	Data is stored in up to buffer level 1.
0	1	Data is stored in up to buffer level 2.
1	0	Data is stored in up to buffer level 3.
1	1	Data is stored in up to buffer level 4 (to full capacity).

- These bits are used to confirm the level of data stored in the buffer that is read from data buffer read register 1n (PWCDBR1n).
- The read value of these bits is valid when the buffer confirmation bit (BFNE1: bit10) is "1".
- n: Channel number

[bit7] Reserved

The read value is "0". Writing has no effect on operation.

[bit6] BFNE2: Buffer confirmation bit

BFNE2	Function
0	There is no data stored in the data buffer.
1	There is data stored in the data buffer.

- This bit is used to confirm that there is data stored in the data buffer.
- If this bit is "0":

  Data buffer read register 2n (PWCDBR2n) reads invalid data.
- If this bit is "1":

  Data buffer read register 2n (PWCDBR2n) reads valid data.
- n: Channel number

[bit5, bit4] BFLV12, BFLV02: Buffer level confirmation bits

BFLV12	BFLV02	Function
0	0	Data is stored in up to buffer level 1.
0	1	Data is stored in up to buffer level 2.
1	0	Data is stored in up to buffer level 3.
1	1	Data is stored in up to buffer level 4 (to full capacity).

- These bits are used to confirm the level of data stored in the buffer that is read from data buffer read register 2n (PWCDBR2n).
- The read value of these bits is valid when the buffer confirmation bit (BFNE2: bit6) is "1".
- n: Channel number

[bit3] Reserved

The read value is "0". Writing has no effect on operation.

[bit2] BFNE3: Buffer confirmation bit

BFNE3	Function
0	There is no data stored in the data buffer.
1	There is data stored in the data buffer.

- This bit is used to confirm that there is data stored in the data buffer.
- If this bit is "0":

  Data buffer read register 3n (PWCDBR3n) reads invalid data.
- If this bit is "1":

  Data buffer read register 3n (PWCDBR3n) reads valid data.
- n: Channel number

[bit1, bit0] BFLV13, BFLV03: Buffer level confirmation bits

BFLV13	BFLV03	Function
0	0	Data is stored in up to buffer level 1.
0	1	Data is stored in up to buffer level 2.
1	0	Data is stored in up to buffer level 3.
1	1	Data is stored in up to buffer level 4 (to full capacity).

- These bits are used to confirm the level of data stored in the buffer that is read from data buffer read register 3n (PWCDBR3n).
- The read value of these bits is valid when the buffer confirmation bit (BFNE3: bit2) is "1".
- n: Channel number

Table 45-5. Combinations of Bits in the Buffer State Registers and Data Types

Buffer State Registers	EXINV Bit Value	Data Buffer Read Register	Data Type
BFNE0, BFLV10, BFLV00	0	PWCDBR0n	"H" pulse width
	1		"L" pulse width
BFNE1, BFLV11, BFLV01	0	PWCDBR1n	Cycle between rising edges
	1		Cycle between falling edges
BFNE2, BFLV12, BFLV02	0	PWCDBR2n	"L" pulse width
	1		"H" pulse width
BFNE3, BFLV13, BFLV03	0	PWCDBR3n	Cycle between falling edges
	1		Cycle between rising edges

n: Channel number

Note:

If you change the setting of the external input level invert bit (EXINV:bit1 in the PWC control register (PWCC)) while the PWC with buffers is operating, the data types that can be confirmed through the buffer state register is mixed. If you invert the external input level during operation, the operation is not assured.

45.4.1.8 Buffer Interrupt Flag Register: PWCBFIRQF0, PWCBFIRQF1

This section shows the bit configuration of the buffer interrupt flag registers.

The buffer interrupt flag registers (PWCBFIRQF) are used to confirm interrupt flags from data buffers.

PWCBFIRQF0: Address 3160H (access: byte, half word, and word)

PWCBFIRQF1: Address 3198H (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ICWF0	ICWF1	ICWF2	ICWF3	ICBF0	ICBF1	ICBF2	ICBF3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUF0	ICUF1	ICUF2	ICUF3	ICLF0	ICLF1	ICLF2	ICLF3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W	R(RM1),W

[bit15 to bit12] ICWF0 to ICWF3: Buffer overrun interrupt flag bits

ICWFk	Function	
	During Read Operation	During Write Operation
0	There is no buffer overrun.	Clear this bit.
1	There is a buffer overrun.	Do not affect this bit.

- This bit is set to "1" when old data is overwritten with new stored data while the buffer that is read from data buffer read register kn (PWCDBRkn) is full.
- If set to "0", this bit is cleared.
- If set to "1", this bit is not affected.
- This bit is cleared when the buffer overrun interrupt clear signal is "H".
- k: Buffer number
- n: Channel number

Note:

The read-modify-write (RMW) instruction always reads "1".
If a buffer overrun occurs, data lost through overwriting cannot be restored.
If a software clear (writing "0") or a clear due to the interrupt clear signal ("H") occurs simultaneously with a hardware set, the hardware set has priority.

[bit11 to bit8] ICBF0 to ICBF3: Data buffer interrupt flag bits

ICBFk	Function	
	During Read Operation	During Write Operation
0	Data has not been stored in the buffer to the set level.	Clear this bit.
1	Data has been stored in the buffer to the set level.	Do not affect this bit.

- This bit is set to "1" when data is stored in the buffer to the level set by IBSL10 to 03: bit15 to 8 in the buffer interrupt control register (PWCBFIRQC). The buffer is one that is read from data buffer read register kn (PWCDBRkn).
- If set to "0", this bit is cleared.
- If set to "1", this bit is not affected.
- This bit is cleared when the data buffer interrupt clear signal is "H".
- k: Buffer number
- n: Channel number

Note:

Clear the data buffer interrupt flag bits (ICBF) while the number of levels of data fetched into the buffer is smaller than the number of levels set by the data buffer interrupt level selection bits (ISBL).
The read-modify-write (RMW) instruction always reads "1".
If a software clear (writing "0") occurs simultaneously with a hardware set, the hardware set has priority. If a clear due to the interrupt clear signal ("H") occurs simultaneously with a hardware set, the clear has priority.

[bit7 to bit4] ICUF0 to ICUF3: Capture data upper limit interrupt flag bits

ICUFk	Function	
	During Read Operation	During Write Operation
0	Captured data has not exceeded the set upper limit value.	Clear this bit.
1	Captured data has exceeded the set upper limit value.	Do not affect this bit.

- This bit is set to "1" when data captured in the buffer that is read from data buffer read register kn (PWCDBRkn) exceeds the set value of the capture data upper limit compare register (PWCCUCkn).
- If set to "0", this bit is cleared.
- If set to "1", this bit is not affected.
- This bit is cleared when the capture data upper limit interrupt clear signal is "H".
- k: Buffer number
- n: Channel number

Note:

The read-modify-write (RMW) instruction always reads "1".
If a software clear (writing "0") or a clear due to the interrupt clear signal ("H") occurs simultaneously with a hardware set, the hardware set has priority.

[bit3 to bit0] ICLF0 to ICLF3: Capture data lower limit interrupt flag bits

ICLFk	Function	
	During Read Operation	During Write Operation
0	Captured data has not fallen below the set lower limit value.	Clear this bit.
1	Captured data has fallen below the set lower limit value.	Do not affect this bit.

- This bit is set to "1" when data captured in the buffer that is read from data buffer read register kn (PWCDBRkn) falls below the set value of the capture data lower limit compare register (PWCCLCkn).
- If set to "0", this bit is cleared.
- If set to "1", this bit is not affected.

- This bit is cleared when the capture data lower limit interrupt clear signal is "H".
- k: Buffer number
- n: Channel number

Note:

The read-modify-write (RMW) instruction always reads "1".
If a software clear (writing "0") or a clear due to the interrupt clear signal ("H") occurs simultaneously with a hardware set, the hardware set has priority.

Table 45-6. Combinations of Bits in the Buffer Interrupt Flag Registers and Data Types

Interrupt Flag Bit	EXINV Bit Value	Data Buffer Read Register	Buffer for Which Flag Bit Is Set
ICWF0, ICBF0, ICUF0, ICLF0	0	PWCDBR0n	"H" pulse width
	1		"L" pulse width
ICWF1, ICBF1, ICUF1, ICLF1	0	PWCDBR1n	Cycle between rising edges
	1		Cycle between falling edges
ICWF2, ICBF2, ICUF2, ICLF2	0	PWCDBR2n	"L" pulse width
	1		"H" pulse width
ICWF3, ICBF3, ICUF3, ICLF3	0	PWCDBR3n	Cycle between falling edges
	1		Cycle between rising edges

n: Channel number

Note:

If you change the setting of the external input level invert bit (EXINV: bit1 in the PWC control register (PWCC)) while the PWC with buffers is operating, operation is not assured.

45.4.1.9 Buffer Interrupt Control Register: PWCBFIRQC0, PWCBFIRQC1

This section shows the bit configuration of the buffer interrupt control registers.

The buffer interrupt control registers (PWCBFIRQC) are used to control enabling/disabling of interrupts from data buffers and control the frequency of data buffer interrupts.

PWCBFIRQC0: Address 3164H (access: byte, half word, and word)

PWCBFIRQC1: Address 319CH (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	ICWE0	ICWE1	ICWE2	ICWE3	ICBE0	ICBE1	ICBE2	ICBE3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	ICUE0	ICUE1	ICUE2	ICUE3	ICLE0	ICLE1	ICLE2	ICLE3
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IBSL10	IBSL00	IBSL11	IBSL01	IBSL12	IBSL02	IBSL13	IBSL03
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

[bit31 to bit28] ICWE0 to ICWE3: Buffer overrun interrupt enable bits

ICWEk	Function
0	Disable buffer overrun interrupts.
1	Enable buffer overrun interrupts.

- This bit is used to enable buffer overrun interrupts for the buffer that is read from data buffer read register kn (PWCDBRkn). The interrupts are enabled when the relevant buffer overrun interrupt flag bit is set (ICWF0 to 3: bit15 to 12 in the buffer interrupt flag register (PWCBFIRQFn) = 1).
- If this bit is set to "0":

  No buffer overrun interrupt is generated when the relevant buffer overrun interrupt flag bit is set.
- If this bit is set to "1":

  A buffer overrun interrupt is generated when the relevant buffer overrun interrupt flag bit is set.
- k: Buffer number
- n: Channel number

[bit27 to bit24] ICBE0 to ICBE3: Data buffer interrupt enable bits

ICBEk	Function
0	Disable data buffer interrupts.
1	Enable data buffer interrupts.

- This bit is used to enable data buffer interrupts for the buffer that is read from data buffer read register kn (PWCDBRkn). The interrupts are enabled when the relevant data buffer interrupt flag bit is set (ICBF0 to 3: bit11 to 8 in the buffer interrupt flag register (PWCBFIRQFn) = 1).
- If this bit is set to "0":

  No data buffer interrupt is generated when the relevant data buffer interrupt flag bit is set.
- If this bit is set to "1":

  A data buffer interrupt is generated when the relevant data buffer interrupt flag bit is set.
- k: Buffer number
- n: Channel number

[bit23 to bit20] ICUE0 to ICUE3: Capture data upper limit interrupt enable bits

ICUEk	Function
0	Disable capture data upper limit interrupts.
1	Enable capture data upper limit interrupts.

- This bit is used to enable capture data upper limit interrupts for the buffer that is read from data buffer read register kn (PWCDBRkn). The interrupts are enabled when the relevant capture data upper limit interrupt flag bit is set (ICUF0 to 3: bit7 to 4 in the buffer interrupt flag register (PWCBFIRQFn) = 1).
- If this bit is set to "0":

  No capture data upper limit interrupt is generated when the relevant capture data upper limit interrupt flag bit is set.
- If this bit is set to "0":

  A capture data upper limit interrupt is generated when the relevant capture data upper limit interrupt flag bit is set.
- k: Buffer number
- n: Channel number

[bit19 to bit16] ICLE0 to ICLE3: Capture data lower limit interrupt enable bits

ICLEk	Function
0	Disable capture data lower limit interrupts.
1	Enable capture data lower limit interrupts.

- This bit is used to enable capture data lower limit interrupts for the buffer that is read from data buffer read register kn (PWCDBRkn). The interrupts are enabled when the relevant capture data lower limit interrupt flag bit is set (ICLF0 to 3: bit3 to 0 in the buffer interrupt flag register (PWCBFIRQFn) = 1).
- If this bit is set to "0":

  No capture data lower limit interrupt is generated when the relevant capture data lower limit interrupt flag bit is set.
- If this bit is set to "0":

  A capture data lower limit interrupt is generated when the relevant capture data lower limit interrupt flag bit is set.
- k: Buffer number

- n: Channel number

[bit15 to bit8] IBSL10 to IBSL03: Data buffer interrupt level selection bits

IBSL1k	IBSL0k	Function
0	0	Do not set an interrupt flag even when data is stored.
0	1	Set an interrupt flag when data is stored in 1 level.
1	0	Set an interrupt flag when data is stored in 2 levels.
1	1	Set an interrupt flag when data is stored in 4 levels (to full capacity).

- These bits are for the buffer that is read from data buffer read register kn (PWCDBRkn). They are used to select a condition for setting the relevant data buffer interrupt flag bit (ICBF0 to 3: bit11 to 8 in the buffer interrupt flag register (PWCBFIRQFn)).
- k: Buffer number
- n: Channel number

[bit7 to bit0] Reserved

The read value is "1". Writing has no effect on operation.

Table 45-7. Combinations of Bits in the Buffer Interrupt Control Registers and Data Types

Interrupt Control Bit	EXINV Bit Value	Data Buffer Read Register	Buffer for Which Interrupt Control Is Set
ICWE0, ICBE0, ICUE0, ICLE0, IBSL10, IBSL00	0	PWCDBR0n	"H" pulse width
	1		"L" pulse width
ICWE1, ICBE1, ICUE1, ICLE1, IBSL11, IBSL01	0	PWCDBR1n	Cycle between rising edges
	1		Cycle between falling edges
ICWE2, ICBE2, ICUE2, ICLE2, IBSL12, IBSL02	0	PWCDBR2n	"L" pulse width
	1		"H" pulse width
ICWE3, ICBE3, ICUE3, ICLE3, IBSL13, IBSL03	0	PWCDBR3n	Cycle between falling edges
	1		Cycle between rising edges

n: Channel number

Note:

If you change the setting of the external input level invert bit (EXINV: bit1 in the PWC control register (PWCC)) while the PWC with buffers is operating, operation is not assured.

45.4.1.10 Capture Data Upper Limit Compare Register: PWCCUC00 to PWCCUC31

This section shows the bit configuration of the capture data upper limit compare registers.

The capture data upper limit compare registers (PWCCUC) are used to set upper limit values for comparison with data stored in data buffers.

PWCCUC00: Address 3168H (access: half word and word)

PWCCUC10: Address 316CH (access: half word and word)

PWCCUC20: Address 3170H (access: half word and word)

PWCCUC30: Address 3174H (access: half word and word)

PWCCUC01: Address 31A0H (access: half word and word)

PWCCUC11: Address 31A4H (access: half word and word)

PWCCUC21: Address 31A8H (access: half word and word)

PWCCUC31: Address 31ACH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	UL15	UL14	UL13	UL12	UL11	UL10	UL09	UL08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	UL07	UL06	UL05	UL04	UL03	UL02	UL01	UL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] UL15 to UL00: Capture data upper limit comparison value bits

UL15 to UL00	Function
	Capture data upper limit comparison value

- The capture data upper limit compare registers are used to set upper limits for data stored in the buffers that are read from data buffer read registers 0n to 3n (PWCDBR0n to 3n).
- The capture data upper limit comparison value is compared with the value of the latest data stored in the relevant data buffer. When the latest data exceeds the upper limit comparison value, the capture data upper limit interrupt flag is set.
- The combination of the set value of each register and the type of data to be compared is predetermined. When the external input level is inverted (EXINV: bit1 in the PWC control register (PWCC) is "1"), the combination for upper limit comparison varies.

Table 45.4-1 Combinations of Capture Data Upper Limit Compare Registers and Data to Be Compared

Capture Data Upper Limit Compare Register	EXINV Bit Value	Data Buffer Read Register	Data to Be Compared
PWCCUC0n	0	PWCDBR0n	"H" pulse width
	1		"L" pulse width
PWCCUC1n	0	PWCDBR1n	Cycle between rising edges
	1		Cycle between falling edges
PWCCUC2n	0	PWCDBR2n	"L" pulse width
	1		"H" pulse width
PWCCUC3n	0	PWCDBR3n	Cycle between falling edges
	1		Cycle between rising edges

n: Channel number

Note:

If you change the setting of the external input level invert bit (EXINV: bit1 in the PWC control register (PWCC)) while the PWC with buffers is operating, operation is not assured.
If the measurement value and the upper limit comparison value are the same, the capture data upper limit interrupt flag is not set.
Do not set an upper limit comparison value that is smaller than the lower limit comparison value.

45.4.1.11 Capture Data Lower Limit Compare Register: PWCCLC00 to PWCCLC31

This section shows the bit configuration of the capture data lower limit compare registers.

The capture data lower limit compare registers (PWCCLC) are used to set lower limit values for comparison with data stored in data buffers.

PWCCLC00: Address 316AH (access: half word and word)

PWCCLC10: Address 316EH (access: half word and word)

PWCCLC20: Address 3172H (access: half word and word)

PWCCLC30: Address 3176H (access: half word and word)

PWCCLC01: Address 31A2H (access: half word and word)

PWCCLC11: Address 31A6H (access: half word and word)

PWCCLC21: Address 31AAH (access: half word and word)

PWCCLC31: Address 31AEH (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	LL15	LL14	LL13	LL12	LL11	LL10	LL09	LL08
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LL07	LL06	LL05	LL04	LL03	LL02	LL01	LL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] LL15 to LL00: Capture data lower limit comparison value bits

LL15 to LL00	Function
	Capture data lower limit comparison value

- The capture data lower limit compare registers are used to set lower limits for data stored in the buffers that are read from data buffer read registers 0n to 3n (PWCDBR0n to 3n).
- The capture data lower limit comparison value is compared with the value of the latest data stored in the relevant data buffer. When the latest data exceeds the lower limit comparison value, the capture data lower limit interrupt flag is set.
- The combination of the set value of each register and the type of data to be compared is predetermined. When the external input level is inverted (EXINV: bit1 in the PWC control register (PWCC) is "1"), the combination for lower limit comparison varies.

Table 45-8. Combinations of Capture Data Lower Limit Compare Registers and Data to Be Compared

Capture Data Upper Limit Compare Register	EXINV Bit Value	Data Buffer Read Register	Data to Be Compared
PWCCLC0n	0	PWCDBR0n	"H" pulse width
	1		"L" pulse width
PWCCLC1n	0	PWCDBR1n	Cycle between rising edges
	1		Cycle between falling edges
PWCCLC2n	0	PWCDBR2n	"L" pulse width
	1		"H" pulse width
PWCCLC3n	0	PWCDBR3n	Cycle between falling edges
	1		Cycle between rising edges

n: Channel number

Note:

If you change the setting of the external input level invert bit (EXINV: bit1 in the PWC control register (PWCC)) while the PWC with buffers is operating, operation is not assured.
If the measurement value and the lower limit comparison value are the same, the capture data lower limit interrupt flag is not set. Do not set a lower limit comparison value that is greater than the upper limit comparison value.

45.5 Explanation of Operation

This section explains the operation of the PWC with buffers.

45.5.1 Interrupts of the PWC with Buffers

45.5.2 Operation of the PWC with Buffers

45.5.1 Interrupts of the PWC with Buffers

This section explains the interrupts of the PWC with buffers.

Table 45-9, Table 45-10 and Table 45-11 show the interrupt control bits and interrupt factors of the PWC with buffers.

Table 45-9. Interrupt Control Bits and Interrupt Factors for 16-bit Timers

	16-bit Timer	
	Compare Clear	**0 Detection**
Interrupt Flag Bit	ICLR: bit9 in timer state control register (PWCTCCS)	IRQZF: bit14 in timer state control register (PWCTCCS)
Interrupt Enable Bit	ICRE: bit8 in timer state control register (PWCTCCS)	IRQZE: bit13 in timer state control register (PWCTCCS)
Interrupt Factor	The 16-bit timer value matches the compare clear register (PWCCPCLR).	The 16-bit timer value becomes "0x0000".

When the value of the 16-bit timer matches the compare clear register (PWCCPCLR), ICLR: bit9 in the timer state control register (PWCTCCS) is set to "1". When interrupt requests are enabled (ICRE: bit8 in PWCTCCS = 1) in the state above, an interrupt request is output to the interrupt controller.

When the timer value becomes "0x0000", IRQZF: bit14 in the timer state control register (PWCTCCS) is set to "1". When interrupt requests are enabled (IRQZE: bit13 in PWCTCCS = 1) in the state above, an interrupt request is output to the interrupt controller. At this time, a clear due to detection of a valid edge of external input is not an interrupt factor.

Table 45-10. Interrupt Control Bits and Interrupt Factors for Data Buffers

	Data Buffer	
	Buffer Overrun	**Data Buffer**
Interrupt Flag Bit	ICWF0 to 3: bit15 to 12 in buffer interrupt flag register (PWCBFIRQFn)	ICBF0 to 3: bit11 to 8 in buffer interrupt flag register (PWCBFIRQFn)
Interrupt Enable Bit	ICWE0 to 3: bit31 to 28 in buffer interrupt control register (PWCBFIRQCn)	ICBE0 to 3: bit27 to 24 in buffer interrupt control register (PWCBFIRQCn)
Interrupt Factor	Data is lost due to overwriting in the buffer.	Data is captured.

n: Channel number

When the buffer that is read from data buffer read register 0n (PWCDBR0n) is full, data may be lost due to overwriting. In this case, the relevant bit of ICWF0 to 3: bit15 to 12 in the buffer interrupt flag register (PWCBFIRQFn) is set to "1". When interrupt requests are enabled (ICWE0 to 3: bit31 to 28 in PWCBFIRQCn = 1) in the state above, an interrupt request is output to the interrupt controller.

When data is captured in the buffer that is read from data buffer read register 0n (PWCDBR0n), the relevant bit of ICBF0 to 3: bit11 to 8 in the buffer interrupt flag register (PWCBFIRQFn) is set to "1". When interrupt requests are enabled (ICBE0 to 3: bit27 to 24 in PWCBFIRQCn = 1) in the state above, an interrupt request is output to the interrupt controller.

Table 45-11. Interrupt Control Bits and Interrupt Factors for Data Comparison

	Data Comparison	
	Capture Data Upper Limit	**Capture Data Lower Limit**
Interrupt Flag Bit	ICUF0 to 3: bit7 to 4 in buffer interrupt flag register (PWCBFIRQFn)	ICLF0 to 3: bit3 to 0 in buffer interrupt flag register (PWCBFIRQFn)
Interrupt Enable Bit	ICUE0 to 3: bit23 to 20 in buffer interrupt control register (PWCBFIRQCn)	ICLE0 to 3: bit19 to 16 in buffer interrupt control register (PWCBFIRQCn)
Interrupt Factor	Capture data exceeds the upper limit.	Capture data falls below the lower limit.

n: Channel number

The latest data stored in the buffer that is read from data buffer read register 0n (PWCDBR0n) may exceed the set value of the capture data upper limit compare register (PWCCUC0n to 3n). In this case, the relevant bit of ICUF0 to 3: bit7 to 4 in the buffer interrupt flag register (PWCBFIRQFn) is set to "1". When interrupt requests are enabled (ICUE0 to 3: bit23 to 20 in PWCBFIRQCn = 1) in the state above, an interrupt request is output to the interrupt controller.

The latest data stored in the buffer that is read from data buffer read register 0n (PWCDBR0n) may fall below the set value of the capture data lower limit compare register (PWCCLC0n to 3n). In this case, the relevant bit of ICLF0 to 3: bit3 to 0 in the buffer interrupt flag register (PWCBFIRQFn) is set to "1". When interrupt requests are enabled (ICLE0 to 3: bit19 to 16 in PWCBFIRQCn = 1) in the state above, an interrupt request is output to the interrupt controller.

45.5.2 Operation of the PWC with Buffers

This section explains the operation of the PWC with buffers.

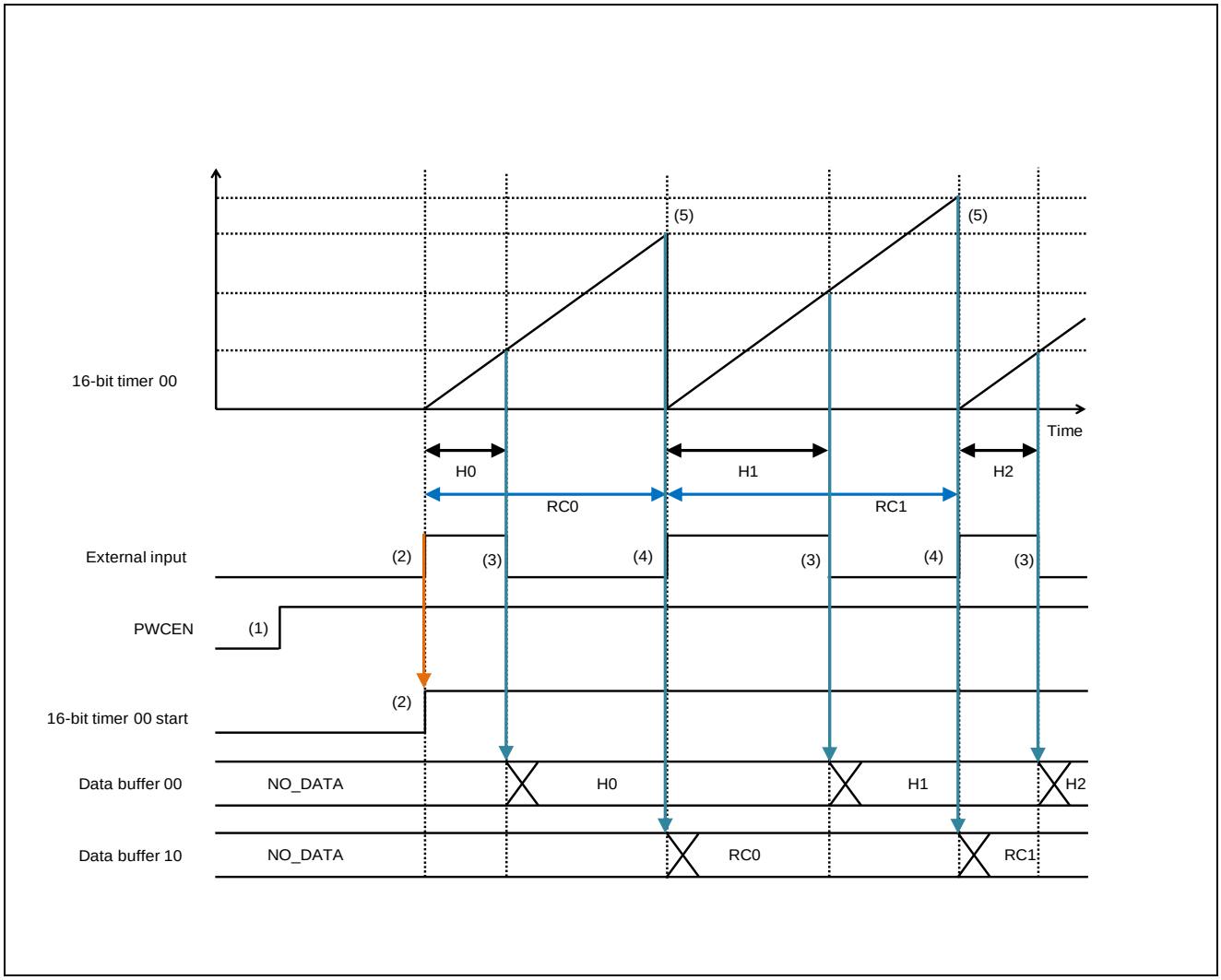
The PWC with buffers starts operation when a valid edge is detected while start is enabled. A 16-bit timer counts up until the count value matches the value of the compare clear register (PWCCPCLR). Then, the counter is cleared to "0x0000" and counts up again. When a valid edge is detected during 16-bit timer operation, the count value of the 16-bit timer is stored in a data buffer and the count value is cleared to "0x0000".

45.5.2.1 Start of the PWC with Buffers

This section explains the start of the PWC with buffers.

Figure 45-2 and Figure 45-3 show the operation when the external input level starts with "L".

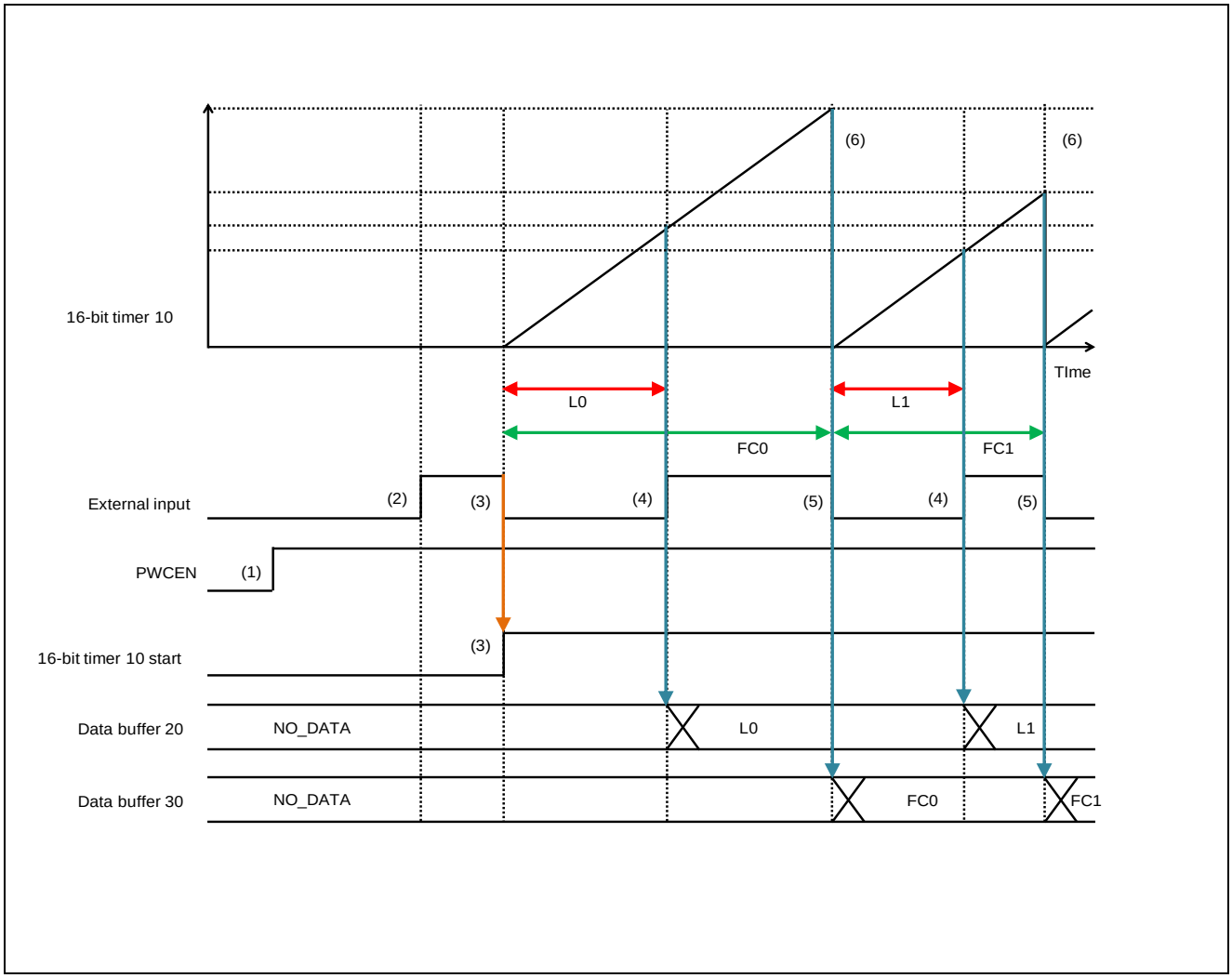
Figure 45-2. Operation of 16-bit Timer 00



Setting and operation procedure

1. Operation is enabled (PWCEN: bit0 in the PWC control register (PWCC) =1)
2. When PWCEN: bit0 is 1 and the first rising edge is detected, 16-bit timer 00 starts.
3. After the start of 16-bit timer 00, when a falling edge is detected, the count value of 16-bit timer 00 is written to data buffer 00 as an "H" pulse width.
4. After the start of 16-bit timer 00, when a rising edge is detected, the count value of 16-bit timer 00 is written to data buffer 10 as a cycle between rising edges.
5. When a rising edge is detected, the count value of 16-bit timer 00 is cleared to "0x0000".
6. (3) to (5) are repeated.

Figure 45-3. Operation of 16-bit Timer 10

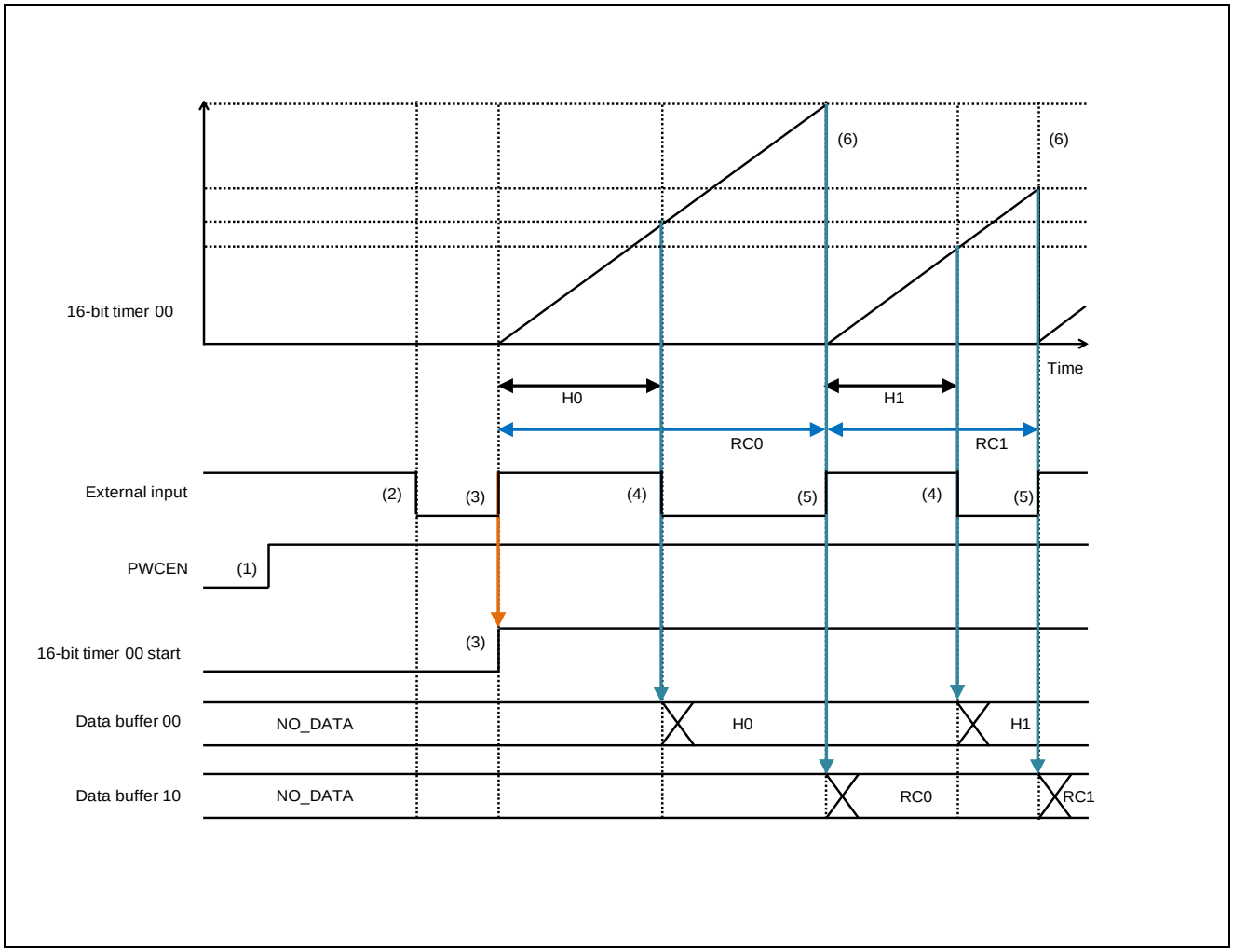


Setting and operation procedure

1. Operation is enabled (PWCEN: bit0 in the PWC control register (PWCC) =1)
2. 16-bit timer 10 does not start at the time of rising detection.
3. When PWCEN: bit0 is 1 and the first falling edge is detected, 16-bit timer 10 starts.
4. After the start of 16-bit timer 10, when a rising edge is detected, the count value of 16-bit timer 10 is written to data buffer 20 as an "L" pulse width.
5. After the start of 16-bit timer 10, when a falling edge is detected, the count value of 16-bit timer 10 is written to data buffer 30 as a cycle between falling edges.
6. When a falling edge is detected, the count value of 16-bit timer 10 is cleared to "0x0000".
7. (4) to (6) are repeated.

Figure 45-4 and Figure 45-5 show the operation when the external input level starts with "H".

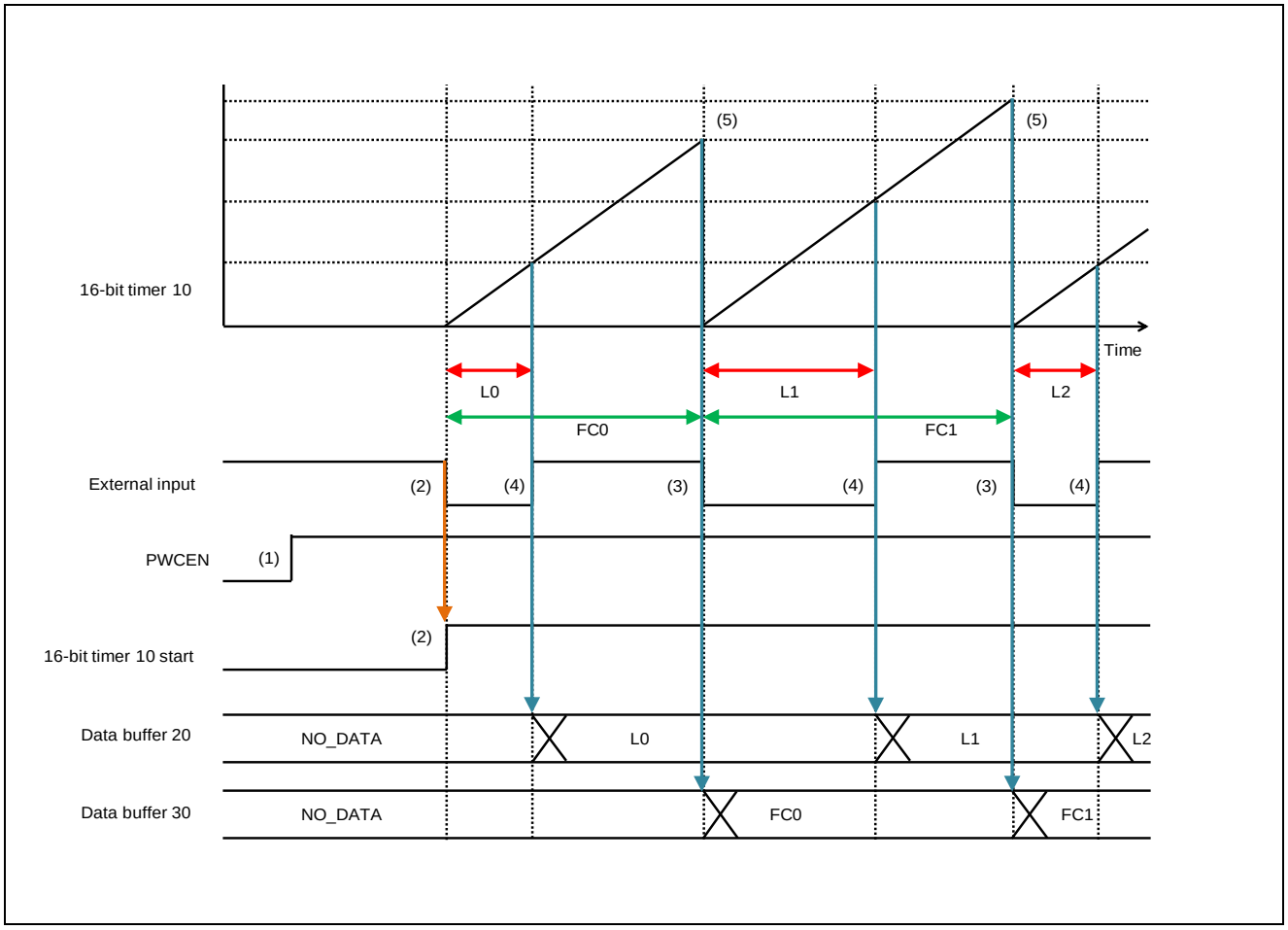
Figure 45-4. Operation of 16-bit Timer 00



Setting and operation procedure

1. Operation is enabled (PWCEN: bit0 in the PWC control register (PWCC) =1)
2. 16-bit timer 00 does not start at the time of falling detection.
3. When PWCEN: bit0 is 1 and the first rising edge is detected, 16-bit timer 00 starts.
4. After the start of 16-bit timer 00, when a falling edge is detected, the count value of 16-bit timer 00 is written to data buffer 00 as an "H" pulse width.
5. After the start of 16-bit timer 00, when a rising edge is detected, the count value of 16-bit timer 00 is written to data buffer 10 as a cycle between rising edges.
6. When a rising edge is detected, the count value of 16-bit timer 00 is cleared to "0x0000".
7. (4) to (6) are repeated.

Figure 45-5. Operation of 16-bit Timer 10



Setting and operation procedure

1. Operation is enabled (PWCEN: bit0 in the PWC control register (PWCC) =1)
2. When PWCEN: bit0 is 1 and the first falling edge is detected, 16-bit timer 10 starts.
3. After the start of 16-bit timer 10, when a rising edge is detected, the count value of 16-bit timer 10 is written to data buffer 20 as an "L" pulse width.
4. After the start of 16-bit timer 10, when a falling edge is detected, the count value of 16-bit timer 10 is written to data buffer 30 as a cycle between falling edges.
5. When a falling edge is detected, the count value of 16-bit timer 10 is cleared to "0x0000".
6. (3) to (5) are repeated.

45.5.2.2 Timer Clear

This section explains timer clear.

The count value of a 16-bit timer is cleared in any of the following cases.

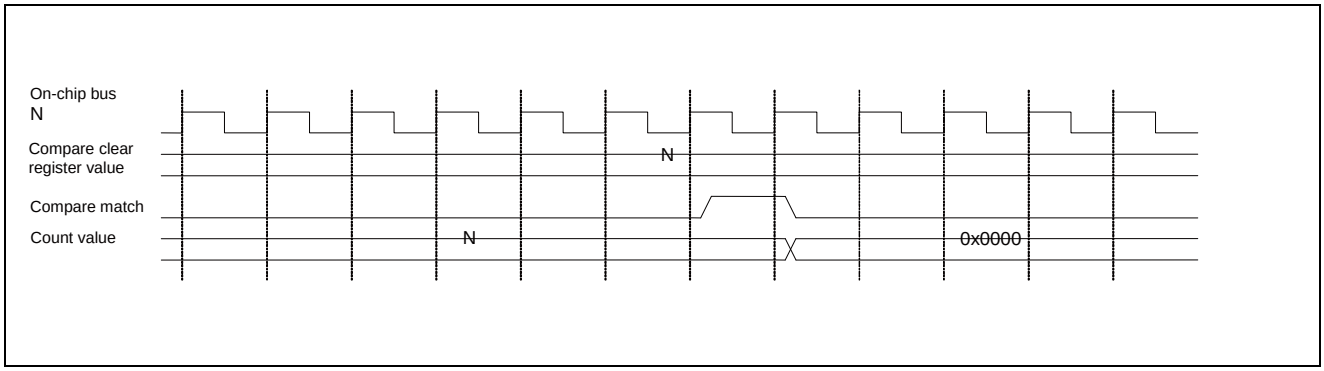
- When a match with the compare clear register is detected
- When the valid edge of external input is detected
- When "1" is written to SCLR: bit4 in the PWCTCCS register during operation
- When "0x0000" is written to the PWCTCDT register while operation is stopped
- When the value is reset

When reset, the counter is immediately cleared. If a software clear occurs or if a match with the compare clear register occurs, the counter is cleared in synchronization with the count timing.

Note:

When "1" is written to SCLR: bit4 in the PWCTCCS register while operation is stopped, the count value of the 16-bit timer is not cleared.

Figure 45-6. 16-bit Timer Clear Timing



45.5.2.3 Compare Clear Buffer

This section explains the compare clear buffer.

The compare clear register (PWCCPCLR) has a buffer function that can be enabled or disabled. If the buffer function is enabled (BFE: bit7 in the PWCTCCS register = 1), data written in the compare clear buffer register (PWCCPCLRB) is transferred to the PWCCPCLR register. This occurs when a 16-bit timer value of "0x0000" is detected. If the buffer function is disabled (BFE: bit7 in PWCTCCS = 0), data can be written directly to the PWCCPCLR register.

Figure 45-7. Operation When the Compare Clear Buffer Is Disabled (PWCTCCS Register BFE: Bit7 = 0)

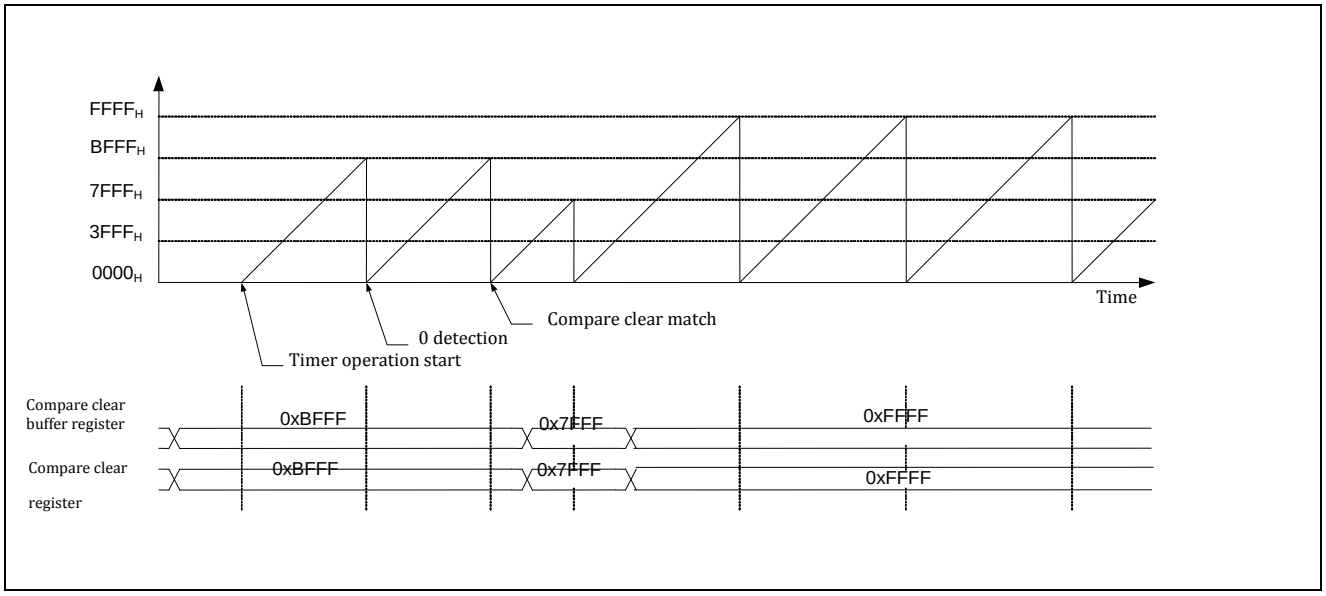
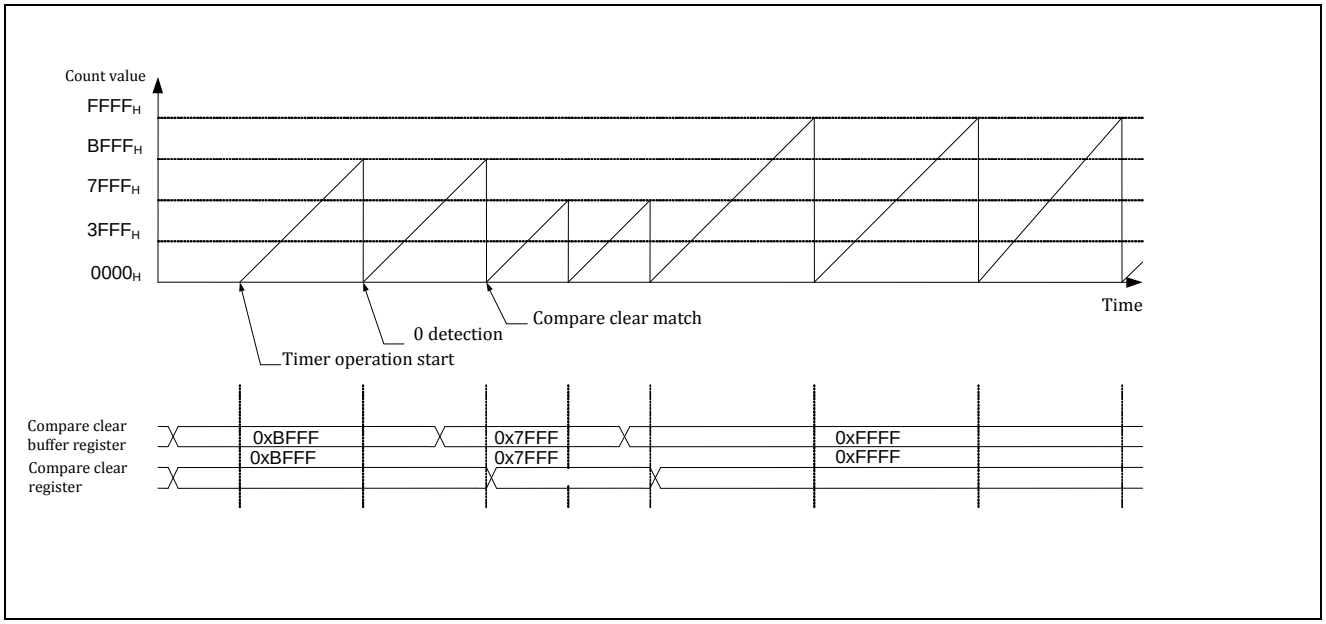


Figure 45-8. Operation When the Compare Clear Buffer Is Enabled (PWCTCCS Register BFE: Bit7 = 1)



45.5.2.4 Timer Interrupts

This section explains timer interrupts.

A 16-bit timer can generate the following 2 types of interrupts.

- Compare clear interrupt
- 0 detection interrupt

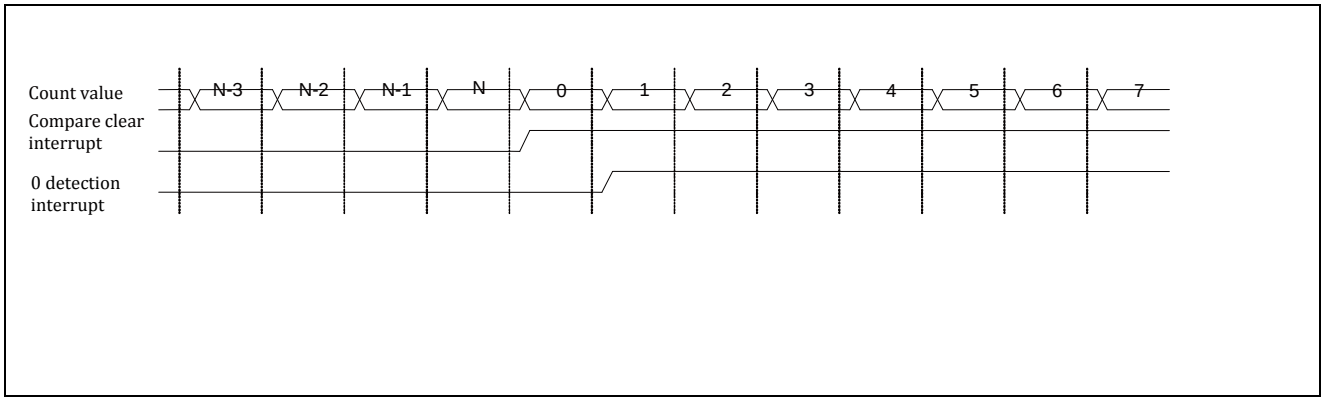
The compare clear interrupt is generated when the timer value matches the value of the compare clear register.

The 0 detection interrupt is generated when the timer value reaches "0x0000".

Note:

A software clear (SCLR: bit4 in the PWCTCCS register = 1) and the detection of a valid edge of external input do not generate the 0 detection interrupt.

Figure 45-9. Interrupts Generated by a 16-bit Timer



45.5.2.5 Interrupt Mask Function

This section explains the interrupt mask function.

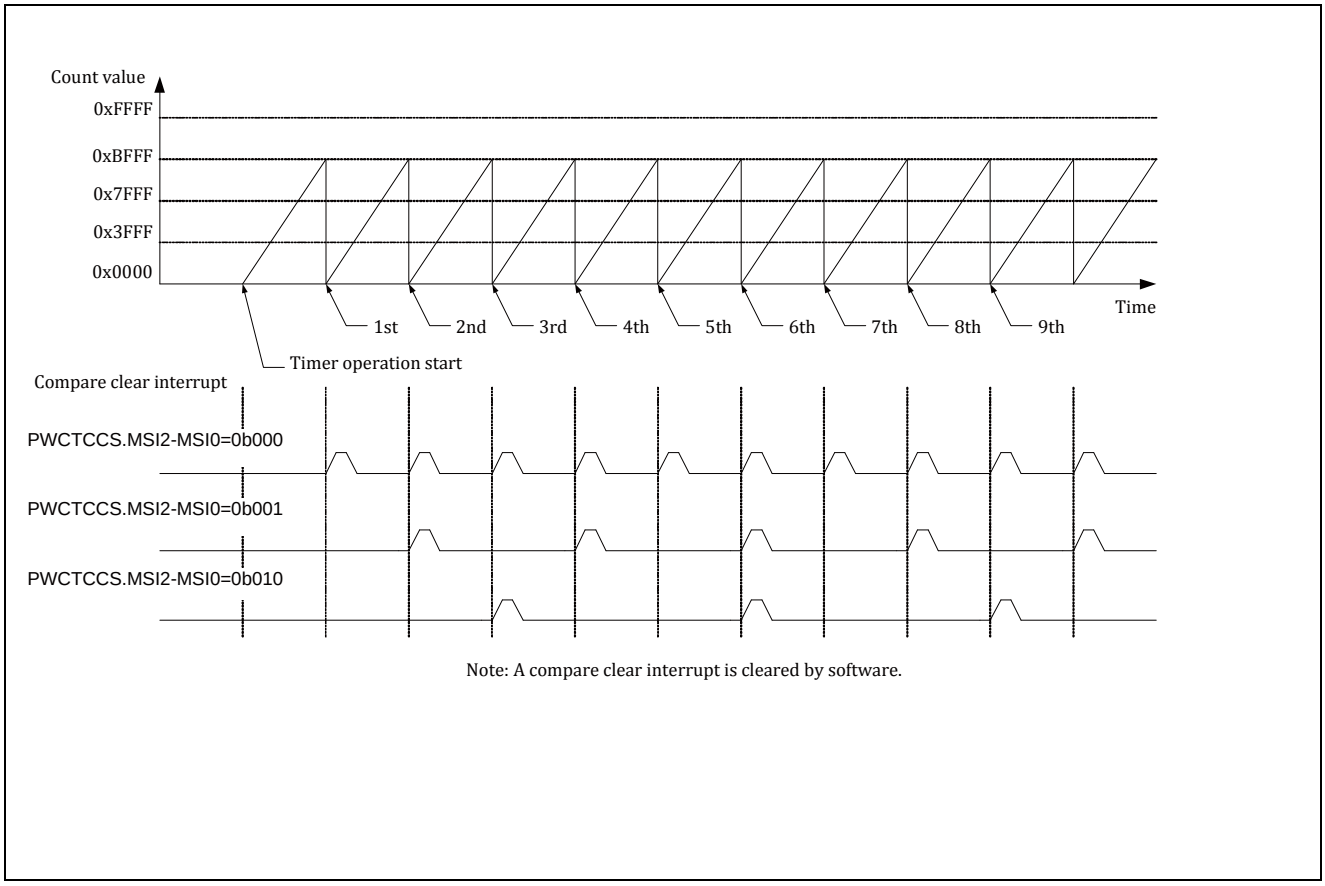
The function can mask compare match interrupts.

- When MSI2 to MSI0: bit12 to bit10 in the PWCTCCS register are set, the interrupt request can be masked. The MSI2 to MSI0 bits are a 3-bit reload down register that reloads a value when the count value reaches "000_B". The count value can be loaded by writing it directly to the MSI2 to MSI0 bits. The mask count is a value set in MSI2 to MSI0. When the MSI2 to MSI0 bits become "000_B", the interrupt request is not masked.

Note:

A software clear (SCLR: bit4 in the PWCTCCS register = 1) and the detection of a valid edge of external input do not generate the 0 detection interrupt.

Figure 45-10. Compare Clear Interrupts to Be Masked

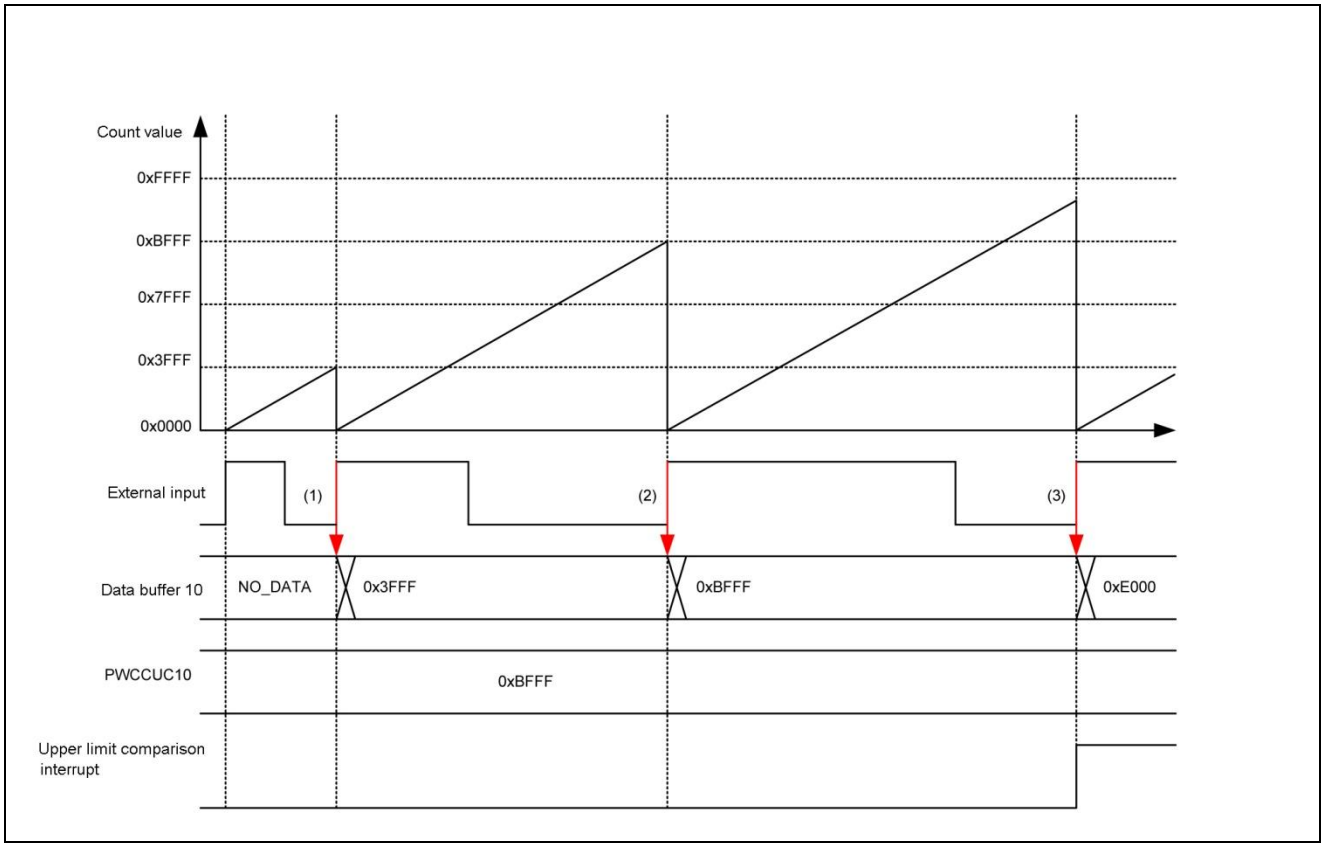


45.5.2.6 Capture Data Upper/Lower Limit Comparison Interrupt

This section explains the capture data upper/lower limit comparison interrupt.

Upper and lower limit comparison for capture data is performed when a 16-bit timer value is stored in a data buffer after the detection of an edge of external input. For example, Figure 45-11 shows upper limit comparison and Figure 45-12 shows lower limit comparison for a cycle between rising edges.

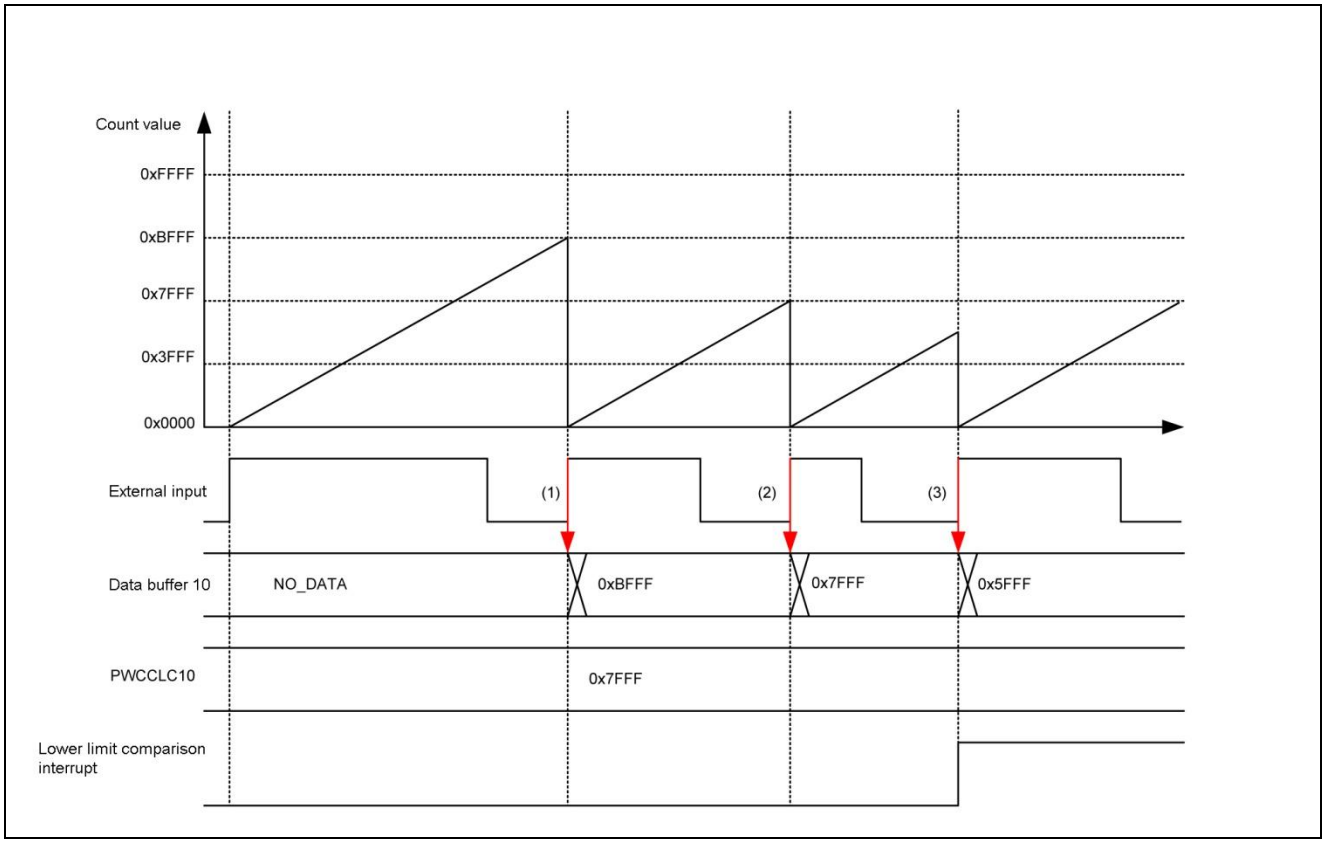
Figure 45-11. Operation During Upper Limit Comparison for Capture Data



Initial settings: External input level inversion is disabled and the upper limit value (PWCCUC10) is 0xBFFF.

1. The value of data buffer 10 is compared with the value of PWCCUC10. If the value of PWCCUC10 is greater, capture data upper limit interrupt 10 is not generated.
2. The value of data buffer 10 is compared with the value of PWCCUC10. If the values are the same, capture data upper limit interrupt 10 is not generated.
3. The value of data buffer 10 is compared with the value of PWCCUC10. If the value of data buffer 10 is greater, capture data upper limit interrupt 10 is generated.

Figure 45-12. Operation During Lower Limit Comparison for Capture Data



Initial settings: External input level inversion is disabled and the lower limit value (PWCCLC10) is 0x7FFF.

1. The value of data buffer 10 is compared with the value of PWCCLC10. If the value of PWCCLC10 is smaller, capture data lower limit interrupt 10 is not generated.
2. The value of data buffer 10 is compared with the value of PWCCLC10. If the values are the same, capture data lower limit interrupt 10 is not generated.
3. The value of data buffer 10 is compared with the value of PWCCLC10. If the value of data buffer 10 is smaller, lower limit comparison interrupt 10 is generated.

45.5.2.7 Data Buffer Interrupt

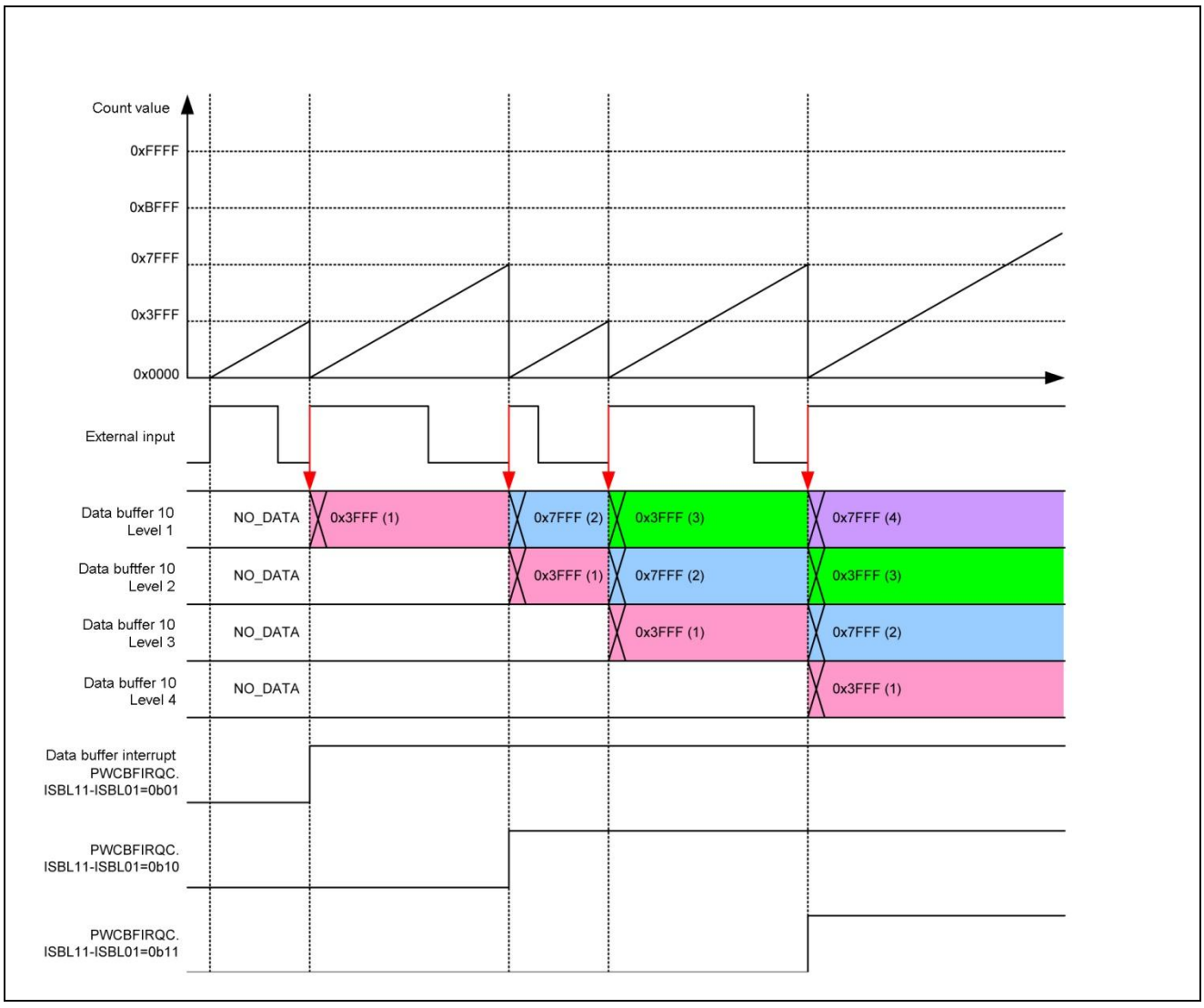
This chapter explains the data buffer interrupt.

When data is stored to the level(s) set by bit15 to bit8 in the buffer interrupt control register (PWCBFIRQC), the data buffer interrupt is generated. For example, Figure 45-13 shows the operation of a data buffer that stores data on the cycle between rising edges.

Note:

When the setting of the data buffer interrupt level selection bits (IBSL1x to 0x in the PWCBFIRQC register) is "0b00", the data buffer interrupt is not generated. (x = 0, 1, 2, 3)

Figure 45-13. Operation When Data Is Stored in a Buffer



45.5.2.8 Buffer Overrun Interrupt

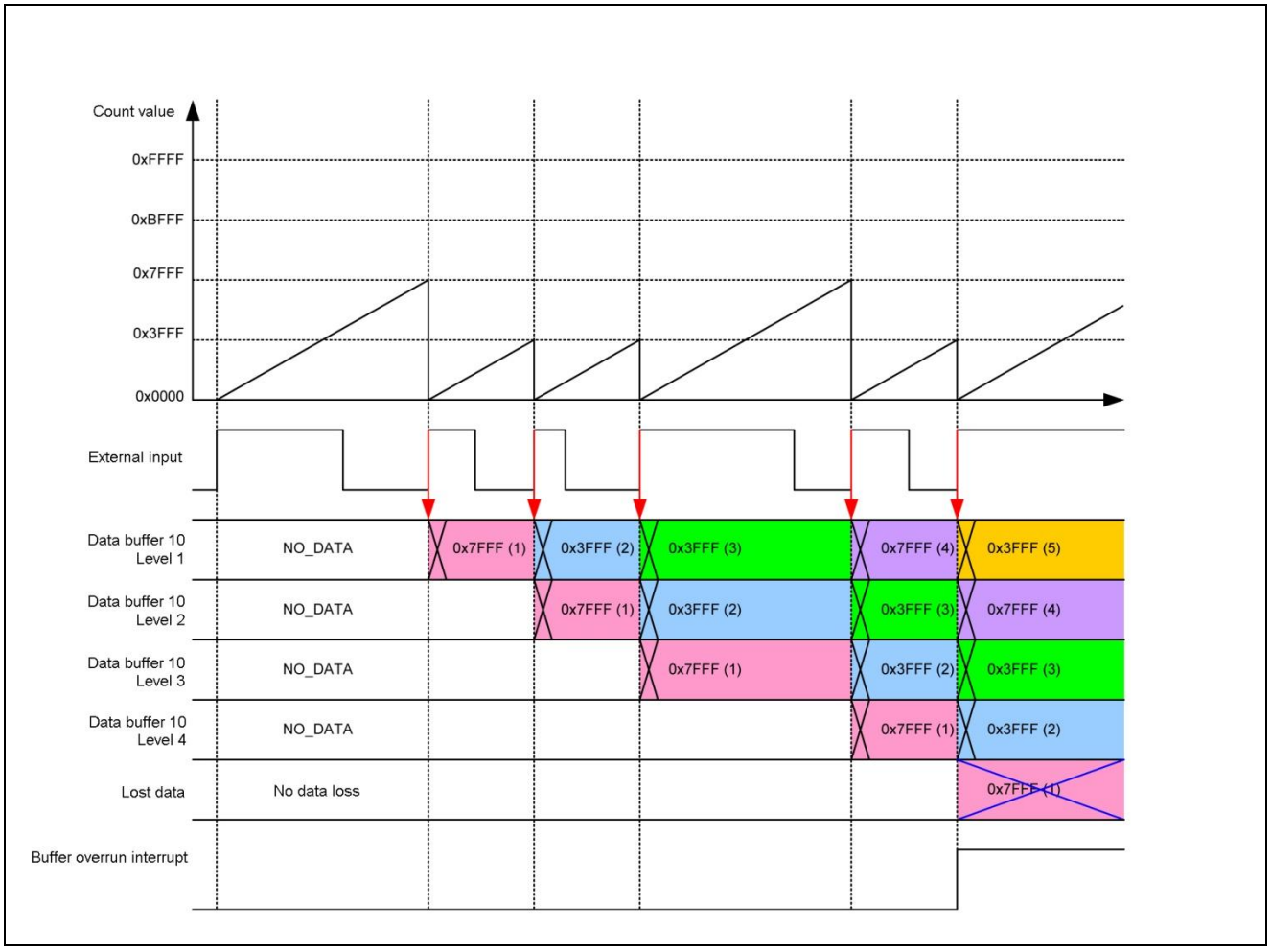
This section explains the buffer overrun interrupt.

When a new timer value is stored in a data buffer while the data buffer is full, a buffer overrun error is generated. If an error occurs, then data overwriting has occurred. The oldest data is lost due to the overwriting. For example, Figure 45-14 shows the operation of a data buffer that stores data on the cycle between rising edges.

Note:

Lost data cannot be restored.

Figure 45-14. Operation When a Buffer Overrun Occurs



45.6 Notes

This section provides notes on the PWC with buffers.

Notes to observe on controlling the PWC with buffers

- Do not initialize the PWC with buffers (PWC initialization register (PWCINIT)) at the same time that you enable its start using the PWC control register.
- Before enabling the start of the PWC (setting PWCEN in the PWC control register to 1), finish all other register settings. When PWCEN is 1, do not change any settings.

Notes to observe on using a program to configure settings

- Although the timer value becomes "0x0000" when a reset is performed, the 0 detection interrupt flag is not set.
- A software clear (SCLR in the PWCTCCS register = 1) initializes the timer, but does not generate the 0 detection interrupt.
- Set a compare value other than "0x0000". Otherwise, the timer value is updated to "0x0000" and fixed to "0x0000". The 0 detection interrupt flag and compare clear flag continue to be set at each count clock.

Notes on interrupts

- Before setting the interrupt request enable bit (IRQZE) in the timer state control register (PWCTCCS) to "1", be sure to clear the interrupt flag (IRQZF).
- Before setting the interrupt request enable bit (ICRE) in the timer state control register (PWCTCCS) to "1", be sure to clear the interrupt flag (ICLR).
- Before setting the interrupt enable bits (ICWE0 to 3) in the buffer interrupt control register (PWCBFIRQC) to "1", be sure to clear the interrupt flags (ICWF0 to 3) in the buffer interrupt flag register (PWCBFIRQF).
- Before setting the interrupt enable bits (ICBE0 to 3) in the buffer interrupt control register (PWCBFIRQC) to "1", be sure to clear the interrupt flags (ICBF0 to 3) in the buffer interrupt flag register (PWCBFIRQF).
- Before setting the interrupt enable bits (ICUE0 to 3) in the buffer interrupt control register (PWCBFIRQC) to "1", be sure to clear the interrupt flags (ICUF0 to 3) in the buffer interrupt flag register (PWCBFIRQF).
- Before setting the interrupt enable bits (ICLE0 to 3) in the buffer interrupt control register (PWCBFIRQC) to "1", be sure to clear the interrupt flags (ICLF0 to 3) in the buffer interrupt flag register (PWCBFIRQF).

Notes to observe when accessing the PWCTCCS register

- A read-modify-write instruction reads the set value from MSI2 to MSI0.
- A normal read operation reads the counter value from MSI2 to MSI0.

Notes to observe when accessing the PWCCPCLR register

The PWCCPCLR register in a 16-bit timer has a buffer function. Do not access this register by using a read-modify-write instruction.

Notes on the external input inversion function

Do not change the setting of the external input level invert bit (EXINV: bit1 in the PWC control register (PWCC)) while the PWC with buffers is operating. If you change it during operation, the operation is not assured.

Notes on the pulse width and cycle of external input

To obtain the measurement results of the pulse width and cycle of external input from the values of the data buffer read registers (PWCDBR00 to PWCDBR31) of the PWC with buffers, perform the following calculations.

Pulse width = (N+1) x T

Cycle = (M+1) x T

N: Value of data buffer read register (PWCDBR0n or PWCDBR2n) (n: Channel number)

M: Value of data buffer read register (PWCDBR1n or PWCDBR3n) (n: Channel number)

T: Count clock cycle

Pulse width and cycle measurement values can be calculated by adding 1 count to the value of the relevant data buffer read register and then multiplying the resulting value by the count clock cycle.

46. 12-bit A/D Converter (4-Channel Simultaneous Sampling)

This chapter explains the 12-bit A/D converter (4-channel simultaneous sampling).

46.1 Overview
46.2 Features
46.3 Configuration
46.4 Registers
46.5 Explanation of Operation
46.6 Notes

46.1 Overview

This section provides an overview of the 12-bit A/D converter (4-channel simultaneous sampling).

The 12-bit A/D converter (4-channel simultaneous sampling) converts the analog input voltage for 4 channels to 12-bit digital values by simultaneously sampling it using the RC successive approximation conversion method. The converted values are stored into a register in the order of AN8, AN9, AN10, and then AN11. A/D conversion always starts from AN8. However, setting the conversion end channel allows conversion to end with an intermediate channel. (Example: End of conversion in 1 channel for AN8 only, end of conversion in 2 channels up to AN8 and AN9)

A/D conversion is performed by the A/D activation trigger. If the A/D activation trigger is input again during A/D conversion, no A/D conversion reactivation occurs. If the A/D activation trigger is input during conversion, an irregular activation interrupt is reported. Moreover, the forced stop function is supported which is performed with the A/D conversion cancel input signal through the bit0 setting of AD4TBUSY:BUSY7 to BUSY0 during A/D conversion. Even if the forced termination by "0" writing of AD4TBUSY:BUSY7 to BUSY0 is set in any register of AD4TBUSY:BUSY7 to BUSY0, the A/D converter is forcibly terminated.

46.2 Features

This section explains the features of the 12-bit A/D converter (4-channel simultaneous sampling).

46.2.1 A/D Activation Control Functions

This section explains the A/D activation control functions.

Activation control section

A/D activation request control performs activation control for 4 channels (AN8 to AN11) by using the 12-bit A/D converter (4-channel simultaneous sampling). The activation control section has 8 blocks.

The activation control section consists of the following registers.

- 4-channel A/D activation trigger control status register
- 4-channel A/D activation trigger extension control register

A/D activation request

- The activation control section issues an A/D activation request with any one of the software, external trigger (falling), reload timer (rising), and activation trigger from PWM (12 types: rising). However, the activation control section does not internally perform reactivation during A/D conversion (activation request).
- Sampling for 4 channels (AN8 to AN11) is simultaneously completed for the 12-bit A/D converter (4-channel simultaneous sampling) by the software activation, external trigger, reload timer, or an activation trigger from PWM. Then, comparison starts.

  A single activation request is made with a single activation factor. A/D sampling is performed for 4 channels (AN8 to AN11) at any one time. An activation request is released at the end of the A/D compare operation for the end setting channel. A/D conversion can be ended each time data is stored in the data register of each of channels AN8 to AN11.
- Conversion data is stored in the A/D data register when A/D conversion ends. The A/D data register is provided for each analog input channel (AN8 to AN11).
- A/D conversion data is always converted and stored in the order of AN8 to AN11.
- Each A/D data register contains the error flag bit and error status bit to enable you to know the A/D conversion data status from these bit values.

Interrupt request

- Each activation channel generates a conversion end interrupt request at compare end for each analog input channel.
- Suppose that the sampling time from the conversion end of the analog input channel set by bit15, bit14 (CHSEL1, CHSEL0) in the A/D activation trigger extension control register (AD4TECS) to the next A/D activation is insufficient. A sampling time insufficiency interrupt request is generated.
- If an activation factor occurs during A/D conversion (compare), an irregular activation interrupt request is generated.

Data protection function

- The A/D data register for each analog input channel can set the data protection function.
- When the data protection function is enabled, A/D data update is not overwritten until A/D data register data reading and interrupt flag clear are performed. However, data reading and interrupt flag clear can be done in any order. Moreover, you can select whether interrupt flag clear can be a protection condition.
- The status of an A/D activation request in progress or the status of conversion in progress is reported by the A/D activation trigger status register (AD4TBUSY:BUSY7 to BUSY0). If you forcibly terminate the present A/D activation request or conversion, write "0" to the A/D activation trigger status register (AD4TBUSY:BUSY7 to BUSY0).

  A/D is forcibly terminated by writing "0" to any of BUSY7 to BUSY0 in the register.

46.2.2 A/D Activation Arbitration Function

This section explains the A/D activation arbitration function.

- A/D activation arbitration consists of an arbitration circuit and the A/D activation trigger.
- A/D activation arbitration arbitrates activation requests made by the activation triggers to generate an activation trigger and 4-channel A/D conversion cancel signal.
- Suppose that another activation trigger is input during A/D conversion (the activation requests for each A/D activation trigger conflict with each other).

  Even if another activation trigger is input before the current A/D conversion has been completed, A/D conversion is not reactivated.

  If another activation trigger is input, as above, the function determines that an unconvertible activation request has been input and sets the irregular activation interrupt flag.

  A/D conversion is not performed for the second activation trigger that is input during A/D conversion, regardless of the activation trigger type.
- If A/D activation triggers of the same type are simultaneously input for another activation control channel, the channel with the smaller number is given priority and an activation trigger is issued. Any A/D activation trigger of a low-priority channel number is ignored and the irregular activation interrupt flag is set.

46.2.3 12-bit A/D Converter (4-channel Simultaneous Sampling) Control Function

This section explains the 12-bit A/D converter (4-channel simultaneous sampling) control function.

The 12-bit A/D converter (4-channel simultaneous sampling) converts an analog voltage (input voltage) input to the analog input pin, to a digital value. It has the following features.

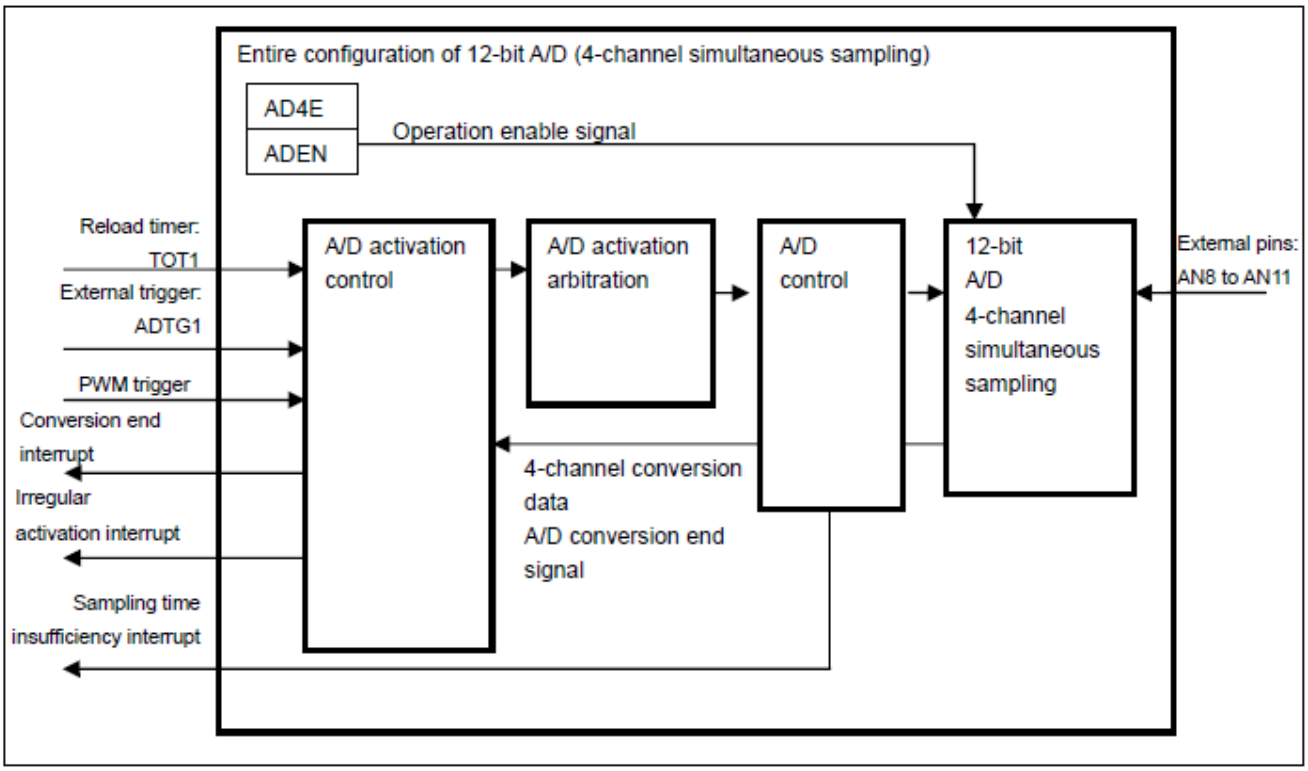
- 4 channels are simultaneously sampled.
- For details on the conversion time, see the data sheet.
- Conversion is done using the RC successive approximation conversion method with a sample hold circuit added.
- The activation signal is input as a pulse signal.
- For A/D conversion, single conversion is performed by the input of a single activation factor.
- If the A/D conversion cancel signal is received during A/D conversion, the current processing stops. (Forced stop function)
- The sampling time limit value and compare time can be set.

46.3 Configuration

This section explains the configuration of the 12-bit A/D converter (4-channel simultaneous sampling).

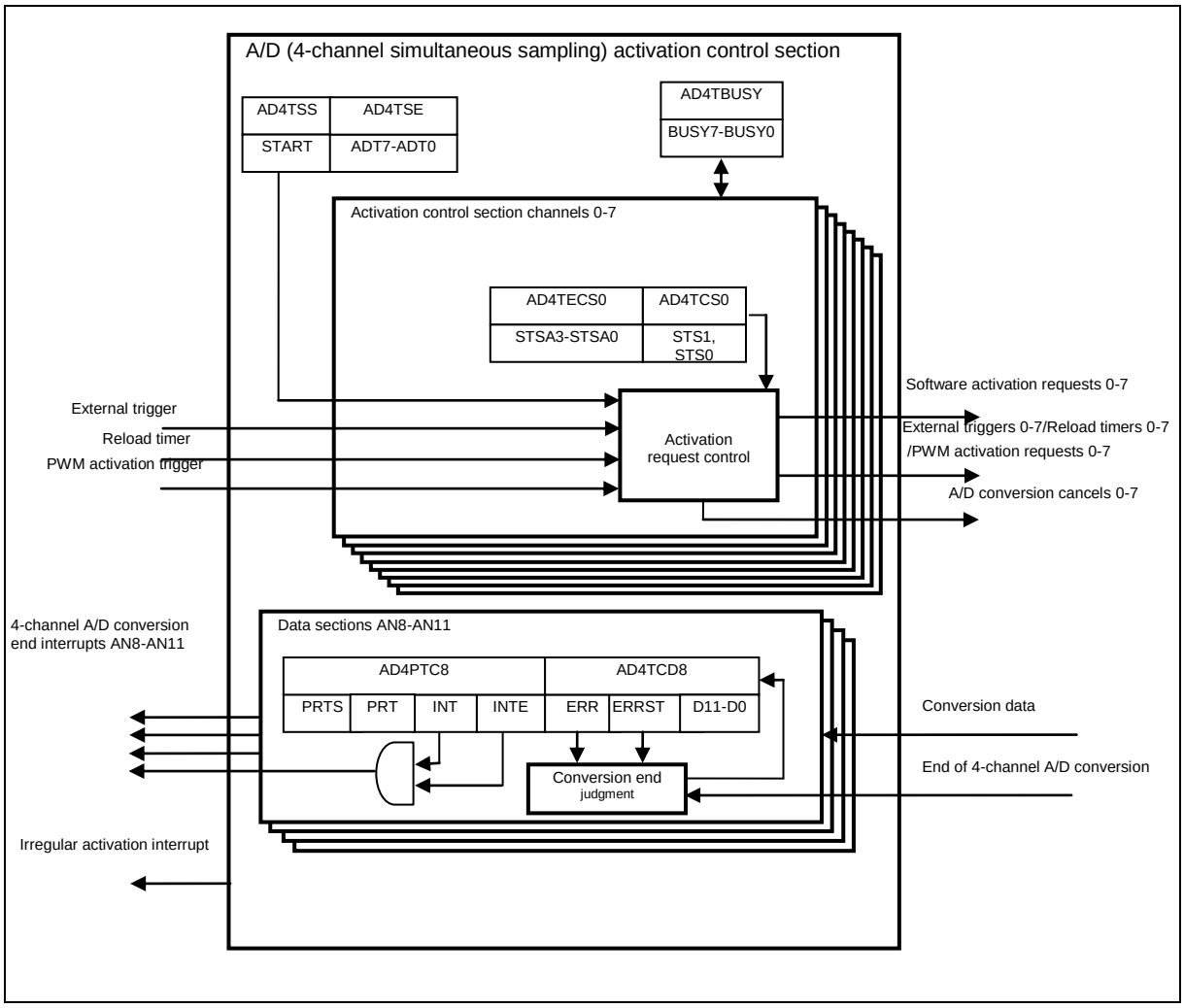
Entire configuration

Figure 46-1. Entire A/D Configuration (12-bit A/D Converter (4-channel Simultaneous Sampling))



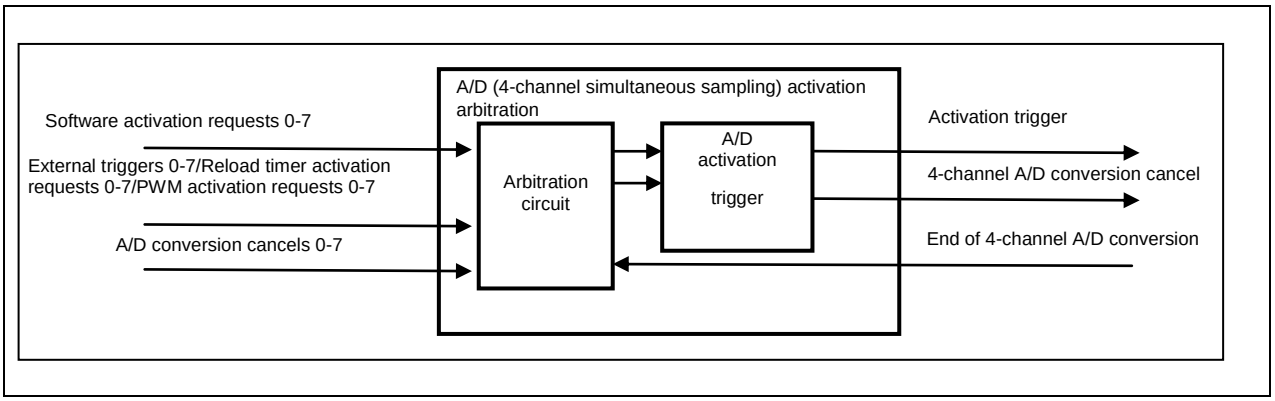
Configuration of A/D activation control section

Figure 46-2. Configuration of A/D Activation Control Section (12-bit A/D Converter (4-channel Simultaneous Sampling))



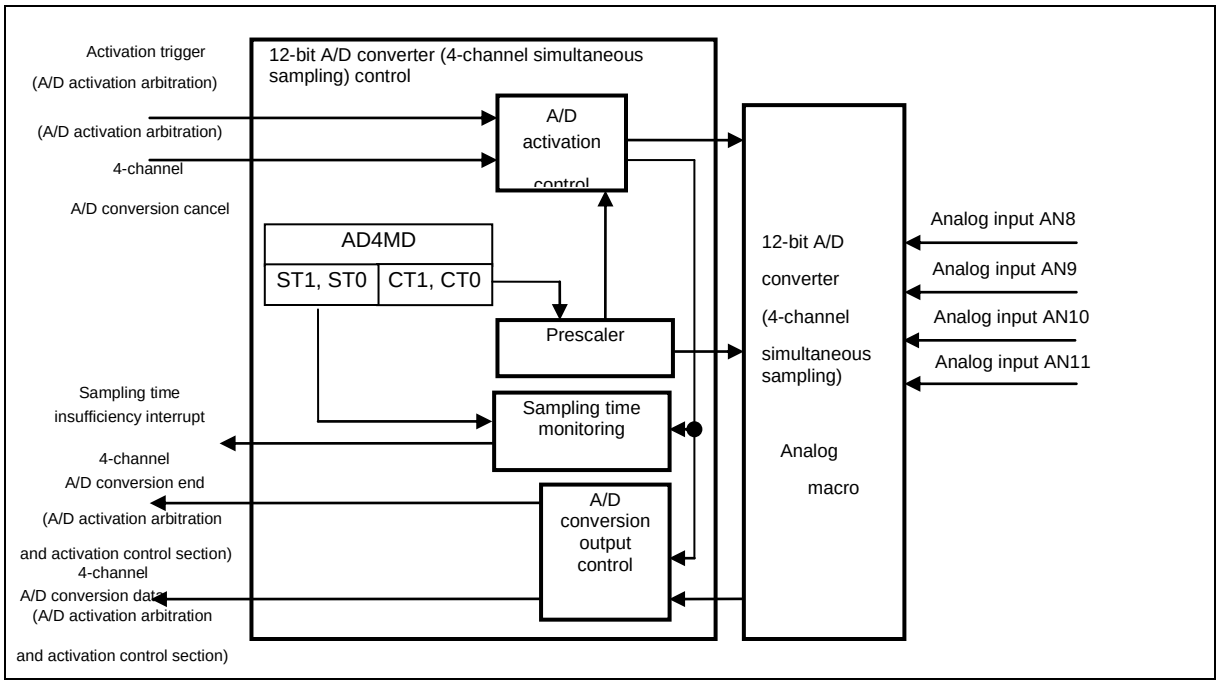
Configuration of A/D activation arbitration

Figure 46-3. A/D Activation Arbitration Configuration (12-bit A/D Converter (4-channel Simultaneous Sampling))



Configuration of 12-bit A/D converter control

Figure 46-4. Configuration of 12-bit A/D Converter (4-channel Simultaneous Sampling) Control



46.4 Registers

This section explains the registers of the 12-bit A/D converter (4-channel simultaneous sampling).

List of 12-bit A/D activation control registers

Table 46-1. List of 12-bit A/D Converter (4-channel Simultaneous Sampling) Activation Control Registers

Address	+0	+1	+2	+3
0x3320	4-channel A/D enable register (AD4EN)	Reserved	Reserved	Reserved
0x3324	4-channel A/D software activation register	Reserved	Reserved	Reserved
0x3328	4-channel A/D software activation channel selection register (AD4TSE)			
0x332C	4-channel A/D activation trigger control status register 0 (AD4TCS0)		4-channel A/D activation trigger control status register 1 (AD4TCS1)	
0x3330	4-channel A/D activation trigger control status register 2 (AD4TCS2)		4-channel A/D activation trigger control status register 3 (AD4TCS3)	
0x3334	4-channel A/D activation trigger control status register 4 (AD4TCS4)		4-channel A/D activation trigger control status register 5 (AD4TCS5)	
0x3338	4-channel A/D activation trigger control status register 6 (AD4TCS6)		4-channel A/D activation trigger control status register 7 (AD4TCS7)	
0x333C	4-channel A/D activation trigger status register (AD4TBUSY)			
0x3340	4-channel A/D activation trigger extension control register 0 (AD4TECS0)		4-channel A/D activation trigger extension control register 1 (AD4TECS1)	
0x3344	4-channel A/D activation trigger extension control register 2 (AD4TECS2)		4-channel A/D activation trigger extension control register 3 (AD4TECS3)	
0x3348	4-channel A/D activation trigger extension control register 4 (AD4TECS4)		4-channel A/D activation trigger extension control register 5 (AD4TECS5)	
0x334C	4-channel A/D activation trigger extension control register 6 (AD4TECS6)		4-channel A/D activation trigger extension control register 7 (AD4TECS7)	
0x3350	4-channel A/D interrupt/protection control register 8 (AD4PTC8)	4-channel A/D interrupt/protection control register 9 (AD4PTC9)	4-channel A/D interrupt/protection control register 10 (AD4PTC10)	4-channel A/D interrupt/protection control register 11 (AD4PTC11)

List of 12-bit A/D data registers

Table 46-2. List of 12-bit A/D Converter (4-channel Simultaneous Sampling) Data Registers

Address	+0	+1	+2	+3
0x3354	4-channel A/D data register AN8 (AD4TCD8)		4-channel A/D data register AN9 (AD4TCD9)	
0x3358	4-channel A/D data register AN10 (AD4TCD10)		4-channel A/D data register AN11 (AD4TCD11)	

List of 12-bit A/D converter control registers

Table 46-3. List of 12-bit A/D Converter (4-channel Simultaneous Sampling) Control Registers

Address	+0	+1	+2	+3
0x335C	4-channel A/D control status register (AD4CS)		Reserved	4-channel A/D mode setting register (AD4MD)
0x3360	4-channel data protection status flag register (AD4PRTF)			

46.4.1 A/D Activation Control Registers

This section explains the A/D activation control registers.

46.4.1.1 4-channel A/D Enable Register: AD4EN

This section shows the bit configuration of the 4-channel A/D enable register.

The 4-channel A/D enable register (AD4EN) places A/D of the 12-bit A/D converter (4-channel simultaneous sampling) in the operation status.

AD4EN: Address 3320H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							ADEN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit7 to bit1] Reserved

[bit0] ADEN: A/D enable bit

ADEN	Function
0	Disable the A/D converter operation.
1	Enable the A/D converter operation.

Writing "1" to this bit causes the A/D converter to enter the operation status. When the A/D converter enters the operation status, it enters the sampling status once the resumption time elapses.

Notes:

- ADEN = "1" indicates that there are some unchangeable registers.
- ADEN is set to "0" in standby mode.

46.4.1.2 4-channel A/D Software Activation Register: AD4TSS

This section shows the bit configuration of the 4-channel A/D software activation register.

The 4-channel A/D software activation register (AD4TSS) issues an A/D activation request for the 12-bit A/D converter (4-channel simultaneous sampling). However, the channel to be activated is set by the 4-channel A/D software activation channel selection register (AD4TSE).

AD4TSS: Address 3324H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							START
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,W

[bit7 to bit1] Reserved

[bit0] START: A/D conversion activation bit (software)

START	Function
0	Do not activate the A/D conversion function.
1	Activate the A/D conversion function.

- This bit activates the A/D conversion operation by software.
- Writing "1" to this bit activates A/D conversion. However, the channel to be activated is set by the 4-channel A/D software activation channel selection register (AD4TSE).

46.4.1.3 4-channel A/D Software Activation Channel Selection Register: AD4TSE

This section shows the bit configuration of the 4-channel A/D software activation channel selection register.

The 4-channel A/D software activation channel selection register (AD4TSE) selects the activation channel that issues an A/D activation request.

AD4TSE: Address 3328H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADT07	ADT06	ADT05	ADT04	ADT03	ADT02	ADT01	ADT00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit31 to bit8] Reserved

[bit7 to bit0] ADT07 to ADT00: Software activation channel selection bits

ADT0n (n=7 to 0)	Function
0	Disable software activation.
1	Enable software activation.

- These bits control the software activation of the activation channel.
- If these bits are "0", software activation is disabled.
- If these bits are "1", software activation is enabled.

Notes:

- Do not enable software activation selection for multiple activation channels at the same time.

  If this is enabled for multiple channels at the same time, the channel with the smaller activation channel number is given priority.
- If software activation is not used, set ADT07 to ADT00 = 8'h00.

46.4.1.4 4-channel A/D Activation Trigger Control Status Registers: AD4TCS0 to AD4TCS7

This section shows the bit configuration of the 4-channel A/D activation trigger control status registers.

The 4-channel A/D activation trigger control status register (AD4TCS) is used for activation factor selection.

AD4TCS0 to AD4TCS7: Address $332C_H$ to $333A_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved			STS1	STS0	Reserved		
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R/W	R0,WX	R0,WX	R0,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

[bit15 to bit13] Reserved

[bit12, bit11] STS1, STS0: A/D activation factor selection bits

STS1	STS0	Function
0	0	Activate software.
0	1	Activate the external pin trigger (falling edge).
1	0	Activate the reload timer (rising edge).
1	1	Activate PWM*1.

*1 The STS1, STS0 bits are used to select an A/D conversion activation factor in combination with bit11 to bit8 (STSA3 to STSA0 in the 4-channel A/D activation trigger extension control register (AD4TECS).

Notes:

- The A/D activation factor selection bits are changed simultaneously with rewriting. So, change these bits when the activation factors of the current selection destination and those of the selection destination to be changed are not active and when no A/D conversion request is in progress (AD4TBUSY:BUSY7 to BUSY0 = 1).
- If you do not issue an A/D activation request, set these bits STS1, STS0 to software activation ("00_B").

[bit10 to bit0] Reserved

46.4.1.5 4-channel A/D Activation Trigger Status Register: AD4TBUSY

This section explains the bit configuration of the 4-channel A/D activation trigger status register.

The 4-channel A/D activation trigger status register (AD4TBUSY) is used to verify the A/D activation request.

AD4TBUSY: Address 333CH (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BUSY7	BUSY6	BUSY5	BUSY4	BUSY3	BUSY2	BUSY1	BUSY0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W

[bit31 to bit8] Reserved

[bit7 to bit0] BUSY7 to BUSY0: Bits indicating that A/D activation is being requested

BUSY7 to BUSY0	Function	
	During Read Operation	During Write Operation
0	Do not request A/D activation.	Forcibly stop the A/D activation request.
1	Indicate that A/D activation is being requested or A/D conversion is in progress.	No change, no effect on other operations

- These bits are used to display the operation of the A/D activation request or the conversion of each activation channel.
- At reading, if this bit is "0", this indicates that A/D conversion is not requested. If it is "1", this indicates that A/D conversion is being requested or is in progress.
- At writing, writing "0" to this bit forcibly stops the A/D activation request or conversion.

  Even if "0" is written to any bit of BUSY7 to BUSY0, the A/D activation request or conversion is forcibly stopped. Writing "1" causes no change and has no effect on the other operations.

Note:

"1" is read upon reading of the read modify write (RMW) instruction.

46.4.1.6 4-channel A/D Activation Trigger Extension Control Registers: AD4TECS0 to AD4TECS7

This section explains the bit configuration of the 4-channel A/D activation trigger extension control registers.

The 4-channel A/D activation trigger extension control register (AD4TECS) is used to select an activation factor for PWM, set the data channel to end A/D conversion, and set the update of each data channel.

AD4TECS0 to AD4TECS7: Address 3340H to 334EH (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CHSEL1	CHSEL0	Reserved	Reserved	STSA3	STSA2	STSA1	STSA0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R0,WX	R0,WX	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DTIH8	DTIH9	DTIH10	DTIH11	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R0,WX

[bit15, bit14] CHSEL1, CHSEL0: A/D conversion end channel selection bits

CHSEL1	CHSEL0	Function
0	0	Select AN8 as the conversion end analog input channel.
0	1	Select AN9 as the conversion end analog input channel.
1	0	Select AN10 as the conversion end analog input channel.
1	1	Select AN11 as the conversion end analog input channel.

- Select the analog input channel to end A/D conversion.
- The selected conversion end channel is converted, after which the A/D converter enters the sampling status.

Note:

The A/D activation factor selection bits are changed simultaneously with rewriting. So, change these bits when the activation factors of the current selection destination and those of the selection destination to be changed are not active and when no A/D conversion request is in progress (AD4TBUSY:BUSY7 to BUSY0 = 1).

[bit13 to bit12] Reserved

[bit11 to bit8] STSA3, STSA2, STSA1, STSA0: A/D activation factor extension selection bits (PWM)

STSA3	STSA2	STSA1	STSA0	Function
0	0	0	0	PWM1: Edge trigger 0
0	0	0	1	PWM2: Edge trigger 1
0	0	1	0	PWM3: Edge trigger 2
0	0	1	1	PWM4: Edge trigger 3
0	1	0	0	PWM5: Fault result 0
0	1	0	1	PWM6: Fault result 1
0	1	1	0	PWM7: Fault result 2
0	1	1	1	PWM8: Fault result 3
1	0	0	0	PWM9: Fault result 4
1	0	0	1	PWM10: Fault result 5
1	0	1	0	PWM11: Special event trigger 0
1	0	1	1	PWM12: Special event trigger 1
1	1	0	0	Setting prohibited
1	1	0	1	
1	1	1	0	
1	1	1	1	

When AD4TCS.STS1 and AD4TCS.STS0 bits select 11_B, you can select the type of the A/D converter activation trigger created with PWM.

Note:

The A/D activation factor selection bits are changed simultaneously with rewriting. So, change these bits when the activation factors of the current selection destination and those of the selection destination to be changed are not active and when no A/D conversion request is in progress (AD4TBUSY:BUSY7 to BUSY0 = 1).

[bit7] DTIH8: Data update disable bit (AN8)

DTIH8	Function
0	AN8: Allow update of this channel data.
1	AN8: Prohibit update of this channel data.

This bit controls update of AN8 channel data.

Note:

The A/D activation factor selection bits are changed simultaneously with rewriting. So, change these bits when the activation factors of the current selection destination and those of the selection destination to be changed are not active and when no A/D conversion request is in progress (AD4TBUSY:BUSY7 to BUSY0 = 1).

[bit6] DTIH9: Data update disable bit (AN9)

DTIH9	Function
0	AN9: Allow update of this channel data.
1	AN9: Prohibit update of this channel data.

This bit controls update of the AN9 channel data.

Note:

The A/D activation factor selection bits are changed simultaneously with rewriting. So, change these bits when the activation factors of the current selection destination and those of the selection destination to be changed are not active and when no A/D conversion request is in progress (AD4TBUSY:BUSY7 to BUSY0 = 1).

[bit5] DTIH10: Data update disable bit (AN10)

DTIH10	Function
0	AN10: Allow update of this channel data.
1	AN10: Prohibit update of this channel data.

This bit controls update of the AN10 channel data.

Note:

The A/D activation factor selection bits are changed simultaneously with rewriting. So, change these bits when the activation factors of the current selection destination and those of the selection destination to be changed are not active and when no A/D conversion request is in progress (AD4TBUSY:BUSY7 to BUSY0 = 1).

[bit4] DTIH11: Data update disable bit (AN11)

DTIH11	Function
0	AN11: Allow update of this channel data.
1	AN11: Prohibit update of this channel data.

This bit controls update of the AN11 channel data.

Note:

The A/D activation factor selection bits are changed simultaneously with rewriting. So, change these bits when the activation factors of the current selection destination and those of the selection destination to be changed are not active and when no A/D conversion request is in progress (AD4TBUSY:BUSY7 to BUSY0 = 1).

[bit3 to bit0] Reserved

46.4.1.7 4-channel A/D Interrupt/Protection Control Registers: AD4PTC8 to AD4PTC11

This section explains the bit configuration of the 4-channel A/D interrupt/protection control registers.

The 4-channel A/D interrupt/protection control register (AD4PTC) is used to enable or disable an interrupt request, verify the interrupt request status, and control the protection function.

AD4PTC8 to AD4PTC11: Address 3350H to 3353H (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	INT	INTE	PRT	PRTS	Reserved			
Initial Value	0	0	0	0	0	0	0	0
Attribute	R(RM1)/W	R/W	R/W	R/W	R0,WX	R0,WX	R0,WX	R0,WX

[bit7] INT: Interrupt request flag bit

INT	Function	
	During Read Operation	During Write Operation
0	Indicate that A/D conversion is not complete.	Clear this bit.
1	End A/D conversion.	No change, no effect on other operations

- This bit is set to "1" when data is set in each A/D data register (AD4TCD8 to AD4TCD11) by A/D conversion. (This bit is set only when data is written.)
- An interrupt request is generated when this bit and the interrupt request enable bit (AD4PTC:INTE) are "1".
- At writing, this bit is cleared with "0". Writing "1" causes no change and no effect on other operations.
- This bit is cleared when the A/D conversion end interrupt clear signal is "H".

Note:

"1" is read upon reading of the read modify write (RMW) instruction.
If software clear ("0" writing) or clear by the interrupt clear signal ("H") and hardware set occur at the same time, the hardware set is given priority.

[bit6] INTE: Interrupt request enable bit

INTE	Function
0	Disable interrupt request output.
1	Enable interrupt request output.

- This bit enables or disables the output of an interrupt to the CPU.
- An interrupt request is generated when this bit and the interrupt request flag bit (AD4PTC:INT) are "1".

[bit5] PRT: A/D data register protection enable bit

PRT	Function
0	Disable protection.
1	Enable protection.

If this bit is set to "1", the A/D data register is protected from overwriting. Suppose that conversion data has been stored in the A/D data register. After that, the A/D data register is protected from overwriting until the factor to be set by the A/D data register protection release selection bit (AD4PTC.PRTS) occurs.

Note:

Set the A/D data register protection enable bit before running A/D conversion. Do not change the A/D data register protection enable bit while A/D conversion is being requested or while the A/D data register is protected.

[bit4] PRTS: A/D data register protection release selection bit

PRTS	Function
0	Read data and clear the interrupt flag.
1	Read data.

- When the A/D data register protection function is enabled (PRT = 1), select a data update protection release condition.
- If this bit is set to "0", A/D data register (AD4TCD) reading and interrupt request flag bit (INT) clear become the data update protection release conditions (in any order).
- If this bit is set to "1", A/D data register (AD4TCD) reading becomes a data update protection release condition.

Note:

Set the A/D data register protection release bit before running A/D conversion.
Do not change the A/D data register protection release selection bit while A/D conversion is being requested or while the A/D data register is protected.

[bit3 to bit0] Reserved

46.4.1.8 4-channel A/D Data Registers: AD4TCD8 to AD4TCD11

This section shows the bit configuration of the 4-channel A/D data registers.

The 4-channel A/D data register (ADT4CD) is used to store the A/D conversion results.

AD4TCD8 to AD4TCD11: Address 3354_H to $335B_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ERR	ERRST	Reserved		D11	D10	D9	D8
Initial Value	1	0	0	0	0	0	0	0
Attribute	R, WX	R,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15, bit14] ERR, ERRST: Conversion data error flag bit/conversion data error status bit

ERR	ERRST	Function
0	0	Store the latest data that has not yet read.
0	1	(Meaningless status)*
1	0	Store old data that has already been read.
1	1	The latest data is stored. However, the data stored before that storing is overwritten without being read.

- Suppose that the 4-channel A/D activation trigger extension control register AD4TECS.DTIH is 0 (enable). In this case, ERR/ERRST indicates the above status if the 4-channel A/D interrupt/protection control register AD4PTC.PRT is 0.
- Suppose that the 4-channel A/D activation trigger extension control register AD4TECS.DTIH is 0 (enable). In this case, "00_B" is read from ERR/ERRST if the 4-channel A/D interrupt/protection control register AD4PTC.PRT is 1.
- When the 4-channel A/D activation trigger extension control register AD4TECS.DTIH is 1 (disable), "00_B" is read from ERR/ERRST.
- The ERR and ERRST bits are set to "10_B" when read.
- When a new conversion result is written to this register, AD4TCD.ERR is set to "0".

*: This status does not exist.

[bit13, bit12] Reserved

[bit11 to bit0] D11 to D0: A/D data bits

D11 to D0	Function
	Conversion data

- The A/D conversion result is stored and the register is rewritten each time conversion ends.
- The last conversion value is generally stored.

Note:

Do not write to this register.

46.4.2 12-bit A/D Converter (4-channel Simultaneous Sampling) Control Registers

This section explains the 12-bit A/D converter (4-channel simultaneous sampling) control registers.

The 12-bit A/D converter (4-channel simultaneous sampling) control registers include the 4-channel A/D control status register and 4-channel A/D mode setting register.

46.4.2.1 4-channel A/D Control Status Register: AD4CS

This section shows the bit configuration of the 4-channel A/D control status register.

The 4-channel A/D control status register (AD4CS) provides a function that enables or disables an interrupt request and verifies the interrupt request status.

AD4CS: Address 335CH (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IRR	SERR	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R(RM1)/W	R(RM1)/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IRRE	SERRE	Reserved					
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit15] IRR: Irregular activation interrupt request bit

IRR	Function	
	During Read Operation	During Write Operation
0	Indicates that no activation is input during A/D conversion.	Clear this bit.
1	Indicates that activation is input during A/D conversion.	No change, no effect on other operations

- This bit is set to "1" when an activation request is input during A/D conversion.
- An interrupt request is generated when this bit and the interrupt request enable bit (AD4CS:IRRE) are "1".
- At reading, if this bit is "0", this indicates that no activation is input during A/D conversion. If this bit is "1", this indicates that activation has been input during A/D conversion.
- At writing, this bit is cleared with "0". Writing "1" causes no change and no effect on other operations.

Note:

"1" is read upon reading of the read modify write (RMW) instruction.
If software clear ("0" writing) and hardware set occur at the same time, the hardware set is given priority.

[bit14] SERR: 4-channel A/D sampling time insufficiency interrupt request bit

SERR	Function	
	During Read Operation	During Write Operation
0	Satisfy the sampling time.	Clear this bit.
1	Do not satisfy the sampling time.	No change, no effect on operation

- This bit indicates the status of the sampling time used at A/D conversion activation for the sampling time set by the 4-channel A/D mode setting register (AD4MD:ST1, ST0).
- If this bit is "0", this indicates that the sampling time is satisfied. If it is "1", this indicates that the sampling time is not satisfied.

- An interrupt request is generated when this bit and the interrupt request enable bit (AD4CS:SERRE) are "1".
- At writing, this bit is cleared with "0". Writing "1" causes no change and has no effect on the operation.

Note:

"1" is read upon reading of the read modify write (RMW) instruction.
If software clear ("0" writing) and hardware set occur at the same time, the hardware set is given priority.

[bit13 to bit8] Reserved

[bit7] IRRE: Irregular activation interrupt request enable bit

IRRE	Function
0	Disable interrupt request output.
1	Enable interrupt request output.

- This bit enables or disables the output of an interrupt to the CPU.
- An interrupt request is generated when this bit and the interrupt request flag bit (AD4CS:IRR) are "1".

[bit6] SERRE: 4-channel A/D sampling time insufficiency interrupt request bit

SERRE	Function
0	Disable interrupt request output.
1	Enable interrupt request output.

- This bit enables or disables the output of an interrupt to the CPU.
- An interrupt request is generated when this bit and the interrupt request flag bit (AD4CS:SERR) are "1".

[bit5 to bit0] Reserved

46.4.2.2 4-channel A/D Mode Setting Register: AD4MD

This section shows the bit configuration of the 4-channel A/D mode setting register.

The 4-channel A/D mode setting register (AD4MD) provides a function that sets the A/D conversion compare time and sampling time monitoring value.

AD4MD: Address 335F$_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				CT1	CT0	ST1	ST0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved

[bit3, bit2] CT1, CT0: Compare time setting bits

CT1	CT0	Function
0	0	28 on-chip bus clock cycle (A/D clock output: On-chip bus clock 2 divisions)
0	1	42 on-chip bus clock cycle (A/D clock output: On-chip bus clock 3 divisions)
1	0	56 on-chip bus clock cycle (A/D clock output: On-chip bus clock 4 divisions)
1	1	112 on-chip bus clock cycle (A/D clock output: On-chip bus clock 8 divisions)

These bits select the compare time for A/D conversion.
After an activation trigger is input, the conversion result data is determined after the time set in this bit has elapsed.

Note:

Set the compare time to 350 ns or more. If the sampling time is less than 350 ns, a normal analog conversion value may not be obtained.
Set bit rewriting only when AD4EN.ADEN before the conversion operation is "0".

Set the A/D clock output so that it is not less than 10 MHz.

[bit1, bit0] ST1, ST0: Sampling time monitoring setting bit

ST1	ST0	Function
0	0	Activation trigger monitoring within the 28 on-chip bus clock cycle
0	1	Activation trigger monitoring within the 32 on-chip bus clock cycle
1	0	Activation trigger monitoring within the 14 on-chip bus clock cycle
1	1	Activation trigger monitoring within the 16 on-chip bus clock cycle

- These bits set the sampling monitoring time for A/D conversion.
- Once A/D is activated, if the activation trigger is input within the time set by these bits, a flag is set.

Note:

Set the sampling time to 350 ns or more. If the sampling time is less than 350 ns, it may not be possible to obtain a normal analog conversion value.
Set bit rewriting only when AD4EN.ADEN before the conversion operation is "0".

46.4.2.3 4-channel Data Protection Status Flag Register: AD4PRTF

This section shows the bit configuration of the data protection status flag register.

The data protection status flag register (ADPRTF) displays the A/D data register protection status for each activation channel.

AD4PRTF: Address 3360H (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				PRTF11	PRTF10	PRTF9	PRTF8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX

[bit31 to bit4] Reserved

[bit3 to bit0] PRTF11 to PRTF8: Data protection status flag bits

PRTF	Description
0	Indicate that data is not protected.
1	Indicate that data is protected.

- These bits show the data protection status for the A/D data register of each activation channel.
- The write operation does not influence the data protection status.

46.5 Explanation of Operation

This section explains the operation.

46.5.1 A/D Activation Control Interrupt

This section shows the A/D activation control interrupt control bits and interrupt factors.

46.5.1.1 A/D Conversion End Interrupt

This section explains the A/D conversion end interrupt.

Table 46-4. A/D Conversion End Interrupt Control Bits and Interrupt Factors

	A/D Conversion End Interrupt
Interrupt Request Flag Bit	INT:bit7 in the 4-channel A/D interrupt/protection control register (AD4PTC)
Interrupt Request Enable Bit	INTE:bit6 in the 4-channel A/D interrupt/protection control register (AD4PTC)
Interrupt Factor	Writing of the 4-channel A/D conversion result to the A/D data register

An A/D conversion end interrupt request can be generated when A/D conversion ends. Moreover, an A/D conversion end interrupt can be controlled for each channel (AN8 to AN11).
When the A/D conversion result is set in the A/D data register (ADT4CD), the INT bit in the 4-channel A/D interrupt/protection control register (AD4PTC) is set to "1". At this time, if the interrupt request is enabled (AD4PTC:INTE = 1), the interrupt request is output to the interrupt controller.

46.5.1.2 Sampling Time Insufficiency Interrupt

This section explains the sampling time insufficiency interrupt.

At A/D conversion, if an activation request is input and operation switches to the compare operation without allowing sufficient sampling time, the sampling time insufficiency interrupt is output.

The sampling monitoring time is set with the sampling time monitoring setting bits AD4MD.ST1, ST0 in the 4-channel A/D mode register.

Table 46-5. ST1, ST0: Sampling Time Monitoring Setting Bits

ST1	ST0	Function
0	0	Activation trigger monitoring within the 28 on-chip bus clock cycle
0	1	Activation trigger monitoring within the 32 on-chip bus clock cycle
1	0	Activation trigger monitoring within the 14 on-chip bus clock cycle
1	1	Activation trigger monitoring within the 16 on-chip bus clock cycle

- These bits set the sampling monitoring time for A/D conversion.
- Once A/D is activated, if the activation trigger is input within the time set by these bits, the sampling time insufficiency interrupt flag is set.
- If no data protection/update setting is specified, conversion data is stored, regardless of the setting of the sampling time insufficiency interrupt flag.

Table 46-6. Sampling Time Insufficiency Interrupt Control Bits and Interrupt Factors

	Sampling Time Insufficiency Interrupt
Interrupt Request Flag Bit	SERR:bit14 in the 4-channel A/D control status register (AD4CS)
Interrupt Request Enable Bit	SERRE:bit6 in the 4-channel A/D control status register (AD4CS)
Interrupt Factor	Detection of sampling time insufficiency

At A/D conversion, if the activation request is input and the operation switches to the compare operation without waiting for the specified sampling time, the SERR bit in the 4-channel A/D control status register (AD4CS) is set to "1". At this time, if the interrupt request is enabled (AD4PTCS:SERRE = 1), the interrupt request is output to the interrupt controller.

A/D conversion continues regardless of whether the sampling time insufficiency interrupt is generated.

46.5.1.3 Irregular Activation Interrupt

This section explains the irregular activation interrupt.

During A/D conversion, if a new activation trigger is input before A/D completes the compare operation, the irregular activation interrupt is generated.

Table 46-7. Irregular Activation Interrupt Control Bits and Interrupt Factors

	Irregular Activation Interrupt
Interrupt Request Flag Bit	IRR:bit15 in the 4-channel A/D control status register (AD4CS)
Interrupt Request Enable Bit	IRRE:bit7 in the 4-channel A/D control status register (AD4CS)
Interrupt Factor	Input of a new activation trigger during A/D conversion

During A/D conversion, if a new activation trigger is input before A/D completes the compare operation, the IRR bit in the 4-channel A/D control status register (AD4CS) is set to "1". At this time, if the interrupt request is enabled (AD4PTCS:IRRE = 1), the interrupt request is output to the interrupt controller.

A/D conversion continues regardless of whether the irregular activation interrupt is generated.

46.5.2 A/D Activation Control Operation

This section explains the A/D activation control operation.

A/D activation can be requested by software, an external trigger, the reload timer, or a trigger from PWM.

46.5.2.1 A/D Activation

This section explains A/D activation.

An activation request is issued to the 12-bit A/D converter (4-channel simultaneous sampling). The activation request signal is generated for the A/D activation arbitration by software, an external trigger (falling), the reload timer (rising), or a trigger from PWM. The A/D activation request signal includes two types: "software activation request" and "external trigger/reload timer/PWM activation request". Either of these is activated.
The activation request is cleared when A/D conversion ends. Conversion data is stored into the A/D data register. At this time, an interrupt can be generated.
However, even if the activation factor is received while activation is being requested, the activation request is not reactivated.

The following table shows the assignment of the pin assigned to this A/D converter.

Table 46-8. Assigned to 12-bit A/DC (4-channel Simultaneous Sampling)

	12-bit A/DC (4-channel Simultaneous Sampling)
MB91F552	AN8, AN9, AN10, AN11

46.5.2.2 A/D Activation Enabling

This section explains the enabling of A/D activation.

An A/D activation factor is selected with the A/D activation factor selection bit (AD4TCS.STS[1:0]) and A/D activation factor extension selection bit (AD4TECS.STSA[3:0]). Any one of software, an external trigger, the reload timer, or a trigger from PWM is selected. If the selected activation factor is generated, the activation request signal is generated for the A/D activation arbitration.
The A/D activation request can be disabled as follows: select software activation for the activation channel that does not activate A/D (AD4TCS.STS1, STS0 = "00_B") and disable software activation for the relevant channel of the 4-channel A/D software activation channel selection register (AD4TSE).

46.5.2.3 Software Activation

This section explains software activation.

Set the A/D activation factor selection bit to software activation. (AD4TCS.STS1, STS0 = "00_B")
The software activation request signal is set as follows: Set activation enable for the channel to be subject to software activation in the 4-channel A/D software activation channel selection register (AD4TSE). Then, write "1" into the START bit in the 4-channel A/D software activation register (AD4TSS).

Here, if the A/D software activation channel selection register (AD4TSE) is set for multiple channels, the channel with the smaller number is given priority, causing an irregular activation trigger interrupt.

46.5.2.4 External Trigger Activation

This section explains external trigger activation.

Set the A/D activation factor selection bit to external trigger activation (AD4TCS.STS1, STS0 = "01_B"). If falling of the external trigger (external pin ADTG1) is detected, the activation request signal for the external trigger is set.

46.5.2.5 Reload Timer Activation

This section explains reload timer activation.

Set the A/D activation factor selection bit to reload timer activation. (AD4TCS.STS1, STS0 = "10_B")
If the reload timer is input to the 12-bit A/D converter (4-channel simultaneous sampling) unit and reload timer rising is detected, the reload timer activation request signal is set.

46.5.2.6 PWM Activation

This section explains the PWM activation.

Set the A/D activation factor selection bit to PWM activation. (AD4TCS.STS1, STS0 = "11_B")
The PWM activation trigger switches the activation factor with AD4TECS.STSA[3:0]. The PWM activation factor trigger is input to the 12-bit A/D converter (4-channel simultaneous sampling). If rising of the PWM signal is detected, the PWM activation request signal is set.

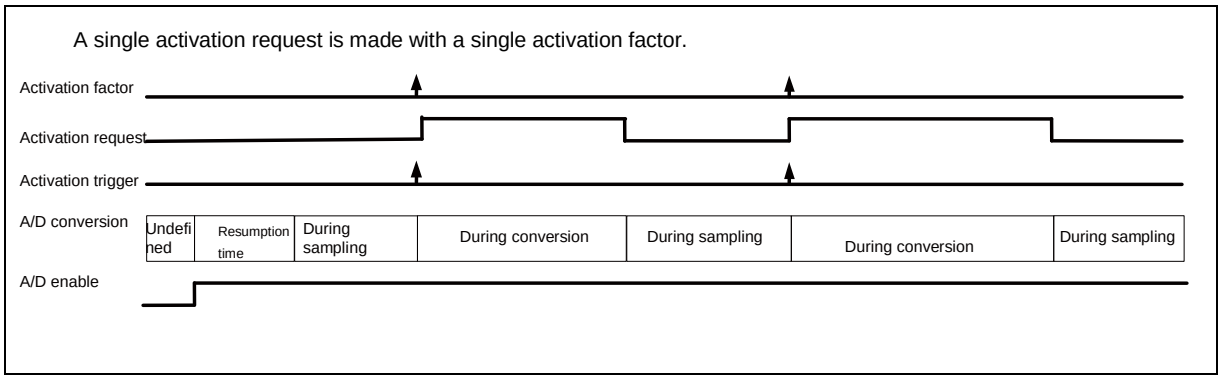
46.5.2.7 Activation Request Mode

This chapter explains activation request mode.

Activation request mode can be set for each activation channel.

- A single activation request is made with a single activation factor. A/D conversion is performed once, and the activation request is released by the end of A/D conversion.
- A/D enters the sampling status when A/D conversion ends.

Figure 46-5. Activation Request Mode



* The fixed time from A/D enable H is required as the resumption time.

46.5.2.8 A/D Conversion Data

This section explains the A/D conversion data.

The A/D conversion result data is stored into the A/D data bit (AD4TCD.D11 to D0) for each activation channel.
Moreover, suppose that the data protection function is disabled (AD4PTC.PRT = "0"). The status of the A/D conversion data stored in the A/D data bit (AD4TCD.D11 to D0) can be confirmed by the conversion data error flag bit (AD4TCD.ERR) and the conversion data error status bit (AD4TCD.ERRST).
When the data protection function is enabled (AD4PTC.PRT = "1"), the conversion data error flag bit (AD4TCD.ERR) and the conversion data error status bit (AD4TCD.ERRST) are fixed to "0".

Table 46-9. Confirmation of A/D Conversion Data Status (When Data Protection Function Is Disabled)

AD4TCD:ERR	AD4TCD:ERRST	A/D Conversion Data Status
0	0	Latest data (not yet read)
0	1	- *1
1	0	Old data (already read)*2
1	1	Latest data/overwriting generated (not yet read)*3

Note:

AD4TCD:ERR and AD4TCD:ERRST are not error flags. These bits indicate the status of the conversion data when the data protection function is disabled (AD4PTC.PRT = "0").

*1: This status does not exist.

*2: Initial value

*3: Data that has not yet been read is discarded.

46.5.2.9 Protection Function

This section explains the protection function.

Each A/D data register can set the data protection function. The protection function is set by the PRT bit in the 4-channel A/D interrupt/protection control register (AD4PTC).

When the protection function is enabled, the data protection status is entered if the conversion result is stored in the A/D data register. The condition for releasing the data protection status can be selected by the PRTS bit in the AD4PTC register.

- If the PRTS bit is "0", no overwriting occurs as a result of storing new data until the A/D data register data is read and the interrupt flag is cleared. However, data reading and interrupt flag clear can be in any order.
- If the PRTS bit is "1", no overwriting occurs as a result of storing new data until the A/D data register data is read.
- Any channel subject to data protection is not overwritten as a result of storing new data, even if the next activation factor is generated.

46.5.2.10 Forced Termination of Activation Request

This section explains the forced termination of the activation request.

Whether an A/D activation request or A/D conversion is in progress can be reported by the bit that indicates that A/D activation is being requested. Moreover, to forcibly terminate the present A/D activation request or conversion, write "0" into the BUSY7 to BUSY0 bits in the 4-channel A/D activation trigger control status register (AD4TBUSY).

Even if "0" is written to any bit of BUSY7 to BUSY0, the A/D activation request or conversion is forcibly terminated.

46.5.3 Operation of A/D Activation Arbitration

This section explains the operation of the A/D activation arbitration.

An A/D activation trigger is generated by arbitrating an A/D activation request issued from the A/D activation control section.

46.5.3.1 A/D Activation Trigger Arbitration

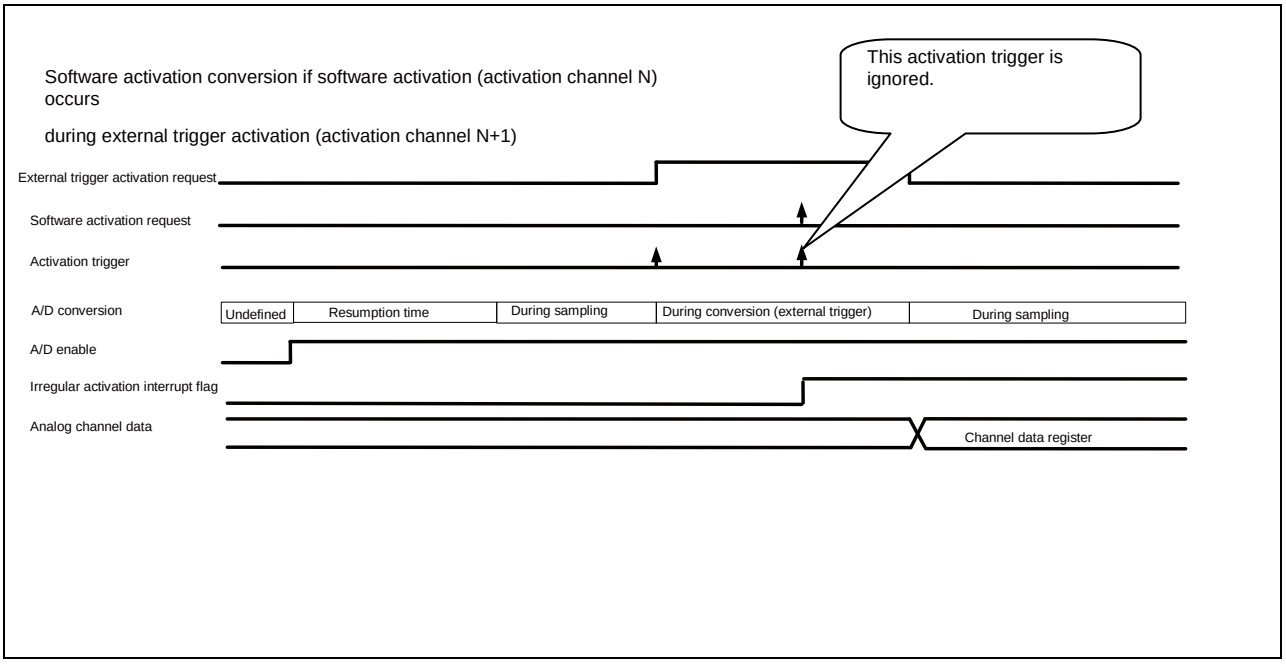
This section explains A/D activation trigger arbitration.

An A/D activation trigger is generated by selecting one activation request from among the activation requests of the A/D activation control section. A respective A/D activation trigger signal is generated by entering a software activation request and external trigger/reload timer activation request/PWM activation from the A/D activation control. While A/D conversion is being performed as a result of the activation trigger that was input first, other triggers are ignored even if they are input.

It is judged that an unconvertible activation request has been input and the irregular activation interrupt flag is set.

A/D conversion is not performed by other activation triggers that are input during A/D conversion, regardless of the activation trigger type.

Figure 46-6. Activation Arbitration



* The fixed time from A/D enable H is required as the resumption time.

46.5.3.2 A/D Conversion Cancel Function

This section explains the A/D conversion cancel function.

During A/D conversion, if the activation request from the request source is deactivated, an A/D conversion cancel signal is generated to forcibly terminate the current conversion processing.

46.5.4 Operation of 12-bit A/D Converter (4-channel Simultaneous Sampling)

This section explains the operation of the 12-bit A/D converter (4-channel simultaneous sampling).

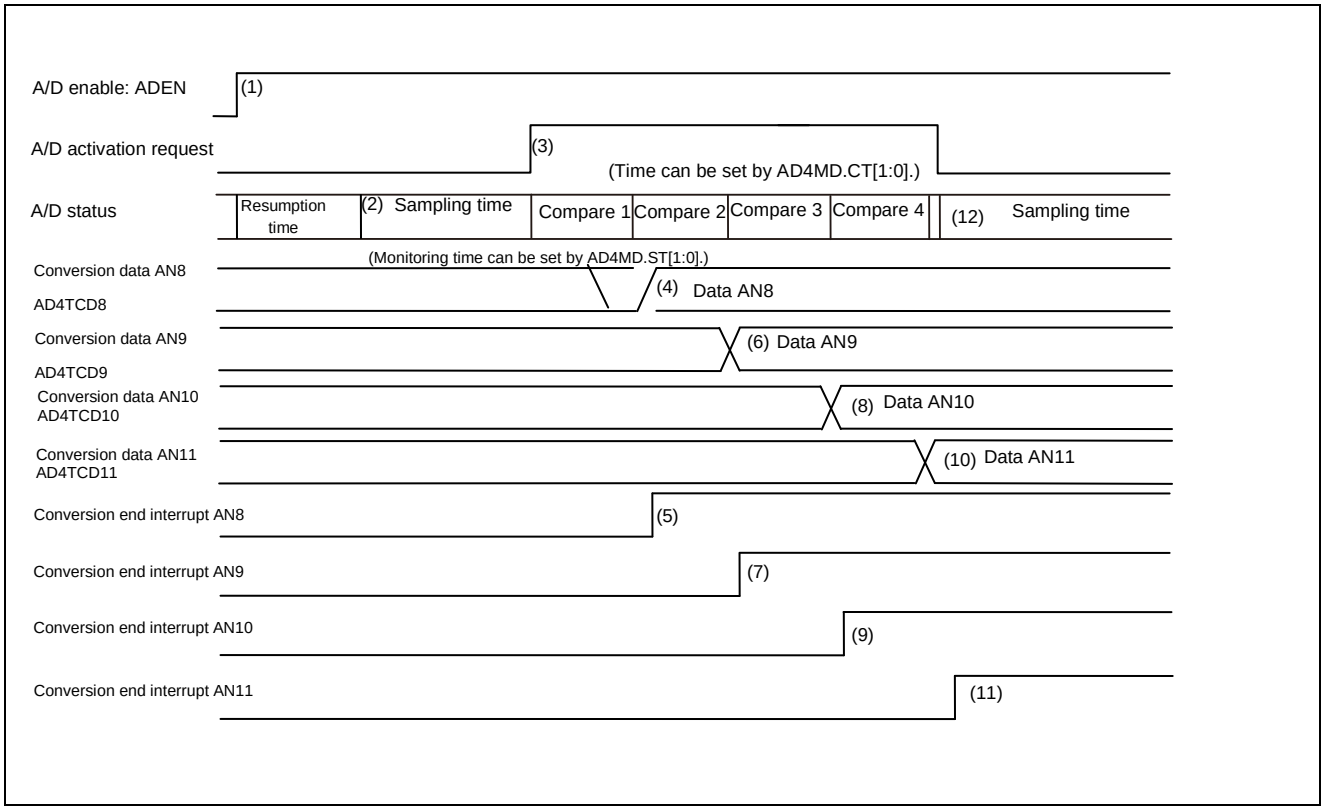
The 12-bit A/D converter (4-channel simultaneous sampling) controls A/D conversion.

46.5.4.1 Operation Timing

This section explains the operation timing.

<Operation when all 4 channels are used for conversion>

Figure 46-7. Operation Timing of 12-bit A/D Converter (4-channel Simultaneous Sampling) (1)

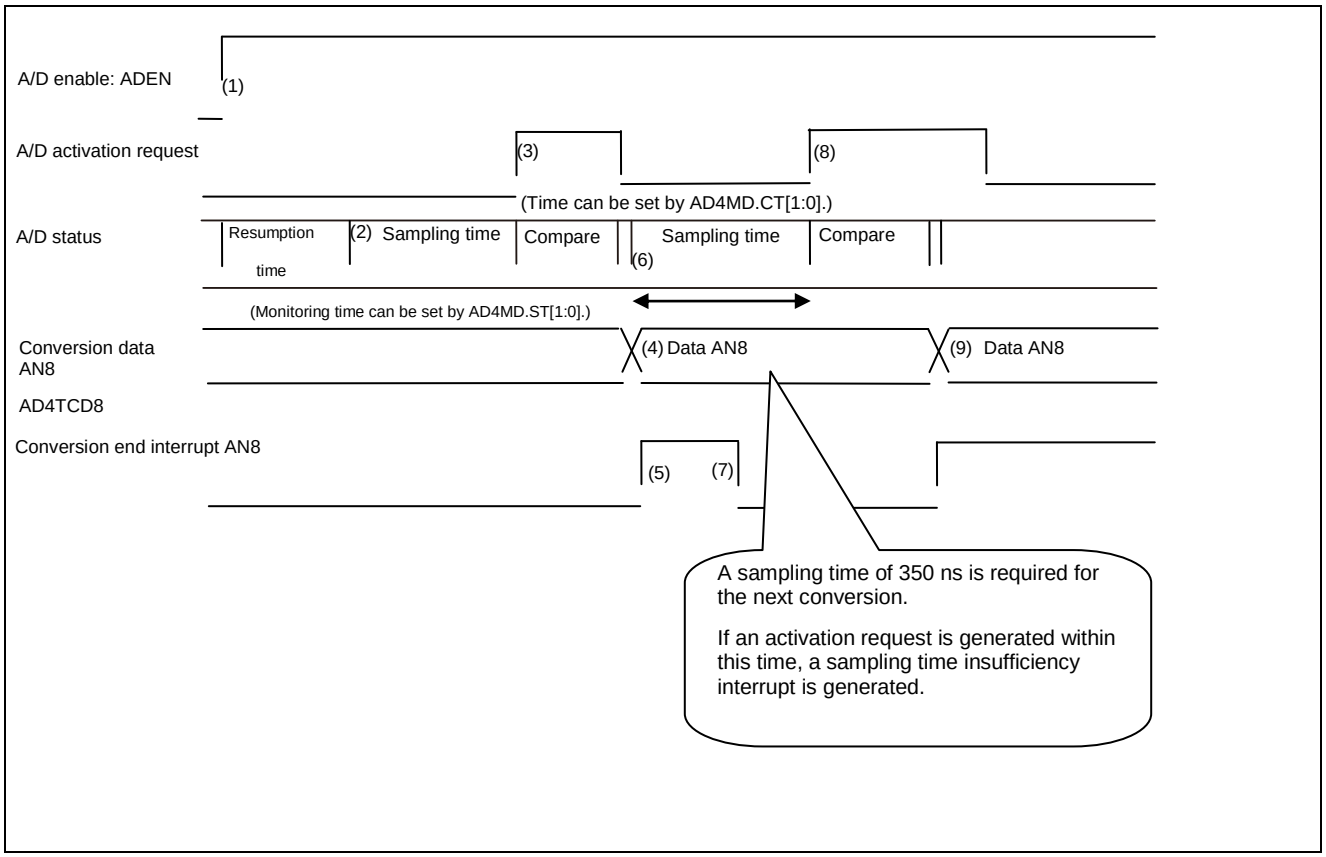


1. (With A/D enable (AD4EN.ADEN):H, A/D is activated to start sampling.
2. Sampling starts.
3. Compare operation starts after activation request input.
4. AN8 conversion data is stored from 4-channel simultaneous sampling.
5. An AN8 conversion end interrupt is output.
6. AN9 conversion data is stored from the 4-channel simultaneous sampling.
7. An AN9 conversion end interrupt is output.
8. AN10 conversion data is stored from the 4-channel simultaneous sampling.
9. An AN10 conversion end interrupt is output.
10. AN11 conversion data is stored from the 4-channel simultaneous sampling.
11. An AN11 conversion end interrupt is output.
12. A/D conversion (sampling) starts at the same time as AN11 conversion ends.

* The fixed time from A/D enable H is required as the resumption time.

Figure 46-8. Operation Timing of 12-bit A/D Converter (4-channel Simultaneous Sampling) (2)

<Two A/D conversions are continuously performed with AN8 input.>

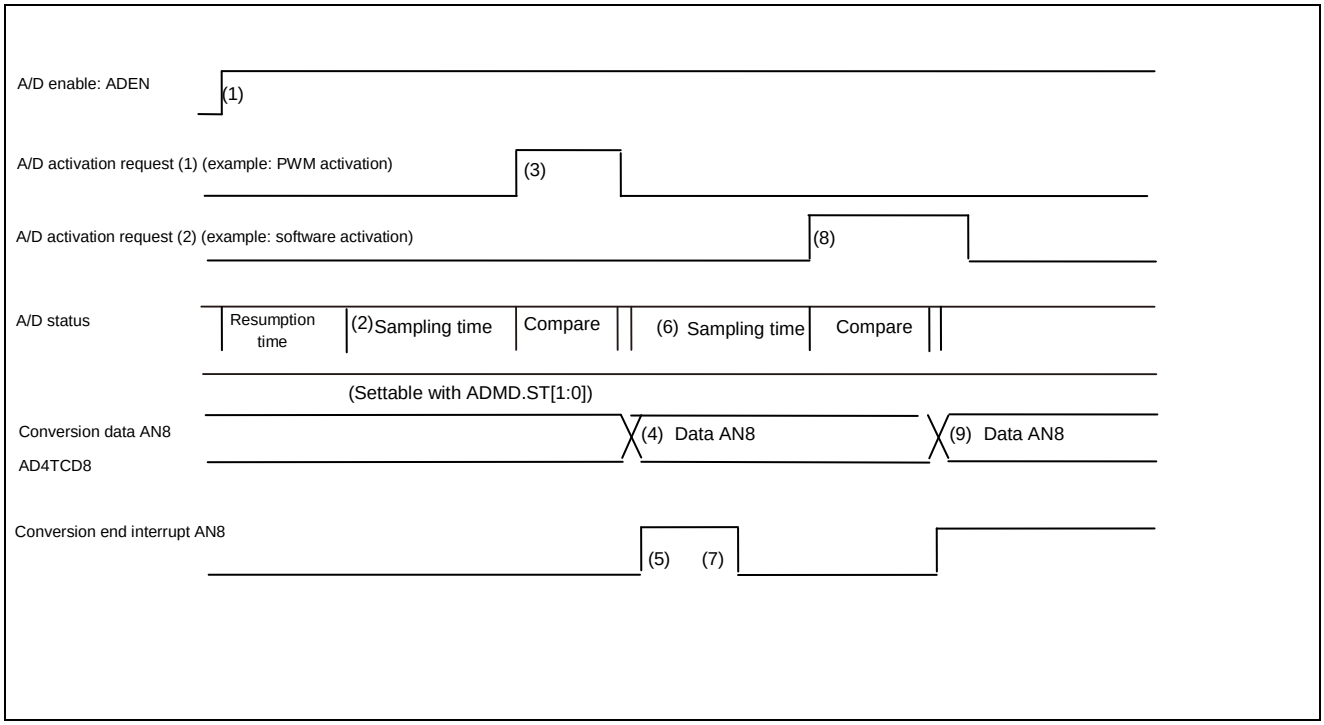


1. With A/D enable (AD4EN.ADEN):H, A/D is activated to start sampling.
2. A/D conversion starts.
3. Compare operation starts after activation trigger input.
4. AN8 conversion data is stored from 4-channel simultaneous sampling.
5. An AN8 conversion interrupt is output.
6. The next A/D conversion (sampling) starts.
7. An interrupt is cleared from software.
8. Compare operation starts after activation request input. (2nd time)
9. AN8 conversion data is stored from 4-channel simultaneous sampling.

- 350 ns is required as the sampling time between (6) and (8).
  If an activation request is generated within this time, an error flag is set.
- The fixed time from A/D enable H is required as the resumption time.

Figure 46-9. Operation Timing of 12-bit A/D Converter (4-channel Simultaneous Sampling) (3)

<When multiple activation requests are generated and data is stored in the same AN8>



1. With A/D enable (AD4EN.ADEN):H, A/D is activated to start sampling.
2. A/D conversion (sampling) starts.
3. Compare operation starts after activation request (1) input.
4. AN8 conversion data is stored from 4-channel simultaneous sampling.
5. An AN8 conversion end interrupt is output.
6. The next A/D conversion (sampling) starts.
7. An interrupt is cleared by software.
8. Compare operation starts after activation request (2) input. (2nd time)
9. AN8 conversion data is stored from 4-channel simultaneous sampling.

- A/D conversion of the same input channel is possible with multiple activation triggers.
- The fixed time from A/D enable H is required as the resumption time.

46.5.4.2 Activation Factors

This section explains the activation factors.

The A/D conversion activation factors include software activation, external trigger activation, reload timer activation, and PWM activation. Each of these is selected with AD4TECS.STSA[3:0] and AD4TCS.STS[1:0].

46.5.4.3 A/D Conversion

This section explains A\D conversion.

For A/D conversion, single conversion is performed by the input of a single activation trigger.

46.5.4.4 Activation Trigger Signal Input during A/D Conversion

This section explains the input of the activation trigger signal during A/D conversion.

If an activation trigger signal is input during A/D conversion, it is ignored and the fact that the activation signal has been input is output as the interrupt flag for irregular activation input.

46.5.4.5 Canceling A/D Conversion

This section explains the cancellation of A/D conversion.

The current conversion stops if the A/D conversion cancel signal is received during A/D conversion.

46.5.4.6 A/D Conversion Time

This section explains the A/D conversion time.

The A/D conversion time is obtained by adding the sampling time to the compare time.

Setting the sampling monitoring time

The sampling monitoring time setting can be used for the following settings.

Table 46-10. Sampling Monitoring Time Setting for the On-chip Bus Clock Frequency

ST1	ST0	Function	Sampling Time (On-chip Bus Clock Frequency/2)	
			(40 MHz)	(20 MHz)
0	0	28 on-chip bus clock cycle	350 ns	Setting prohibited
0	1	32 on-chip bus clock cycle	400 ns	Setting prohibited
1	0	14 on-chip bus clock cycle	Setting prohibited	350 ns
1	1	16 on-chip bus clock cycle	Setting prohibited	400 ns

Note:

Set the sampling time to 350 ns or more. If the sampling time is less than 350 ns, it may not be possible to obtain a normal analog conversion value.
Rewrite these bits while A/D operation is stopped before conversion operation.

Compare time

The compare time is set with the compare time setting bits (AD4MD.CT1, CT0).

Table 46-11. Compare Time for the On-chip Bus Clock Frequency

CT1	CT0	Function	Compare Time for Each Channel (On-chip Bus Clock Frequency)		
			(80 MHz)	(40 MHz)	(20 MHz)
0	0	28 on-chip bus clock cycle	350 ns	700 ns	1400 ns
0	1	42 on-chip bus clock cycle	525 ns	1050 ns	Setting prohibited
1	0	56 on-chip bus clock cycle	700 ns	1400 ns	Setting prohibited
1	1	112 on-chip bus clock cycle	1400 ns	Setting prohibited	Setting prohibited

Note:

Set the compare time to 350 ns or more. If the sampling time is less than 350 ns, a normal analog conversion value may not be obtained.
Rewrite these bits while A/D operation is stopped before conversion operation.

46.5.4.7 A/D Conversion End and A/D Data Retrieval

This section explains the A/D conversion end and A/D data retrieval.

When A/D conversion ends normally (the specified number of cycles have passed), the received conversion data is retrieved and output. At this time, the A/D conversion completion signal is generated.

46.5.4.8 Power Down

This section explains power down.

When standby mode is set, the 12-bit A/D converter (4-channel simultaneous sampling) is powered down.

When ADEN = "0" is set, the converter is powered down in the same way.

Upon restoration from the power down status, the setting value is retained and the operation status is cleared.

46.6 Notes

This section explains the notes.

Notes on using A/D activation compare

- Setting A/D data register protection

  Set the PRT and PRTS bits before running A/D conversion. Do not change the setting of these bits while A/D conversion is in progress and the A/D data register is protected.
  To release the A/D data register protection function, perform the protection release operation set in the PRTS bit after A/D conversion stops, or disable the protection function with the PRT bit.
  Suppose that the PRTS bit has been changed while the A/D data register is protected. To release the A/D data register protection, perform the following protection release operations set in the PRTS bit after the PRTS bit has been changed: A/D data register reading and clear operation by writing "0" to the interrupt request flag bit. For example, suppose that PRT = 1 and PRTS = 1 are set, the A/D data register is protected, and PRTS = 0 is set after the interrupt request flag bit is cleared. To release the protection, the following operations are required: A/D data register reading and clear operation by writing "0" to the interrupt request flag bit.

- Setting a sampling monitoring time and compare time

  The required sampling time is 350 ns or more. If the sampling time is less than 350 ns, a normal analog conversion value may not be obtained.
  Set the CT1 and CT0 bits so that the compare time is 350 ns or more for each channel. If the A/D output clock is less than 10 MHz, a normal analog conversion value may not be obtained.

- Setting the AD4MD register

  Rewrite the A/D mode setting register (AD4MD) bits while AD4EN.ADEN = 0 is set.

- Resumption time

  The resumption time is required before the A/D converter starts sampling after the A/D enable (AD4EN.ADEN) is set to H. For details, see the data sheet.

- Power consumption control

  Before changing the operation mode to "bus sleep mode" or "standby mode (watch mode, stop mode)" of the power consumption control, set the 12-bit A/D converter (4-channel simultaneous sampling) to AD4EN.ADEN = 0.

- On-chip bus clock

  Note that if the on-chip bus clock is not provided, the 12-bit A/D converter (4-channel simultaneous sampling) module does not perform register access, DMA transfer, or interrupt processing.

- DMAC transfer

  To perform interrupt clear by DMAC transfer, set interrupt enable for the last channel (set by AD4TECS.CHSEL1, CHSEL0) in which the conversion data is stored.

Use example: For AD4TECS.CHSEL1, 0 = 2'b10 (end for AN10 conversion), set AD4PTC8.INTE, AD4PTC9.INTE, and AD4PTC11.INTE to 0 and ADPTC11.INTE to 1.

  During DMAC transfer, use the AD4PTC.PRT, PRTS protection setting for the PRT = 1 and PRTS = 0 setting (for releasing protection by data reading and interrupt flag clear).

47. Comparator

This chapter explains the comparator.

47.1 Overview
47.2 Features
47.3 Configuration
47.4 Registers
47.5 Explanation of Operation
47.6 Notes

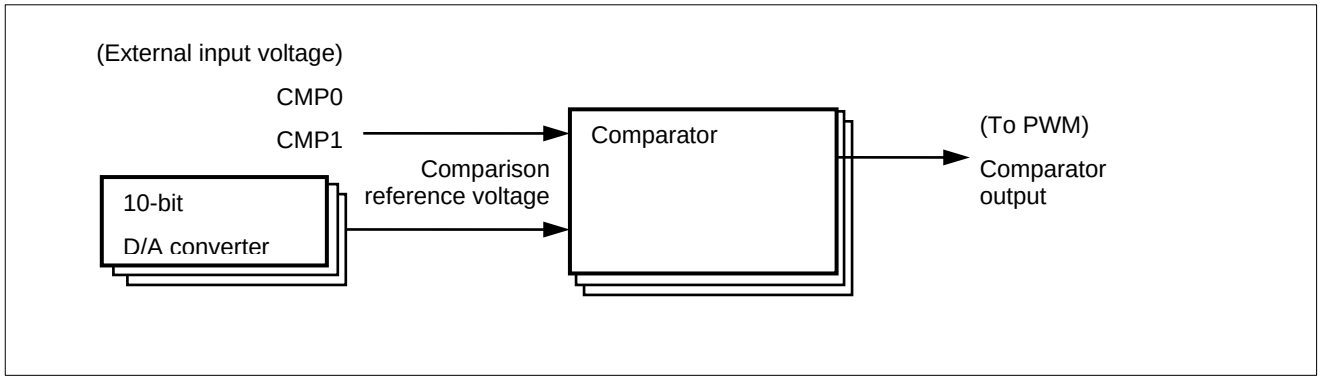
47.1 Overview

This section explains an overview of the comparator.

The comparator circuit compares the input voltage entered from an external pin with the comparison reference voltage. According to the comparison results, it outputs "L" or "H".

The comparison reference voltage is set with the 10-bit D/A converter.

Figure 47-1. Block Diagram (Overview)



47.2 Features

This section explains the features of the comparator.

Number of channels

The comparator has 3 channels.

Comparison reference voltage

- The comparison reference voltage is generated with the 10-bit D/A converter.
- The 10-bit D/A comparator has 3 channels. The comparison reference voltage can be set for each channel of the comparator.

Comparison input voltage

- This voltage is obtained by adding the slope compensation voltage to the channel 0 = external input pin (CMP0).

  (You can also specify a setting to which no slope compensation is added. For details, see "Chapter: Slope Compensation.")
- Channel 1 = voltage of the external input pin (CMP1)
- Channel 2 = voltage of the external input pin (CMP2)

Operating frequency

The comparator runs with the division ratio set with the clock division register (CMPDIV) and with the frequency obtained by dividing the PWM clock.

If the clock division register sets division by 4 and the PWM clock is 200 MHz, the comparator runs at 50 MHz.

Output polarity invert

The polarity of the comparator output signal can be inverted for each channel according to the setting of the comparator control register (CMPCTL).

Noise removal

The noise of the comparator output can be removed by the glitch filter.

Interrupt request

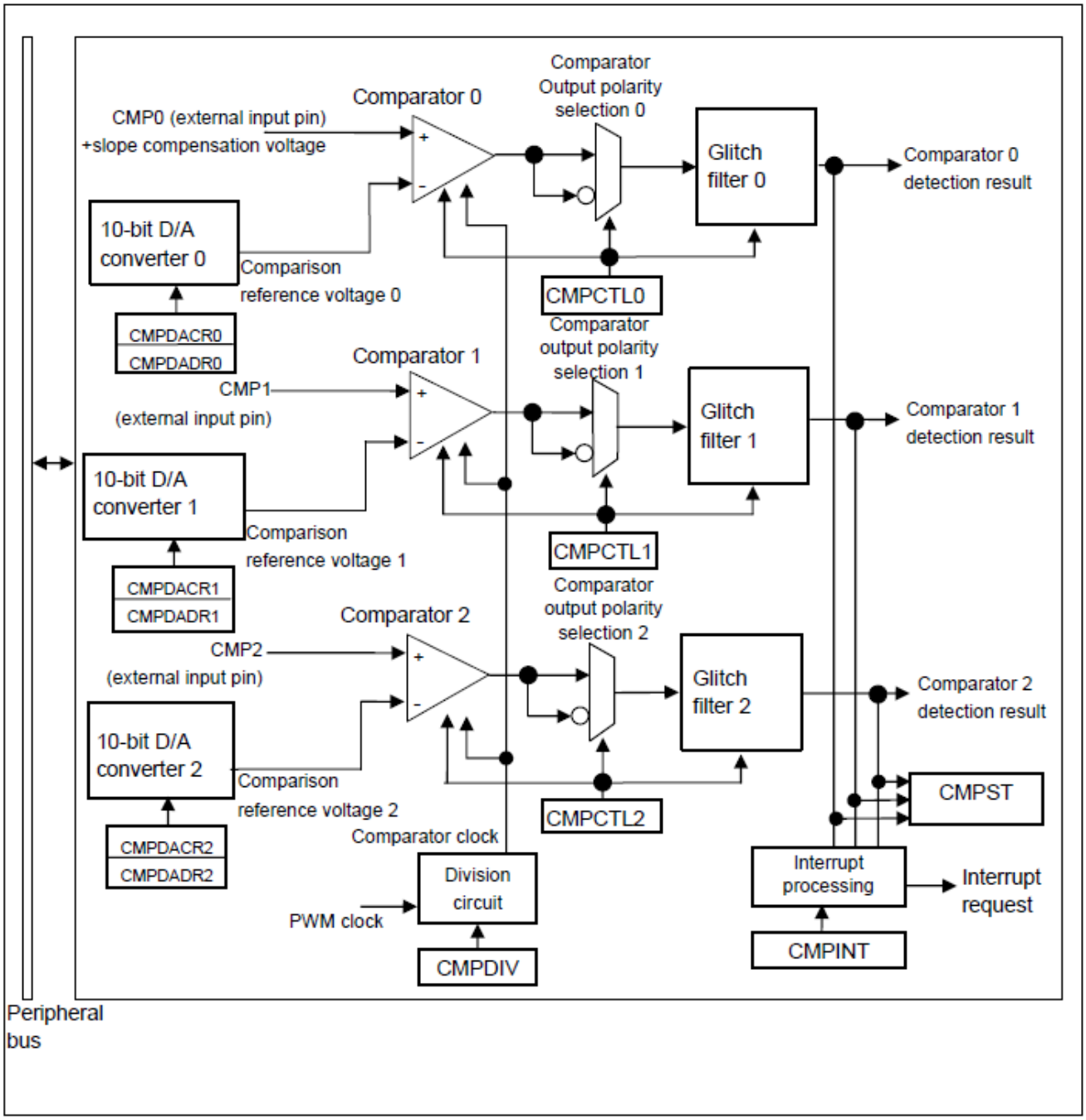
An interrupt request can be generated when the output of the comparator n detection results are as follows: (n=0,1, 2)

The output of the comparator n detection result has changed from "L" to "H".
The output of the comparator n detection result has changed from "H" to "L".

47.3 Configuration

This section explains the configuration of the comparator.

Figure 47-2. Configuration Diagram



47.4 Registers

This section explains the comparator registers.

Table 47-1. Register Map

Address	Registers				Register functions
	+0	+1	+2	+3	
0x05E0	CMPCTL0	CMPCTL1	CMPCTL2	CMPDIV	Comparator 0 control register (CMPCTL0) Comparator 1 control register (CMPCTL1) Comparator 2 control register (CMPCTL2) Clock division setting register (CMPDIV)
0x05E4	CMPDACR0	Reserved	CMPDADR0		Comparator 0 D/A converter control register (CMPDACR0) Comparator 0 D/A converter output setting register (CMPDADR0)
0x05E8	CMPDACR1	Reserved	CMPDADR1		Comparator 1 D/A converter control register (CMPDACR1) Comparator 1 D/A converter output setting register (CMPDADR1)
0x05EC	CMPDACR2	Reserved	CMPDADR2		Comparator 2 D/A converter control register (CMPDACR2) Comparator 2 D/A converter output setting register (CMPDADR2)
0x05F0	CMPINT	Reserved	Reserved	Reserved	Comparator output interrupt flag register (CMPINT)
0x05F4	CMPST	Reserved	Reserved	Reserved	Comparator output status register (CMPST)

47.4.1 Comparator Control Register: CMPCTL

This section shows the bit configuration of the comparator register.

This register controls the comparator operation.

CMPCTL0: Address $05E0_H$ (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved		GF02	GF01	GF00	CMPIE0	CMPPOL0	CMPEN0
Initial Value	0	0	0	0	1	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

CMPCTL1: Address $05E1_H$ (access: byte, half word, and word)

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved		GF12	GF11	GF10	CMPIE1	CMPPOL1	CMPEN1
Initial Value	0	0	0	0	1	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

CMPCTL2: Address $05E2_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved		GF22	GF21	GF20	CMPIE2	CMPPOL2	CMPEN2
Initial Value	0	0	0	0	1	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

CMPDIV: Address $05E3_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						DIV1	DIV0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit31 to bit30] (Reserved)

[bit29 to bit27] GF02 to GF00: Glitch filter coefficient setting bits for comparator 0

GF02 to GF00	Function
000	Less than the comparator clock 1 cycle (through)
001	Less than the comparator clock 2 cycle
010	Less than the comparator clock 3 cycle
011	Less than the comparator clock 4 cycle
100	Less than the comparator clock 5 cycle
101	Less than the comparator clock 6 cycle
110	Less than the comparator clock 7 cycle
111	Less than the comparator clock 8 cycle

These bits set the maximum width of the noise to be removed by the glitch filter.

Note:

During comparator operation, do not change the setting of the glitch filter coefficient.

[bit26] CMPIE0: Comparator 0 interrupt request enable bit

CMPIE0	Function
0	Disable
1	Enable

Interrupt request enable is set when the output of the comparator 0 detection results are as follows:

The output of the comparator 0 detection result has changed from "L" to "H" or from "H" to "L".

"0": Disable the interrupt request.

"1": Enable the interrupt request.

[bit25] CMPPOL0: Comparator 0 output polarity selection bit

CMPPOL0	Function
0	Normal polarity
1	Inverse polarity

This bit sets the output polarity of comparator 0.

"0": Indicate that output has normal polarity.

"1": Indicate that output has inverse polarity.

Note:

During comparator operation, do not change the polarity.

If CMPEN0 is "0", the comparator polarity setting is invalid. In this case, "L" is output from the comparator.

[bit24] CMPEN0: Comparator 0 output enable bit

CMPEN0	Function
0	Disable
1	Enable

This bit sets the output enabling of the comparator 0 detection result.

"0": Disable the output of the comparison result.

"1": Enable the output of the comparison result.

Note:

Before writing "1" to CMPEN0, set the output value in the comparator 0 D/A converter setting register. Then, write "1" to the DAE bit to enable the output of the D/A converter.

If "1" is written to CMPEN0 before the D/A converter is set, comparator 0 may output "H" to PWM.

To wait for the comparator to stabilize, the comparator output is fixed to "L" during a comparator clock of 5 to 6 cycles after "1" is written to CMPEN0.

Set a time of 6 cycles of the comparator clock after rewriting CMPEN0 from "1" to "0" until writing "1" to CMPEN0.

[bit23, bit22] (Reserved)

[bit21 to bit19] GF12 to GF10: Glitch filter coefficient setting bits for comparator 1

GF12 to GF10	Function
000	Less than the comparator clock 1 cycle (through)
001	Less than the comparator clock 2 cycle
010	Less than the comparator clock 3 cycle
011	Less than the comparator clock 4 cycle
100	Less than the comparator clock 5 cycle
101	Less than the comparator clock 6 cycle
110	Less than the comparator clock 7 cycle
111	Less than the comparator clock 8 cycle

These bits set the maximum width of the noise to be removed by the glitch filter.

Note:

During comparator operation, do not change the setting of the glitch filter coefficient.

[bit18] CMPIE1: Comparator 1 interrupt request enable bit

CMPIE1	Function
0	Disable
1	Enable

Interrupt request enable is set when the output of the comparator 1 detection results are as follows:

The output of the comparator 1 detection result has changed from "L" to "H" or from "H" to "L".

"0": Disable the interrupt request.

"1": Enable the interrupt request.

[bit17] CMPPOL1: Comparator 1 output polarity selection bit

CMPPOL1	Function
0	Normal polarity
1	Inverse polarity

This bit sets the output polarity of comparator 1.

"0": Indicate that output is has normal polarity.

"1": Indicate that output has inverse polarity.

Note:

During comparator operation, do not change the polarity.

If CMPEN1 is "0", the comparator polarity setting is invalid. In this case, "L" is output from the comparator.

[bit16] CMPEN1: Comparator 1 output enable bit

CMPEN1	Function
0	Disable
1	Enable

This bit sets the output enabling of the comparator 1 detection result.

"0": Disable the output of the comparison result.

"1": Enable the output of the comparison result.

Note:

Before writing "1" to CMPEN1, set the output value in the comparator 1 D/A converter setting register. Then, write "1" to the DAE bit to enable the output of the D/A converter.
If "1" is written to CMPEN1 before the D/A converter is set, comparator 1 may output "H" to PWM.

To wait for the comparator to stabilize, the comparator output becomes "L" during a comparator clock of 5 to 6 cycles after "1" is written to CMPEN1.

Set a time of 6 cycles of the comparator clock after rewriting CMPEN1 from "1" to "0" until writing "1" to CMPEN1.

[bit15, bit14] (Reserved)

[bit13 to bit11] GF22 to GF20: Glitch filter coefficient setting bits for comparator 2

GF22 to GF20	Function
000	Less than the comparator clock 1 cycle (through)
001	Less than the comparator clock 2 cycle
010	Less than the comparator clock 3 cycle
011	Less than the comparator clock 4 cycle
100	Less than the comparator clock 5 cycle
101	Less than the comparator clock 6 cycle
110	Less than the comparator clock 7 cycle
111	Less than the comparator clock 8 cycle

These bits set the maximum width of the noise to be removed by the glitch filter.

Note:

During comparator operation, do not change the setting of the glitch filter coefficient.

[bit10] CMPIE2: Comparator 2 interrupt request enable bit

CMPIE2	Function
0	Disable
1	Enable

Interrupt request enable is set when the output of the comparator 2 detection results are as follows:

The output of the comparator 2 detection result has changed from "L" to "H" or from "H" to "L".

"0": Disable the interrupt request.

"1": Enable the interrupt request.

[bit9] CMPPOL2: Comparator 2 output polarity selection bit

CMPPOL2	Function
0	Normal polarity
1	Inverse polarity

This bit sets the output polarity of comparator 2.

"0": Indicate that output is has normal polarity.

"1": Indicate that output has inverse polarity.

Note:

During comparator operation, do not change the polarity.

If CMPEN2 is "0", the comparator polarity setting is invalid. In this case, "L" is output from the comparator.

[bit8] CMPEN2: Comparator 2 output enable bit

CMPEN2	Function
0	Disable
1	Enable

This bit sets the output enabling of the comparator 2 detection result.

"0": Disable the output of the comparison result.

"1": Enable the output of the comparison result.

Note:

Before writing "1" to CMPEN2, set the output value in the comparator 2 D/A converter setting register. Then, write "1" to the DAE bit to enable the output of the D/A converter.
If "1" is written to CMPEN2 before the D/A converter is set, comparator 2 may output "H" to PWM.

To wait for the comparator to stabilize, the comparator output becomes "L" during a comparator clock of 5 to 6 cycles after "1" is written to CMPEN2.

Set a time of 6 cycles of the comparator clock after rewriting CMPEN2 from "1" to "0" until writing "1" to CMPEN2.

[bit7 to bit2] (reserved)

[bit1, bit0] DIV1, DIV0: Clock division setting bits

DIV1 to DIV0	Function
00	Divided by 4
01	Divided by 2
10	Not divided
11	

These bits select a comparator clock division ratio.

Note:

During comparator operation, do not change the division.

Do not input a comparator clock exceeding a frequency of 50 MHz.

47.4.2 Comparator 0 D/A Converter Control Register: CMPDACR0

This section shows the bit configuration of the comparator 0 D/A converter control register.

This register controls the operation of the D/A converter for comparison reference voltage generation for comparator 0.

CMPDACR0: Address $05E4_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							DAE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit7 to bit1] (Reserved)

[bit0] DAE: Comparison reference voltage output enable bit for comparator 0

DAE	Function
0	Disable
1	Enable

This bit sets the output enabling of the comparison reference voltage for comparator 0.

"0": Disable the output of the D/A converter that sets the comparison reference voltage.

"1": Enable the output of the D/A converter that sets the comparison reference voltage.

Note:

The output stability wait time is required to allow the output of the D/A converter to stabilize after "1" is written to DAE.

47.4.3 Comparator 0 D/A Converter Output Setting Register: CMPDADR0

This section shows the bit configuration of the comparator 0 D/A converter output setting register.

This register controls the operation of the D/A converter for comparison reference voltage generation for comparator 0.

CMPDADR0: Address $05E6_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						DA9	DA8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit10] (Reserved)

[bit9 to bit0] DA9 to DA0: Comparator 0 comparison reference voltage setting bits

These bits set the output value of the D/A converter that generates the comparison reference voltage for comparator 0.

The following shows the relationship between the D/A converter input codes and comparison reference voltage values.

DA9 to DA0	Comparison Reference Voltage Value [V]
00 0000 0000	$0/1024 \times AV_{CC}2$
00 0000 0001	$1/1024 \times AV_{CC}2$
00 0000 0010	$2/1024 \times AV_{CC}2$
...	...
11 1111 1101	$1021/1024 \times AV_{CC}2$
11 1111 1110	$1022/1024 \times AV_{CC}2$
11 1111 1111	$1023/1024 \times AV_{CC}2$

Note:

The maximum value that can be specified for the output voltage of the D/A converter is 1023/1024 × AVCC2. The upper limit of the input operation assurance range for comparator is the supply voltage - 1 V.

The output stability wait time is required to allow the output of the D/A converter to stabilize after a value is written to DA9 to DA0.
To access the comparator 0 D/A converter output setting register, use a half-word or word access instruction.

47.4.4 Comparator 1 D/A Converter Control Register: CMPDACR1

This section shows the bit configuration of the comparator 1 D/A converter control register.

This register controls the operation of the D/A converter for comparison reference voltage generation for comparator 1.

CMPDACR1: Address $05E8_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							DAE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit7 to bit1] (Reserved)

[bit0] DAE: Comparison reference voltage output enable bit for comparator 1

DAE	Function
0	Disable
1	Enable

This bit sets the output enabling of the comparison reference voltage for comparator 1.

"0": Disable the output of the D/A converter that sets the comparison reference voltage.

"1": Enable the output of the D/A converter that sets the comparison reference voltage.

Note:

The output stability wait time is required to allow the output of the D/A converter to stabilize after "1" is written to DAE.

47.4.5 Comparator 1 D/A Converter Output Setting Register: CMPDADR1

This section shows the bit configuration of the comparator 1 D/A converter output setting register.

This register controls the operation of the D/A converter for comparison reference voltage generation for comparator 1.

CMPDADR1: Address $05EA_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						DA9	DA8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit10] (Reserved)

[bit9 to bit0] DA9 to DA0: Comparator 1 comparison reference voltage setting bits

These bits set the output value of the D/A converter that generates the comparison reference voltage for comparator 1.

The following shows the relationship between the D/A converter input codes and comparison reference voltage values.

DA9 to DA0	Comparison Reference Voltage Value [V]
00 0000 0000	$0/1024 \times AV_{CC}2$
00 0000 0001	$1/1024 \times AV_{CC}2$
00 0000 0010	$2/1024 \times AV_{CC}2$
...	...
11 1111 1101	$1021/1024 \times AV_{CC}2$
11 1111 1110	$1022/1024 \times AV_{CC}2$
11 1111 1111	$1023/1024 \times AV_{CC}2$

Note:

The maximum value that can be specified for the output voltage of the D/A converter is 1023/1024 × AVCC2. The upper limit of the input operation assurance range for the comparator is the supply voltage - 1 V.

The output stability wait time is required for the output of the D/A converter to stabilize after a value is written to DA9 to DA0.
To access the comparator 1 D/A converter output setting register, use a half-word or word access instruction.

47.4.6 Comparator 2 D/A Converter Control Register: CMPDACR2

This section shows the bit configuration of the comparator 2 D/A converter control register.

This register controls the operation of the D/A converter for comparison reference voltage generation for comparator 2.

CMPDACR2: Address $05EC_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							DAE
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[bit7 to bit1] (Reserved)

[bit0] DAE: Comparison reference voltage output enable bit for comparator 2

DAE	Function
0	Disable
1	Enable

This bit sets the output enabling of the comparison reference voltage for comparator 2.

"0": Disable the output of the D/A converter that sets the comparison reference voltage.

"1": Enable the output of the D/A converter that sets the comparison reference voltage.

Note:

The output stability wait time is required for the output of the D/A converter to stabilize after "1" is written to DAE.

47.4.7 Comparator 2 D/A Converter Output Setting Register: CMPDADR2

This section shows the bit configuration of the comparator 2 D/A converter output setting register.

This register controls the operation of the D/A converter for comparison reference voltage generation for comparator 2.

CMPDADR2: Address $05EE_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						DA9	DA8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit10] (Reserved)

[bit9 to bit0] DA9 to DA0: Comparator 2 comparison reference voltage setting bits

These bits set the output value of the D/A converter that generates the comparison reference voltage for comparator 2.

The following shows the relationship between the D/A converter input codes and comparison reference voltage values.

DA9 to DA0	Comparison Reference Voltage Value [V]
00 0000 0000	$0/1024 \times AV_{CC}2$
00 0000 0001	$1/1024 \times AV_{CC}2$
00 0000 0010	$2/1024 \times AV_{CC}2$
...	...
11 1111 1101	$1021/1024 \times AV_{CC}2$
11 1111 1110	$1022/1024 \times AV_{CC}2$
11 1111 1111	$1023/1024 \times AV_{CC}2$

Note:

The maximum value that can be specified for the output voltage of the D/A converter is 1023/1024 × AVCC2. The upper limit of the input operation assurance range for comparator is the supply voltage - 1 V.

The output stability wait time is required to enable the output of the D/A converter to stabilize after a value is written to DA9 to DA0.To access the comparator 2 D/A converter output setting register, use a half-word or word access instruction.

47.4.8 Comparator Output Interrupt Flag Register: CMPINT

This section shows the bit configuration of the comparator output interrupt flag register.

This register shows the interrupt flag that depends on comparator output changes. The INTn bit is set to "1" when the comparator n detection results are as follows:

The output of the comparator n detection result has changed from "L" to "H" or from "H" to "L". (n=0,1, 2)

CMPINT: Address $05F0_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					INT2	INT1	INT0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1),W	R(RM1),W	R(RM1),W

[bit7 to bit3] (Reserved)

[bit2] INT2: Comparator 2 interrupt request flag bit

INT2	Function	
	During Read Operation	During Write Operation
0	Indicate no interrupt request.	Clear this bit.
1	Indicate that an interrupt request is issued.	Do not affect this bit.

- This bit is set to "1" when the output of the comparator 2 detection results are as follows:
- The output of the comparator 2 detection result has changed from "L" to "H" or from "H" to "L".
- "0": Clear this bit.
- "1": Do not affect this bit.
- This bit is cleared when the interrupt clear signal is "H".

Note:

"1" is read upon reading of the read modify write (RMW) instruction.
If software clear ("0" writing) or clear by the interrupt clear signal ("H") and hardware set occur at the same time, the hardware set is given priority.

[bit1] INT1: Comparator 1 interrupt request flag bit

INT1	Function	
	During Read Operation	During Write Operation
0	Indicate no interrupt request.	Clear this bit.
1	Indicate that an interrupt request is issued.	Do not affect this bit.

- This bit is set to "1" when the output of the comparator 1 detection results are as follows:
- The output of the comparator 1 detection result has changed from "L" to "H" or from "H" to "L".
- "0": Clear this bit.
- "1": Do not affect this bit.

- This bit is cleared when the interrupt clear signal is "H".

Note:

"1" is read at reading of the read modify write (RMW) instruction.
If software clear ("0" writing) or clear by the interrupt clear signal ("H") and hardware set occur at the same time, the hardware set is given priority.

[bit0] INT0: Comparator 0 interrupt request flag bit

INT0	Function	
	During Read Operation	During Write Operation
0	Indicate no interrupt request.	Clear this bit.
1	Indicate that an interrupt request is issued.	Do not affect this bit.

- This bit is set to "0" when the output of the comparator 1 detection results are as follows:
- The output of the comparator 0 detection result has changed from "L" to "H" or from "H" to "L".
- "0": Clear this bit.
- "1": Do not affect this bit.
- This bit is cleared when the interrupt clear signal is "H".

Note:

"1" is read at reading of the read modify write (RMW) instruction.
If software clear ("0" writing) or clear by the interrupt clear signal ("H") and hardware set occur at the same time, the hardware set is given priority.

47.4.9 Comparator Output Status Register: CMPST

This section shows the bit configuration of the comparator output status register.

This status register shows the output status of comparator n. (n=0, 1, 2)

CMPST: Address $05F4_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					ST2	ST1	ST0
Initial Value	0	0	0	0	0	X	X	X
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX	R,WX

[bit7 to bit3] (Reserved)

[bit2] ST2: Comparator 2 output status bit

ST2	Function
0	Indicate that the comparator 2 detection result is "L".
1	Indicate that the comparator 2 detection result is "H".

- Writing to this bit is invalid.
- The value after this bit is reset depends on the pin status of the comparator input after the reset.

[bit1] ST1: Comparator 1 output status bit

ST1	Function
0	Indicate that the comparator 1 detection result is "L".
1	Indicate that the comparator 1 detection result is "H".

- Writing to this bit is invalid.
- The value after this bit is reset depends on the pin status of the comparator input after reset.

[bit0] ST0: Comparator 0 output status bit

ST0	Function
0	Indicate that the comparator 0 detection result is "L".
1	Indicate that the comparator 0 detection result is "H".

- Writing to this bit is invalid.
- The value after this bit is reset depends on the pin status of the comparator input after reset.

47.5 Explanation of Operation

This section explains the operations of the comparator.

47.5.1 Comparator
47.5.2 D/A Converter for Comparison Reference Voltage
47.5.3 Comparator Output Polarity Selection
47.5.4 Glitch Filter
47.5.5 Interrupt Request

47.5.1 Comparator

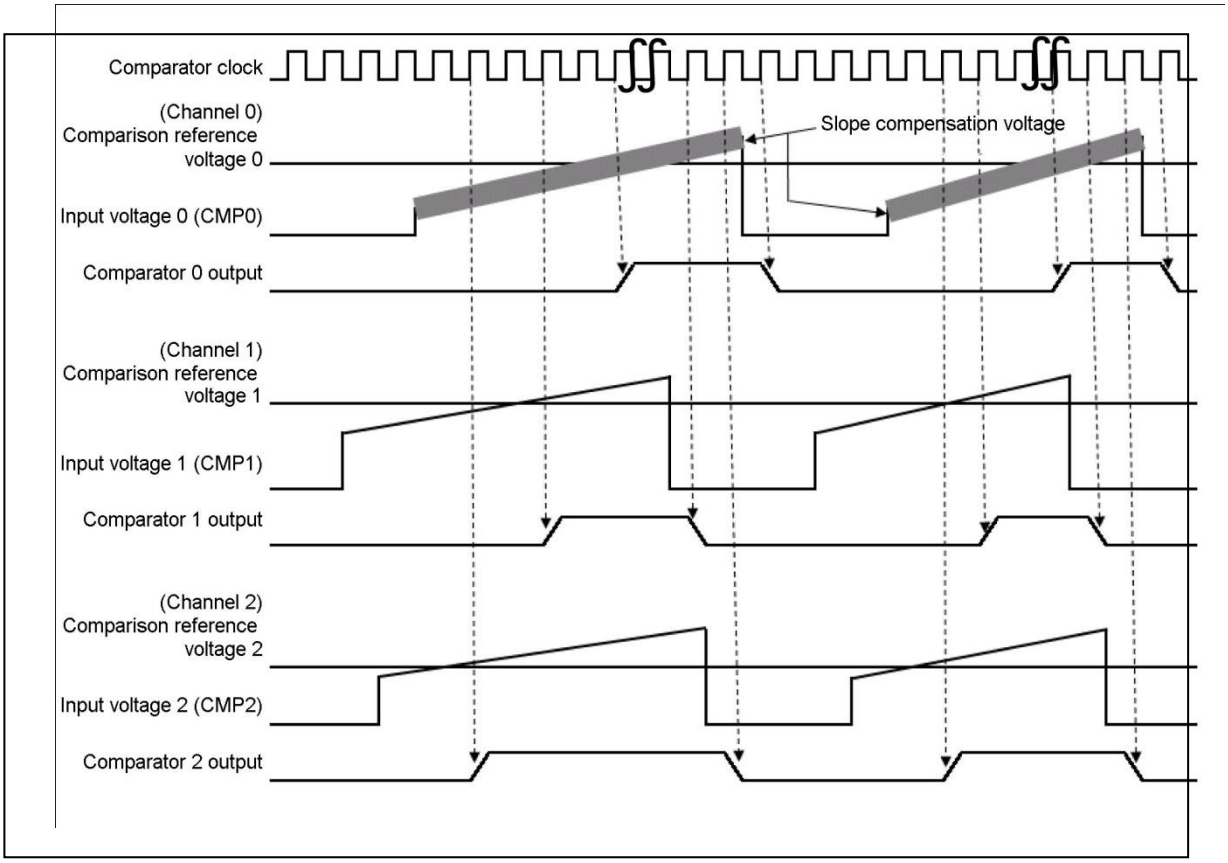
This section explains the comparator.

The comparator compares the input voltage and the comparison reference voltage, and outputs "H" when the input voltage is higher than the comparison reference voltage, or "L" when the former is lower than the latter (for CMPCTLn.CMPPOLn = 0).

When the input voltage and comparison reference voltage are equal, "L" or "H" is output.
This comparison runs with the clock obtained by dividing the PWM clock by the division ratio set in the clock division register (CMPDIV). When the PWM clock is 200 MHz, this comparison is performed for each rising edge of the 50-MHz clock signal.

The comparator has 3 channels (channels 0 to 2). Channel 0 can compare the voltage obtained by adding the slope compensation voltage to the input voltage with the comparison reference voltage.
Moreover, according to the register setting, the input voltage can be compared with the comparison reference voltage without adding the slop compensation voltage. For details, see "Chapter: Slope Compensation."

Figure 47-3. Comparison between Comparator Input Voltage and Comparison Reference Voltage

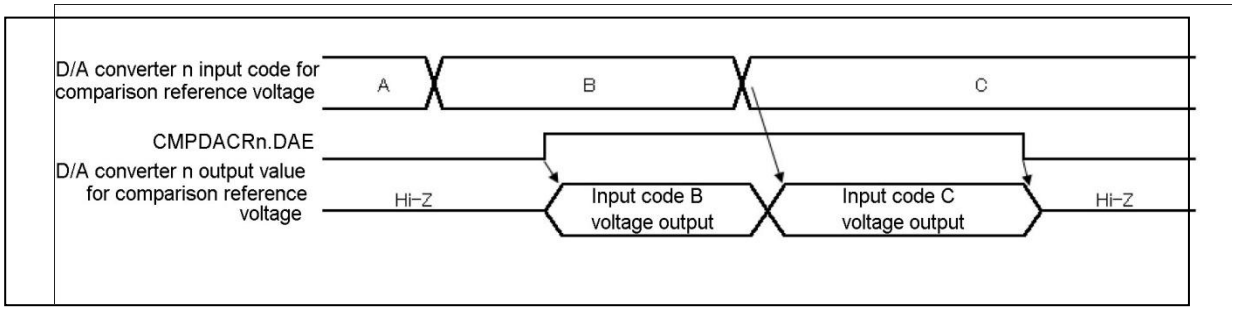


47.5.2 D/A Converter for Comparison Reference Voltage

This section explains the D/A converter for comparison reference voltage.

The comparison reference voltage is supplied from the output voltage of the 10-bit D/A converter and can be set separately for each channel.
This setting is specified with the comparator D/A converter output setting register (CMPDACRn). (n=0, 1, 2)

Figure 47-4. Input Codes and Output Voltages of D/A Converter for Comparison Reference Voltage



47.5.3 Comparator Output Polarity Selection

This section explains the comparator output polarity selection.

The polarity of the comparator output can be inverted with the comparator control register (CMPCTLn.CMPPOLn). (n=0,1, 2)

However, the comparator polarity setting is invalid when the comparator control register (CMPCTLn.CMPENn) is "0" or during the 5 to 6 cycles of the comparator clock after "1" is written to the comparator control register (CMPCTLn.CMPENn). In this case, the comparator outputs "L". (n=0,1, 2)

Figure 47-5. Comparator Output (for CMPPOLn = "0")

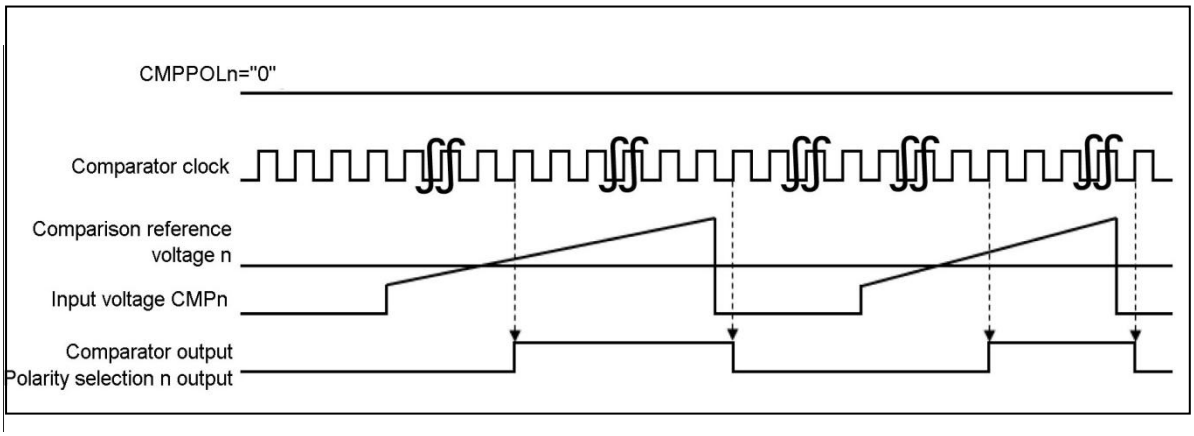
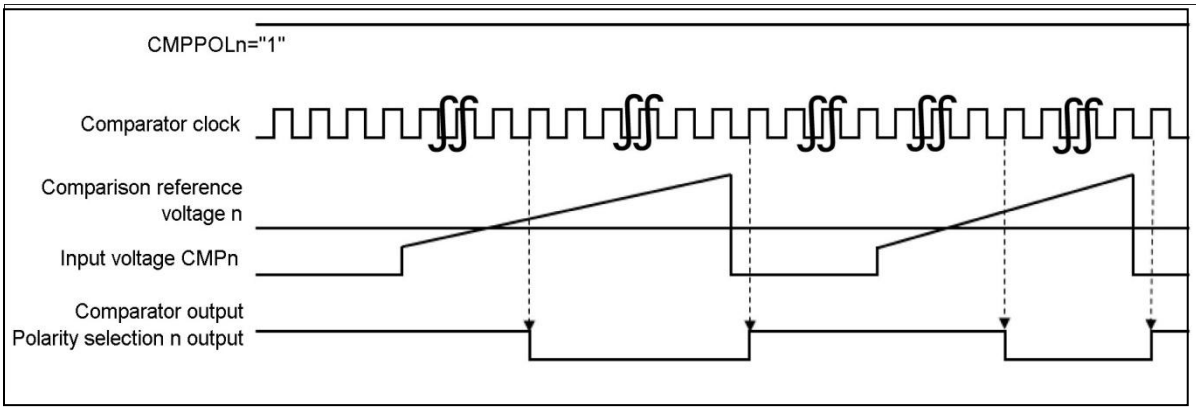


Figure 47-6. Comparator Output (for CMPPOLn = "1")



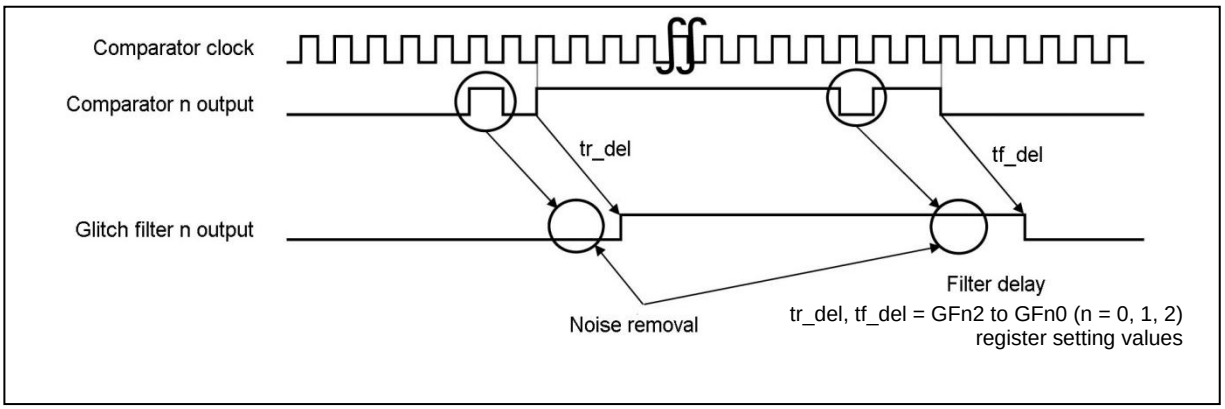
47.5.4 Glitch Filter

This section explains the glitch filter.

By performing glitch filter processing, a signal in which comparator output noise has been removed is output to PWM.

The glitch filter removes the noise in the time set in the comparator control register (CMPCTL).

Figure 47-7. Noise Removal by Glitch Filter



47.5.5 Interrupt Request

This section explains the interrupt factors.

The bit corresponding to the comparator output interrupt flag register is set to "1" when the output of the comparator n detection results are as follows: (n=0,1, 2)

- The output of the comparator n detection result has changed from "L" to "H".
- The output of the comparator n detection result has changed from "H" to "L".

The channel from which an interrupt request has been generated can be determined by reading the value of the comparator output interrupt request flag bit (CMPINT.INTn).
Moreover, the interrupt request flag can be cleared to "0" by writing "0" to the interrupt request flag bit.

47.6 Notes

This section explains the notes on the comparator.

Note on interrupts

Before setting the comparator output enable bit (CMPCTLn.CMPIEn (n = 0, 1, 2), clear the interrupt flag (CMPINT.INTn (n = 0, 1, 2)) of the associated channel.

Note on stop mode transition

To change the operation mode to "stop mode" of power dissipation control, put the comparator D/A converter and the comparator in standby mode.

Note on register access

Upon access to a comparator register, a wait of PCLK cycle × 3 + PWMCLK cycle × 7 is inserted. (This is because synchronization between the bus clock (PCLK) and macro clock (PWMCLK) must be established.)

48. Slope Compensation

This chapter explains slope compensation.

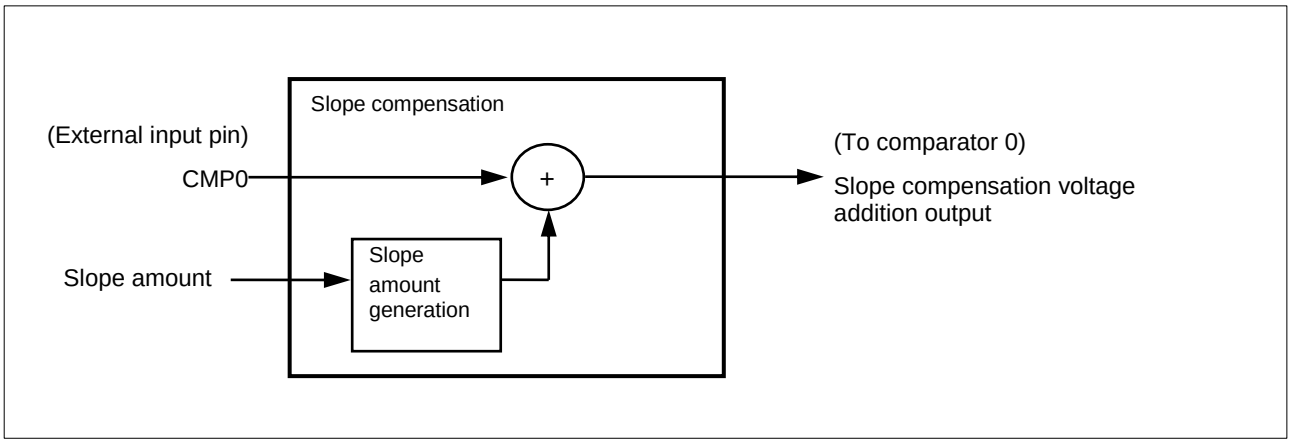
48.1 Overview
48.2 Features
48.3 Configuration
48.4 Registers
48.5 Explanation of Operation
48.6 Notes

48.1 Overview

This section provides an overview of slope compensation.

The slope compensation circuit adds a slope compensation voltage to the input voltage supplied from the external pin (CMP0).

Figure 48-1. Block Diagram (Overview)



48.2 Features

This section explains features of slope compensation.

Adding slope compensation voltage

Slope compensation adds the slope compensation voltage to the input voltage by charging the capacitor with the slope current.

Slope amount

The slope current used to charge the capacitor can be set from the slope amount setting register (SLPDADR).

Slope compensation operation status

- Slope compensation charge/discharge can be set by combining the operation control switches.
- The operation status of the slope compensation depends on the ON/OFF setting of the 3 operation control switches (SWI/SWS/SWE).
- Up to 3 types of "slope compensation in progress" operation status can be set with the operation control switches.

Operation status transition by PWM signal trigger

- 4 operation status transition triggers can be arbitrarily selected from the following PWM signals.

1. PWM0H
2. PWM0L
3. PWM1H
4. PWM1L
5. PWM2H
6. PWM2L
7. PWM master clock 0
8. PWM master clock 1

For the 4 triggers selected above, a rising or falling edge can be selected by the setting of the status transition trigger edge selection register (SLPEDGESEL1, SLPEDGESEL2).

Operation status transition based on comparator 0 detection results

- Aside from the operation status transition by the PWM signal triggers, the operation status transition can be achieved by using the comparator 0 detection result edge as a trigger.
- For operation status transition using the comparator 0 detection results as a trigger, you can select an operation status to which a transition is made; by setting the status transition trigger transition enable setting register (SLPSTMSEL1).

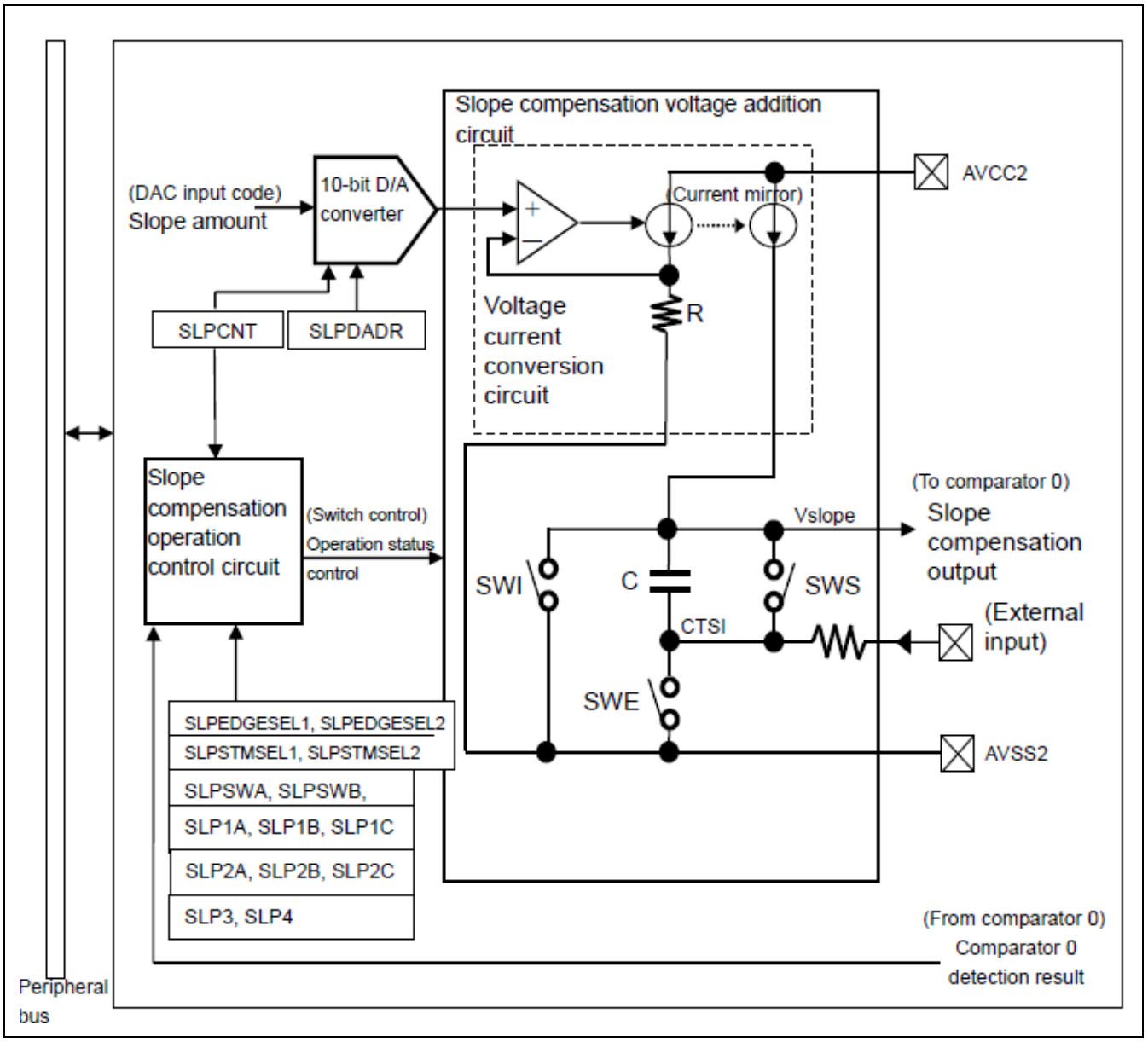
Slope compensation through mode

External input voltage (CMP0) can be output as is to comparator 0 with no slope compensation voltage added.

48.3 Configuration

This section explains the configuration of slope compensation.

Figure 48-2. Configuration Diagram



48.4 Registers

This section explains the slope compensation registers.

Table 48-1. Register Map

Address	Registers				Register functions
	+0	+1	+2	+3	
0x05A0	SLPCNT	Reserved	Reserved	Reserved	Operation control register (SLPCNT)
0x05A4	SLPEDGESEL1		SLPEDGESEL2		Status transition trigger edge selection register 1 (SLPEDGESEL1) Status transition trigger edge selection register 2 (SLPEDGESEL2)
0x05A8	SLPSWA	SLPSWB	SLPSWC	Reserved	Operation control switch setting register (SLPSWA) Operation control switch setting register (SLPSWB) Operation control switch setting register (SLPSWC)
0x05AC	SLP1A		SLP1B		Status transition trigger 1A time setting register (SLP1A) Status transition trigger 1B time setting register (SLP1B)
0x05B0	SLP1C		SLP3		Status transition trigger 1C time setting register (SLP1C) Status transition trigger 3 time setting register (SLP3)
0x05B4	SLP2A		SLP2B		Status transition trigger 2A time setting register (SLP2A) Status transition trigger 2B time setting register (SLP2B)
0x05B8	SLP2C		SLP4		Status transition trigger 2C time setting register (SLP2C) Status transition trigger 4 time setting register (SLP4)
0x05BC	Reserved	Reserved	SLPDADR		Slope amount setting register (SLPDADR)
0x05C0	SLPSTMSEL1		SLPSTMSEL2		Status transition trigger transition enable setting register 1 (SLPSTMSEL1) Status transition trigger transition enable setting register 2 (SLPSTMSEL2)
0x05C4	SLPCVE		Reserved	Reserved	Slope transmission error correction amount display register (SLPCVE)

48.4.1 Operation Control Register: SLPCNT

This section shows the bit configuration of the operation control register.

This register sets the slope compensation operation control.

SLPCNT: Address $05A0_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						SLPP	SLPEN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W

[bit7 to bit2] (Reserved)

[bit1] SLPP: Slope compensation voltage addition through mode bit

SLPP	Function
0	Disable addition through.
1	Enable addition through.

This bit sets addition through for the slope compensation voltage.

"0": Disable slope compensation voltage through to the input voltage (CMP0).

"1": Enable slope compensation voltage through to the input voltage (CMP0).

Input voltage (CMP0) is output as is to the comparator.

This function is not affected by a value set in the operation control switch setting register (SLPSWA, SLPSWB, SLPSWC).

[bit0] SLPEN: Slope compensation operation enable bit

SLPEN	Function
0	Disable
1	Enable

This bit sets the slope compensation operation.

"0": Disable the slope compensation operation.

"1": Enable the operation of slope compensation.

Note:

To avoid the output of an unintended signal to PWM, set the slope compensation register before writing "1" to SLPEN.

When "1" is written to SLPEN, slope compensation runs according to the status of the switch set in the operation control switch setting register (SLPSWA).

The output stability wait time is required to allow the output of the slope amount to stabilize after "1" is written to SLPEN.

Whenever you set SLPP to "1", set SLPEN to "0".

When SLPEN is "1", do not rewrite the slope compensation registers other than SLPEN and SLPDADR.

48.4.2 Status Transition Trigger Edge Selection Register 1: SLPEDGESEL1

This section describes the bit configuration of the status transition trigger edge selection register 1.

This register selects the signal used as the trigger for status transition of the slope compensation operation and its edge.

SLPEDGESEL1: Address $05A4_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved			EDGTINV0	Reserved	PWMSEL02	PWMSEL01	PWMSEL00
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R0,WX	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			EDGTINV1	Reserved	PWMSEL12	PWMSEL11	PWMSEL10
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R0,WX	R/W	R/W	R/W

[bit15 to bit13] (Reserved)

[bit12] EDGTINV0: Trigger edge detection polarity setting bit (trigger 1)

EDGTINV0	Function
0	Rising edge
1	Falling edge

This bit sets the edge polarity of the signal that detects trigger 1.

"0": Use the rising edge of the signal selected by PWMSEL02 to PWMSEL00 as a trigger.

"1": Use the falling edge of the signal selected by PWMSEL02 to PWMSEL00 as a trigger.

[bit11] (Reserved)

[bit10 to bit8] PWMSEL02 to PWMSEL00: Trigger signal selection bits (trigger 1)

PWMSEL 02 to 00	Function
000	PWM0H
001	PWM0L
010	PWM1H
011	PWM1L
100	PWM2H
101	PWM2L
110	PWM master clock 0
111	PWM master clock 1

These bits select the signal used as a trigger that enables status transition of the slope compensation operation. (Trigger 1)

[bit7 to bit5] (Reserved)

[bit4] EDGTINV1: Trigger edge detection polarity setting bit (trigger 2)

EDGTINV1	Function
0	Rising edge
1	Falling edge

This bit sets the edge polarity of the signal that detects trigger 2.

"0": Use the rising edge of the signal selected by PWMSEL12 to PWMSEL10 as a trigger.

"1": Use the falling edge of the signal selected by PWMSEL12 to PWMSEL10 as a trigger.

[bit3] (Reserved)

[bit2 to bit0] PWMSEL12 to PWMSEL10: Trigger signal selection bits (trigger 2)

PWMSEL 12 to 10	Function
000	PWM0H
001	PWM0L
010	PWM1H
011	PWM1L
100	PWM2H
101	PWM2L
110	PWM master clock 0
111	PWM master clock 1

These bits select the signal used as a trigger that enables status transition of the slope compensation operation. (Trigger 2)

48.4.3 Status Transition Trigger Edge Selection Register 2: SLPEDGESEL2

This section shows the bit configuration of status transition trigger edge selection register 2.

This register selects the signal used as the trigger for status transition of the slope compensation operation and its edge.

SLPEDGESEL2: Address $05A6_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved			EDGTINV2	Reserved	PWMSEL22	PWMSEL21	PWMSEL20
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R0,WX	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			EDGTINV3	Reserved	PWMSEL32	PWMSEL31	PWMSEL30
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R/W	R0,WX	R/W	R/W	R/W

[bit15 to bit13] (Reserved)

[bit12] EDGTINV2: Trigger edge detection polarity setting bit (trigger 3)

EDGTINV2	Function
0	Rising edge
1	Falling edge

This bit sets the edge polarity of the signal that detects trigger 3.

"0": Use the rising edge of the signal selected by PWMSEL22 to PWMSEL20 as a trigger.

"1": Use the falling edge of the signal selected by PWMSEL22 to PWMSEL20 as a trigger.

[bit11] (Reserved)

[bit10 to bit8] PWMSEL22 to PWMSEL20: Trigger signal selection bits (trigger 3)

PWMSEL 22 to 20	Function
000	PWM0H
001	PWM0L
010	PWM1H
011	PWM1L
100	PWM2H
101	PWM2L
110	PWM master clock 0
111	PWM master clock 1

These bits select the signal used as a trigger that enables the status transition of slope compensation operation. (Trigger 3)

[bit7 to bit5] (Reserved)

[bit4] EDGTINV3: Trigger edge detection polarity setting bit (trigger 4)

EDGTINV3	Function
0	Rising edge
1	Falling edge

This bit sets the edge polarity of the signal that detects trigger 4.

"0": Use the rising edge of the signal selected by PWMSEL32 to PWMSEL30 as a trigger.

"1": Use the falling edge of the signal selected by PWMSEL32 to PWMSEL30 as a trigger.

[bit3] (Reserved)

[bit2 to bit0] PWMSEL32 to PWMSEL30: Trigger signal selection bits (trigger 4)

PWMSEL 32 to 30	Function
000	PWM0H
001	PWM0L
010	PWM1H
011	PWM1L
100	PWM2H
101	PWM2L
110	PWM master clock 0
111	PWM master clock 1

These bits select the signal used as a trigger that enables the status transition of slope compensation. (Trigger 4)

48.4.4 Operation Control Switch Setting Register: SLPSWA, SLPSWB, SLPSWC

This section shows the bit configuration of the operation control switch setting registers.

These registers set ON/OFF of the operation control switches.

SLPSWA: Address $05A8_H$ (access: byte, half word, and word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved					SWIA	SWSA	SWEA
Initial Value	0	0	0	0	0	1	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

SLPSWB: Address $05A9_H$ (access: byte, half word, and word)

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	Reserved					SWIB	SWSB	SWEB
Initial Value	0	0	0	0	0	1	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

SLPSWC: Address $05AA_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved					SWIC	SWSC	SWEC
Initial Value	0	0	0	0	0	1	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

Reserved: Address $05AB_H$ (access: byte, half word, and word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	1	1	1	1	1	1	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

[bit31 to bit27] (Reserved)

[bit26] SWIA: Status A operation control switch (SWI) setting bit

SWIA	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWI) in slope compensation operation status A.

"0": Set the operation control switch (SWI) to OFF.

"1": Set the operation control switch (SWI) to ON.

[bit25] SWSA: Status A operation control switch (SWS) setting bit

SWSA	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWS) in slope compensation operation status A.

"0": Set the operation control switch (SWS) to OFF.

"1": Set the operation control switch (SWS) to ON.

[bit24] SWEA: Status A operation control switch (SWE) setting bit

SWEA	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWE) in slope compensation operation status A.

"0": Set the operation control switch (SWE) to OFF.

"1": Set the operation control switch (SWE) to ON.

[bit23 to bit19] (Reserved)

[bit18] SWIB: Status B operation control switch (SWI) setting bit

SWIB	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWI) in slope compensation operation status B.

"0": Set the operation control switch (SWI) to OFF.

"1": Set the operation control switch (SWI) to ON.

[bit17] SWSB: Status B operation control switch (SWS) setting bit

SWSB	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWS) in slope compensation operation status B.

"0": Set the operation control switch (SWS) to OFF.

"1": Set the operation control switch (SWS) to ON.

[bit16] SWEB: Status B operation control switch (SWE) setting bit

SWEB	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWE) in slope compensation operation status B.

"0": Set the operation control switch (SWE) to OFF.

"1": Set the operation control switch (SWE) to ON.

[bit15 to bit11] (Reserved)

[bit10] SWIC: Status C operation control switch (SWI) setting bit

SWIC	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWI) in slope compensation operation status C.

"0": Set the operation control switch (SWI) to OFF.

"1": Set the operation control switch (SWI) to ON.

[bit9] SWSC: Status C operation control switch (SWS) setting bit

SWSC	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWS) in slope compensation operation status C.

"0": Set the operation control switch (SWS) to OFF.

"1": Set the operation control switch (SWS) to ON.

[bit8] SWEC: Status C operation control switch (SWE) setting bit

SWEC	Function
0	OFF status
1	ON status

This bit sets ON/OFF of the operation control switch (SWE) in slope compensation operation status C.

"0": Set the operation control switch (SWE) to OFF.

"1": Set the operation control switch (SWE) to ON.

[bit7 to bit0] (Reserved)

Note:

- During transition from discharge status to charge status, set the LEB status so that a SWE OFF switching time of at least 5 ns is obtained after SWI/SWS is set to OFF.
  The following gives examples of register settings in each status.
  - ☐ Charge status: SWI = OFF, SWS = OFF, SWE = OFF
  - ☐ Discharge status: SWI = ON, SWS = ON, SWE = ON
  - ☐ LEB status: SWI = OFF, SWS = OFF, SWE = ON

48.4.5 Status Transition Trigger 1A Time Setting Register: SLP1A

This section shows the bit configuration of the status transition trigger 1A time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLPSWA using trigger 1 as a reference point.

SLP1A: Address 05AC$_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP1 A15	SLP1 A14	SLP1 A13	SLP1 A12	SLP1 A11	SLP1 A10	SLP1 A9	SLP1 A8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP1 A7	SLP1 A6	SLP1 A5	SLP1 A4	SLP1 A3	SLP1 A2	SLP1 A1	SLP1 A0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP1A15 to SLP1A0: Trigger 1 status A transition time setting bits

SLP1A15 to 0	Function
	Setting value × PWM division clock cycle

These bits set the time used from trigger 1 (signal set by PWMSEL02 to PWMSEL00) until slope compensation switch enters the status set by SLPSWA.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.6 Status Transition Trigger 1B Time Setting Register: SLP1B

This section shows the bit configuration of the status transition trigger 1B time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLPSWB using trigger 1 as a reference point.

SLP1B: Address $05AE_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP1 B15	SLP1 B14	SLP1 B13	SLP1 B12	SLP1 B11	SLP1 B10	SLP1 B9	SLP1 B8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP1 B7	SLP1 B6	SLP1 B5	SLP1 B4	SLP1 B3	SLP1 B2	SLP1 B1	SLP1 B0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP1B15 to SLP1B0: Trigger 1 status B transition time setting bits

SLP1B15 to 0	Function
	Setting value × PWM division clock cycle

These bits set the time used from trigger 1 (signal set by PWMSEL02 to PWMSEL00) until slope compensation switch enters the status set by SLPSWB.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.7 Status Transition Trigger 1C Time Setting Register: SLP1C

This section shows the bit configuration of the status transition trigger 1C time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLPSWC using trigger 1 as a reference point.

SLP1C: Address $05B0_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP1 C15	SLP1 C14	SLP1 C13	SLP1 C12	SLP1 C11	SLP1 C10	SLP1 C9	SLP1 C8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP1 C7	SLP1 C6	SLP1 C5	SLP1 C4	SLP1 C3	SLP1 C2	SLP1 C1	SLP1 C0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP1C15 to SLP1C0: Trigger 1 status C transition time setting bits

SLP1C15 to 0	Function
	Setting value × PWM division clock cycle

These bits set the time used from trigger 1 (signal set by PWMSEL02 to PWMSEL00) until the slope compensation switch enters the status set by SLPSWC.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.8 Status Transition Trigger 3 Time Setting Register: SLP3

This section shows the bit configuration of the status transition trigger 3 time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLP3SEL using trigger 3 as a reference point.

SLP3: Address $05B2_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP3 15	SLP3 14	SLP3 13	SLP3 12	SLP3 11	SLP3 10	SLP3 9	SLP3 8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP3 7	SLP3 6	SLP3 5	SLP3 4	SLP3 3	SLP3 2	SLP3 1	SLP3 0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP315 to SLP30: Trigger 3 status transition time setting bits

SLP3 15 to 0	Function
	Setting value × PWM division clock cycle

These bits set the time used from trigger 3 (signal set by PWMSEL32 to PWMSEL30) until the slope compensation switch enters the status set by SLP3SEL.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.9 Status Transition Trigger 2A Time Setting Register: SLP2A

This section shows the bit configuration of the status transition trigger 2A time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLPSWA using trigger 2 as a reference point.

SLP2A: Address $05B4_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP2 A15	SLP2 A14	SLP2 A13	SLP2 A12	SLP2 A11	SLP2 A10	SLP2 A9	SLP2 A8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP2 A7	SLP2 A6	SLP2 A5	SLP2 A4	SLP2 A3	SLP2 A2	SLP2 A1	SLP2 A0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP2A15 to SLP2A0: Trigger 2 status A transition time setting bits

SLP2A15 to 0	Function
	Setting value × PWM division clock cycle

These bits set the time used from trigger 2 (signal set by PWMSEL12 to PWMSEL10) until the slope compensation switch enters the status set by SLPSWA.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.10 Status Transition Trigger 2B Time Setting Register: SLP2B

This section shows the bit configuration of the status transition trigger 2B time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLPSWB using trigger 2 as a reference point.

SLP2B: Address $05B6_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP2 B15	SLP2 B14	SLP2 B13	SLP2 B12	SLP2 B11	SLP2 B10	SLP2 B9	SLP2 B8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP2 B7	SLP2 B6	SLP2 B5	SLP2 B4	SLP2 B3	SLP2 B2	SLP2 B1	SLP2 B0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP2B15 to SLP2B0: Trigger 2 status B transition time setting bits

SLP2B15 to 0	Function
	Setting value × PWM division clock cycle

These bits set the time used from trigger 2 (signal set by PWMSEL12 to PWMSEL10) until the slope compensation switch enters the status set by SLPSWB.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.11 Status Transition Trigger 2C Time Setting Register: SLP2C

This section shows the bit configuration of the status transition trigger 2C time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLPSWC using trigger 2 as a reference point.

SLP2C: Address $05B8_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP2C15	SLP2C14	SLP2C13	SLP2C12	SLP2C11	SLP2C10	SLP2C9	SLP2C8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP2C7	SLP2C6	SLP2C5	SLP2C4	SLP2C3	SLP2C2	SLP2C1	SLP2C0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP2C15 to SLP2C0: Trigger 2 status C transition time setting bits

SLP2C15 to 0	Function
	Setting value × PWM division clock cycle

These bits set the time used from trigger 2 (signal set by PWMSEL12 to PWMSEL10) until the slope compensation switch enters the status set by SLPSWC.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.12 Status Transition Trigger 4 Time Setting Register: SLP4

This section shows the bit configuration of the status transition trigger 4 time setting register.

This register sets the time used until slope compensation enters the status of the switch set in SLP4SEL using trigger 4 as a reference point.

SLP4: Address 05BA$_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	SLP4 15	SLP4 14	SLP4 13	SLP4 12	SLP4 11	SLP4 10	SLP4 9	SLP4 8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SLP4 7	SLP4 6	SLP4 5	SLP4 4	SLP4 3	SLP4 2	SLP4 1	SLP4 0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit0] SLP415 to SLP40: Trigger 4 status transition time setting bits

SLP4 15 to 0	Function
	· Setting value × PWM division clock cycle

These bits set the time used from trigger 4 (signal set by PWMSEL32 to PWMSEL30) until the slope compensation switch enters the status set by SLP4SEL.

Note:

- Do not set any value that is greater than the cycle of the PWM signal used as the trigger.

48.4.13 Slope Amount Setting Register: SLPDADR

This section shows the bit configuration of the slope amount setting register.

This register sets the slope amount.

SLPDADR: Address $05BE_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						DA9	DA8
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Initial Value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit10] (Reserved)

[bit9 to bit0] DA9 to DA0: Slope amount setting bits

These bits set the slant of the slope compensation current.
The relationship between the slope setting values and slope compensation current slant is as follows (under the condition of $AV_{CC}2$ = 5.0 V, Tj = 25°C, process = Typical).

DA9 to DA0	Slope Amount dVslope/dt [V/usec][*1]	
00 0000 0000 ... 00 0010 0011	Setting prohibited	
00 0010 0100	0.0502	
00 0010 0101	0.0516	
...	...	Settable range[*2]
10 1100 1011	0.9975	
10 1100 1100	0.9989	
10 1100 1101	Setting prohibited	
...		
11 1111 1111		

*1 For details on the relationship between slope amount dVslope/dt and DA9 to DA0, see Section 48.5.2

*2 The settable range differs depending on the operation cycle. For details, see Section 48.5.2

Note:

The output stability wait time is required to allow the output of the slope amount to stabilize after values are written to DA9 to DA0.
To access the slope amount setting register, use a half-word or word access instruction.

This register can be rewritten when SLPEN is "0" or the slope is in the discharge status. Moreover, do not rewrite this register from when writing "1" to SLPEN until the slope becomes stable.

48.4.14 Status Transition Trigger Transition Enable Setting Register 1: SLPSTMSEL1

This section shows the bit configuration of status transition trigger transition enable setting register 1.

This register enables/disables status transition of the slope compensation operation.

SLPSTMSEL1: Address $05C0_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	SLP1CEN	SLP1BEN	SLP1AEN	Reserved	SLP2CEN	SLP2BEN	SLP2AEN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					CMPSEL1	CMPSEL0	CMPENB
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W	R/W	R/W

[bit15] (Reserved)

[bit14] SLP1CEN: Trigger 1 status C transition enable bit

SLP1CEN	Function
0	Disable
1	Enable

This bit sets enabling of transition to status C of the slope after the time set by SLP1C15 to SLP1C0 from trigger 1 generation has elapsed.

"0": Disable the transition to status C of slope operation.

"1": Enable the transition to status C of slope operation.

[bit13] SLP1BEN: Trigger 1 status B transition enable bit

SLP1BEN	Function
0	Disable
1	Enable

This bit sets enabling of transition to status B of the slope after the time set by SLP1B15 to SLP1B0 from trigger 1 generation has elapsed.

"0": Disable the transition to status B of slope operation.

"1": Enable the transition to status B of slope operation.

[bit12] SLP1AEN: Trigger 1 status A transition enable bit

SLP1AEN	Function
0	Disable
1	Enable

This bit sets enabling of transition to status A of the slope after the time set by SLP1A15 to SLP1A0 from trigger 1 generation has elapsed.

"0": Disable the transition to status A of slope operation.

"1": Enable the transition to status A of slope operation.

Note:

- Verify that the status transition enable bit of any unused operation status is set to "0".
  If the status transition enable bit of the unused operation status is set to "1", an undefined value may be output to PWM.

[bit11] (Reserved)

[bit10] SLP2CEN: Trigger 2 status C transition enable bit

SLP2CEN	Function
0	Disable
1	Enable

This bit sets enabling of transition to status C of the slope after the time set by SLP2C15 to SLP2C0 from trigger 2 generation has elapsed.

"0": Disable the transition to status C of slope operation.

"1": Enable the transition to status C of slope operation.

[bit9] SLP2BEN: Trigger 2 status B transition enable bit

SLP2BEN	Function
0	Disable
1	Enable

This bit sets enabling of transition to status B of the slope after the time set by SLP2B15 to SLP2B0 from trigger 2 generation has elapsed.

"0": Disable the transition to status B of slope operation.

"1": Enable the transition to status B of slope operation.

[bit8] SLP2AEN: Trigger 2 status A transition enable bit

SLP2AEN	Function
0	Disable
1	Enable

This bit sets enabling of transition to status A of the slope after the time set by SLP2A15 to SLP2A0 from trigger 2 generation has elapsed.

"0": Disable the transition to status A of slope operation.

"1": Enable the transition to status A of slope operation.

Note:

- Verify that the status transition enable bit of any unused operation status is set to "0".
  If the status transition enable bit of the unused operation status is set to "1", an undefined value may be output to PWM.

[bit7 to bit3] (Reserved)

[bit2, bit1] CMPSEL1, CMPSEL0: Comparator trigger status transition setting bits

CMPSEL1 to 0	Function
00	SLPSWA status
01	SLPSWA status
10	SLPSWB status
11	SLPSWC status

These bits select the operation control switch setting to which transition is made when the comparator 0 detection result has changed.
The output polarity when the detection result has changed is determined by the setting of the comparator 0 control register (CMPCTL0.CMPPOL0). If CMPCTL0.CMPPOL0 is "0", status transition occurs when the comparator 0 detection result has changed from "L" to "H". If CMPCTL0.CMPPOL0 is "1", status transition occurs when the comparator 0 detection result has changed from "H" to "L". For details, see "Chapter: Comparator."

[bit0] CMPENB: Comparator status transition enable bit

CMPENB	Function
0	Disable
1	Enable

This bit sets enabling of transition to the status selected by CMPSEL1, CMPSEL0 when the comparator 0 detection result has changed.
"0": Disable the status transition of slope operation.
"1": Enable the status transition of slope operation.

48.4.15 Status Transition Trigger Transition Enable Setting Register 2: SLPSTMSEL2

This section shows the bit configuration of the status transition trigger transition enable setting register 2.

This register enables/disables status transition of the slope compensation operation.

SLPSTMSEL2: Address 05C2$_H$ (access: byte, half word, and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	SLP3SEL1	SLP3SEL0	SLP3EN	Reserved	SLP4SEL1	SLP4SEL0	SLP4EN
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R/W	R/W	R/W	R0,WX	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							
Initial Value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX

[bit15] (Reserved)

[bit14, bit13] SLP3SEL1, SLP3SEL0: Trigger 3 status transition setting bits

SLP3SEL1 to 0	Function
00	SLPSWA status
01	SLPSWA status
10	SLPSWB status
11	SLPSWC status

These bits set the operation control switch setting to which transition is made after the time set by SLP3 15 to SLP3 0 from trigger 3 generation has elapsed.

[bit12] SLP3EN: Trigger 3 status transition enable bit

SLP3EN	Function
0	Disable
1	Enable

This bit sets the enabling of transition to the status selected by SLP3SEL1, SLP3SEL0 after the time set by SLP3 15 to SLP3 0 from trigger 3 generation has elapsed.

"0": Disable the status transition of slope operation.

"1": Enable the status transition of slope operation.

[bit11] (Reserved)

[bit10, bit9] SLP4SEL1, SLP4SEL0: Trigger 4 status transition setting bits

SLP4SEL1 to 0	Function
00	SLPSWA status
01	SLPSWA status
10	SLPSWB status
11	SLPSWC status

These bits set the operation control switch setting to which transition is made after the time set by SLP4 15 to SLP4 0 from trigger 4 generation has elapsed.

[bit8] SLP4EN: Trigger 4 status transition enable bit

SLP4EN	Function
0	Disable
1	Enable

These bits set the enabling of transition to the status selected by SLP4SEL1, SLP4SEL0 after the time set by SLP4 15 to SLP4 0 from trigger 4 generation has elapsed.

"0": Disable the status transition of slope operation.

"1": Enable the status transition of slope operation.

[bit7 to bit0] (Reserved)

48.4.16 Slope Transmission Error Correction Amount Display Register: SLPCVE

This section shows the bit configuration of the slope transmission error correction amount display register.

This register indicates the amount by which slope compensation transmission errors are corrected.

SLPCVE: Address $05C4_H$ (access: half word and word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						CVE9	CVE8
Initial Value	0	0	0	0	0	0	X	X
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CVE7	CVE6	CVE5	CVE4	CVE3	CVE2	CVE1	CVE0
Initial Value	X	X	X	X	X	X	X	X
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15 to bit10] (Reserved)

[bit9 to bit0] CVE9 to CVE0: Slope transmission error correction amount setting bits

CVE9 to CVE0	Reference Voltage Correction Amount
00 0000 0000	$2^{10}/2^{10}$ (=100.00%)
00 0000 0001	$(2^{10}-1)/2^{10}$ (=99.90%)
00 0000 0010	$(2^{10}-2)/2^{10}$ (=99.81%)
00 0000 0011	$(2^{10}-3)/2^{10}$ (=99.71%)
...	...
11 1111 1100	$(2^{10}-1020)/2^{10}$ (=0.39%)
11 1111 1101	$(2^{10}-1021)/2^{10}$ (=0.29%)
11 1111 1110	$(2^{10}-1022)/2^{10}$ (=0.20%)
11 1111 1111	$(2^{10}-1023)/2^{10}$ (=0.10%)

To correct slope transmission errors, this register indicates the reference voltage correction amount of the comparator having slope compensation.
Transmission errors are corrected by multiplying this register value as a coefficient to the reference voltage of the comparator having slope compensation.

Comparison reference voltage value [V] after transmission error correction

= comparison reference voltage value [V] before transmission error correction × (1 - (CVE register value)/2^{10})

Set the CMPDADR0 register using "comparison reference voltage value after transmission error correction" obtained from this calculation formula.

48.5 Explanation of Operation

This section explains the slope compensation operation.

48.5.1 Slope Compensation

48.5.2 Setting the Slope Amount

48.5.1 Slope Compensation

This section explains slope compensation.

The slope compensation voltage is generated by charging the capacitor with the current corresponding to the slope amount set in the slope amount setting register (SLPDADR).

While slop compensation is stopped (SLPEN = "0"), slope operation status R in Table 48-2 is entered. SWI is set to ON, SWS to OFF, and SWE to OFF.
If SLPEN is set to "1", slope compensation starts and the operation status becomes status A. In this case, ON/OFF of the operation control switch conforms to the value set in the operation control switch setting register (SLPSWA).

Table 48-2. Operation Control Switch Statuses in Each Slope Compensation Operation Status

Slope Operation Status	Operation Control Switch Name and Switch Status		
	SWI	SWS	SWE
A	Set by SWIA	Set by SWSA	Set by SWEA
B	Set by SWIB	Set by SWSB	Set by SWEB
C	Set by SWIC	Set by SWSC	Set by SWEC
R*1	ON	OFF	OFF
E*2	OFF	ON	OFF

*1 R indicates SLPEN = "0" (operation is stopping).

The setting value of the operation control switch setting register (SLPSW) is not applied.

*2 E indicates SLPP = "1" (slope compensation voltage not added) mode. The input pin (CMP0) is output as is to the comparator.

The setting value of the operation control switch setting register is not applied.

The operation control switch statuses during slope compensation) include 3 statuses: A, B, and C.

4 signals used as the trigger for transition to 3 statuses can be arbitrarily selected from the following.

PWM0H, PWM0L, PWM1H, PWM1L, PWM2H, PWM2L, PWM master clock 0, and PWM master clock 1

Moreover, a rising/falling edge can be selected as the trigger for the selected signal.

For example, to set the first trigger as the PWM0H rising edge and the second as the PWM0L falling edge, set the following:

- PWMSEL02 to PWMSEL00 = "000_B", EDGTINV0 = "0"
- PWMSEL02 to PWMSEL10 = "001_B", EDGTINV0 = "1"

The operation control switch status makes a transition when a certain time has elapsed from trigger generation.

The transition time of the status for trigger 1 is determined by the value set in the status transition trigger 1 time setting register (SLP1A, SLP1B, SLP1C). The transition time of the status for trigger 2 is determined by the value set in the status transition trigger 2 time setting register (SLP2A, SLP2B, SLP2C). The transition time of the status for trigger 3 is determined by the value set in the status transition trigger 3 time setting register (SLP3). The transition time of the status for trigger 4 is determined by the value set in the status transition trigger 4 time setting register (SLP4).

Moreover, enabling/disabling of transition to 3 statuses A, B, and C can be set for each of triggers 1 and 2 with the status transition trigger transition enable setting register 1 (SLPSTMSEL1).

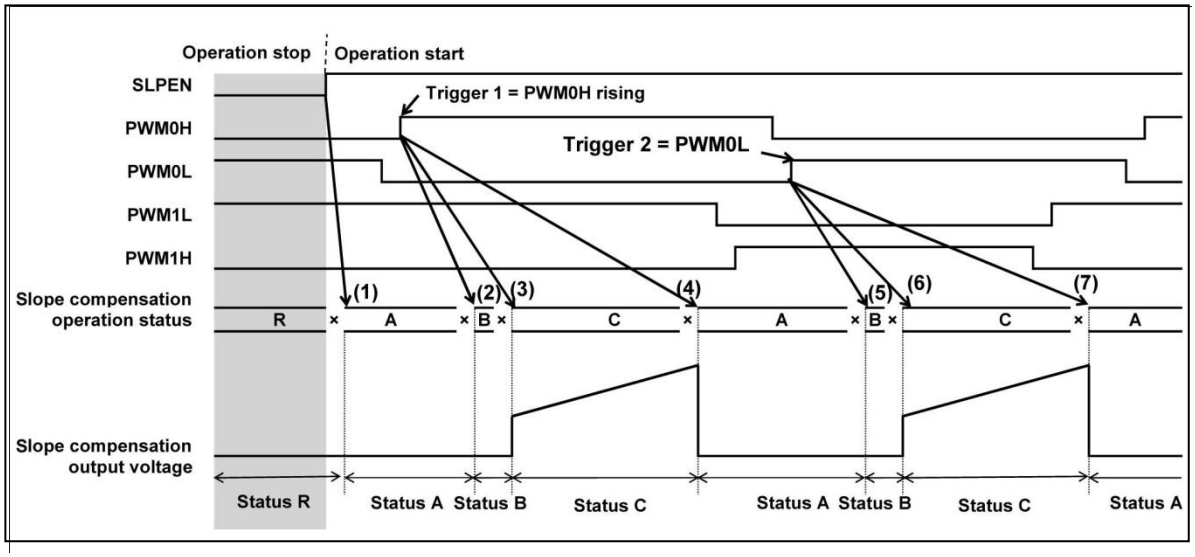
For triggers 3 and 4, the status transition enable setting register 2 (SLPSTMSEL2) can be used to specify the settings for the status to which transition is made. The register can also be used to set enabling of the selected status transition. However, if multiple transition conditions occur at the same time, transition to status R is given priority. Hereafter, priority is given to status A, status B, and then status C.

The following provides some slope operation setting examples.

Slope operation setting example (1)

- Select the PWM0H rising edge (PWMSEL02 to PWMSEL00 = "000_B" EGDTINV0 = "0") as trigger 1.
- Select the PWM0L rising edge (PWMSEL12 to PWMSEL10 = "001_B" EGDTINV1 = "0") as trigger 2.
- Trigger 1 and trigger 2 both use status A, status B, and status C.
  Comparator output edge detection is invalid (SLPSTMSEL1 = "7700_H").
- The operation control switch setting is as follows:
  - ☐ SLPSWA="111_B"
  - ☐ SLPSWB="001_B"
  - ☐ SLPSWC="000_B"

Figure 48-3. Time Chart of Slope Operation Setting Example (1)



1. Start of slope compensation operation (SLPEN = "0" -> "1")
   Transition to the status of the switch set with operation status = SLPSWA
2. Passing of the time set with SLP1B15 to SLP1B0 from trigger 1
   Transition to the status of the switch set with operation status = SLPSWB
3. Passing of the time set with SLP1C15 to SLP1C0 from trigger 1
   Transition to the status of the switch set with operation status = SLPSWC
4. Passing of the time set with SLP1A15 to SLP1A0 from trigger 1
   Transition to the status of the switch set with operation status = SLPSWA
5. Passing of the time set with SLP2B15 to SLP2B0 from trigger 2
   Transition to the status of the switch set with operation status = SLPSWB
6. Passing of the time set with SLP2C15 to SLP2C0 from trigger 2
   Transition to the status of the switch set with operation status = SLPSWC
7. Passing of the time set with SLP2A15 to SLP2A0 from trigger 2
   Transition to the status of the switch set with operation status = SLPSWA

Figure 48-4 shows the operation status transition of slope operation setting example (1).

The operation status during slope compensation stop (SLPEN = "0") is status R. Transition is made to status A when slope compensation starts (SLPEN = "1").

Transition is made to status B when SLP1B15 to SLP1B0 or SLP2B15 to SLP2B0 has elapsed (b) from when trigger 1 or 2 occurs in status A. Similarly, transition is made to status C when SLP1C15 to SLP1C0 or SLP2C15 to SLP2C0 has elapsed (c) from when trigger 1 or 2 occurs.

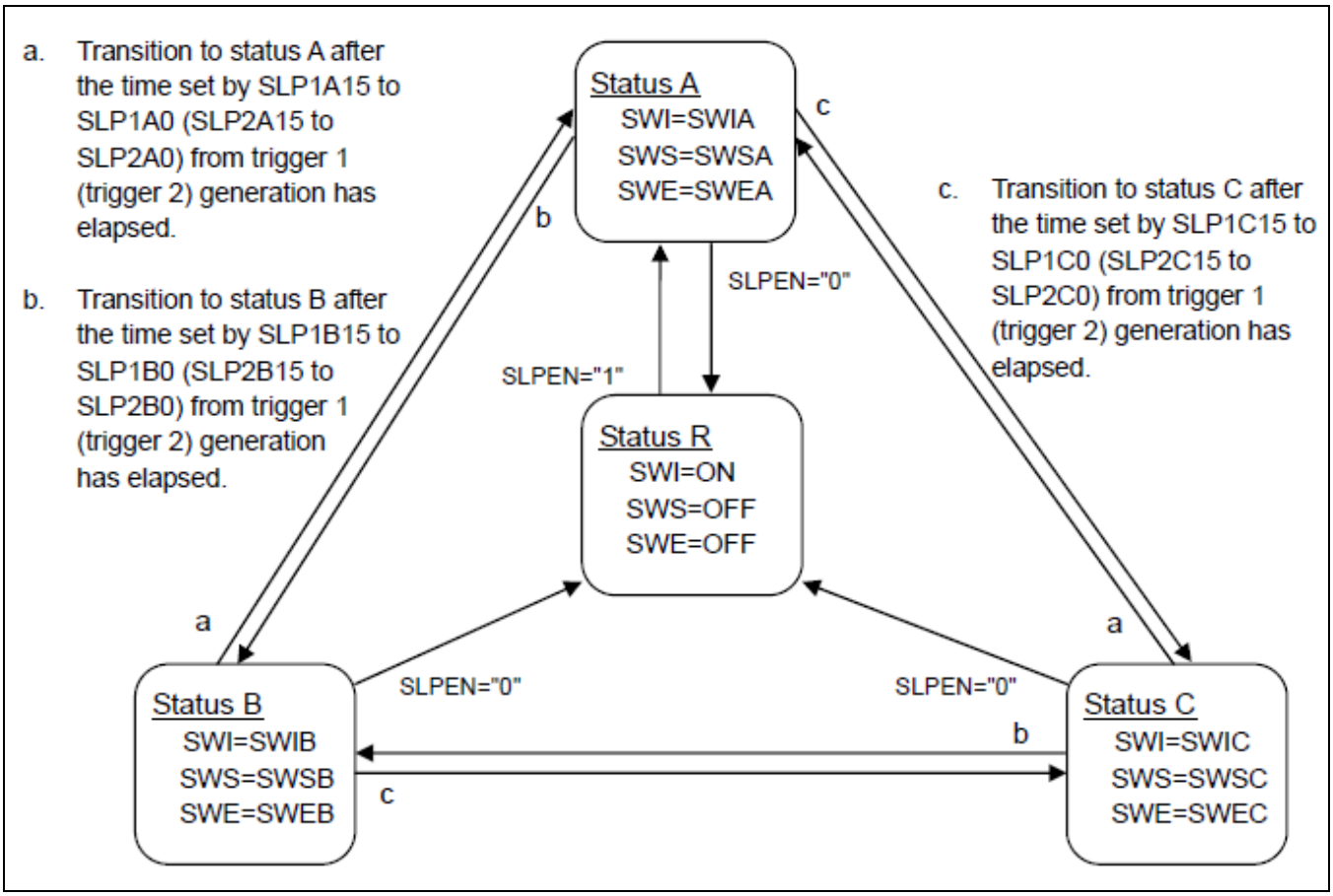
Transition is made to status A when SLP1A15 to SLP1A0 or SLP2A15 to SLP2A0 has elapsed (a) from when trigger 1 or 2 occurs in status B. Similarly, transition is made to status C when SLP1C15 to SLP1C0 or SLP2C15 to SLP2C0 has elapsed (c) from when trigger 1 or 2 occurs.

Transition is made to status A when SLP1A15 to SLP1A0 or SLP2A15 to SLP2A0 has elapsed (a) from when trigger 1 or 2 occurs in status C. Similarly, transition is made to status B when SLP1B15 to SLP1B0 or SLP2B15 to SLP2B0 has elapsed (b) from when trigger 1 or 2 occurs.

Transition is made to status R when slope compensation stops (SLPEN = "0") in status A, B, or C.

For CMPENB = "0", status transition is invalid when comparator output has changed.

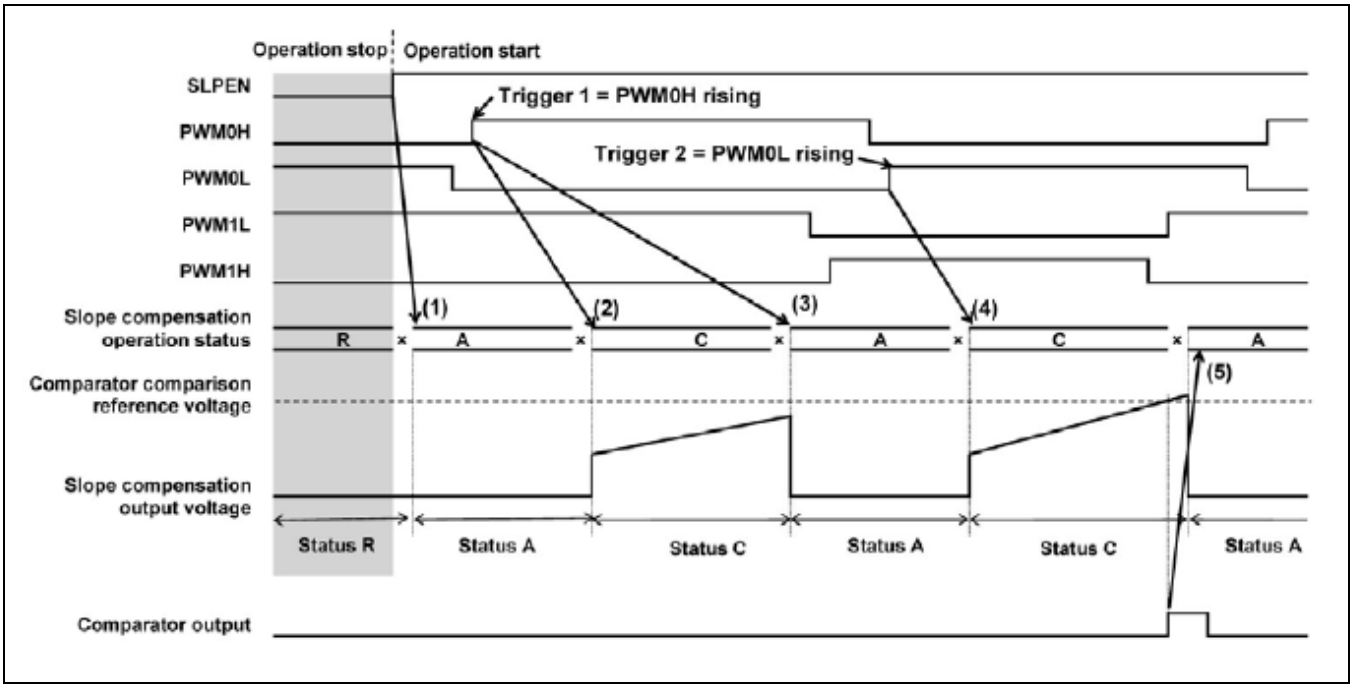
Figure 48-4. Status Transition (for CMPENB = "0") of Slope Operation Setting Example (1)



Slope operation setting example (2)

- Select the PWM0H rising edge (PWMSEL02 to PWMSEL00 = "000_B" EGDTINV0 = "0") as trigger 1.
- Select the PWM0L rising edge (PWMSEL12 to PWMSEL10 = "001_B" EGDTINV1 = "0") as trigger 2.
- Trigger 1 and trigger 2 both use status A and status C.
  Comparator output edge detection is valid, and transition is made to status A at edge detection (SLPSTMSEL1 = "5503_H").
- The operation control switch setting is as follows:
  - ☐ SLPSWA="111_B"
  - ☐ SLPSWB="001_B"
  - ☐ SLPSWC="000_B"

Figure 48-5. Time Chart of Slope Operation Setting Example (2)



1. Start of slope compensation operation (SLPEN = "0" -> "1")
   Transition to the status of the switch set with operation status = SLPSWA
2. Passing of the time set with SLP1C15 to SLP1C0 from trigger 1
   Transition to the status of the switch set with operation status = SLPSWC
3. Passing of the time set with SLP1A15 to SLP1A0 from trigger 1
   Transition to the status of the switch set with operation status = SLPSWA
4. Passing of the time set with SLP2C15 to SLP2C0 from trigger 2
   Transition to the status of the switch set with operation status = SLPSWC
5. Generation of the comparator output change trigger
   Transition to the status of the switch set with operation status = SLPSWA

Figure 48-6 shows the operation status transition of slope operation setting example (2).

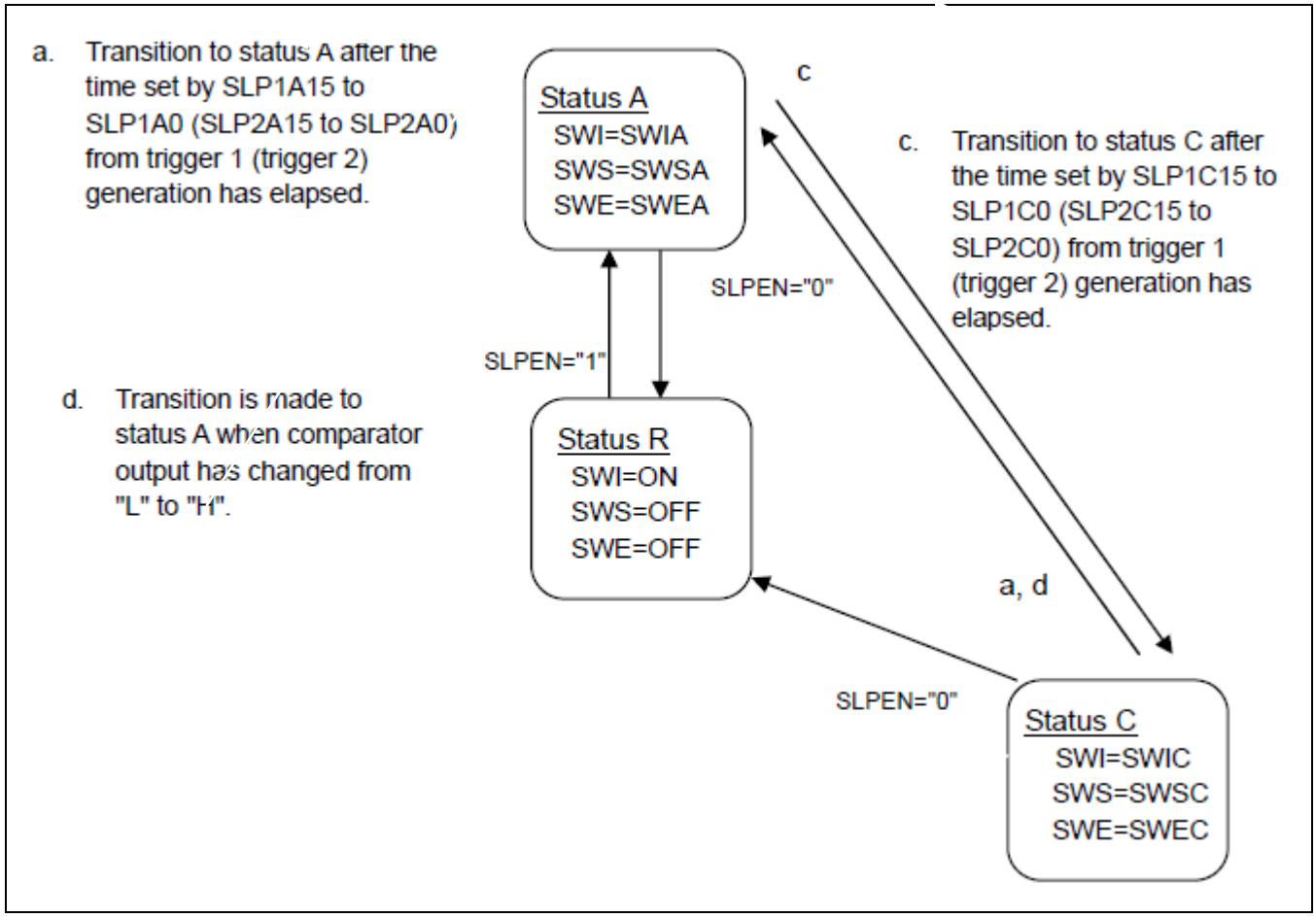
The status transition for slope compensation stop/operation (SLPEN = "0" or "1") and transitions (a, c) of each status are the same as in.Figure 48-5 Slope operation setting example (2) uses only status A and status C. So, no transition is made to status B regardless of the value of the status transition trigger 1 time setting register (SLP1B) or of the status transition trigger 2 time setting register (SLP2B).

For CMPENB = "1", when comparator output has changed, status transition occurs according to the value set in the comparator trigger status transition setting bit (CMPSEL1 to CMPSEL0). In slope operation setting example (2), transition is made to status A.

Transition is made to status R when slope compensation stops (SLPEN = "0") in status A, or C.

Figure 48-6. Status Transition of Slope Operation Setting Example (2)

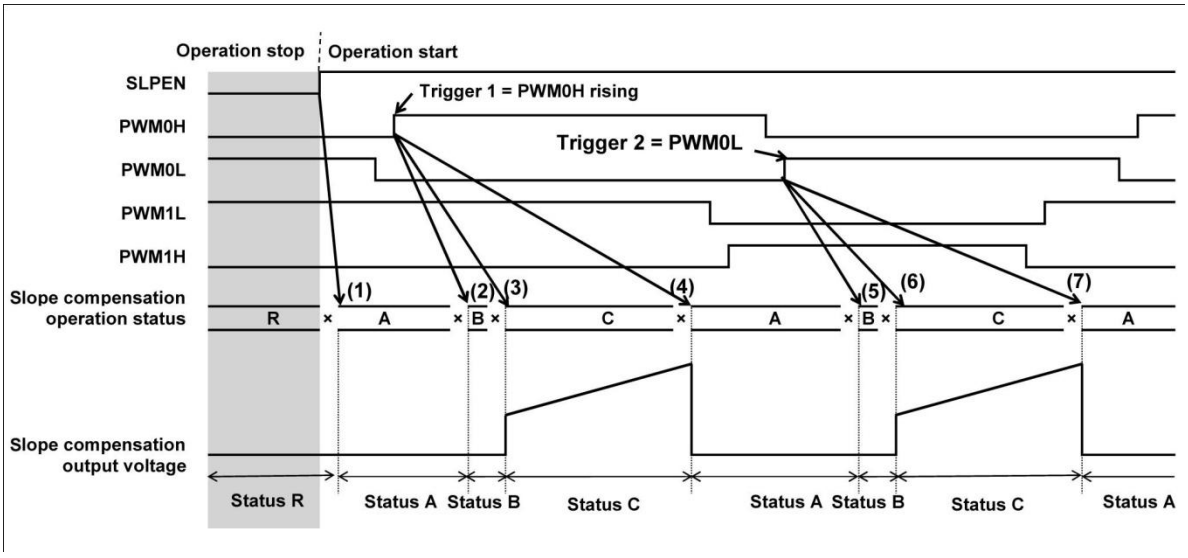
(For CMPENB = "1", CMPSEL1, CMPSEL0 = "01_B", and CMPPOL0 = "0")



Slope operation setting example (3)

- Select the PWM0H rising edge (PWMSEL02 to PWMSEL00 = "000_B" EGDTINV0 = "0") as trigger 1.
- Select the PWM0L rising edge (PWMSEL12 to PWMSEL10 = "001_B" EGDTINV1 = "0") as trigger 2.
- Select the PWM1L falling edge (PWMSEL22 to PWMSEL20 = "011_B" EGDTINV2 = "1") as trigger 3.
- Select the PWM1H falling edge (PWMSEL32 to PWMSEL30 = "010_B" EGDTINV3 = "1") as trigger 4.
- Trigger 1 and trigger 2 both use status B and status C.
  Comparator output edge detection is invalid (SLPSTMSEL1 = "6600_H").
- Edge detection is valid for both trigger 3 and trigger 4, and transition is made to status A at edge detection.
  (SLPSTMSEL2 = "3300_H")
- The operation control switch setting is as follows:
  - ☐ SLPSWA="111_B"
  - ☐ SLPSWB="001_B"
  - ☐ SLPSWC="000_B"

Figure 48-7. Time Chart of Slope Operation Setting Example (3)



1. Start of slope compensation operation (SLPEN = "0" -> "1")
   Transition to the status of the switch set with operation status = SLPSWA
2. Passing of the time set with SLP1B15 to SLP1B0 from trigger 1
   Transition to the status of the switch set with operation status = SLPSWB
3. Passing of the time set with SLP1C15 to SLP1C0 from trigger 1
   Transition to the status of the switch set with operation status = SLPSWC
4. Passing of the time set with SLP315 to SLP30 from trigger 3
   Transition to the status of the switch set with operation status = SLPSWA
5. Passing of the time set with SLP2B15 to SLP2B0 from trigger 2
   Transition to the status of the switch set with operation status = SLPSWB
6. Passing of the time set with SLP2C15 to SLP2C0 from trigger 2

Transition to the status of the switch set with operation status = SLPSWC

7. Passing of the time set with SLP415 to SLP40 from trigger 4

   Transition to the status of the switch set with operation status = SLPSWA

Figure 48-8. shows the operation status transition of slope operation setting example (3).

The operation status during slope compensation stop (SLPEN = "0") is status R. Transition is made to status A when slope compensation starts (SLPEN = "1").

Transition is made to status B when SLP1B15 to SLP1B0 or SLP2B15 to SLP2B0 has elapsed (b) from when trigger 1 or 2 occurs in status A. Similarly, transition is made to status C when SLP1C15 to SLP1C0 or SLP2C15 to SLP2C0 has elapsed (c) from when trigger 1 or 2 occurs.

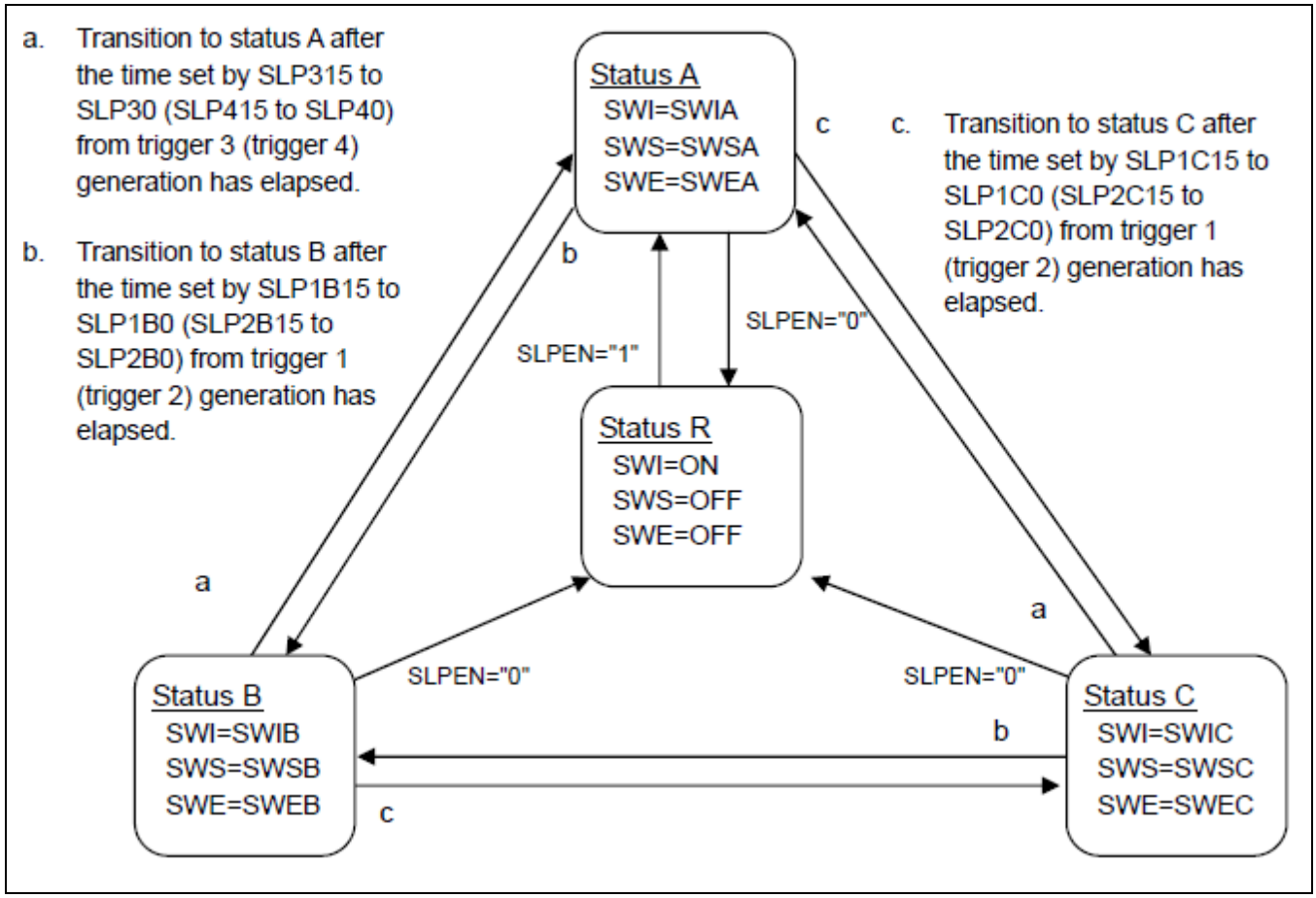
Transition is made to status A when SLP315 to SLP30 or SLP415 to SLP40 has elapsed (a) from when trigger 3 or 4 occurs in status B. Similarly, transition is made to status C when SLP1C15 to SLP1C0 or SLP2C15 to SLP2C0 has elapsed (c) from when trigger 1 or 2 occurs.

Transition is made to status A when SLP315 to SLP30 or SLP415 to SLP40 has elapsed (a) from when trigger 3 or 4 occurs in status C. Similarly, transition is made to status B when SLP1B15 to SLP1B0 or SLP2B15 to SLP2B0 has elapsed (b) from when trigger 1 or 2 occurs.

Transition is made to status R when slope compensation stops (SLPEN = "0") in status A, B, or C.

For CMPENB = "0", status transition is invalid when comparator output has changed.

Figure 48-8. Status Transition (for CMPENB = "0") of Slope Operation Setting Example (3)



48.5.2 Setting the Slope Amount

This section explains the slope amount.

The D/A converter for slope compensation generates the voltage used to determine the current generated by the voltage current conversion circuit.

Relationship between D/A converter input codes (DA9 to DA0) and the slope amount

The slope amount is as follows:

$dVslope/dt = AV_{CC}2/1024 \times 1/3.5 \times$ input code [V/usec]

Moreover, the relationship between the input code setting value and slope compensation voltage Vslope is as follows:

$Vslope = AV_{CC}2/1024 \times 1/3.5 \times$ input code $\times$ charge time ($t_{charge[us]}$) [V]

Note that the 3.5 in these formulas is a time constant for CR.

Upper/lower limits for slope amount and for D/A converter input codes

Suppose that the upper/lower limit values of the input codes are used within the range for which the resolution is compensated. They are determined by the maximum value t_{chmax} of the capacitor charge time assumed for each operation cycle and by the upper/lower limits on the slope compensation voltage range. Their relationships are as follows:

Input code upper limit value (Vslope_max) = slope amount upper limit value $\times (1024/AV_{CC}2) \times 3.5$

Input code lower limit value (Vslope_min) = slope amount lower limit value $\times (1024/AV_{CC}2) \times 3.5$

Slope amount upper limit value = $Vslope_max/t_{chmax} \times 10^{-6}$ [V/usec]

Slope amount lower limit value = $Vslope_min/t_{chmax} \times 10^{-6}$ [V/usec]

For details on the resolution compensation range, see the data sheet.

48.6 Notes

This section explains the notes on slope compensation.

Note on watch mode transition

To enable operation transition to "watch mode" of power consumption control, make a transition from when SLPEN is "0" or the slope is in the discharge status.

Note on stop mode transition

To enable operation transition to "stop mode" of power consumption control, put the slope in standby mode.

Note on register access

Upon access to a slope compensation register, a wait of PCLK cycle × 3 + PWM division clock cycle × 7 is inserted. (This is because synchronization must be established between the bus clock (PCLK) and macro clock (PWM division clock).)

Appendix

Appendix is shown.

A.1 I/O Map

A.2 List of Interrupt Vector

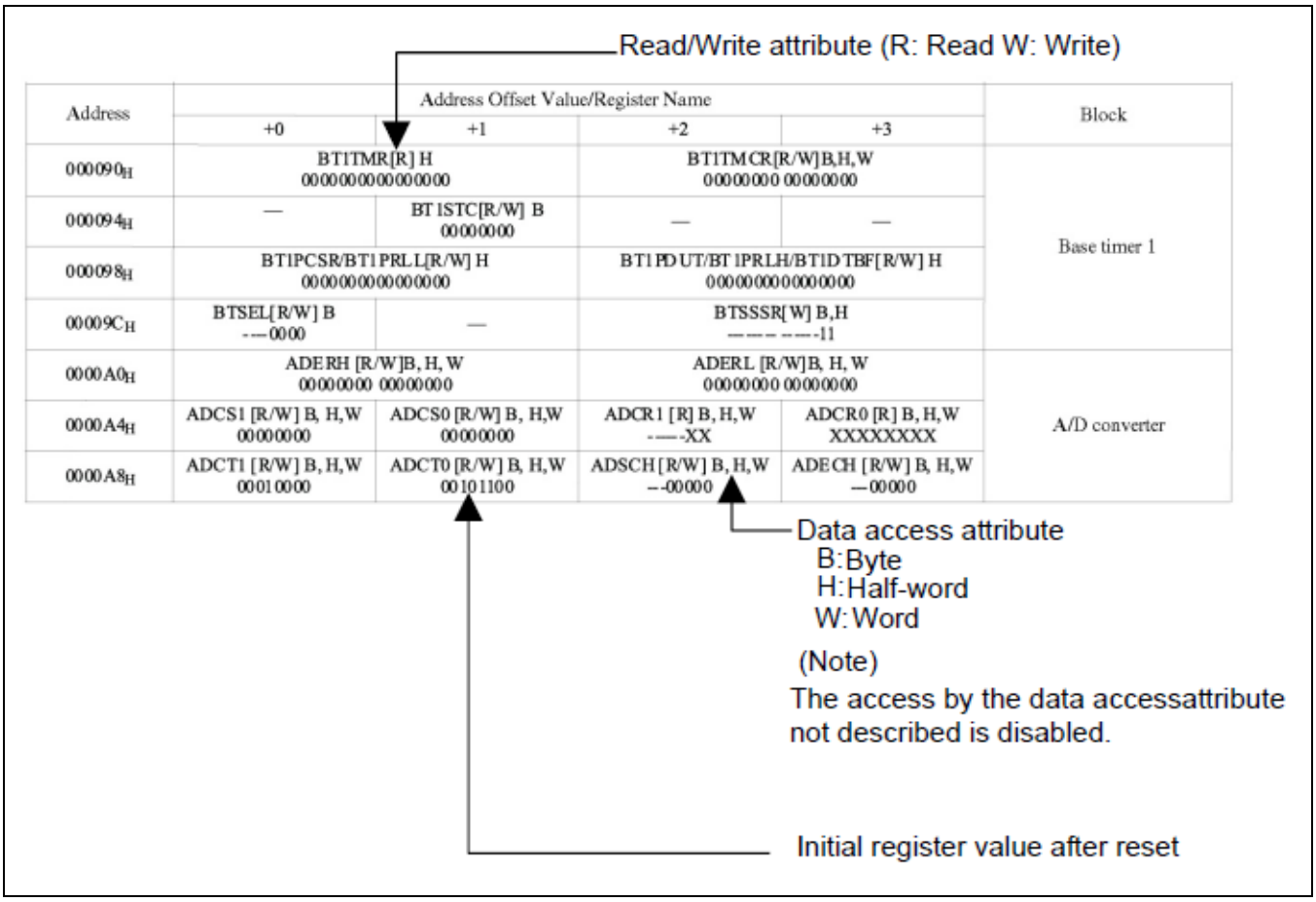
A.3 Pin States by CPU States

A.1 I/O Map

I/O map is shown.

The following I/O map shows the relationship between memory space and registers for peripheral resources.

Figure A-1 Legend of I/O Map



The initial register value after reset indicates as follows:

"1": Initial value "1"

"0": Initial value "0"

"X": Initial value undefined

"-": Reserved bit/Undefined bit

"*": Initial value "0" or "1" according to the setting

Note: The access to addresses not described is disabled.

Table A 1. I/O Map

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000000H	PDR00 [R/W] B,H,W --XXXXXX	—	PDR02 [R/W] B,H,W XXXXXXXX	PDR03 [R/W] B,H,W XXXXXXXX	Port Data Register
000004H	PDR04 [R/W] B,H,W XXXXXXXX	—	—	—	
000008H to 000034H	—	—	—	—	Reserved
000038H	WDTECR0 [R/W] B,H,W ---00000	—	—	—	Watchdog Timer [S]
00003CH	WDTCR0 [R/W] B,H,W -0--0000	WDTCPR0 [W] B,H,W 00000000	WDTCR1 [R] B,H,W ----0110	WDTCPR1 [W] B,H,W 00000000	
000040H	—	—	—	—	Reserved
000044H	DICR [R/W] B -------0	—	—	—	Delayed Interrupt
000048H to 00005CH	—	—	—	—	Reserved
000060H	TMRLRA0 [R/W] H XXXXXXXX XXXXXXXX		TMR0 [R] H XXXXXXXX XXXXXXXX		Reload Timer 0
000064H	TMRLRB0 [R/W] H XXXXXXXX XXXXXXXX		TMCSR0 [R/W] B,H,W 00000000 0-000000		
000068H to 00007CH	—	—	—	—	Reserved
000080H	BT0TMR [R] H 00000000 00000000		BT0TMCR [R/W] H -000--00 -000-000		Base Timer 0
000084H	BT0TMCR2 [R/W] B -------0	BT0STC [R/W] B -0-0-0-0	—	—	
000088H	BT0PCSR/BT0PRLL [R/W] H XXXXXXXX XXXXXXXX		BT0PDUT/BT0PRLH/BT0DTBF [R/W] H XXXXXXXX XXXXXXXX		
00008CH	—	—	—	—	Reserved
000090H	BT1TMR [R] H 00000000 00000000		BT1TMCR [R/W] H -000--00 -000-000		Base Timer 1
000094H	BT1TMCR2 [R/W] B -------0	BT1STC [R/W] B -0-0-0-0	—	—	
000098H	BT1PCSR/BT1PRLL [R/W] H XXXXXXXX XXXXXXXX		BT1PDUT/BT1PRLH/BT1DTBF [R/W] H XXXXXXXX XXXXXXXX		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
00009CH	BTSEL01 [R/W] B ----0000	—	BTSSSR [W] B,H -------- ------11		Base Timer 0,1
0000A0H to 0000FCH	—	—	—	—	Reserved
000100H	TMRLRA1 [R/W] H XXXXXXXX XXXXXXXX		TMR1 [R] H XXXXXXXX XXXXXXXX		Reload Timer 1
000104H	TMRLRB1 [R/W] H XXXXXXXX XXXXXXXX		TMCSR1 [R/W] B, H,W 00000000 0-000000		
000108H	TMRLRA2 [R/W] H XXXXXXXX XXXXXXXX		TMR2 [R] H XXXXXXXX XXXXXXXX		Reload Timer 2
00010CH	TMRLRB2 [R/W] H XXXXXXXX XXXXXXXX		TMCSR2 [R/W] B,H,W 00000000 0-000000		
000110H	TMRLRA3 [R/W] H XXXXXXXX XXXXXXXX		TMR3 [R] H XXXXXXXX XXXXXXXX		Reload Timer 3
000114H	TMRLRB3 [R/W] H XXXXXXXX XXXXXXXX		TMCSR3 [R/W] B,H,W 00000000 0-000000		
000118H	BT2TMR [R] H 00000000 00000000		BT2TMCR [R/W] H -000--00 -000-000		Base Timer 2
00011CH	BT2TMCR2 [R/W] B -------0	BT2STC [R/W] B -0-0-0-0	—	—	
000120H	BT2PCSR/BT2PRLL [R/W] H XXXXXXXX XXXXXXXX		BT2PDUT/BT2PRLH/BT2DTBF [R/W] H XXXXXXXX XXXXXXXX		
000124H	BT3TMR [R] H 00000000 00000000		BT3TMCR [R/W] H -000--00 -000-000		Base Timer 3
000128H	BT3TMCR2 [R/W] B -------0	BT3STC [R/W] B -0-0-0-0	—	—	
00012CH	BT3PCSR/BT3PRLL [R/W] H XXXXXXXX XXXXXXXX		BT3PDUT/BT3PRLH/BT3DTBF [R/W] H XXXXXXXX XXXXXXXX		
000130H	BTSEL23 [R/W] B ----0000	—	BTSSSRA [W] B,H -------- ------11		Base Timer 2,3
000134H to 0001B4H	—	—	—	—	Reserved
0001B8H	—	EPFR65 [R/W] B,H,W --000000	—	—	Extended port function register
0001BCH to 0001C8H	—	—	—	—	Reserved
0001CCH	—	—	EPFR86 [R/W] B,H,W -----0--	—	Extended port function register
0001D0H	EPFR88 [R/W] B,H,W -------0	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0001D4H	—	—	—	—	Reserved
0001D8H	TMRLRA4 [R/W] H XXXXXXXX XXXXXXXX		TMR4 [R] H XXXXXXXX XXXXXXXX		Reload Timer 4
0001DCH	TMRLRB4 [R/W] H XXXXXXXX XXXXXXXX		TMCSR4 [R/W] B,H,W 00000000 0-000000		
0001E0H to 00030CH	—	—	—	—	Reserved
000310H	—	—	MPUCR [R/W] H 000000-0 ----0100		MPU [S] (Only CPU core can access this area)
000314H	—	—	—	—	
000318H	—				
00031CH	—	—	—	—	
000320H	DPVAR [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000324H	—	—	DPVSR [R/W] H -------- 00000--0		
000328H	DEAR [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00032CH	—	—	DESR [R/W] H -------- 00000--0		
000330H	PABR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000334H	—	—	PACR0 [R/W] H 000000-0 00000--0		
000338H	PABR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
00033CH	—	—	PACR1 [R/W] H 000000-0 00000--0		
000340H	PABR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000344H	—	—	PACR2 [R/W] H 000000-0 00000--0		
000348H	PABR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
00034CH	—	—	PACR3 [R/W] H 000000-0 00000--0		
000350H	PABR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000354H	—	—	PACR4 [R/W] H 000000-0 00000--0		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000358H	PABR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				MPU [S] (Only CPU core can access this area)
00035CH	—	—	PACR5 [R/W] H 000000-0 00000--0		
000360H	PABR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000364H	—	—	PACR6 [R/W] H 000000-0 00000--0		
000368H	PABR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
00036CH	—	—	PACR7 [R/W] H 000000-0 00000--0		
000370H to 0003FCH	—	—	—	—	Reserved [S]
000400H	ICSEL0 [R/W] B,H,W -----000	—	ICSEL2 [R/W] B,H,W ------00	ICSEL3 [R/W] B,H,W -------0	DMA request generation and clear
000404H	—	ICSEL5 [R/W] B,H,W -----000	—	—	
000408H	—	—	—	ICSEL11 [R/W] B,H,W -----000	
00040CH	—	ICSEL13 [R/W] B,H,W ------00	ICSEL14 [R/W] B,H,W ------00	ICSEL15 [R/W] B,H,W ----0000	
000410H	ICSEL16 [R/W] B,H,W ------00	ICSEL17 [R/W] B,H,W ----0000	ICSEL18 [R/W] B,H,W ------00	ICSEL19 [R/W] B,H,W -------0	
000414H	ICSEL20 [R/W] B,H,W -------0	ICSEL21 [R/W] B,H,W ------00	ICSEL22 [R/W] B,H,W ------00	ICSEL23 [R/W] B,H,W ------00	
000418H	IRPR0H [R] B,H,W 000-----	IRPR0L [R] B,H,W 00------	IRPR1H [R] B,H,W 00------	IRPR1L [R] B,H,W 00------	Interrupt Request Batch Reading Register
00041CH	IRPR2H [R] B,H,W 00------	IRPR2L [R] B,H,W 00------	IRPR3H [R] B,H,W 00------	IRPR3L [R] B,H,W 00------	
000420H	IRPR4H [R] B,H,W 0000----	IRPR4L [R] B,H,W 0000----	IRPR5H [R] B,H,W 0000----	IRPR5L [R] B,H,W 00------	
000424H	IRPR6H [R] B,H,W -0------	IRPR6L [R] B,H,W 0-0-----	IRPR7H [R] B,H,W 0000----	IRPR7L [R] B,H,W 00000000	
000428H	IRPR8H [R] B,H,W 000-----	IRPR8L [R] B,H,W 0000----	IRPR9H [R] B,H,W 00000000	IRPR9L [R] B,H,W 00000000	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
00042CH	IRPR10H [R] B,H,W 00000000	IRPR10L [R] B,H,W 00000000	IRPR11H [R] B,H,W 0000----	IRPR11L [R] B,H,W 0000----	
000430H	IRPR12H [R] B,H,W 00------	IRPR12L [R] B,H,W 00------	IRPR13H [R] B,H,W 00------	IRPR13L [R] B,H,W 00------	
000434H	IRPR14H [R] B,H,W 00000000	—	IRPR15H [R] B,H,W 0000----	IRPR15L [R] B,H,W -000----	
000438H	—	ICSEL25 [R/W] B,H,W ---00000	—	—	DMA request generation and clear
00043CH	—	—	—	—	Reserved [S]
000440H	ICR00 [R/W] B,H,W ---11111	ICR01 [R/W] B,H,W ---11111	ICR02 [R/W] B,H,W ---11111	ICR03 [R/W] B,H,W ---11111	Interrupt Controller [S]
000444H	ICR04 [R/W] B,H,W ---11111	ICR05 [R/W] B,H,W ---11111	ICR06 [R/W] B,H,W ---11111	ICR07 [R/W] B,H,W ---11111	
000448H	ICR08 [R/W] B,H,W ---11111	ICR09 [R/W] B,H,W ---11111	ICR10 [R/W] B,H,W ---11111	ICR11 [R/W] B,H,W ---11111	
00044CH	ICR12 [R/W] B,H,W ---11111	ICR13 [R/W] B,H,W ---11111	ICR14 [R/W] B,H,W ---11111	ICR15 [R/W] B,H,W ---11111	
000450H	ICR16 [R/W] B,H,W ---11111	ICR17 [R/W] B,H,W ---11111	ICR18 [R/W] B,H,W ---11111	ICR19 [R/W] B,H,W ---11111	
000454H	ICR20 [R/W] B,H,W ---11111	ICR21 [R/W] B,H,W ---11111	ICR22 [R/W] B,H,W ---11111	ICR23 [R/W] B,H,W ---11111	
000458H	ICR24 [R/W] B,H,W ---11111	ICR25 [R/W] B,H,W ---11111	ICR26 [R/W] B,H,W ---11111	ICR27 [R/W] B,H,W ---11111	
00045CH	ICR28 [R/W] B,H,W ---11111	ICR29 [R/W] B,H,W ---11111	ICR30 [R/W] B,H,W ---11111	ICR31 [R/W] B,H,W ---11111	
000460H	ICR32 [R/W] B,H,W ---11111	ICR33 [R/W] B,H,W ---11111	ICR34 [R/W] B,H,W ---11111	ICR35 [R/W] B,H,W ---11111	
000464H	ICR36 [R/W] B,H,W ---11111	ICR37 [R/W] B,H,W ---11111	ICR38 [R/W] B,H,W ---11111	ICR39 [R/W] B,H,W ---11111	
000468H	ICR40 [R/W] B,H,W ---11111	ICR41 [R/W] B,H,W ---11111	ICR42 [R/W] B,H,W ---11111	ICR43 [R/W] B,H,W ---11111	
00046CH	ICR44 [R/W] B,H,W ---11111	ICR45 [R/W] B,H,W ---11111	ICR46 [R/W] B,H,W ---11111	ICR47 [R/W] B,H,W ---11111	
000470H to 00047CH	—	—	—	—	Reserved[S]
000480H	RSTRR [R] B,H,W XXXX--XX	RSTCR [R/W] B,H,W 111----0	STBCR [R/W] B,H,W * 000---11	—	Reset Control [S] Power Control [S] *: Writing STBCR by DMA is forbidden

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000484H	—	—	—	—	Reserved [S]
000488H	DIVR0 [R/W] B,H,W 000-----	—	DIVR2 [R/W] B,H,W 0011----	—	Clock Control [S]
00048CH	—	—	—	—	Reserved [S]
000490H	IORR0 [R/W] B,H,W -0000000	IORR1 [R/W] B,H,W -0000000	IORR2 [R/W] B,H,W -0000000	IORR3 [R/W] B,H,W -0000000	DMA request by peripheral[S]
000494H	IORR4 [R/W] B,H,W -0000000	IORR5 [R/W] B,H,W -0000000	IORR6 [R/W] B,H,W -0000000	IORR7 [R/W] B,H,W -0000000	
000498H	—	—	—	—	
00049CH	—	—	—	—	
0004A0H	—	—	—	—	Reserved
0004A4H	CANPRE [R/W] B,H,W ---00000	—	—	—	CAN prescaler
0004A8H	—	—	CSCFG[R/W]B,H,W ---0----	CMCFG[R/W]B,H,W 00000000	Clock monitor control register
0004ACH	—	—	ADERL0[R/W] B,H -------- 11111111		Analog input control register
0004B0H to 0004C0H	—	—	—	—	Reserved
0004C4H	CUCR1 [R/W] B,H,W -------- ---0--00		CUTD1 [R/W] B,H,W 11000011 01010000		WDT1 calibration
0004C8H	CUTR1 [R] B,H,W -------- 00000000 00000000 00000000				
0004CCH to 0004E4H	—	—	—	—	Reserved
0004E8H	PLL2DIVM [R/W] B,H,W ----0000	PLL2DIVN [R/W] B,H,W -0000000	—	—	PWM Clock control
0004ECH	—	PLL2DIVK [R/W] B,H,W -------0	CLKR2 [R/W] B,H,W 000--000	—	
0004F0H	—	PWMCGRCR0 [R/W] B,H,W 00----00	PWMCGRCR1 [R/W] B,H,W 00000000	PWMCGRCR2 [R/W] B,H,W 00000000	
0004F4H to 00050CH	—	—	—	—	Reserved
000510H	CSELR [R/W] B,H,W -01---00	CMONR [R] B,H,W -01---00	MTMCR [R/W] B,H,W 00001111	—	Clock Control[S]
000514H	PLLCR [R/W] B,H,W -------- 11110000		CSTBR [R/W] B,H,W ----0000	PTMCR [R/W] B,H,W 00------	
000518H	—	—	CPUAR [R/W] B,H,W 0----XXX	—	Reset clock[S]
00051CH	—	—	—	—	Reserved[S]

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000520H	CCPSSELR [R/W] B,H,W -------0	—	—	CCPSDIVR [R/W] B,H,W -000-000	Clock Control 2[S]
000524H	—	CCPLLFBR [R/W] B,H,W -0000000	CCSSFBR0 [R/W] B,H,W --000000	CCSSFBR1 [R/W] B,H,W ---00000	
000528H	—	CCSSCCR0 [R/W] B,H,W ----0000	CCSSCCR1 [R/W] H,W 000----- --------		
00052CH	—	CCCGRCR0 [R/W] B,H,W 00----00	CCCGRCR1 [R/W] B,H,W 00000000	CCCGRCR2 [R/W] B,H,W 00000000	
000530H to 00053CH	—	—	—	—	
000540H to 00054CH	—	—	—	—	Reserved
000550H	EIRR0 [R/W] B,H,W ----XXXX	ENIR0 [R/W] B,H,W ----0000	ELVR0 [R/W] B,H,W -------- 00000000		External Interrupt (INT0 to 3)
000554H to 000568H	—	—	—	—	Reserved
00056CH	—	CSVCR [R/W] B -0-11--0	—	—	Clock Supervisor
000570H to 00057CH	—	—	—	—	Reserved
000580H	REGSEL [R/W] B,H,W 01------	—	—	—	Regulator Control / Low Voltage Detection
000584H	LVD5R [R/W] B,H,W -------1	LVD5F [R/W] B,H,W 00110001	LVD [R/W] B,H,W 01000--0	—	
000588H to 00059CH	—	—	—	—	Reserved
0005A0H	SLPCNT [R/W] B,H,W ------00	—	—	—	Slope Compensation Control
0005A4H	SLPEDGESEL1 [R/W] B,H,W ---0-000 ---0-000		SLPEDGESEL2 [R/W] B,H,W ---0-000 ---0-000		
0005A8H	SLPSWA [R/W] B,H,W -----100	SLPSWB [R/W] B,H,W -----100	SLPSWC [R/W] B,H,W -----100	—	
0005ACH	SLP1A [R/W] H,W 00000000 00000000		SLP1B [R/W] H,W 00000000 00000000		
0005B0H	SLP1C [R/W] H,W 00000000 00000000		SLP3 [R/W] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0005B4H	SLP2A [R/W] H,W 00000000 00000000		SLP2B [R/W] H,W 00000000 00000000		Slope Compensation Control
0005B8H	SLP2C [R/W] H,W 00000000 00000000		SLP4 [R/W] H,W 00000000 00000000		
0005BCH	—	—	SLPDADR [R/W] H,W ------00 00000000		
0005C0H	SLPSTMSEL1 [R/W] B,H,W -000-000 -----000		SLPSTMSEL2 [R/W] B,H,W -000-000 --------		
0005C4H	SLPCVE [R] H,W ------XX XXXXXXXX		—	—	
0005C8H to 0005DCH	—	—	—	—	Reserved
0005E0H	CMPCTL0 [R/W] B,H,W --001000	CMPCTL1 [R/W] B,H,W --001000	CMPCTL2 [R/W] B,H,W --001000	CMPDIV [R/W] B,H,W ------00	Comparator Control
0005E4H	CMPDACR0 [R/W] B,H,W -------0	—	CMPDADR0 [R/W] H,W ------00 00000000		
0005E8H	CMPDACR1 [R/W] B,H,W -------0	—	CMPDADR1 [R/W] H,W ------00 00000000		
0005ECH	CMPDACR2 [R/W] B,H,W -------0	—	CMPDADR2 [R/W] H,W ------00 00000000		
0005F0H	CMPINT [R/W] B,H,W -----000	—	—	—	
0005F4H	CMPST [R] B,H,W -----XXX	—	—	—	
0005F8H to 0005FCH	—	—	—	—	Reserved
000600H to 0006FCH	—	—	—	—	Reserved[S]
000700H to 00070CH	—	—	—	—	Reserved
000710H	BPCCRA [R/W] B 00000000	BPCCRB [R/W] B 00000000	BPCCRC [R/W] B 00000000	—	Bus Performance Counter
000714H	BPCTRA [R/W] W 00000000 00000000 00000000 00000000				
000718H	BPCTRB [R/W] W 00000000 00000000 00000000 00000000				
00071CH	BPCTRC [R/W] W 00000000 00000000 00000000 00000000				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000720H to 0007F8H	—	—	—	—	Reserved
0007FCH	BMODR [R] B, H, W XXXXXXXX	—	—	—	Operation Mode
000800H to 00083CH	—	—	—	—	Reserved[S]
000840H	FCTLR [R/W] H -0--1000 0--0----		—	FSTR [R/W] B -----001	Flash Memory Register [S]
000844H to 000854H	—	—	—	—	Reserved[S]
000858H	—	—	WREN [R/W] H 00000000 00000000		Wild Register [S]
00085CH to 00087CH	—	—	—	—	Reserved[S]
000880H	WRAR00 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild Register [S]
000884H	WRDR00 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000888H	WRAR01 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
00088CH	WRDR01 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000890H	WRAR02 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
000894H	WRDR02 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000898H	WRAR03 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
00089CH	WRDR03 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008A0H	WRAR04 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008A4H	WRDR04 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008A8H	WRAR05 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008ACH	WRDR05 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0008B0H	WRAR06 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild Register [S]
0008B4H	WRDR06 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008B8H	WRAR07 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008BCH	WRDR07 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008C0H	WRAR08 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008C4H	WRDR08 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008C8H	WRAR09 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008CCH	WRDR09 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008D0H	WRAR10 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008D4H	WRDR10 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008D8H	WRAR11 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008DCH	WRDR11 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008E0H	WRAR12 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008E4H	WRDR12 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008E8H	WRAR13 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				]
0008ECH	WRDR13 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008F0H	WRAR14 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008F4H	WRDR14 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0008F8H	WRAR15 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
0008FCH	WRDR15 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000900H	TPUUNLOCK [R/W] W 00000000 00000000 00000000 00000000				Time Protection Unit[S]
000904H	TPULST [R] B,H,W -------0	—	TPUVST [R/W] B,H,W -----000	—	
000908H	TPUCFG [R/W] B,H,W -------0 0-000000 -------- -------0				
00090CH	TPUTIR [R] B,H,W 00000000	—	—	—	
000910H	TPUTST [R] B,H,W 00000000	—	—	—	
000914H	TPUTIE [R/W] B,H,W 00000000	—	—	—	
000918H	TPUTMID [R] B,H,W 00000000 00000000 00000000 00000000				
00091CH to 00092CH	—	—	—	—	
000930H	TPUTCN00 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000934H	TPUTCN01 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000938H	TPUTCN02 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
00093CH	TPUTCN03 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000940H	TPUTCN04 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000944H	TPUTCN05 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000948H	TPUTCN06 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
00094CH	TPUTCN07 [R/W] B,H,W 000000-- 00000000 00000000 00000000				
000950H	TPUTCN10 [R/W] B,H,W ---00000	—	—	—	
000954H	TPUTCN11 [R/W] B,H,W ---00000	—	—	—	
000958H	TPUTCN12 [R/W] B,H,W ---00000	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
00095CH	TPUTCN13 [R/W] B,H,W ---00000	—	—	—	Time Protection Unit[S]
000960H	TPUTCN14 [R/W] B,H,W ---00000	—	—	—	
000964H	TPUTCN15 [R/W] B,H,W ---00000	—	—	—	
000968H	TPUTCN16 [R/W] B,H,W ---00000	—	—	—	
00096CH	TPUTCN17 [R/W] B,H,W ---00000	—	—	—	
000970H	TPUTCC0 [R] B,H,W -------- 00000000 00000000 00000000				
000974H	TPUTCC1 [R] B,H,W -------- 00000000 00000000 00000000				
000978H	TPUTCC2 [R] B,H,W -------- 00000000 00000000 00000000				
00097CH	TPUTCC3 [R] B,H,W -------- 00000000 00000000 00000000				
000980H	TPUTCC4 [R] B,H,W -------- 00000000 00000000 00000000				
000984H	TPUTCC5 [R] B,H,W -------- 00000000 00000000 00000000				
000988H	TPUTCC6 [R] B,H,W -------- 00000000 00000000 00000000				
00098CH	TPUTCC7 [R] B,H,W -------- 00000000 00000000 00000000				
000990H to 0009FCH	—	—	—	—	
000A00H to 000BECH	—	—	—	—	Reserved
000BF0H	HSCFR [R/W] B,H,W -------- ------00 00000000 00000000				OCDU
000BF4H	—	—	—	—	
000BF8H	—	—	MBR [R/W] B,H,W 00------ XXXXXXXX		
000BFCH	—	—	UER [W] B,H,W -------- -------X		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000C00H	DCCR0 [R/W] W 0----000 --00--00 00000000 0-000000				DMA Controller[S]
000C04H	DCSR0 [R/W] H 0------- -----000		DTCR0 [R/W] H 00000000 00000000		
000C08H	DSAR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C0CH	DDAR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C10H	DCCR1 [R/W] W 0----000 --00--00 00000000 0-000000				
000C14H	DCSR1 [R/W] H 0------- -----000		DTCR1 [R/W] H 00000000 00000000		
000C18H	DSAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C1CH	DDAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C20H	DCCR2 [R/W] W 0----000 --00--00 00000000 0-000000				
000C24H	DCSR2 [R/W] H 0------- -----000		DTCR2 [R/W] H 00000000 00000000		
000C28H	DSAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C2CH	DDAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C30H	DCCR3 [R/W] W 0----000 --00--00 00000000 0-000000				
000C34H	DCSR3 [R/W] H 0------- -----000		DTCR3 [R/W] H 00000000 00000000		
000C38H	DSAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C3CH	DDAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C40H	DCCR4 [R/W] W 0----000 --00--00 00000000 0-000000				
000C44H	DCSR4 [R/W] H 0------- -----000		DTCR4 [R/W] H 00000000 00000000		
000C48H	DSAR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C4CH	DDAR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C50H	DCCR5 [R/W] W 0----000 --00--00 00000000 0-000000				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000C54H	DCSR5 [R/W] H 0------- -----000		DTCR5 [R/W] H 00000000 00000000		DMA Controller[S]
000C58H	DSAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C5CH	DDAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C60H	DCCR6 [R/W] W 0----000 --00--00 00000000 0-000000				
000C64H	DCSR6 [R/W] H 0------- -----000		DTCR6 [R/W] H 00000000 00000000		
000C68H	DSAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C6CH	DDAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C70H	DCCR7 [R/W] W 0----000 --00--00 00000000 0-000000				
000C74H	DCSR7 [R/W] H 0------- -----000		DTCR7 [R/W] H 00000000 00000000		
000C78H	DSAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C7CH	DDAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000C80H to 000DF0H	—	—	—	—	Reserved[S]
000DF4H	—	—	DNMIR [R/W] B 0------0	DILVR [R/W] B ---11111	DMA Controller[S]
000DF8H	DMACR[R/W] W 0------- -------- 0------- --------				
000DFCH	—	—	—	—	Reserved[S]
000E00H	—	—	DDR02 [R/W] B,H,W 00000000	DDR03 [R/W] B,H,W 00000000	Data Direction Register
000E04H	DDR04 [R/W] B,H,W 00000000	—	—	—	
000E08H to 000E1CH	—	—	—	—	Reserved
000E20H	PFR00 [R/W] B,H,W --000000	—	PFR02 [R/W] B,H,W 00000000	PFR03 [R/W] B,H,W 00000000	Port Function Register
000E24H	PFR04 [R/W] B,H,W 00000000	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
000E28H to 000E3CH	—	—	—	—	Reserved
000E40H	PDDR00 [R] B,H,W --XXXXXX	—	PDDR02 [R] B,H,W XXXXXXXX	PDDR03 [R] B,H,W XXXXXXXX	Port Direct Read Register
000E44H	PDDR04 [R] B,H,W XXXXXXXX	—	—	—	
000E48H to 000E5CH	—	—	—	—	Reserved
000E60H	—	—	EPFR02 [R/W] B,H,W ---00000	—	Extended Port Function Register
000E64H to 000E68H	—	—	—	—	
000E70H	EPFR16 [R/W] B,H,W -------0	—	—	—	
000E74H	—	—	—	—	
000E78H	—	—	EPFR26 [R/W] B,H,W -0-0-0-0	—	
000E7CH	—	—	—	—	
000E80H	—	—	—	EPFR35 [R/W] B,H,W --000000	
000E84H to 000EBCH	—	—	—	—	Reserved
000EC0H	PPER00 [R/W] B,H,W --000000	—	PPER02 [R/W] B,H,W 00000000	PPER03 [R/W] B,H,W 00000000	Port Pull-up/down Enable Register
000EC4H	PPER04 [R/W] B,H,W 00000000	—	—	—	
000EC8H to 000F3CH	—	—	—	—	Reserved
000F40H	PORTEN [R/W] B,H,W -------0	—	—	—	Port Enable Register
000F44H	KEYCDR [R/W] H 00000000 00000000		—	—	Key Code Register
000F48H to 000FFCH	—	—	—	—	Reserved
001000H	SACR [R/W] B,H,W -------0	PICD [R/W] B,H,W ----0011	—	—	Clock Control
001004H to 00112CH	—	—	—	—	Reserved

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001130H	—	—	—	CRCCR [R/W] B,H,W -0000000	CRC calculation unit
001134H	CRCINIT [R/W] B,H,W 11111111 11111111 11111111 11111111				
001138H	CRCIN [R/W] B,H,W 00000000 00000000 00000000 00000000				
00113CH	CRCR [R] B,H,W 11111111 11111111 11111111 11111111				
001140H to 001200H	—	—	—	—	Reserved
001204H	CPCLRB0/CPCLR0 [W] H,W 11111111 11111111		TCDT0 [R/W] H,W 00000000 00000000		Free-run Timer 0 (16bit)
001208H	TCCS0 [R/W] B,H,W 00000000 01000000 ----0000 --------				
00120CH to 001278H	—	—	—	—	Reserved
00127CH	IPCP0 [R] H,W 00000000 00000000		—	—	Input Capture 0 (16bit)
001280H	ICS01 [R/W] B,H,W -------0 -0-0--00		—	LSYNS [R/W] B,H,W -----000	
001284H to 001300H	—	—	—	—	Reserved
001304H	ADTSS0[R/W]B,H,W -------0	—	—	—	12 bit A/D converter
001308H	ADTSE0[R/W] B,H,W -------- -------- -------- 00000000				
00130CH	—		—		
001310H	—		—		
001314H	—		—		
001318H	—		—		
00131CH to 001348H	—	—	—	—	
00134CH	ADTCS0[R/W] B,H,W 00000000 --------		ADTCS1[R/W] B,H,W 00000000 --------		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001350H	ADTCS2[R/W] B,H,W 00000000 --------		ADTCS3[R/W] B,H,W 00000000 --------		12 bit A/D converter
001354H	ADTCS4[R/W] B,H,W 00000000 --------		ADTCS5[R/W] B,H,W 00000000 --------		
001358H	ADTCS6[R/W] B,H,W 00000000 --------		ADTCS7[R/W] B,H,W 00000000 --------		
00135CH to 001388H	—	—	—	—	
00138CH	ADTCD0[R] B,H,W 10--0000 00000000		ADTCD1[R] B,H,W 10--0000 00000000		
001390H	ADTCD2[R] B,H,W 10--0000 00000000		ADTCD3[R] B,H,W 10--0000 00000000		
001394H	ADTCD4[R] B,H,W 10--0000 00000000		ADTCD5[R] B,H,W 10--0000 00000000		
001398H	ADTCD6[R] B,H,W 10--0000 00000000		ADTCD7[R] B,H,W 10--0000 00000000		
00139CH to 0013C8H	—	—	—	—	
0013CCH	ADTECS0[R/W] B,H,W -------0 ---00000		ADTECS1[R/W] B,H,W -------0 ---00000		
0013D0H	ADTECS2[R/W] B,H,W -------0 ---00000		ADTECS3[R/W] B,H,W -------0 ---00000		
0013D4H	ADTECS4[R/W] B,H,W -------0 ---00000		ADTECS5[R/W] B,H,W -------0 ---00000		
0013D8H	ADTECS6[R/W] B,H,W -------0 ---00000		ADTECS7[R/W] B,H,W -------0 ---00000		
0013DCH to 001454H	—	—	—	—	
001458H	ADPRTF0[R] B,H,W -------- -------- -------- 00000000				
00145CH	—				
001460H	ADCS0[R] B,H,W 0------- --------		ADCH0[R] B,H,W ---00000	ADMD0[R/W] B,H,W 0---0000	
001464H	ADSTPCS0[R/W] B,H,W 00000000	ADSTPCS1[R/W] B,H,W 00000000	—	—	
001468H	—	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
00146CH	EADTCS0 [R/W] B,H,W ----0000	EADTCS1 [R/W] B,H,W ----0000	EADTCS2 [R/W] B,H,W ----0000	EADTCS3 [R/W] B,H,W ----0000	
001470H	EADTCS4 [R/W] B,H,W ----0000	EADTCS5 [R/W] B,H,W ----0000	EADTCS6 [R/W] B,H,W ----0000	EADTCS7 [R/W] B,H,W ----0000	
001474H to 00174CH	—	—	—	—	Reserved
001750H	SCR0[R/W] B,H,W 0--00000	SMR0[R/W] B,H,W 000-00-0	SSR0[R/W] B,H,W 0-000011	ESCR0[R/W] B,H,W 00000000	Multi-UART0 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because CSIO mode is not set immediately after reset. *3: Reserved because LIN2.1 mode is not set immediately after reset. * 4: Byte access in CSIO mode is prohibited.
001754H	— /(RDR10/(TDR10))[R/W] H,W -------- --------*2		RDR00/(TDR00)[R/W] B,H,W*4 -------0 00000000*1		
001758H	SACSR0[R/W] B,H,W 00-----0 0--00000		STMR0[R] B,H,W 00000000 00000000		
00175CH	STMCR0[R/W] B,H,W 00000000 00000000		— /(SCSCR0/SFUR0)[R/W] B,H,W -------- --------*2 *3		
001760H	— /(SCSTR30)/ (LAMSR0) [R/W] B,H,W --------*2	— /(SCSTR20)/ (LAMCR0) [R/W] B,H,W --------*2	— /(SCSTR10) /(SFLR10) [R/W] B,H,W --------*2	— /(SCSTR00)/ (SFLR00) [R/W] B,H,W --------*2	
001764H	—	—	—	— /(SCSFR00) [R/W] B,H,W --------*2	
001768H	—/(LAMESR0) [R/W] B,H,W --------*2	—/(LAMERT0) [R/W] B,H,W --------*2	—/(TBYTE10)/ (LAMIER0) [R/W] B,H,W --------*2	TBYTE00/(LAMRID0)/ (LAMTID0) [R/W] B,H,W 00000000	
00176CH	BGR0[R/W] H, W 00000000 00000000		—	—	
001770H	FCR10[R/W] B,H,W ---00100	FCR00[R/W] B,H,W -0000000	FBYTE0[R/W] B,H,W 00000000 00000000		
001774H	FTICR0[R/W] B,H,W 00000000 00000000		—	—	
001778H	SCR1[R/W] B,H,W 0--00000	SMR1[R/W] B,H,W 000-00-0	SSR1[R/W] B,H,W 0-000011	ESCR1[R/W] B,H,W 00000000	Multi-UART1 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because CSIO mode is not set
00177CH	— /(RDR11/(TDR11))[R/W] H,W -------- --------*2		RDR01/(TDR01)[R/W] B,H,W *4 -------0 00000000*1		
001780H	SACSR1[R/W] B,H,W 00-----0 0--00000		STMR1[R] B,H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
001784H	STMCR1[R/W] B,H,W 00000000 00000000		— /(SCSCR1/SFUR1)[R/W] B,H,W -------- --------*2 *3		immediately after reset. *3: Reserved because LIN2.1 mode is not set immediately after reset. * 4: Byte access in CSIO mode is prohibited.
001788H	— /(SCSTR31)/ (LAMSR1) [R/W] B,H,W --------*2	— /(SCSTR21)/ (LAMCR1) [R/W] B,H,W --------*2	— /(SCSTR11)/ (SFLR11) [R/W] B,H,W --------*2	— /(SCSTR01)/ (SFLR01) [R/W] B,H,W --------*2	
00178CH	—	—	—	— /(SCSFR01) [R/W] B,H,W --------*2	
001790H	—/(LAMESR1) [R/W] B,H,W --------*2	—/(LAMERT1) [R/W] B,H,W --------*2	—/(TBYTE11)/ (LAMIER1) [R/W] B,H,W --------*2	TBYTE01/(LAMRID1)/ (LAMTID1) [R/W] B,H,W 00000000	
001794H	BGR1[R/W] H,W 00000000 00000000		—	—	
001798H	FCR11[R/W] B,H,W ---00100	FCR01[R/W] B,H,W -0000000	FBYTE1[R/W] B,H,W 00000000 00000000		
00179CH	FTICR1[R/W] B,H,W 00000000 00000000		—	—	
0017A0H	SCR2[R/W] B,H,W 0--00000	SMR2[R/W] B,H,W 000-00-0	SSR2[R/W] B,H,W 0-000011	ESCR2[R/W] B,H,W 00000000	Multi-UART2 *1: Byte access is possible only for access to lower 8 bits. *2: Reserved because CSIO mode is not set immediately after reset. *3: Reserved because LIN2.1 mode is not set immediately after reset. * 4: Byte access in CSIO mode is prohibited.
0017A4H	— /(RDR12/(TDR12))[R/W] H,W -------- --------*2		RDR02/(TDR02)[R/W] B,H,W*4 -------0 00000000*1		
0017A8H	SACSR2[R/W] B,H,W 00-----0 0--00000		STMR2[R] B,H,W 00000000 00000000		
0017ACH	STMCR2[R/W] B,H,W 00000000 00000000		— /(SCSCR2/SFUR2)[R/W] B,H,W -------- --------*2 *3		
0017B0H	— /(SCSTR32)/ (LAMSR2) [R/W] B,H,W --------*2	— /(SCSTR22)/ (LAMCR2) [R/W] B,H,W --------*2	— /(SCSTR12)/ (SFLR12) [R/W] B,H,W --------*2	— /(SCSTR02)/ (SFLR02) [R/W] B,H,W --------*2	
0017B4H	—	—	—	— /(SCSFR02) [R/W] B,H,W --------*2	
0017B8H	—/(LAMESR2) [R/W] B,H,W --------*2	—/(LAMERT2) [R/W] B,H,W --------*2	—/(TBYTE12)/ (LAMIER2) [R/W] B,H,W --------*2	TBYTE02/(LAMRID2)/ (LAMTID2) [R/W] B,H,W 00000000	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0017BCH	BGR2[R/W] H, W 00000000 00000000		—	—	
0017C0H	FCR12[R/W] B,H,W ---00100	FCR02[R/W] B,H,W -0000000	FBYTE2[R/W] B,H,W 00000000 00000000		
0017C4H	FTICR2[R/W] B,H,W 00000000 00000000		—	—	
0017C8H to 0020FCH	—	—	—	—	Reserved
002100H	CTRLR1 [R/W] B,H,W -------- 000-0001		STATR1 [R/W] B,H,W -------- 00000000		CAN (64msb)
002104H	ERRCNT1 [R] B,H,W 00000000 00000000		BTR1 [R/W] B,H,W -0100011 00000001		
002108H	INTR1 [R] H,W 00000000 00000000		TESTR1 [R/W] B,H,W -------- X00000--		
00210CH	BRPER1 [R/W] B,H,W -------- ----0000		—	—	
002110H	IF1CREQ1 [R/W] B,H,W 0------- 00000001		IF1CMSK1 [R/W] B,H,W -------- 00000000		
002114H	IF1MSK21 [R/W] B,H,W 11-11111 11111111		IF1MSK11 [R/W] B,H,W 11111111 11111111		
002118H	IF1ARB21 [R/W] B,H,W 00000000 00000000		IF1ARB11 [R/W] B,H,W 00000000 00000000		
00211CH	IF1MCTR1 [R/W] B,H,W 00000000 0---0000		—	—	
002120H	IF1DTA11 [R/W] B,H,W 00000000 00000000		IF1DTA21 [R/W] B,H,W 00000000 00000000		
002124H	IF1DTB11 [R/W] B,H,W 00000000 00000000		IF1DTB21 [R/W] B,H,W 00000000 00000000		
002128H	—	—	—	—	
00212CH	—	—	—	—	
002130H, 002134H	Reserved (IF1 data mirror)				
002138H	—	—	—	—	
00213CH	—	—	—	—	
002140H	IF2CREQ1 [R/W] B,H,W 0------- 00000001		IF2CMSK1 [R/W] B,H,W -------- 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
002144H	IF2MSK21 [R/W] B,H,W 11-11111 11111111		IF2MSK11 [R/W] B,H,W 11111111 11111111		CAN (64msb)
002148H	IF2ARB21 [R/W] B,H,W 00000000 00000000		IF2ARB11 [R/W] B,H,W 00000000 00000000		
00214CH	IF2MCTR1 [R/W] B,H,W 00000000 0---0000		—	—	
002150H	IF2DTA11 [R/W] B,H,W 00000000 00000000		IF2DTA21 [R/W] B,H,W 00000000 00000000		
002154H	IF2DTB11 [R/W] B,H,W 00000000 00000000		IF2DTB21 [R/W] B,H,W 00000000 00000000		
002158H	—	—	—	—	
00215CH	—	—	—	—	
002160H, 002164H	Reserved (IF2 data mirror)				
002168H to 00217CH	—				
002180H	TREQR21 [R] B,H,W 00000000 00000000		TREQR11 [R] B,H,W 00000000 00000000		
002184H	TREQR41 [R] B,H,W 00000000 00000000		TREQR31 [R] B,H,W 00000000 00000000		
002188H	—	—	—	—	
00218CH	—	—	—	—	
002190H	NEWDT21 [R] B,H,W 00000000 00000000		NEWDT11 [R] B,H,W 00000000 00000000		
002194H	NEWDT41 [R] B,H,W 00000000 00000000		NEWDT31 [R] B,H,W 00000000 00000000		
002198H	—	—	—	—	
00219CH	—	—	—	—	
0021A0H	INTPND21 [R] B,H,W 00000000 00000000		INTPND11 [R] B,H,W 00000000 00000000		
0021A4H	INTPND41 [R] B,H,W 00000000 00000000		INTPND31 [R] B,H,W 00000000 00000000		
0021A8H	—	—	—	—	
0021ACH	—	—	—	—	

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0021B0H	MSGVAL21 [R] B,H,W 00000000 00000000		MSGVAL11 [R] B,H,W 00000000 00000000		CAN (64msb)
0021B4H	MSGVAL41 [R] B,H,W 00000000 00000000		MSGVAL31 [R] B,H,W 00000000 00000000		
0021B8H	—	—	—	—	
0021BCH	—	—	—	—	
0021C0H to 0022FCH	—				Reserved
002300H	DFCTLR [R/W] B,H,W -0------ --------		—	DFSTR [R/W] B,H,W -----001	WorkFlash
002304H	—	—	—	—	
002308H	FLIFCTLR [R/W] B,H,W ---0--00	—	FLIFFER1 [R/W] B,H,W --------	FLIFFER2 [R/W] B,H,W --------	Flash / WorkFlash
00230CH to 0023FCH	—				Reserved
002400H	SEEARX [R] B,H,W -0000000 00000000		DEEARX [R] B,H,W -0000000 00000000		XBS RAM ECC controll
002404H	EECSRX [R/W] B,H,W ----00-0	—	EFEARX [R/W] B,H,W -0000000 00000000		
002408H	—	EFECRX [R/W] B,H,W -------0 00000000 00000000			
00240CH to 003008H	—				Reserved
00300CH	TEAR0X[R] B,H,W 000----- -------- -0000000 00000000				RAM diagnosis XBS RAM
003010H	TEAR1X[R] B,H,W 000----- -------- -0000000 00000000				
003014H	TEAR2X[R] B,H,W 000----- -------- -0000000 00000000				
003018H	TAEARX [R/W] B,H,W -1111111 11111111		TASARX [R/W] B,H,W -0000000 00000000		
00301CH	TFECRX [R/W] B,H,W ----0000	TICRX [R/W] B,H,W ----0000	TTCRX [R/W] B,H,W ------00 00001100		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
003020H	TSRCRX [W] B,H,W 0-------	—	—	TKCCRX [R/W] B,H,W 00----00	
003024H to 0030FCH	—				Reserved
003100H	BUSDIGSR0[R/W] H,W 00000000 0-----00		BUSDIGSR1[R/W] H,W 00000000 0-----00		Bus diagnosis
003104H	BUSDIGSR2[R/W] H,W 00000000 0-----00		BUSTSTR0[R/W] H,W 00--0000 00000000		
003108H	BUSADR0 [R] W 00000000 00000000 00000000 00000000				
00310CH	BUSADR1 [R] W 00000000 00000000 00000000 00000000				
003110H	BUSADR2 [R] W 00000000 00000000 00000000 00000000				
003114H	—	—	BUSDIGSR3[R/W] H,W 00000000 0-----00		
003118H	BUSDIGSR4[R/W] H,W 00000000 0-----00		BUSTSTR1[R/W] H,W 00--000- 00000000		
00311CH	—	—	—	—	
003120H	BUSADR3 [R] W 00000000 00000000 00000000 00000000				
003124H	BUSADR4 [R] W 00000000 00000000 00000000 00000000				
003128H to 00313CH	—				Reserved
003140H	PWCINIT0 [R/W] B,H,W -------0	PWCC0 [R/W] B,H,W ---00000	—	—	PWC 0ch
003144H	PWCCPCLRB00/PWCCPCLR00 [W] H,W 11111111 11111111		PWCTCDT00 [R/W] H,W 00000000 00000000		
003148H	PWCTCCS00 [R/W] B,H,W -0000000 0--0-000		—	—	
00314CH	PWCCPCLRB10/PWCCPCLR10 [W] H,W 11111111 11111111		PWCTCDT10 [R/W] H,W 00000000 00000000		
003150H	PWCTCCS10 [R/W] B,H,W -0000000 0--0----		—	—	
003154H	PWCDBR00 [R] H,W 00000000 00000000		PWCDBR10 [R] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
003158H	PWCDBR20 [R] H,W 00000000 00000000		PWCDBR30 [R] H,W 00000000 00000000		PWC 0ch
00315CH	PWCDBS0 [R] B,H,W -000-000 -000-000		—	—	
003160H	PWCBFIRQF0 [R/W] B,H,W 00000000 00000000		—	—	
003164H	PWCBFIRQC0 [R/W] B,H,W 00000000 00000000 00000000			—	
003168H	PWCCUC00 [R/W] H,W 00000000 00000000		PWCCLC00 [R/W] H,W 00000000 00000000		
00316CH	PWCCUC10 [R/W] H,W 00000000 00000000		PWCCLC10 [R/W] H,W 00000000 00000000		
003170H	PWCCUC20 [R/W] H,W 00000000 00000000		PWCCLC20 [R/W] H,W 00000000 00000000		
003174H	PWCCUC30 [R/W] H,W 00000000 00000000		PWCCLC30 [R/W] H,W 00000000 00000000		
003178H	PWCINIT1 [R/W] B,H,W -------0	PWCC1 [R/W] B,H,W ---00000	—	—	PWC 1ch
00317CH	PWCCPCLRB01/PWCCPCLR01 [W] H,W 11111111 11111111		PWCTCDT01 [R/W] H,W 00000000 00000000		
003180H	PWCTCCS01 [R/W] B,H,W -0000000 0--0-000		—	—	
003184H	PWCCPCLRB11/PWCCPCLR11 [W] H,W 11111111 11111111		PWCTCDT11 [R/W] H,W 00000000 00000000		
003188H	PWCTCCS11 [R/W] B,H,W -0000000 0--0----		—	—	
00318CH	PWCDBR01 [R] H,W 00000000 00000000		PWCDBR11 [R] H,W 00000000 00000000		
003190H	PWCDBR21 [R] H,W 00000000 00000000		PWCDBR31 [R] H,W 00000000 00000000		
003194H	PWCDBS1 [R] B,H,W -000-000 -000-000		—	—	
003198H	PWCBFIRQF1 [R/W] B,H,W 00000000 00000000		—	—	
00319CH	PWCBFIRQC1 [R/W] B,H,W 00000000 00000000 00000000			—	
0031A0H	PWCCUC01 [R/W] H,W 00000000 00000000		PWCCLC01 [R/W] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0031A4H	PWCCUC11 [R/W] H,W 00000000 00000000		PWCCLC11 [R/W] H,W 00000000 00000000		PWC 1ch
0031A8H	PWCCUC21 [R/W] H,W 00000000 00000000		PWCCLC21 [R/W] H,W 00000000 00000000		
0031ACH	PWCCUC31 [R/W] H,W 00000000 00000000		PWCCLC31 [R/W] H,W 00000000 00000000		
0031B0H to 0031BCH	—		—		Reserved
0031C0H	PWMTCGS [R/W] B,H,W ------00	—	—	PWMTCGSE [R/W] B,H,W ------00	PWM master clock generation
0031C4H	PWMCPCLRB0/PWMCPCLR0 [W] H,W 11111111 11111111		PWMTCDT0 [R/W] H,W 00000000 00000000		
0031C8H	PWMTCCS0 [R/W] B,H,W -0000000 0100-000 ----0000			—	
0031CCH	PWMCPCLRB1/PWMCPCLR1 [W] H,W 11111111 11111111		PWMTCDT1 [R/W] H,W 00000000 00000000		
0031D0H	PWMTCCS1 [R/W] B,H,W -0000000 0100---- ----0000			—	
0031D4H	PWMTRC [R/W] B,H,W -0000000 -000-000 -----000			—	
0031D8H	PWMSYNCP0 [R/W] H,W 00000000 00000000		PWMSYNCP1 [R/W] H,W 00000000 00000000		
0031DCH	PWMSEVCON [R/W] B,H,W ---00000 ---00000		—	—	
0031E0H	PWMSEVST [R/W] B,H,W ------00	—	—	—	
0031E4H	PWMSEVCP0 [R/W] H,W 00000000 00000000		PWMSEVCP1 [R/W] H,W 00000000 00000000		
0031E8H	PWMMCD0B [R/W] H,W 00000000 00000000		PWMMCD1B [R/W] H,W 00000000 00000000		
0031ECH	PWMST0 [R/W] B,H,W ----0000	PWMST1 [R/W] B,H,W ----0000	PWMST2 [R/W] B,H,W ----0000	PWMFLTST [R] B,H,W --XXXXXX	
0031F0H	PWMCMD [R/W] H,W 00000000 00000000		PWMPCGS [R/W] B,H,W -----000	—	PWM PWM generation
0031F4H	PWMPCN01 [R/W] B,H,W 00000000 ------00		—	—	
0031F8H	PWMCC0B [R/W] H,W 00000000 00000000		PWMCP0B [R/W] H,W 00000000 00000000		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0031FCH	PWMCD0B [R/W] H,W 00000000 00000000		PWMPTMR0 [R] H,W 00000000 00000000		PWM PWM generation
003200H	PWMCC1B [R/W] H,W 00000000 00000000		PWMCP1B [R/W] H,W 00000000 00000000		
003204H	PWMCD1B [R/W] H,W 00000000 00000000		PWMPTMR1 [R] H,W 00000000 00000000		
003208H	PWMPCN23 [R/W] B,H,W 00000000 ------00		—	—	
00320CH	PWMCC2B [R/W] H,W 00000000 00000000		PWMCP2B [R/W] H,W 00000000 00000000		
003210H	PWMCD2B [R/W] H,W 00000000 00000000		PWMPTMR2 [R] H,W 00000000 00000000		
003214H	PWMCC3B [R/W] H,W 00000000 00000000		PWMCP3B [R/W] H,W 00000000 00000000		
003218H	PWMCD3B [R/W] H,W 00000000 00000000		PWMPTMR3 [R] H,W 00000000 00000000		
00321CH	PWMPCN45 [R/W] B,H,W 00000000 ------00		—	—	
003220H	PWMCC4B [R/W] H,W 00000000 00000000		PWMCP4B [R/W] H,W 00000000 00000000		
003224H	PWMCD4B [R/W] H,W 00000000 00000000		PWMPTMR4 [R] H,W 00000000 00000000		
003228H	PWMCC5B [R/W] H,W 00000000 00000000		PWMCP5B [R/W] H,W 00000000 00000000		
00322CH	PWMCD5B [R/W] H,W 00000000 00000000		PWMPTMR5 [R] H,W 00000000 00000000		
003230H	PWMFLTCON00 [R/W] B,H,W 00000000 -0000-00		PWMFLTCON01 [R/W] B,H,W 00000000 -0000-00		PWM fault
003234H	PWMFLTRCON0 [R/W] B,H,W -000-000 -000-000		PWMFLTCAPCON0 [R/W] B,H,W --00--00 -0000000		
003238H	PWMFLTSR0 [R/W] B,H,W -------0 -------0		—	—	
00323CH	PWMCAPITH0 [R/W] H,W 00000000 00000000		—	—	
003240H	PWMFLTRDCON00 [R/W] H,W 00000000 00000000 00000000 00000000				
003244H	PWMFLTRDCON01 [R/W] H,W 00000000 00000000 00000000 00000000				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
003248H	PWMFLTCAPRDCON0 [R/W] H,W 00000000 00000000 00000000 00000000				PWM fault
00324CH	PWMFLTCAPD0 [R] H,W 00000000 00000000 00000000 00000000				
003250H	PWMFLTCON10 [R/W] B,H,W 00000000 -0000-00		PWMFLTCON11 [R/W] B,H,W 00000000 -0000-00		
003254H	PWMFLTRCON1 [R/W] B,H,W -000-000 -000-000		PWMFLTCAPCON1 [R/W] B,H,W --00--00 -0000000		
003258H	PWMFLTSR1 [R/W] B,H,W -------0 -------0		—	—	
00325CH	PWMCAPITH1 [R/W] H,W 00000000 00000000		—	—	
003260H	PWMFLTRDCON10 [R/W] H,W 00000000 00000000 00000000 00000000				
003264H	PWMFLTRDCON11 [R/W] H,W 00000000 00000000 00000000 00000000				
003268H	PWMFLTCAPRDCON1 [R/W] H,W 00000000 00000000 00000000 00000000				
00326CH	PWMFLTCAPD1 [R] H,W 00000000 00000000 00000000 00000000				
003270H	PWMFLTCON20 [R/W] B,H,W 00000000 -0000-00		PWMFLTCON21 [R/W] B,H,W 00000000 -0000-00		
003274H	PWMFLTRCON2 [R/W] B,H,W -000-000 -000-000		PWMFLTCAPCON2 [R/W] B,H,W --00--00 -0000000		
003278H	PWMFLTSR2 [R/W] B,H,W -------0 -------0		—	—	
00327CH	PWMCAPITH2 [R/W] H,W 00000000 00000000		—	—	
003280H	PWMFLTRDCON20 [R/W] H,W 00000000 00000000 00000000 00000000				
003284H	PWMFLTRDCON21 [R/W] H,W 00000000 00000000 00000000 00000000				
003288H	PWMFLTCAPRDCON2 [R/W] H,W 00000000 00000000 00000000 00000000				
00328CH	PWMFLTCAPD2 [R] H,W 00000000 00000000 00000000 00000000				
003290H	PWMSOWCON0 [R/W] B,H,W 00000000 -------0 00000000 00000000				PWM soft overwrite control

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
003294H	PWMSOWCON1 [R/W] B,H,W 00000000 -------0 00000000 00000000				PWM soft overwrite control
003298H	PWMSOWCON2 [R/W] B,H,W 00000000 -------0 00000000 00000000				
00329CH	PWMDMOD [R/W] B,H,W --000000	—	—	—	PWM dead time
0032A0H	PWMHRTMRR0 [R/W] H,W 00000000 00000000		PWMHFTMRR0 [R/W] H,W 00000000 00000000		
0032A4H	PWMLRTMRR0 [R/W] H,W 00000000 00000000		PWMLFTMRR0 [R/W] H,W 00000000 00000000		
0032A8H	PWMHRTMRR1 [R/W] H,W 00000000 00000000		PWMHFTMRR1 [R/W] H,W 00000000 00000000		
0032ACH	PWMLRTMRR1 [R/W] H,W 00000000 00000000		PWMLFTMRR1 [R/W] H,W 00000000 00000000		
0032B0H	PWMHRTMRR2 [R/W] H,W 00000000 00000000		PWMHFTMRR2 [R/W] H,W 00000000 00000000		
0032B4H	PWMLRTMRR2 [R/W] H,W 00000000 00000000		PWMLFTMRR2 [R/W] H,W 00000000 00000000		
0032B8H	PWMLEBCON0 [R/W] B,H,W -000-000 -000-000 ----0000			—	PWM Blanking
0032BCH	PWMLEBSDCON00 [R/W] H,W 00000000 00000000		PWMLEBSDCON01 [R/W] H,W 00000000 00000000		
0032C0H	PWMLEBSDCON02 [R/W] H,W 00000000 00000000		PWMLEBSDCON03 [R/W] H,W 00000000 00000000		
0032C4H	PWMLEBTCON00 [R/W] H,W 00000000 00000000 00000000 00000000				
0032C8H	PWMLEBTCON01 [R/W] H,W 00000000 00000000 00000000 00000000				
0032CCH	PWMLEBCON1 [R/W] B,H,W -000-000 -000-000 ----0000			—	
0032D0H	PWMLEBSDCON10 [R/W] H,W 00000000 00000000		PWMLEBSDCON11 [R/W] H,W 00000000 00000000		
0032D4H	PWMLEBSDCON12 [R/W] H,W 00000000 00000000		PWMLEBSDCON13 [R/W] H,W 00000000 00000000		
0032D8H	PWMLEBTCON10 [R/W] H,W 00000000 00000000 00000000 00000000				
0032DCH	PWMLEBTCON11 [R/W] H,W 00000000 00000000 00000000 00000000				

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
0032E0H	PWMLEBCON2 [R/W] B,H,W -000-000 -000-000 ----0000			—	PWM Blanking
0032E4H	PWMLEBSDCON20 [R/W] H,W 00000000 00000000		PWMLEBSDCON21 [R/W] H,W 00000000 00000000		
0032E8H	PWMLEBSDCON22 [R/W] H,W 00000000 00000000		PWMLEBSDCON23 [R/W] H,W 00000000 00000000		
0032ECH	PWMLEBTCON20 [R/W] H,W 00000000 00000000 00000000 00000000				
0032F0H	PWMLEBTCON21 [R/W] H,W 00000000 00000000 00000000 00000000				
0032F4H	PWMADTCON [R/W] B,H,W 00000000 00000000 --000000 ----0000				PWM trigger
0032F8H	PWMADTST [R/W] B,H,W ----0000	—	—	—	
0032FCH	PWMADTDCON0 [R/W] H,W 00000000 00000000		PWMADTDCON1 [R/W] H,W 00000000 00000000		
003300H	PWMADTDCON2 [R/W] H,W 00000000 00000000		PWMADTDCON3 [R/W] H,W 00000000 00000000		
003304H to 00331CH	—				Reserved
003320H	AD4EN [R/W] B,H,W -------0	—	—	—	12 bit 4ch A/D converter
003324H	AD4TSS [R/W] B,H,W -------0	—	—	—	
003328H	AD4TSE [R/W] B,H,W -------- -------- -------- 00000000				
00332CH	AD4TCS0 [R/W] B,H,W ---00--- --------		AD4TCS1 [R/W] B,H,W ---00--- --------		
003330H	AD4TCS2 [R/W] B,H,W ---00--- --------		AD4TCS3 [R/W] B,H,W ---00--- --------		
003334H	AD4TCS4 [R/W] B,H,W ---00--- --------		AD4TCS5 [R/W] B,H,W ---00--- --------		

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
003338H	AD4TCS6 [R/W] B,H,W ---00--- --------		AD4TCS7 [R/W] B,H,W ---00--- --------		12 bit 4ch A/D converter
00333CH	AD4TBUSY [R/W] B,H,W -------- -------- -------- 00000000				
003340H	AD4TECS0 [R/W] B,H,W 00--0000 0000----		AD4TECS1 [R/W] B,H,W 00--0000 0000----		
003344H	AD4TECS2 [R/W] B,H,W 00--0000 0000----		AD4TECS3 [R/W] B,H,W 00--0000 0000----		
003348H	AD4TECS4 [R/W] B,H,W 00--0000 0000----		AD4TECS5 [R/W] B,H,W 00--0000 0000----		
00334CH	AD4TECS6 [R/W] B,H,W 00--0000 0000----		AD4TECS7 [R/W] B,H,W 00--0000 0000----		
003350H	AD4PTC8 [R/W] B,H,W 0000----	AD4PTC9 [R/W] B,H,W 0000----	AD4PTC10 [R/W] B,H,W 0000----	AD4PTC11 [R/W] B,H,W 0000----	
003354H	AD4TCD8 [R] B,H,W 10--0000 00000000		AD4TCD9 [R] B,H,W 10--0000 00000000		
003358H	AD4TCD10 [R] B,H,W 10--0000 00000000		AD4TCD11 [R] B,H,W 10--0000 00000000		
00335CH	AD4CS [R/W] B,H,W 00------ 00------		—	AD4MD [R/W] B,H,W ----0000	
003360H	AD4PRTF [R] B,H,W -------- -------- -------- ----0000				
003364H to 00EFFCH	—				Reserved
00F000H to 00FEFCH	—	—	—	—	Reserved[S]

Address	Address offset value / Register name				Block
	+0	+1	+2	+3	
00FF00H	DSUCR [R/W] B,H,W -------- -------0		—	—	OCDU [S]
00FF04H to 00FF0CH	—				
00FF10H	PCSR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00FF14H	PSSR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00FF18H to 00FFF4H	—				
00FFF8H	EDIR1 [R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00FFFCH	EDIR0 [R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

[S]: It is a system register. The illegal instruction exception (data access error) is generated in these registers in the user mode when reading and writing to it.

A.2 List of Interrupt Vector

List of interrupt vector is shown.

This list shows the assignments of interrupt factors and interrupt vectors/interrupt control registers.

Table A 2. Interrupt Vector MB91F552 (64 pin)

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexadecimal				
Reset	0	0	-	3FCH	000FFFFCH	-
System reserved	1	1	-	3F8H	000FFFF8H	-
System reserved	2	2	-	3F4H	000FFFF4H	-
System reserved	3	3	-	3F0H	000FFFF0H	-
System reserved	4	4	-	3ECH	000FFFECH	-
FPU exception	5	5	-	3E8H	000FFFE8H	-
Exception of instruction access protection violation	6	6	-	3E4H	000FFFE4H	-
Exception of data access protection violation	7	7	-	3E0H	000FFFE0H	-
Data access error interrupt	8	8	-	3DCH	000FFFDCH	-
INTE instruction	9	9	-	3D8H	000FFFD8H	-
Instruction break	10	0A	-	3D4H	000FFFD4H	-
System reserved	11	0B	-	3D0H	000FFFD0H	-
System reserved	12	0C	-	3CCH	000FFFCCH	-
System reserved	13	0D	-	3C8H	000FFFC8H	-
Exception of invalid instruction	14	0E	-	3C4H	000FFFC4H	-
NMI request	15	0F	15(FH) Fixed	3C0H	000FFFC0H	-
Error generation during internal bus diagnosis						
XBS RAM double-bit error generation						
TPU violation						
External interrupt 0-3	16	10	ICR00	3BCH	000FFFBCH	0
External low-voltage detection interrupt	17	11	ICR01	3B8H	000FFFB8H	-
Reload timer 0/1/4	18	12	ICR02	3B4H	000FFFB4H	2
Reload timer 2/3	19	13	ICR03	3B0H	000FFFB0H	3
Multi-function serial interface ch.0 (reception completed)	20	14	ICR04	3ACH	000FFFACH	4(*1)
Multi-function serial interface ch.0 (status)						
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	3A8H	000FFFA8H	5(*1)
Multi-function serial interface ch.1 (reception completed)	22	16	ICR06	3A4H	000FFFA4H	6(*1)
Multi-function serial interface ch.1(status)						
Multi-function serial interface ch.1 (transmission completed)	23	17	ICR07	3A0H	000FFFA0H	7(*1)
Multi-function serial interface ch.2 (reception completed)	24	18	ICR08	39CH	000FFF9CH	8(*1)
Multi-function serial interface ch.2(status)						
Multi-function serial interface ch.2 (transmission completed)	25	19	ICR09	398H	000FFF98H	9(*1)
-	26	1A	ICR10	394H	000FFF94H	-

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexadecimal				
-	27	1B	ICR11	390H	000FFF90H	-
-	28	1C	ICR12	38CH	000FFF8CH	-
-	29	1D	ICR13	388H	000FFF88H	-
-	30	1E	ICR14	384H	000FFF84H	-
-	31	1F	ICR15	380H	000FFF80H	-
-	32	20	ICR16	37CH	000FFF7CH	-
-	33	21	ICR17	378H	000FFF78H	-
CAN	34	22	ICR18	374H	000FFF74H	-
RAM diagnosis end RAM initialization completion Error generation during RAM diagnosis	35	23	ICR19	370H	000FFF70H	-
4ch A/D converter irregular activation interrupt/insufficient sampling time interrupt	36	24	ICR20	36CH	000FFF6CH	-
-	37	25	ICR21	368H	000FFF68H	-
PWM special event interrupt 0/1	38	26	ICR22	364H	000FFF64H	-
16-bit Free-run timer 0 (0 detection) / (compare clear)	39	27	ICR23	360H	000FFF60H	23
PWM 0 detection interrupt 0, compare clear interrupt 0	40	28	ICR24	35CH	000FFF5CH	-
PWM 0 detection interrupt 1, compare clear interrupt 1	41	29	ICR25	358H	000FFF58H	-
PWM SOW interrupt 0, fault interrupt 0/1, capture interrupt 0	42	2A	ICR26	354H	000FFF54H	-
PWM SOW interrupt 1, fault interrupt 2/3, capture interrupt 1	43	2B	ICR27	350H	000FFF50H	-
PWM SOW interrupt 2, fault interrupt 4/5, capture interrupt 2	44	2C	ICR28	34CH	000FFF4CH	-
16bit ICU 0 (fetching)	45	2D	ICR29	348H	000FFF48H	29
Main timer PLL timer	46	2E	ICR30	344H	000FFF44H	30
PWM trigger interrupt 0/1/2/3	47	2F	ICR31	340H	000FFF40H	-
A/D converter 0/1/2/3/4/5/6/7	48	30	ICR32	33CH	000FFF3CH	32
Clock calibration unit (CR oscillation)	49	31	ICR33	338H	000FFF38H	-
PLL alarm interrupt for PWM	50	32	ICR34	334H	000FFF34H	-
reserved	51	33	ICR35	330H	000FFF30H	-
Comparator output detection interrupt 0/1/2	52	34	ICR36	32CH	000FFF2CH	36
PWC0 0 detection interrupt 00/10, compare clear interrupt 00/10	53	35	ICR37	328H	000FFF28H	-(*2)
PWC0 capture data upper limit interrupt 00/10/20/30, PWC0 capture data lower limit interrupt 00/10/20/30, PWC0 data buffer interrupt 00/10/20/30, PWC0 buffer overrun interrupt 00/10/20/30	54	36	ICR38	324H	000FFF24H	-(*2)
PWC1 0 detection interrupt 01/11, compare clear interrupt 01/11	55	37	ICR39	320H	000FFF20H	-(*2)
PWC1 capture data upper limit interrupt 01/11/21/31, PWC1 capture data lower limit interrupt 01/11/21/31, PWC1 data buffer interrupt 01/11/21/31, PWC1 buffer overrun interrupt 01/11/21/31	56	38	ICR40	31CH	000FFF1CH	-(*2)
A/D converter 8/9/10/11	57	39	ICR41	318H	000FFF18H	-(*2)
Base Timer 2 IRQ0 Base Timer 2 IRQ1	58	3A	ICR42	314H	000FFF14H	42
Base Timer 3 IRQ0	59	3B	ICR43	310H	000FFF10H	43

Interrupt factor	Interrupt number		Interrupt level	Offset	Default address for TBR	RN *
	Decimal	Hexadecimal				
Base Timer 3 IRQ1						
Base Timer 0 IRQ0 Base Timer 0 IRQ1	60	3C	ICR44	30CH	000FFF0CH	44
Base Timer 1 IRQ0 Base Timer 1 IRQ1	61	3D	ICR45	308H	000FFF08H	45
DMAC0/1/2/3/4/5/6/7	62	3E	ICR46	304H	000FFF04H	-
Delay interrupt	63	3F	ICR47	300H	000FFF00H	-
System reserved (Used for REALOSTM*3)	64	40	-	2FCH	000FFEFCH	-
System reserved (Used for REALOS)	65	41	-	2F8H	000FFEF8H	-
Used with the INT instruction	66 | 255	42 | FF	-	2F4H | 000H	000FFEF4H | 000FFC00H	-

*: DMA transfer request by interrupt from peripherals without an assigned RN number is not supported.

*1: DMA transfer by the multi-function serial interface status is not supported.

*2: DMA transfer request by on-chip bus IP interrupt is supported.

*3: REALOS is a trademark of Spansion LLC.

A.3 Pin States by CPU States

Pin states by CPU states are shown.

Pin No.	Pin Name	Pin Function	Input Level Output Level Circuit Type	Pin States after Reset Factor Generated and after Reset Released									Pin States in Sleep and Standby Modes						#Bidirectionality PORTEN Setting?
				Reset Factor 1 (*1)			Reset Factor 2 (*2)					Internal Reset Factor (*4)	Sleep Mode		Stop Mode		Watch Mode		
				External Factor Being Generated	After External Factor Released		External Factor Being Generated		After External Factor Released										
				Internal Reset Being Issued (Waiting for Oscillation Stabilization)	Internal Reset Being Issued, or is after Its Release (Waiting for Oscillation Stabilization)	After Internal Reset Released	Before Internal Reset Issued	Internal Reset Being Issued	Before Internal Reset Issued	Internal Reset Being Issued, or is after Its Release (Waiting for Oscillation Stabilization)	After Internal Reset Released	Internal Reset Being Issued, or is after Its Release (Waiting for Oscillation Stabilization)	After Internal Reset Released	SLVL[1:0]=2'b0x	SLVL[1:0]=2'b1x	SLVL[1:0]=2'b0x	SLVL[1:0]=2'b1x	SLVL[1:0]=2'b0x	SLVL[1:0]=2'b1x	
1	VSS	GND	-	-			-			-		-	-		-		-		–
2	P000 PWM0H	General-purpose I/O port PWM ch.0-H	Input: Automotive Output: CMOS A	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
3	P001 PWM0L	General-purpose I/O port PWM ch.0-L	Input: Automotive Output: CMOS A	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
4	P002 PWM1H	General-purpose I/O port PWM ch.1-H	Input: Automotive Output: CMOS A	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
5	P003 PWM1L	General-purpose I/O port PWM ch.1-L	Input: Automotive Output: CMOS A	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
6	P004 PWM2H	General-purpose I/O port PWM ch.2-H	Input: Automotive Output: CMOS A	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
7	P005 PWM2L	General-purpose I/O port PWM ch.2-L	Input: Automotive Output: CMOS A	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
8	VCC	+5.0 V power supply	-	-			-			-		-	-		-		-		–
9	VSS	GND	-	-			-			-		-	-		-		-		–
10	P046 SYNCIN SYNCOUT SCS00	General-purpose I/O port Master/Slave input for PWM parallel operation drive Master/Slave output for PWM parallel operation drive Multi-function serial ch.0 serial chip select 00 I/O	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
11	P047 MONCLK TIN4 ADTG1 SCS01	General-purpose I/O port Clock monitor output Reload timer ch.4 event input A/D converter ch.8-ch.11 external trigger input pin Multi-function serial ch.0 serial chip select 01 output	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
12	NMIX	Non-maskable interrupt input	Input: CMOS hysteresis L	Input enabled			Input enabled			Input enabled		Input enabled	Input enabled						–
13	CMP0	Comparator ch.0 input	Analog input C	Analog input			Analog input			Analog input		Analog input	Analog input						–
14	CMP1	Comparator ch.1 input	Analog input C	Analog input			Analog input			Analog input		Analog input	Analog input						–
15	CMP2	Comparator ch.2 input	Analog input C	Analog input			Analog input			Analog input		Analog input	Analog input						–
16	AVCC2	Comparator, slope compensation circuit analog power supply	-	-			-			-		-	-						–
17	AVSS2	Comparator, slope compensation circuit GND	-	-			-			-		-	-						–
18	AN11	A/D converter ch.11 analog input (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)	Analog input C	Analog input			Analog input			Analog input		Analog input	Analog input						–
19	AN10	A/D converter ch.10 analog input (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)	Analog input C	Analog input			Analog input			Analog input		Analog input	Analog input						–
20	AN9	A/D converter ch.9 analog input (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)	Analog input C	Analog input			Analog input			Analog input		Analog input	Analog input						–
21	AN8	A/D converter ch.8 analog input (Simultaneous sampling possible with ch.8, ch.9, ch.10, and ch.11)	Analog input C	Analog input			Analog input			Analog input		Analog input	Analog input						–
22	AVCC1	AD converter analog power supply with built-in 4ch simultaneous sampling/hold function	-	-			-			-		-	-						–
23	AVSS1	AD converter GND with built-in 4ch simultaneous sampling/hold function	-	-			-			-		-	-						–
24	VR1	Intermediate reference voltage for AD converter with built-in 4ch simultaneous sampling/hold function	-	-			-			-		-	-						–
25	AVRL1	Lower limit reference voltage for AD converter with built-in 4ch simultaneous sampling/hold function	-	-			-			-		-	-						–
26	AVRH1	Upper limit reference voltage for AD converter with built-in 4ch simultaneous sampling/hold function	-	-			-			-		-	-						–
27	AVCC0	Analog power supply for ADC0	-	-			-			-		-	-						–
28	AVSS0	GND for ADC0	-	-			-			-		-	-						–
29	AVRL0	Lower limit reference voltage for ADC0	-	-			-			-		-	-						–
30	AVRH0	Upper limit reference voltage for ADC0	-	-			-			-		-	-						–
31	P020 AN0 TOT0 TRCK	General-purpose I/O port A/D converter ch.0 analog input Reload timer ch.0 output Free-run timer clock input	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked)(*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
32	VCC	+5.0 V power supply	-	-			-			-		-	-		-		-		–
33	VSS	GND	-	-			-			-		-	-		-		-		–
34	P021 AN1 TIN0	General-purpose I/O port A/D converter ch.1 analog input Reload timer ch.0 event input	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
35	P022 AN2 TOT1	General-purpose I/O port A/D converter ch.2 analog input Reload timer ch.1 output	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
36	P023 AN3 TIN1	General-purpose I/O port A/D converter ch.3 analog input Reload timer ch.1 event input	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
37	P024 AN4 TOT2	General-purpose I/O port A/D converter ch.4 analog input Reload timer ch.2 output	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
38	P025 AN5 TIN2	General-purpose I/O port A/D converter ch.5 analog input Reload timer ch.2 event input	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
39	P026 AN6 TOT3	General-purpose I/O port A/D converter ch.6 analog input Reload timer ch.3 output	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
40	P027 AN7 TIN3 SCS21	General-purpose I/O port A/D converter ch.7 analog input Reload timer ch.3 event input Multi-function serial ch.2 serial chip select 21 output	Input: Automotive Output: CMOS D	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
41	P030 TIOA0 ADTG0 SCS10	General-purpose I/O port TIOA output of base timer ch.0 A/D converter ch.0-ch.7 external trigger input pin Multi-function serial ch.1 serial chip select 10 I/O	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
42	P031 SOT1 TIOB0	General-purpose I/O port Multi-function serial ch.1 serial data output TIOB input of base timer ch.0	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
43	P032 SCK1 PWC1 TIOA1	General-purpose I/O port Multi-function serial ch.1 clock I/O PWC ch.1 input TIOA I/O of base timer ch.1	Input: CMOS hysteresis Output: CMOS E	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
44	P033 SIN1 TIOB1 INT2	General-purpose I/O port Multi-function serial ch.1 serial data input TIOB input of base timer ch.1 INT2 external interrupt input	Input: CMOS hysteresis Output: CMOS E	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
45	P034 PWC0 TIOA2 SCS11	General-purpose I/O port PWC ch.0 input TIOA output of base timer ch.2 Multi-function serial ch.1 serial chip select 11 output	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
46	P035 SOT0 TIOB2	General-purpose I/O port Multi-function serial ch.0 serial data output TIOB input of base timer ch.2	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
47	P036 SCK0 TIOA3	General-purpose I/O port Multi-function serial ch.0 clock I/O TIOA I/O of base timer ch.3	Input: CMOS hysteresis Output: CMOS E	Input enabled			Input enabled			Input enabled		Input enabled	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	No
48	P037 SIN0 TIOB3 INT1	General-purpose I/O port Multi-function serial ch.0 serial data input TIOB input of base timer ch.3 INT1 external interrupt input	Input: CMOS hysteresis Output: CMOS E	Input enabled			Input enabled			Input enabled		Input enabled	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	No
49	DEBUGIF	Debugger (OCD) MDI I/O	Input: TTL Output: 25 mA (Nch open drain) F	Hi-Z/input enabled	Input enabled	Input disabled (*9)	Input disabled	Input enabled(*10)	Input disabled	Input enabled(*10)	Input disabled (*9)	Input enabled(*10)	Input disabled (*9)			Input disabled (*9)			–
50	P040 MM ICU SCS20	General-purpose I/O port Clock supervisor main clock stop detection output pin Input capture input Multi-function serial ch.2 serial chip select 20 I/O	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
51	MD0	Mode pin 0	Flash test pin dual mode I/O Input: CMOS hysteresis G	Input enabled			Input enabled			Input enabled		Input enabled	Input enabled						–
52	MD1	Mode pin 1	Flash test pin dual mode I/O Input: CMOS hysteresis G	Input enabled			Input enabled			Input enabled		Input enabled	Input enabled						–
53	X0	Main clock oscillation input	H	-			-			-		-	-		-		-		–
54	X1	Main clock oscillation output	H	-			-			-		-	-		-		-		–
55	VSS	GND	-	-			-			-		-	-		-		-		–
56	P041 TX	General-purpose I/O port CAN transmission data output	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
57	P042 RX INT0	General-purpose I/O port CAN reception data input INT0 external interrupt input	Input: CMOS hysteresis Output: CMOS E	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
58	RSTX	External reset input	Input: CMOS hysteresis L	Input enabled			Input enabled			Input enabled		Input enabled	Input enabled						–
59	VSS	GND	-	-			-			-		-	-		-		-		–
60	C	External capacitor connect output	-	-			-			-		-	-		-		-		–
61	P043 SOT2 TOT4	General-purpose I/O port Multi-function serial ch.2 serial data output Reload timer ch.4 output	Input: Automotive Output: CMOS B	Hi-Z/input blocked			Hi-Z/input enabled	Hi-Z/input blocked	Hi-Z/input enabled	Hi-Z/input blocked		Hi-Z/input blocked	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	Yes
62	P044 SCK2	General-purpose I/O port Multi-function serial ch.2 clock I/O	Input: CMOS hysteresis Output: CMOS E	Input enabled			Input enabled			Input enabled		Input enabled	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	No
63	P045 SIN2 INT3	General-purpose I/O port Multi-function serial ch.2 serial data input INT3 external interrupt input	Input: CMOS hysteresis Output: CMOS E	Input enabled			Input enabled			Input enabled		Input enabled	Last status retained (*7)	Last status retained	Last status retained	Hi-Z (input blocked) (*5)	Last status retained	Hi-Z (input blocked) (*5)	No
64	VCC	+5.0 V power supply	-	-			-			-		-	-		-		-		–

(*1) Factor: Power-on reset, low-voltage detection (internal low-voltage detection), and NMIX+RSTX
(*2) Factor: Low-voltage detection (external low-voltage detection) and external reset
(*4) Factor: Software reset, software/hardware watchdog reset (including timeout)
(*5) If external interrupts are enabled, the input blocking of the following pins is disabled. (GPORTEN is cleared only for pins whose ENIR.EN bits are set for channels supporting external interrupts and for pins affected by the PORTEN setting.)
P042, P037, P033, P045
(*7) Operation continues according to the peripheral function.
(*9) Emulation mode: Input enabled; Free-run mode: Input disabled
(*10) Input is enabled only when an INT level reset has been issued. Otherwise, input is disabled.

Revision History

Document Revision History

Document Title: MB91550 Series FR81S MB91F552 Hardware Manual Document Number: 002-05582			
Revision	**ECN No.**	**Origin of Change**	**Description of Change**
**	—	NNAS	Migrated to Cypress and assigned document number 002-05582 from Spansion MB91F552_MN705-00023-1v0-E. No change to document contents or format.
*A	5571834	NNAS	Updated to Cypress format.