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

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

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



CY91570 Series

FR Family FR81S Hardware Manual

Document Number: 002-05574 Rev. *B

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

© Cypress Semiconductor Corporation, 2011-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, “Security Breach”). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. “High-Risk Device” means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. “Critical Component” means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, 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 any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress’s published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

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

Preface

Thank you for your continued use of Cypress products.

Read this manual and "CY91570 series Datasheet" thoroughly before using products in the CY91570 series.

Purpose of this Manual and Intended Reader

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.
It is best suited for application control for automotive.
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 CY91570 series. Feel free to use them to verify the operational specifications and usage of this Cypress microcontroller.

- Microcontroller support information:

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

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

Target Products

CY91F575B/F575BS/F575BH/F575BHS/F575C/F575CS/F575CH/F575CHS

CY91F577B/F577BS/F577BH/F577BHS/F577CS/F577CH/F577CHS

CY91F578C(M)/F578CS(M)/F578CH(M)/F578CHS(M)

CY91F579C(M)/F579CS(M)/F579CH(M)/F579CHS(M)

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

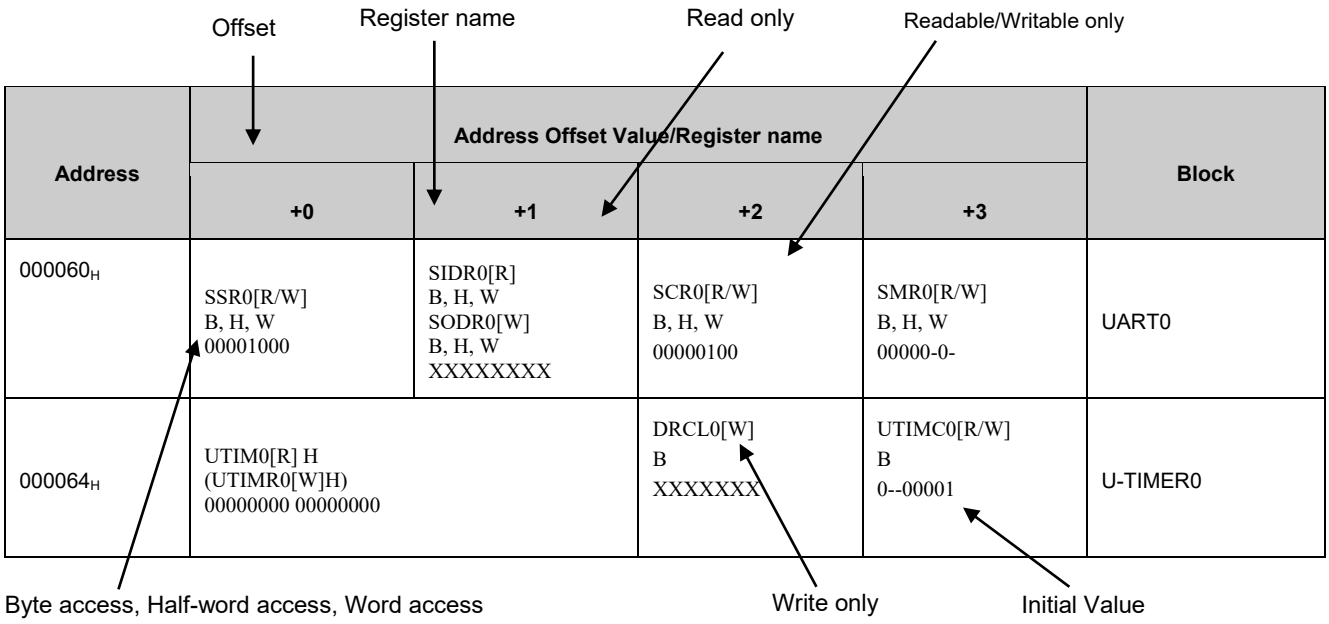
Primary Terms

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 an external bus interface, CAN, 16/32-bit peripheral bus bridge and others. The bus slave of only DMA layer has an access to the XBS.
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.
External bus (External bus)	8/16-bit width, low-speed external bus. It connects to memory devices, ASIC and others. This series is the bus master, and a device connected to the external bus is a bus slave.
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.
Sub clock (SBCLK)	This is the reference clock for LSI operation, and it is supplied from the low-speed system oscillator. It is connected to the timer for sub oscillation stabilization wait and others. It can be used by the dual clock products only.
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.
External bus clock (TCLK)	The reference clock for an external bus interface connected to the X-bus and for the external clock output. It is generated from the base clock by the clock generator.
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.

Term	Explanation
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 (1026+3) PCLK cycles.
Power shutdown	The power supply to the target circuit is stopped, and power consumption is decreased.
Always power supply ON block	It is not a target division for the power shutdown.
PMU Power management unit	The power shutdown is controlled. PMU exists in always ON block.
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. It is a technology to suppress the peak of EMI to low. SSCG is a technology that suppresses the peak of EMI to low by the clock frequency change slightly and oscillates it (= frequency modulation). 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 mak
ADC	A/D converter

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)

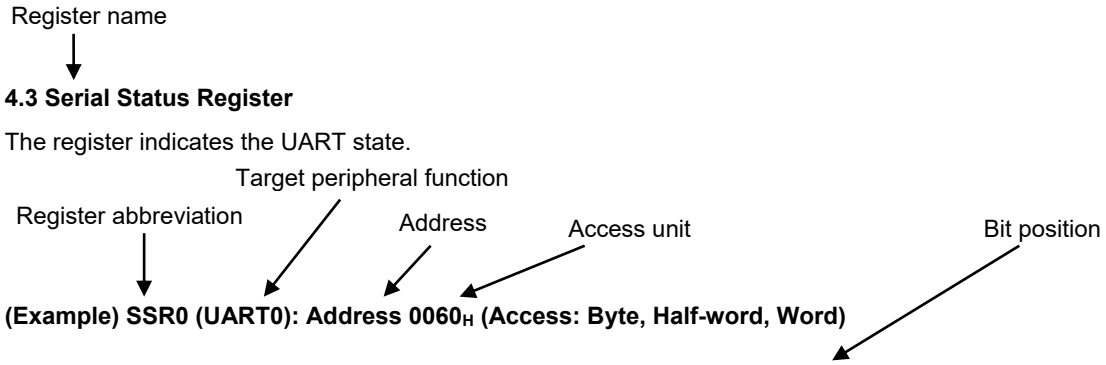
The following explains the address position during access.

- 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 set to the Half-word access, for example, SSR0 + SIDR0 (SODR0) register at address 060_H is 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



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

1. Overview 43

1.1 Overview 44
1.2 Features 45
1.2.1 FR81S CPU Core 46
1.2.2 Peripheral Functions 47
1.3 Product Line-up 51
1.4 Function Overview 55
1.5 Block Diagram 60
1.6 CPU 61
1.6.1 General-purpose Registers 62
1.6.2 Dedicated Registers 63
1.7 Pin Assignment 64
1.8 Package Dimensions 65
1.9 Explanation of Pin Functions 67
1.10 Pins of Each Function 84
1.10.1 Pins of A/D convertor 85
1.10.2 Pins of Base Timer (ch.0, ch.1) 87
1.10.3 Pins of CAN (ch.0 to ch.2) 88
1.10.4 Pins of DAC (ch.0, ch.1) 89
1.10.5 Pins of External Bus Interface 90
1.10.6 Pins of External Interrupt Request Input (ch.0 to ch.15) 92
1.10.7 Pins of Free-run Timer (ch.0 to ch.5) 93
1.10.8 Pins of General-purpose I/O Port 94
1.10.9 Pins of HS_SPI 97
1.10.10 Pins of Input Capture (ch.0 to ch.11) 98
1.10.11 Pins of LCDC 99
1.10.12 Pins of LIN-UART (ch.2 to ch.7) 101
1.10.13 Pins of Multi-function Serial Interface (ch.0 to ch.1, ch.8 to ch.9) 102
1.10.14 Pins of Output Compare (ch.0 to ch.11) 103
1.10.15 Pins of PPG (ch.0 to ch.23) 104
1.10.16 Pins of Reload Timer (ch.0 to ch.3) 106
1.10.17 Pins of Real Time Clock 107
1.10.18 Pins of Stepping Motor Controller (ch.0 to ch.5) 108
1.10.19 Pins of Sound Generator (ch.0 to ch.4) 109
1.10.20 Pins of Up/Down Counter (ch.0, ch.1) 110
1.10.21 Pins of Clock 111
1.10.22 Pins of Mode 112
1.10.23 Pins of Reset 113
1.10.24 Pins of Power Supply 114

1.11 I/O Circuit Types 115

2. Handling the Device 121

2.1 Handling Precautions 122
2.2 Handling Device 126
2.3 Application Notes 129
2.3.1 Function Switching of a Multiplexed Port 130
2.3.2 Low-power Consumption Mode 131
2.3.3 Notes When Writing Data in a Register Having the Status Flag 132

3. CPU 133

3.1 Overview 134
3.2 Features 135
3.3 CPU Operating Description 137
3.4 Pipeline Operation 139
3.5 Floating Point Operation Processing 140
3.6 Data Structure 141
3.7 Addressing 142
3.8 Programming Model 143
3.8.1 General-purpose Registers, Dedicated Registers, and Floating Point Registers 143
3.8.2 System Register 144
3.9 Reset and EIT Processing 145
3.9.1 Reset 145
3.9.2 EIT Processing 145
3.9.3 Vector Table 146
3.10 Memory Protection Function (MPU) 148
3.10.1 Overview 149
3.10.2 List of Registers 150
3.10.3 Description of Registers 151
3.10.3.1 MPU Control Register (MPUCR) 152
3.10.3.2 Instruction Access Protection Violation Address Register (IPVAR) 156
3.10.3.3 Instruction Access Protection Violation Status Register (IPVSR) 157
3.10.3.4 Data Access Protection Violation Address Register (DPVAR) 159
3.10.3.5 Data Access Protection Violation Status Register (DPVSR) 160
3.10.3.6 Data Access Error Address Register (DEAR) 162
3.10.3.7 Data Access Error Status Register (DESR) 163
3.10.3.8 Protection Area Base Address Register 0 to 7 (PABR0 to PABR7) 165
3.10.3.9 Protection Area Control Register 0 to 7 (PACR0 to PACR7) 166
3.10.4 Operations of Memory Protection Function (MPU) 170
3.10.4.1 Setting Up Memory Protection Areas 171
3.10.4.2 Instruction Access Protection Violation 171
3.10.4.3 Data Access Protection Violation 171
3.10.4.4 Data Access Errors 172
3.10.4.5 Memory Protection Operation by Delay Slot 172
3.10.4.6 DEAR and DESR Update 173
3.10.4.7 Notes 173

4. Operation Mode 175

4.1 Overview 176
4.2 Features 177
4.3 Configuration 178
4.4 Register 179
4.5 Operation 180
4.5.1 MD0, MD1, MD2, P127 Pins Settings 181
4.5.2 Fetching the Operation Mode 182
4.5.3 Explanation of Each Operation Mode 183

5. Clock 185

5.1 Overview 186
5.2 Features 188
5.3 Configuration 189
5.4 Register 193
5.4.1 Division Configuration Register 0: DIVR0 (Division clock configuration Register 0) 194
5.4.2 Division Configuration Register 1: DIVR1 (Division clock configuration Register 1) 195
5.4.3 Division Configuration Register 2: DIVR2 (Division clock configuration Register 2) 196
5.4.4 Clock Source Selection Register: CSELR 197
5.4.5 Clock Source Monitor Register: CMONR 200
5.4.6 Main Timer Control Register: MTMCR (Main Clock Timer Control Register) 202
5.4.7 Sub Timer Control Register: STMCR (Sub Clock Timer Control Register) 205
5.4.8 PLL Setting Register: PLLCR (PLL Configuration Register) 208
5.4.9 Clock Stabilization Selection Register: CSTBR 211
5.4.10 PLL Clock Oscillation Timer Control Register: PTMCR 213
5.4.11 PLL/SSCG Clock Selection Register: CCPSSELR (CCTL PLL/SSCG clock Selection Register) 214
5.4.12 PLL/SSCG Output Clock Division Setting Register: CCPSDIVR (CCTL PLL/SSCG Clock Division Register) 215
5.4.13 PLL Feedback Division Setting Register: CCPLLFBR (CCTL PLL FB clock division Register) 217
5.4.14 SSCG Feedback Division Setting Register 0: CCSSFBR0 (CCTL SSCG FB clock division Register 0) 218
5.4.15 SSCG Feedback Division Setting Register 1: CCSSFBR1 (CCTL SSCG FB clock division Register 1) 219
5.4.16 SSCG Configuration Setting Register 0: CCSSCCR0 (CCTL SSCG Config. Register 0) 220
5.4.17 SSCG Configuration Setting Register 1: CCSSCCR1 (CCTL SSCG Config. Register 1) 222
5.4.18 Clock Gear Configuration Setting Register 0: CCCGRCR0 (CCTL Clock Gear Config. Register 0) 223
5.4.19 Clock Gear Configuration Setting Register 1: CCCGRCR1 (CCTL Clock Gear Config. Register 1) 225
5.4.20 Clock Gear Configuration Setting Register 2: CCCGRCR2 (CCTL Clock Gear Config. Register 2) 226
5.4.21 RTC/PMU Clock Selection Register: CCRTSELR (CCTL RTC PMU clock Selection Register) 227
5.4.22 PMU Clock Division Setting Register 0: CCPMUCR0 (CCTL PMU Clock division Register 0) 229
5.4.23 PMU Clock Division Setting Register 1: CCPMUCR1 (CCTL PMU Clock division Register 1) 230
5.4.24 Sync/Async Control Register: SACR 232
5.4.25 Peripheral Interface Clock Divider: PICD (Peripheral Interface Clock Divider) 233
5.5 Operation 234
5.5.1 Oscillation Control 235
5.5.1.1 Main Clock (MCLK) 236
5.5.1.2 Sub Clock (SBCLK) 236
5.5.1.3 PLL/SSCG Clock (PLLSSCLK) 237

5.5.1.4 Limitations When PLL/SSCG Clock is Used 240
5.5.2 Oscillation Stabilization Wait 241
5.5.2.1 Conditions for Generating Stabilization Wait Time 241
5.5.2.2 Selecting Stabilization Wait Time 241
5.5.2.3 End of the Stabilization Wait Time 241
5.5.3 Selecting the Source Clock (SRCCLK) 242
5.5.3.1 Selecting the Source Clock at the Time of Initialization 242
5.5.3.2 Procedure of Switching the Source Clock 243
5.5.4 Timer 248
5.5.4.1 Main Clock Oscillation Stabilization Wait Timer (Main Timer) 249
5.5.4.2 Sub Clock Oscillation Stabilization Wait Timer (Sub Timer) 249
5.5.4.3 PLL/SSCG Clock Oscillation Stabilization Wait Timer (PLL Timer) 249
5.5.4.4 Setting 249
5.5.4.5 Procedure for Setting the Timer Interrupt 250
5.5.4.6 Timer Operations 251
5.5.4.7 Watch Mode and Timer Interrupt 251
5.5.5 Notes When Clocks Conflict 252
5.5.6 The Clock Gear Circuit 253
5.5.6.1 Procedure of Gear Up 253
5.5.6.2 Procedure of Gear Down 253
5.5.7 Operations During MDI Communications 254
5.5.8 About PMU Clock (PMUCLK) 255

6. Clock Reset State Transitions 257

6.1 Overview 258
6.2 Device States and Transitions 259
6.2.1 Diagram of State Transitions 260
6.2.2 Explanation of Each States 262
6.2.3 Priority of State Transition Requests 264
6.3 Device State and Regulator Mode Corresponding to those States 265

7. Reset 267

7.1 Overview 268
7.2 Features 269
7.3 Configuration 270
7.4 Registers 273
7.4.1 Reset Source Register: RSTRR (Reset Result Register) 274
7.4.2 Reset Control Register: RSTCR 277
7.4.3 CPU Abnormal Operation Register: CPUAR 278
7.4.4 PMU Status Register: PMUSTR (Power Management Unit STatus register) 280
7.5 Operation 282
7.5.1 Reset Level 283
7.5.1.1 Initialize Reset (INIT) 283
7.5.1.2 Reset (RST) 283
7.5.2 Reset Factor 284

7.5.2.1 Power-on Reset 285
7.5.2.2 RSTX Pin Input 285
7.5.2.3 Watchdog Reset 0 285
7.5.2.4 Watchdog Reset 1 285
7.5.2.5 External Low-voltage Detection Reset 285
7.5.2.6 Illegal Standby Mode Transition Detection Reset 285
7.5.2.7 Internal Low-voltage Detection Reset 286
7.5.2.8 Flash Security Violation Reset 286
7.5.2.9 Software Reset (RSTCR:SRST) 286
7.5.2.10 Recovery from Standby (Power Interception) 286
7.5.3 Reset Acceptance 287
7.5.3.1 Generation of Reset Request 288
7.5.3.2 Acceptance of Reset Request 288
7.5.3.3 Reset Issue Delay Counter 288
7.5.3.4 Irregular Reset 288
7.5.4 Reset Issue 289
7.5.4.1 Super Initialize Reset (SINIT) 290
7.5.4.2 Initialize Reset (INIT) 292
7.5.4.3 Reset (RST) 293
7.5.5 Reset Sequence 294
7.5.5.1 Reset Cycle 295
7.5.5.2 Reset Release 295
7.5.5.3 Operating Mode Fix 295
7.5.5.4 Transition of Bus Control 295
7.5.5.5 Reset Vector Fetch 295
7.5.5.6 Reset and Forced Break 295
7.5.6 Notes 296

8. DMA Controller (DMAC) 297

8.1 Overview 298
8.2 Features 299
8.3 Configuration 300
8.4 Registers 301
8.4.1 DMA Control Register: DMACR 305
8.4.2 DMA Channel Control Register 0 to 15: DCCR0 to 15 307
8.4.3 DMA Channel Status Register 0 to 15: DCSR0 to 15 313
8.4.4 DMA Transfer Count Register 0 to 15: DTCR0 to 15 315
8.4.5 DMA Transfer Source Register 0 to 15: DSAR0 to 15: (DMA Source Address Register 0 to 15) 316
8.4.6 DMA Transfer Destination Register 0 to 15: DDAR0 to 15 (DMA Destination Address Register 0 to 15) 318
8.4.7 DMA Transfer Suppression NMI Flag Register: DNMIR (DMA-halt by NMI Register) 320
8.4.8 DMA Transfer Suppression Level Register: DILVR (DMA-halt by Interrupt Level Register) 321
8.5 Operation 323
8.5.1 DMA Operation Enable 323

8.5.2 Separate Items for Each Channel 324
8.5.3 Operations 328
8.6 DMA Usage Examples 342

9. Generation and Clearing of DMA Transfer Requests 345

9.1 Overview 346
9.2 Features 347
9.3 Configuration 348
9.4 Registers 349
9.4.1 DMA Request Clear Register 0: ICSEL0 (Interrupt Clear Select Register 0) 351
9.4.2 DMA Request Clear Register 1: ICSEL1 (Interrupt Clear Select Register 1) 352
9.4.3 DMA Request Clear Register 2: ICSEL2 (Interrupt Clear Select Register 2) 353
9.4.4 DMA Request Clear Register 3: ICSEL3 (Interrupt Clear Select Register 3) 354
9.4.5 DMA Request Clear Register 4: ICSEL4 (Interrupt Clear Select Register 4) 355
9.4.6 DMA Request Clear Register 5: ICSEL5 (Interrupt Clear Select Register 5) 356
9.4.7 DMA Request Clear Register 6: ICSEL6 (Interrupt Clear Select Register 6) 357
9.4.8 DMA Request Clear Register 7: ICSEL7 (Interrupt Clear Select Register 7) 358
9.4.9 DMA Request Clear Register 8: ICSEL8 (Interrupt Clear Select Register 8) 359
9.4.10 DMA Request Clear Register 9: ICSEL9 (Interrupt Clear Select Register 9) 360
9.4.11 DMA Request Clear Register 10: ICSEL10 (Interrupt Clear Select Register 10) 361
9.4.12 DMA Request Clear Register 11: ICSEL11 (Interrupt Clear Select Register 11) 362
9.4.13 DMA Request Clear Register 12: ICSEL12 (Interrupt Clear Select Register 12) 363
9.4.14 DMA Request Clear Register 13: ICSEL13 (Interrupt Clear Select Register 13) 364
9.4.15 DMA Request Clear Register 14: ICSEL14 (Interrupt Clear Select Register 14) 365
9.4.16 DMA Request Clear Register 15: ICSEL15 (Interrupt Clear Select Register 15) 366
9.4.17 DMA Request Clear Register 16: ICSEL16 (Interrupt Clear Select Register 16) 367
9.4.18 DMA Request Clear Register 17: ICSEL17 (Interrupt Clear Select Register 17) 368
9.4.19 DMA Request Clear Register 18: ICSEL18 (Interrupt Clear Select Register 18) 369
9.4.20 DMA Request Clear Register 19: ICSEL19 (Interrupt Clear Select Register 19) 370
9.4.21 DMA Request Clear Register 20: ICSEL20 (Interrupt Clear Select Register 20) 371
9.4.22 DMA Request Clear Register 21: ICSEL21 (Interrupt Clear Select Register 21) 372
9.4.23 DMA Request Clear Register 22: ICSEL22 (Interrupt Clear Select Register 22) 373
9.4.24 IO Transfer Request Setting Register 0 to 15: IORR0 to 15 (IO triggered DMA Request Register for ch. 0 to 15) 374
9.5 Operation 376
9.5.1 Configuration 377
9.5.2 Notes 378

10. FixedVector Function 379

10.1 Overview 380
10.2 Operation 381

11. I/O Ports 383

11.1 Overview 384
11.2 Features 385
11.3 Configuration 386
11.4 Registers 387
11.4.1 Port Data Register 00 to 13: PDR00-13 389
11.4.2 Data Direction Register 00 to 13: DDR00-13 390

11.4.3 Port Function Register 00 to 13: PFR00-13 391
11.4.4 Input Data Direct Register 00 to 13: PDDR00-13 (Port Data Direct Register 00-13) 392
11.4.5 Port Pull-up/down Control Register 00 to 13: PPCR00-13 393
11.4.6 Port Pull-up/down Enable Register 00 to 13: PPER00-13 394
11.4.7 Port Input Level Selection Register 00 to 13: PILR00-13 (Port Input Level Register 00-13) 395
11.4.8 Extended Port Input Level Selection Register 00 to 13: EPILR00-13 (Extended Port Input Level Register 00-13) 396
11.4.9 Port Output Drive Register 00 to 13: PODR00-13 397
11.4.9.1 Pins P010 to P036 397
11.4.9.2 Pins P060 to P087 398
11.4.9.3 Other Pins than P010 to P036 nor P060 to P087 398
11.4.10 Extended Port Output Drive Register 01-03,06-08: EPODR01-03,06-08 399
11.4.11 Extended Port Function Register 00 to 54: EPFR00-54 400
11.4.11.1 Extended Port Function Register 00, 01, 39: EPFR00, 01, 39 401
11.4.11.2 Extended Port Function Register 02 to 05: EPFR02-05 403
11.4.11.3 Extended Port Function Register 06 to 09, 33, 34: EPFR06-09, 33, 34 405
11.4.11.4 Extended Port Function Register 10 to 15, 45, 46: EPFR10-15,45,46 408
11.4.11.5 Extended Port Function Register 16 to 20: EPFR16-20 412
11.4.11.6 Extended Port Function Register 21 to 23: EPFR21-23 414
11.4.11.7 Extended Port Function Register 24: EPFR24 415
11.4.11.8 Extended Port Function Register 25: EPFR25 416
11.4.11.9 Extended Port Function Register 26: EPFR26 417
11.4.11.10 Extended Port Function Register 27, 30: EPFR27, 30 418
11.4.11.11 Extended Port Function Register 28: EPFR28 419
11.4.11.12 Extended Port Function Register 29, 31, 32: EPFR29, 31, 32 420
11.4.11.13 Extended Port Function Register 35 to 38: EPFR35-38 421
11.4.11.14 Extended Port Function Register 40, 41: EPFR40, 41 424
11.4.11.15 Extended Port Function Register 42: EPFR42 425
11.4.11.16 Extended Port Function Register 43, 44: EPFR43, 44 426
11.4.11.17 Extended Port Function Register 47: EPFR47 427
11.4.11.18 Extended Port Function Register 48 to 51: EPFR48-51 428
11.4.11.19 Extended Port Function Register 52: EPFR52 429
11.4.11.20 Extended Port Function Register 53, 54: EPFR53, 54 430
11.4.12 Port Input Enable Register: PORTEN (Port Enable Register) 431
11.5 Operation 432
11.5.1 Pin I/O Assignment 433
11.5.1.1 Peripheral I/O (Bidirectional) Pin Assignment 434
11.5.1.2 Peripheral Input Assignment 435
11.5.1.3 Peripheral Output Assignment 436
11.5.1.4 External Bus Assignment 437
11.5.1.5 Port Function (Input) Assignment 438
11.5.1.6 Port Function (Output) Assignment 439

11.5.1.7 A/D Converter Input Assignment 440
11.5.1.8 D/A Converter Output Assignment....440
11.5.1.9 LCD Controller V0, V1, V2, V3 (LCDC Reference Voltage Input) Assignment 440
11.5.2 EPFR Setting Priority....441
11.5.3 Notes on Input I/O Relocation Setting....441
11.5.4 Input Blocked by GPORTEN 441
11.5.5 Notes on Pins with the A/D Converter Function....441
11.5.6 Setting when Using the Base Timer TIOA1 Pin 442
11.5.7 Operation at Wake Up from Power Shutdown 442
11.5.8 Notes on Switching the I/O Port Function 442
11.5.9 Port Enable Function at Using Specification Resources....442

12. Interrupt Control (Interrupt Controller)....443

12.1 Overview 444
12.2 Features 445
12.3 Configuration 446
12.4 Registers 447
12.4.1 Interrupt Control Registers 00 to 47: ICR00 to ICR47....448
12.5 Operation....449

13. External Interrupt Input....451

13.1 Overview 452
13.2 Features 453
13.3 Configuration 454
13.4 Registers 455
13.4.1 External Interrupt Factor Register 0/1: EIRR0/EIRR1 (External Interrupt Request Register 0/1) 457
13.4.2 External Interrupt Enable Register 0/1: ENIR0/ENIR1 (Enable Interrupt Request Register 0/1) 458
13.4.3 External Interrupt Request Level Register 0/1: ELVR0/ELVR1 459
13.5 Operation....460
13.6 Setting 462
13.7 Q&A....463
13.8 Notes 464

14. NMI Input 465

14.1 Overview 466
14.2 Features 467
14.3 Configuration 468
14.4 Register 469
14.5 Operation....470
14.6 Usage Example 471

15. Delay Interrupt 473

15.1 Overview 474
15.2 Features 475
15.3 Configuration 476
15.4 Registers 477
15.5 Operation....478
15.6 Restrictions....479

16. Interrupt Request Batch Read 481

16.1 Overview 482
16.2 Features 483
16.3 Configuration 484
16.4 Registers 485
16.4.1 Interrupt Request Batch Read Register 0 Upper-order: IRPR0H (Interrupt Request Peripheral Read register 0H) 486
16.4.2 Interrupt Request Batch Read Register 0 lower-order: IRPR0L (Interrupt Request Peripheral Read register 0L) 487
16.4.3 Interrupt Request Batch Read Register 1 Upper-order: IRPR1H (Interrupt Request Peripheral Read register 1H) 488
16.4.4 Interrupt Request Batch Read Register 1 Lower-order: IRPR1L (Interrupt Request Peripheral Read register 1L) 489
16.4.5 Interrupt Request Batch Read Register 2 Upper-order: IRPR2H (Interrupt Request Peripheral Read register 2H) 490
16.4.6 Interrupt Request Batch Read Register 2 Lower-order: IRPR2L (Interrupt Request Peripheral Read register 2L) 491
16.4.7 Interrupt Request Batch Read Register 3 Upper-order: IRPR3H (Interrupt Request Peripheral Read register 3H) 492
16.4.8 Interrupt Request Batch Read Register 3 Lower-order: IRPR3L (Interrupt Request Peripheral Read register 3L) 493
16.4.9 Interrupt Request Batch Read Register 4 Upper-order: IRPR4H (Interrupt Request Peripheral Read register 4H) 494
16.4.10 Interrupt Request Batch Read Register 4 Lower-order: IRPR4L (Interrupt Request Peripheral Read register 4L) 495
16.4.11 Interrupt Request Batch Read Register 5 Upper-order: IRPR5H (Interrupt Request Peripheral Read register 5H) 496
16.4.12 Interrupt Request Batch Read Register 5 Lower-order: IRPR5L (Interrupt Request Peripheral Read register 5L) 497
16.4.13 Interrupt Request Batch Read Register 6 Upper-order: IRPR6H (Interrupt Request Peripheral Read register 6H) 498
16.4.14 Interrupt Request Batch Read Register 6 Lower-order: IRPR6L (Interrupt Request Peripheral Read register 6L) 499
16.4.15 Interrupt Request Batch Read Register 7 Upper-order: IRPR7H (Interrupt Request Peripheral Read register 7H) 500
16.4.16 Interrupt Request Batch Read Register 7 Lower-order: IRPR7L (Interrupt Request Peripheral Read register 7L) 501
16.4.17 Interrupt Request Batch Read Register 8 Upper-order: IRPR8H (Interrupt Request Peripheral Read register 8H) 502
16.4.18 Interrupt Request Batch Read Register 8 Lower-order: IRPR8L (Interrupt Request Peripheral Read register 8L) 503
16.4.19 Interrupt Request Batch Read Register 9 Upper-order: IRPR9H (Interrupt Request Peripheral Read register 9H) 504
16.4.20 Interrupt Request Batch Read Register 9 Lower-order: IRPR9L (Interrupt Request Peripheral Read register 9L) 505
16.4.21 Interrupt Request Batch Read Register 10 Upper-order: IRPR10H (Interrupt Request Peripheral Read register 10H) 506
16.4.22 Interrupt Request Batch Read Register 10 Lower-order: IRPR10L (Interrupt Request Peripheral Read register 10L) 507
16.4.23 Interrupt Request Batch Read Register 11 Upper-order: IRPR11H (Interrupt Request Peripheral Read register 11H) 508
16.4.24 Interrupt Request Batch Read Register 11 Lower-order: IRPR11L (Interrupt Request Peripheral Read register 11L) 509

16.4.25 Interrupt Request Batch Read Register 12 Upper-order: IRPR12H (Interrupt Request Peripheral Read register 12H) 510
16.4.26 Interrupt Request Batch Read Register 12 Lower-order: IRPR12L (Interrupt Request Peripheral Read register 12L) 511
16.4.27 Interrupt Request Batch Read Register 13 Upper-order: IRPR13H (Interrupt Request Peripheral Read register 13H) 512
16.4.28 Interrupt Request Batch Read Register 13 Lower-order: IRPR13L (Interrupt Request Peripheral Read register 13L) 513
16.4.29 Interrupt Request Batch Read Register 14 Upper-order: IRPR14H (Interrupt Request Peripheral Read register 14H) 514
16.4.30 Interrupt Request Batch Read Register 14 Lower-order: IRPR14L (Interrupt Request Peripheral Read register 14L) 515
16.4.31 Interrupt Request Batch Read Register 15 Upper-order: IRPR15H (Interrupt Request Peripheral Read register 15H) 516
16.5 Operation 517

17. PPG 519

17.1 Overview 520
17.2 Features 521
17.3 Configuration 522
17.4 Registers 523
17.4.1 PPG Cycle Setting Register: PCSR 529
17.4.2 PPG Duty Setting Register: PDUT 530
17.4.3 PPG Control Status Register: PCN 531
17.4.4 General Control Register 10-13: GCN10 to GCN13 535
17.4.5 General Control Register 14, 15: GCN14, GCN15 537
17.4.6 General Control Register 20-25: GCN20 to GCN25 539
17.4.7 PPG Timer Register: PTMR 540
17.4.8 PPG0 Output Division Setting Register: PPGDIV 541
17.5 Operation 542
17.5.1 PWM Operation 543
17.5.2 One-shot Operation 545
17.5.3 Restart Operation 547
17.6 Setting 548
17.7 Q&A 550
17.7.1 How to Set (Rewrite) Cycle and Duty Values 551
17.7.2 How to Enable/Stop PPG Operation 552
17.7.3 How to Set PPG Operation Mode (PWM/One-shot) 553
17.7.4 How to Restart 554
17.7.5 Type and Selection of Count Clock 555
17.7.6 How to Fix the PPG Pin Output Level 556
17.7.7 Type and Selection of Activation Trigger 557
17.7.8 How to Reverse the Output Polarity 559
17.7.9 How to Change a Pin to a PPG Output Pin 560
17.7.10 How to Generate Activation Trigger 561
17.7.11 How to Stop PPG Operation 562
17.7.12 Interrupt-related Registers 563
17.7.13 Type and Selection of Interrupts 565
17.7.14 How to Enable/Disable/Clear Interrupt 566
17.8 Sample Programs 567

17.9 Notes 571

18. Watchdog Timer 573

18.1 Overview 574
18.2 Features 575
18.3 Configuration 576
18.4 Registers 577
18.4.1 Watchdog Control Register 0: WDTCR0 (Watchdog Timer Configuration Register 0) 578
18.4.2 Watchdog Timer 0 Clear Register: WDTCPR0 (Watchdog Timer Clear Pattern Register 0) 580
18.4.3 Watchdog Timer 1 Cycle information Register: WDTCR1 (Watchdog Timer Cycle Information Register 1) 581
18.4.4 Watchdog Timer 1 Clear Register: WDTCPR1 (Watchdog Timer Clear Pattern Register 1) 582
18.5 Operation 583
18.6 Usage Example 585

19. Base Timer 587

19.1 Overview 588
19.2 Features 589
19.2.1 16-/32-bit Reload Timer 590
19.2.2 16-bit PWM Timer 591
19.2.3 16-/32-bit PWC Timer 592
19.2.4 16-bit PPG Timer 593
19.3 Configuration 594
19.4 Registers 595
19.4.1 Common Registers 597
19.4.1.1 Timer Registers 0, 1: BTxTMR (Base Timer 0/1 Timer Register) 597
19.4.1.2 Timer Control Registers 0, 1: BTxTMCR (Base Timer 0/1 Timer Control Register) 598
19.4.1.3 I/O Selection Register: BTSEL01 (Base Timer Select register ch.0 and ch.1) 604
19.4.1.4 Simultaneous Software Activation Register: BTSSSR (Base Timer Software Synchronous Start Register) 605
19.4.2 Registers for 16-/32-bit Reload Timer 606
19.4.2.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control) 607
19.4.2.2 Cycle Setting Registers 0, 1: BTxPCSR (Base Timer 0/1 Pulse Counter Start Register) 608
19.4.3 Registers for 16-bit PWM Timer 609
19.4.3.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control) 610
19.4.3.2 Cycle Setting Registers 0, 1: BTxPCSR (Base Timer 0/1 Pulse Counter Start Register) 612
19.4.3.3 Duty Setting Registers 0, 1: BTxPDUT (Base Timer 0/1 Pulse Duty register) 613
19.4.4 Registers for 16-bit PPG Timer 614
19.4.4.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control) 615
19.4.4.2 L Width Setting Registers 0, 1: BTxPRLL (Base Timer 0/1 Pulse Length of "L" register) 616
19.4.4.3 H Width Setting Registers 0, 1: BTxPRLH (Base Timer 0/1 Pulse Length of "H" register) 617
19.4.5 16-/32-bit PWC Timer Register 618
19.4.5.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control) 619
19.4.5.2 Data Buffer Registers 0, 1: BTxDTBF (Base Timer 0/1 Data Buffer Register) 621
19.5 Operation 622
19.5.1 Selection of Timer Function 623

19.5.2 I/O Allocation 624
19.5.3 32-bit Mode Operation 627
19.5.3.1 32-bit Mode Function 627
19.5.3.2 32-bit Mode Setting 627
19.5.3.3 32-bit Mode Operation 627
19.5.4 16-/32-bit Reload Timer Operation 628
19.5.4.1 Overview 630
19.5.4.2 Operation in Reload Mode 631
19.5.4.3 Operation in One-shot Mode 634
19.5.4.4 32-bit Timer Mode Operation 636
19.5.4.5 Interrupts 638
19.5.4.6 Precautions for Using this Device 639
19.5.5 16-bit PWM Timer Operation 640
19.5.5.1 Overview 641
19.5.5.2 Operation in Reload Mode 642
19.5.5.3 Operation in One-Shot Mode 646
19.5.5.4 Interrupt 648
19.5.5.5 Precautions for Using this Device 649
19.5.6 16-bit PPG Timer Operation 650
19.5.6.1 Overview 651
19.5.6.2 Pulse Width Calculation Method 651
19.5.6.3 Operation in Reload Mode 652
19.5.6.4 Operation in One-shot Mode 656
19.5.6.5 Interrupts 658
19.5.6.6 Application Notes 659
19.5.7 16-/32-bit PWC Timer Operation 660
19.5.7.1 Overview 662
19.5.7.2 Operation During PWC Measurement 665
19.5.7.3 32-bit Timer Mode Operation 668
19.5.7.4 Interrupt 670
19.5.7.5 Application Notes 671

20. Reload Timer 673

20.1 Overview 674
20.2 Features 675
20.3 Configuration 676
20.4 Registers 677
20.4.1 Control Status Register: TMCSR (Timer Control and Status Register) 679
20.4.2 16-bit Timer Register: TMR 684
20.4.3 16-bit Timer Reload Register A, 16-bit Timer Reload Register B: TMRLRA, TMRLRB 685
20.5 Operation 688
20.5.1 Setting 689
20.5.1.1 Count Source 689

20.5.1.2 Timer Underflow period 690
20.5.1.3 Trigger 690
20.5.1.4 Gate 690
20.5.1.5 Counter Operation Selection 691
20.5.1.6 TOUT Pin Level Setting 692
20.5.2 Operation Procedure 694
20.5.2.1 Activation 695
20.5.2.2 Retrigger 697
20.5.2.3 Underflow/Reload 699
20.5.2.4 Generation of Interrupt Requests 699
20.5.2.5 Concurrent Operation of Register Write and a Timer Activation 700
20.5.3 Operations of Each Counter 701
20.5.3.1 Single One-shot Operation 702
20.5.3.2 Single Reload Operation 704
20.5.3.3 Dual One-shot Operation 706
20.5.3.4 Dual Reload Operation 708
20.5.3.5 Compare One-shot Operation 710
20.5.3.6 Compare Reload Operation 713
20.5.3.7 Capture Mode 716
20.5.4 Cascade Input 719
20.5.5 Priority of Concurrent Operations 720
20.6 Application Note 721
20.6.1 Single One-shot Timer 723
20.6.2 Reload Timer 726
20.6.3 PPG 729
20.6.4 PWM 732
20.6.5 PWC 735

21. Free-run Timer 737

21.1 Overview 738
21.2 Features 739
21.3 Configuration 740
21.4 Registers 741
21.4.1 Timer Control Register (Upper Bit): TCCSH 743
21.4.2 Timer Control Register (Lower Bit): TCCSL 745
21.4.3 Compare Clear Register: CPCLR 748
21.4.4 Timer Data Register: TCDT 749
21.5 Operation 750
21.5.1 Count Operation of the Free-run Timer 751
21.5.2 Counting Up 753
21.5.3 Timer Clear 754
21.5.4 Each Clear Operations of the Free-run Timer 755
21.5.5 Timer Interrupt 756
21.6 Setting 757
21.7 Q&A 758

21.7.1 How to Select Internal Clock Dividers 759
21.7.2 How to Select the External Clock 760
21.7.3 How to Enable/Disable the Count Operation of the Free-run Timer 761
21.7.4 How to Clear the Free-run Timer 762
21.7.5 About Interrupt Related Registers 763
21.7.6 How to Enable Compare Clear Interrupt 764
21.7.7 How to Stop the Free-run Timer Operation 765
21.8 Sample Program 766
21.9 Notes 767

22. Output Compare 769

22.1 Overview 770
22.2 Features 771
22.3 Configuration Diagram 773
22.4 Registers 774
22.4.1 Free-run Timer Selection Register: OCFS 776
22.4.2 Output Control Register (Upper Bit): OCSH 777
22.4.3 Output Control Register (Lower Bit): OCSL 779
22.4.4 Compare Register: OCCP 781
22.5 Operation 782
22.5.1 Output Compare Output (Independent Invert) CMOD = "0" 783
22.5.2 Output Compare Output (Coordinated Invert) CMOD = "1" 784
22.5.3 Output Compare Operation Timing 785
22.5.3.1 Compare Register Write 785
22.5.3.2 Compare match, Interrupt 786
22.5.3.3 Pin Output 786
22.6 Setting 787
22.7 Q&A 788
22.7.1 How Can I Set the Compare Value 789
22.7.2 How Can I Set the Compare Mode (Example with OCU1) 790
22.7.3 How Can I Enable/Disable the Compare Operation (Example with OCU0, OCU1) 791
22.7.4 How Can I Set the Compare Pin Output Initial Level (Example with OCU0, OCU1) 792
22.7.5 How Can I Set the Compare Pin OCU0, OCU1 for Output 793
22.7.6 How Can I Clear the Free-run Timer 794
22.7.7 How Can I Enable the Compare Operation 795
22.7.8 Interrupt Related Register 796
22.7.9 Interrupt Type 797
22.7.10 How Can I Enable the Interrupt 798
22.7.11 Calculation Method for the Compare Value 799
22.7.11.1 Toggle Output Pulse 800
22.7.11.2 PWM Output 801
22.8 Sample Program 802
22.9 Notes 804

23. Up/Down Counter 805

23.1 Overview 806
23.2 Features 807
23.3 Configuration 809

23.4 Registers 811
23.4.1 Reload Compare Register (RCR0, RCR1) 812
23.4.2 Up/Down Count Register (UDCR0, UDCR1) 813
23.4.3 Counter Control Register (CCR0, CCR1) 814
23.4.4 Counter Status Register (CSR0, CSR1) 820
23.5 Interrupt 823
23.6 Operation and Setting Procedure Examples 825
23.6.1 Operation in Timer Mode 829
23.6.2 Operation in Up/Down Count Mode 831
23.6.3 Operation in the Phase Difference Count Mode (Multiply-by-Two) 834
23.6.4 Operation in the Phase Difference Count Mode (Multiply-by-four) 836

24. Input Capture 839

24.1 Overview 840
24.2 Features 841
24.3 Configuration 842
24.4 Registers 843
24.4.1 Input Capture Data Register: IPCP 845
24.4.2 Free-run Timer Selection Register: ICFS 846
24.4.3 Input Capture Control Register: ICS 847
24.4.4 LIN SYNCH FIELD Switching Register: LSYNS 849
24.5 Operation 851
24.5.1 Capture and Interrupt Timings 852
24.5.2 Edge Detection Specifications for Input Capture and their Operations 853
24.6 Setting 855
24.7 Q&A 856
24.7.1 Effective Edge Polarity of External Input: Types and How to Select 857
24.7.2 How to Enable External Input Pins (ICU0, ICU1, ICU2, ICU3, ICU4, ICU5, …) 858
24.7.3 About Interrupt Related Registers 859
24.7.4 About Interrupt Types 860
24.7.5 How to Enable Interrupt 861
24.7.6 How to Measure the Pulse Width of the Input Signal 862
24.8 Sample Program 863
24.9 Notes 864

25. Real-time Clock (RTC) 865

25.1 Overview 866
25.2 Features 867
25.3 Configuration 868
25.4 Registers 869
25.4.1 RTC Control Register: WTCR 870
25.4.2 Sub-second Register: WTBR 875
25.4.3 Day/Hour/Minute/Second Register: WTDR/WTHR/WTMR/WTSR 876
25.5 Operation 879
25.6 Setting 881
25.7 Q&A 882
25.7.1 How to Set the 0.5 Second Count Interval 883
25.7.2 How to Initialize the Real-time Clock 884
25.7.3 How to Set/Update Number of Days (Day) and Time (Hour/Minute/Second) 885

25.7.4 How to Start/Stop the Count of the Real-time Clock 886
25.7.5 How to Confirm That the Real-time Clock is Running 887
25.7.6 How to Know the Number of Days and Time 888
25.7.7 How to Stop the Real-time Clock 889
25.7.8 How to Calibrate the Real-time Clock 890
25.7.9 What Are Interrupt Related Registers 891
25.7.10 What are the Interrupt Types and How to Select Them 892
25.7.11 How to Enable Interrupts 893
25.8 Sample Program 894
25.9 Notes 895

26. RTC/WDT1 Calibration 897

26.1 Overview 898
26.2 Features 899
26.3 Configuration 900
26.4 Registers 901
26.4.1 Calibration Unit Control Register 0: CUCR0 (Calibration Unit Control Register 0) 902
26.4.2 Sub Clock Timer Data Register: CUTD0 (Calibration Unit Timer Data register 0) 904
26.4.3 Main Oscillation Timer Result Register 0: CUTR0 (Calibration Unit Timer Result register 0) 905
26.4.4 Calibration Unit Control Register 1: CUCR1 (Calibration Unit Control Register 1) 906
26.4.5 CR Clock Timer Data Register: CUTD1 (Calibration Unit Timer Data register 1) 908
26.4.6 Main Oscillation Timer Result Register 1: CUTR1 (Calibration Unit Timer Result register 1) 909
26.4.7 CR Oscillation Trimming Setting Register: CRTR (CR oscillator calibration Trimming Register) 910
26.5 Operation 911
26.5.1 Real-Time Clock (RTC) Calibration 912
26.5.2 WDT1 Calibration (CR Clock Calibration) 913
26.5.3 Notes 914

27. Power Consumption Control 915

27.1 Overview 916
27.2 Features 917
27.3 Configuration 918
27.4 Registers 920
27.4.1 Standby Control Register: STBCR (Standby mode Control Register) 921
27.4.2 PMU Control Register: PMUCTLR (Power Management Unit Control register) 923
27.4.3 Power on Timing Control Register: PWRTMCTL 925
27.4.4 PMU Interrupt Flag Register 0: PMUINTF0 (Power Management Unit Interrupt Flag0 register) 926
27.4.5 PMU Interrupt Flag Register 1: PMUINTF1 (Power Management Unit Interrupt Flag1 register) 927
27.4.6 PMU Interrupt Flag Register 2: PMUINTF2 (Power Management Unit Interrupt Flag2 register) 928
27.5 Operation 930
27.5.1 Clock Control 931
27.5.2 List of Clock Supply in Low-power Consumption Mode 932
27.5.3 Sleep Mode 933
27.5.4 Standby Mode: Watch Mode 935
27.5.5 Standby Mode: Watch Mode with Power-shutdown 937
27.5.6 Standby Mode: Stop Mode 940
27.5.7 Standby Mode: Stop Mode with Power-shutdown 942
27.5.8 Stop State of Microcontroller 946
27.5.9 Transition to Illegal Standby Mode 947

27.5.10 Restrictions on Power-shutdown and Normal Standby Control 948
27.6 Example of Use 950

28. Low-voltage Detection (Internal Low-voltage Detection) 951

28.1 Overview 952
28.2 Features 953
28.3 Configuration 954
28.4 Registers 955
28.4.1 Internal Low-Voltage Detection Register: LVD (Low-Voltage Detect Internal Power Fall Register) 956
28.5 Operation 958
28.5.1 Internal Low-voltage Detection 959
28.6 Notes 960

29. Low-voltage Detection (External Low-voltage Detection) 961

29.1 Overview 962
29.2 Features 963
29.3 Configuration 964
29.4 Registers 965
29.4.1 External Low-voltage Detection Rise Detection Register: LVD5R (Low-voltage Detect external 5 V Rise register) 966
29.4.2 External Low-voltage Detection Fall Detection Register: LVD5F (Low-voltage Detect external 5 V Fall register) 967
29.5 Operation 969
29.5.1 External Low-voltage Detection 970
29.6 Notes 971

30. Wild Register 973

30.1 Overview 974
30.2 Features 975
30.3 Configuration 976
30.4 Registers 977
30.4.1 Wild Register Data Enable Register: WREN 979
30.4.2 Wild Register Address Register 00 to 15: WRAR00 to 15 980
30.4.3 Wild Register Data Register 00 to 15: WRDR00 to 15 981
30.5 Operation 982
30.6 Usage Example 983

31. Clock Supervisor 985

31.1 Overview 986
31.2 Configuration 987
31.3 Registers 988
31.4 Operation 991
31.4.1 Initial State 992
31.4.2 Stopping CR Oscillator and the Clock Supervisor Function 993
31.4.3 Re-enabling the Clock Supervisor 994
31.4.4 Sub Clock Mode 995
31.4.5 Stop Mode 996
31.4.6 Watch Mode 997
31.4.7 Checking the Reset Factor Using the Clock Supervisor 998
31.4.8 Return from CR Clock 999

32. Sound Generator 1001
32.1 Overview 1002
32.2 Features 1003
32.3 Configuration 1004
32.4 Registers 1005
32.4.1 DMA Transfer Update Enable Register: SGDER (SG DMA Enable Register) 1007
32.4.2 Sound Control Register: SGCR (SG Control Register) 1009
32.4.3 Amplitude Data Register: SGAR (SG Amplitude Register) 1012
32.4.4 Frequency Data Register: SGFR (SG Frequency Register) 1013
32.4.5 Tone Outputs Number Register: SGNR (SG tone Number Register) 1014
32.4.6 Cycle Register: SGTCR (SG Tone Cycle Register) 1015
32.4.7 Increment Decrement Data Register: SGIDR (SG Increment Decrement Register) 1016
32.4.8 PWM Cycles Number Data Register: SGPCR (SG PWM Cycle Register) 1017
32.4.9 DMA Transfer Indirect Register: SGDMAR (SG DMA Register) 1018
32.5 Operation 1019
32.5.1 Relation of Amplitude Data Register (SGAR) and PWM Pulse 1020
32.5.2 Relation of Frequency Data Register (SGFR) and Tone Pulse Signals 1021
32.5.3 Relation of PWM Cycles Number Data Register (SGPCR) and PWM Cycle 1022
32.5.4 Relation of DMA Transfer Update Enable Register (SGDER) and DMA Transfers Count/DMA Transfer Size/Transfer Byte Location 1023
32.5.4.1 DMA Transfers Count 1023
32.5.4.2 DMA Transfer Size 1023
32.5.4.3 Transfer Byte Location for DMA Transfer Indirect Register 1024
32.5.4.4 DMA Transfer Image 1027
32.5.5 Operation of Sound Generator 1028
32.5.6 Sound Generator Continuous Operation by CPU 1031
32.5.7 Sound Generator Operation Coordinated with DMA 1034
32.5.8 When DMA Transfer of 4 Bytes × 2 is Performed N Times 1035
32.5.8.1 When DMA Transfer of 2 Bytes × 2 Is Performed N Times 1037
32.5.8.2 When DMA Transfer of 1 Byte × 1 is Performed N Times 1039
32.5.8.3 For DMA Transfer of 4 Bytes × 2 N Times and DMA Transfer of 2 Bytes × 1 M Times (Transfer Bytes Number Change During Sound Output) 1041
32.5.8.4 For DMA Transfer of 4 Bytes × 2 N Times and DMA Transfer of 4 Bytes × 2 M Times (Transfer Bytes Number and Increment Decrement Setting Change During Sound Output) 1044

33. Stepping Motor Controller 1047
33.1 Overview 1048
33.2 Features 1049
33.3 Configuration 1050
33.4 Registers 1051
33.4.1 PWM Control Register: PWC 1053
33.4.2 PWM1&2 Compare Register: PWC1/PWC2 1055
33.4.3 PWM1 Selection Register: PWS1 1056
33.4.4 PWM2 Selection Register: PWS2 1057
33.5 Operation 1059
33.5.1 PWM Operation 1060
33.5.2 PWM Compare Register Loading with the BS Bit 1061

33.5.3 Selection of Motor Drive Signals 1063
33.6 Setting 1064
33.7 Q&A 1065
33.7.1 How to Set Cycle and Duty 1066
33.7.2 How to Enable/Stop PWM Operation 1067
33.7.3 How to Reflect the Duty Change 1068
33.7.4 Type and Selection of Operating Clock 1069
33.7.5 How to Change the Motor Drive Signals 1070
33.7.6 How to Assign a Pin as a PWM Output Pin 1071
33.7.7 How to Assign a Pin as an A/D Converter Analog Input Pin 1072
33.8 Sample Programs 1073
33.9 Notes 1074

34. Regulator Control 1075

34.1 Overview 1076
34.2 Features 1077
34.3 Configuration 1078
34.4 Registers 1079
34.4.1 Regulator Output Voltage Select Register: REGSEL 1080
34.5 Operation 1082

35. LCD Controller 1083

35.1 Overview 1084
35.2 Features 1085
35.3 Configuration 1086
35.4 Registers 1087
35.4.1 LCD Control Register 0: LCR0 1088
35.4.2 Data Memory for Display: VRAM 1090
35.4.3 LCDC Control Register 1: LCR1 1092
35.4.4 Common Pin Switching Register: LCDCMR 1093
35.4.5 LCDC Static Control Register: LCRS 1094
35.4.6 Static LCD Display Data Register: LDR 1096
35.5 Operation 1098
35.5.1 Operation of LCD Controller/Driver (LCDC) 1099
35.5.2 1/2 Duty Output Waveform 1101
35.5.3 1/3 Duty Output Waveform 1104
35.5.4 1/4 Duty Output Waveform 1107
35.5.5 Static Drive Output Waveform 1110
35.6 Setting 1113
35.7 Q&A 1115
35.7.1 How Can I Set Pins to COM Output Pins or SEG Output Pins 1116
35.7.2 How to Set VRAM 1119
35.7.3 How Can I Set the Frame Cycle 1120
35.7.4 How Can I Set the Bias 1122
35.7.5 How Can I Set the Duty 1123
35.7.6 How Can I Control the LCD Operation Start/Stop 1124
35.7.7 How Can I Execute/Cancel the Display 1125
35.7.8 How Can I Display During the Watch Mode 1126
35.7.9 How Can I Select Either Internal or External for the Division Resistor 1127

35.7.10 How Can I Select Pin of V3 Voltage 1128
35.7.11 How Can I Select either Internal or External for the Division Resistor 1129
35.7.12 How Can I Adjust the Brightness When the Internal Division Resistor is Used 1131
35.7.13 How Can I Block the Current with the External Division Resistor When the LCD Stops 1132
35.7.14 How Can I Display/Non-display the LCD with Static Drive (ST0 to ST8) 1133
35.8 Sample Program 1134
35.9 Notes 1139

36. External Bus Interface 1141

36.1 Overview 1142
36.2 Features 1143
36.3 Configuration 1144
36.4 Registers 1145
36.4.1 CS Area Setting Registers: ASR0 to ASR3 (Area Setting Register 0-3) 1146
36.4.2 CS Bus Setting Registers: ACR0 to ACR3 (Area Configuration Register 0-3) 1149
36.4.3 CS Wait Registers: AWR0 to AWR3 (Area Wait Register 0-3) 1151
36.4.4 External DMA Transfer Registers: DMAR0-3 (DMA transfer Register 0-3) 1157
36.5 Operation 1159
36.5.1 External Pin Table 1160
36.5.2 External Bus Signal Protocol 1161
36.5.3 Address Alignment 1168
36.5.4 Split Access 1169
36.5.5 Data Alignment 1170
36.5.6 Address Information 1174
36.5.7 Idle Cycle Insertion Function 1175
36.5.8 External Bus Output Signal Timing Settings 1176
36.5.9 RDY Pin Access Cycle Extension Function 1181
36.5.10 CS Setting Flow 1182
36.5.11 Example of Connecting to Asynchronous Memory 1188
36.5.12 Example of Connection to Little Endian Device 1190

37. Bus Performance Counters 1191

37.1 Overview 1192
37.2 Features 1193
37.3 Configuration 1194
37.4 Registers 1195
37.4.1 BPC-A Control Register: BPCCRA (Bus Performance Counter Control Register A) 1196
37.4.2 BPC-B Control Register: BPCCRB (Bus Performance Counter Control Register B) 1198
37.4.3 BPC-C Control Register: BPCCRC (Bus Performance Counter Control Register C) 1199
37.4.4 BPC-A Count Register: BPCTRA (Bus Performance Counter Register A) 1200
37.4.5 BPC-B Count Register: BPCTRB (Bus Performance Counter Register B) 1201
37.4.6 BPC-C Count Register: BPCTRC (Bus Performance Counter Register C) 1202
37.5 Operation 1203
37.5.1 Setting 1204
37.5.2 Starting and Stopping 1206
37.5.3 Operation 1207
37.5.4 Measurement and Result Processing 1208

38. CRC 1211

38.1 Overview 1212
38.2 Features 1213
38.3 Configuration 1214
38.4 Registers 1215
38.4.1 CRC Control Register: CRCCR 1216
38.4.2 CRC Initial Value Register: CRCINIT 1218
38.4.3 CRC Input Data Register: CRCIN 1219
38.4.4 CRC Register: CRCR 1220
38.5 Operation 1221
38.5.1 CRC Definition 1222
38.5.2 Reset Operation 1223
38.5.3 Initialization 1224
38.5.4 Byte and Bit Orders 1225
38.5.5 CRC Calculation Sequence 1226
38.5.6 Examples 1227
38.5.6.1 Example 1 CRC16, Fixed Byte Input 1228
38.5.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths 1231
38.5.6.3 Example 3 CRC32, Byte Order, Big-endian 1232
38.5.6.4 Example 4 CRC32, Byte Order, Little-endian 1233

39. RAMECC 1235

39.1 Overview 1236
39.2 Features 1237
39.3 Configuration 1238
39.4 Registers 1240
39.4.1 ECC Error Control Register XBS RAM: EECSRX (ECC Error Control and Status Register XBS RAM) 1241
39.4.2 Single-bit ECC Error Address Register XBS RAM: SEEARX 1243
39.4.3 Double-bit ECC Error Address Register XBS RAM: DEEARX 1244
39.4.4 ECC False Error Generation Address Register XBS RAM: EFEARX (ECC False Error Address Register XBS RAM) 1245
39.4.5 ECC False Error Generation Control Register XBS RAM: EFECRX (ECC False Error Control Register XBS RAM) 1246
39.4.6 ECC Error Control Register Backup-RAM: EECSRA (ECC Error Control and Status Register Backup-RAM) 1248
39.4.7 Single-bit ECC Error Address Register Backup-RAM: SEEARA 1250
39.4.8 Double-bit ECC Error Address Register BACKUP-RAM: DEEARA 1251
39.4.9 ECC False Error Generation Address Register Backup-RAM: EFEARA (ECC False Error Address Register Backup-RAM) 1252
39.4.10 ECC False Error Generation Control Register Backup-RAM: EFECRA (ECC False Error Control Register Backup-RAM) 1253
39.5 Operation 1255
39.5.1 ECC Generation 1256
39.5.2 ECC Inspection 1257
39.5.3 Interrupt by Error Detection 1258
39.5.4 Test Function 1259

40. Multi Function Serial Interface 1261

40.1 Overview 1262
40.2 Features 1263

40.2.1 UART 1264
40.2.2 CSIO 1265
40.2.3 LIN-UART 1266
40.2.4 I²C 1267
40.2.5 Notes 1268
40.3 Configuration 1269
40.4 Registers 1273
40.4.1 Common Registers 1277
40.4.1.1 Serial Mode Register: SMR 1278
40.4.1.2 FIFO Control Register 1: FCR1 1281
40.4.1.3 FIFO Control Register 0: FCR0 1283
40.4.1.4 FIFO BYTE Register: FBYTE 1287
40.4.2 Registers for UART 1289
40.4.2.1 Serial Control Register: SCR 1290
40.4.2.2 Serial Status Register: SSR 1292
40.4.2.3 Extended Serial Control Register: ESCR 1294
40.4.2.4 Receive Data Register/Transmit Data Register: RDR/TDR 1296
40.4.2.5 Baud Rate Generator Register: BGR 1299
40.4.3 Registers for CSIO 1300
40.4.3.1 Serial Control Register: SCR 1301
40.4.3.2 Serial Status Register: SSR 1303
40.4.3.3 Extended Serial Control Register: ESCR 1305
40.4.3.4 Receive Data Register/Transmit Data Register: RDR/TDR 1306
40.4.3.5 Baud Rate Generator Register: BGR 1309
40.4.4 Registers for LIN-UART 1310
40.4.4.1 Serial Control Register: SCR 1311
40.4.4.2 Serial Status Register: SSR 1313
40.4.4.3 Extended Serial Control Register: ESCR 1316
40.4.4.4 Receive Data Register/Transmit Data Register: RDR/TDR 1318
40.4.4.5 Baud Rate Generator Register: BGR 1320
40.4.5 Registers for I²C 1321
40.4.5.1 I²C Bus Control Register: IBCR 1322
40.4.5.2 Serial Status Register: SSR 1327
40.4.5.3 I²C Bus Status Register: IBSR 1330
40.4.5.4 Receive Data Register/Transmit Data Register: RDR/TDR 1334
40.4.5.5 Baud Rate Generator Register: BGR 1336
40.4.5.6 I²C 7-bit Slave Address Mask Register: ISMK 1337
40.4.5.7 I²C 7-bit Slave Bus Address Register: ISBA 1338
40.5 Operation of UART 1339
40.5.1 Interrupt of UART 1340
40.5.1.1 List of Interrupt of UART 1341
40.5.1.2 Reception Interrupts and Flag Setting Timing 1342

40.5.1.3 Interrupts When Using Reception FIFO and Flag Setting Timing 1344
40.5.1.4 Trans Mission Interrupts and Flag Setting Timing 1346
40.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing 1347
40.5.2 Operation of UART 1348
40.5.2.1 Transmission/Reception Data Format 1349
40.5.2.2 Transmission Operation 1350
40.5.2.3 Reception Operation 1351
40.5.2.4 Clock Selection 1352
40.5.2.5 Start Bit Detection 1352
40.5.2.6 Stop Bit 1352
40.5.2.7 Error Detection 1352
40.5.2.8 Parity Bit 1353
40.5.2.9 Data Signaling Method 1353
40.5.2.10 Data Transfer Method 1353
40.5.2.11 UART Baud Rate Selection/Setting 1354
40.5.3 Setup Procedure and Program Flow 1359
40.5.3.1 Operation Mode 0 (One-to-One Connection) 1360
40.5.3.2 Operation Mode 1 (One-to-N Connection) 1362
40.6 Operation of CSIO 1365
40.6.1 Interrupts of CSIO 1366
40.6.1.1 List of Interrupts of CSIO 1367
40.6.1.2 Reception Interrupts and Flag Setting Timing 1368
40.6.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing 1369
40.6.1.4 Transmission Interrupts and Flag Setting Timing 1371
40.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing 1372
40.6.2 Operation of CSIO 1373
40.6.2.1 Normal Transfer (I) 1374
40.6.2.2 Normal Transfer (II) 1378
40.6.2.3 SPI Transfer (I) 1382
40.6.2.4 SPI Transfer (II) 1386
40.6.2.5 Baud Rate Generation 1390
40.6.3 Setup Procedure and Program Flow 1392
40.6.3.1 Connections between Chips 1393
40.6.3.2 Flowchart 1394
40.7 Operation of LIN-UART 1396
40.7.1 Interrupts of LIN-UART 1397
40.7.1.1 List of Interrupts of LIN-UART Interface 1398
40.7.1.2 Reception Interrupts and Flag Setting Timing 1400
40.7.1.3 Interrupts When Using Reception FIFO and Flag Setting Timing 1402
40.7.1.4 Transmission Interrupts and Flag Setting Timing 1404
40.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing 1405

40.7.2 Operation of LIN-UART 1406
40.7.2.1 Master Device Operation 1406
40.7.2.2 Slave Device Operation 1411
40.7.2.3 LIN-UART Baud Rate Selection/Setting 1416
40.7.3 Setup Procedure and Program Flow 1417
40.7.3.1 Inter-CPU Connection 1417
40.7.3.2 Flowchart Example 1418
40.8 Operation of I²C 1422
40.8.1 Interrupts of I²C 1423
40.8.2 Operation for I²C Interface Communication 1425
40.8.2.1 I²C Bus Start Condition 1425
40.8.2.2 I²C Bus Stop Condition 1425
40.8.2.3 I²C Bus Repeated Start Condition 1425
40.8.2.4 I²C Bus Error 1426
40.8.2.5 Baud Rate Generation 1427
40.8.3 I²C Master Mode 1429
40.8.3.1 Start Condition Generation 1430
40.8.3.2 Slave Address Output 1431
40.8.3.3 Acknowledge Reception by Transmitting First Byte 1433
40.8.3.4 Data Transmission by Master 1439
40.8.3.5 Data Reception by Master System 1453
40.8.3.6 Arbitration Lost 1461
40.8.3.7 Wait of the Master Mode 1461
40.8.3.8 Repetition Start Condition Issue When DMA Mode Enabled (SSR:DMA=1) 1462
40.8.4 I²C Slave Mode 1463
40.8.4.1 Detection of Slave Address Matching 1464
40.8.4.2 Data Direction Bit 1465
40.8.4.3 Reception by Slave Device 1466
40.8.4.4 Transmission by Slave Device 1471
40.8.5 Bus Error 1472
40.8.5.1 Bus Error Generation Condition 1472
40.8.5.2 Bus Error Operation 1472
40.8.6 Example of I²C Flowchart 1473
40.8.6.1 Example of I²C Flowchart (FIFO Memory Not Used) (When DMA Mode is Disable (SSR: DMA=0)) 1474
40.8.6.2 Example of I²C Flowchart (FIFO Memory Not Used) (When DMA Mode is Enable (SSR: DMA=1)) 1477

41. LIN-UART 1481

41.1 Overview 1482
41.2 Features 1483
41.2.1 Functions 1483
41.2.2 Operation Mode 1485

41.3 Configuration 1487
41.3.1 Block Diagram of the LIN-UART 1487
41.3.2 Explanation of Each Block 1489
41.3.2.1 Reload Counter 1490
41.3.2.2 Reception Control Circuit 1490
41.3.2.3 Reception Shift Register 1490
41.3.2.4 Reception Data Register (RDR) 1490
41.3.2.5 Transmission Control Circuit 1490
41.3.2.6 Transmission Shift Register 1491
41.3.2.7 Transmission Data Register (TDR) 1491
41.3.2.8 Error Detection Circuit 1491
41.3.2.9 Over-sampling Circuit 1491
41.3.2.10 Interrupt Generation Circuit 1491
41.3.2.11 LIN Synch Break/LIN Synch Field Detection Circuit 1491
41.3.2.12 LIN Synch Break Generation Circuit 1491
41.3.2.13 Bus Idle Detection Circuit 1492
41.3.2.14 Serial Mode Register (SMR) 1492
41.3.2.15 Serial Control Register (SCR) 1492
41.3.2.16 Serial Status Register (SSR) 1492
41.3.2.17 Extended Status Control Register (ESCR) 1493
41.3.2.18 Extended Communication Control Register (ECCR) 1493
41.4 Registers 1494
41.4.1 Serial Control Register: SCR 1497
41.4.2 Serial Mode Register: SMR 1500
41.4.3 Serial Status Register: SSR 1503
41.4.4 Reception Data Register/Transmission Data Register: RDR/TDR 1507
41.4.4.1 Reception Data Register: RDR 1507
41.4.4.2 Transmission Data Register: TDR 1508
41.4.5 Extended Status Control Register: ESCR 1509
41.4.6 Extended Communication Control Register: ECCR 1513
41.4.7 Baud Rate Generator Register: BGR 1516
41.5 Interrupts 1517
41.5.1 Overview 1518
41.5.1.1 Interrupts of LIN-UART 1519
41.5.1.2 Reception Interrupt 1520
41.5.1.3 Transmission Interrupt 1521
41.5.1.4 LIN Synch Break Interrupt 1521
41.5.1.5 LIN Synch Field Edge Detection Interrupt 1522
41.5.2 Generation of Reception Interrupt and Flag Setting Timing 1523
41.5.2.1 Generation of Reception Interrupt and Flag Setting Timing 1524
41.5.3 Occurrence of Transmission Interrupt and Flag Timing 1525
41.5.3.1 Occurrence of Transmission Interrupt and Flag Timing 1526

41.5.3.2 Transmission Interrupt Request Generation Timing 1527
41.6 Baud Rates 1528
41.6.1 Selection of Baud Rates 1529
41.6.1.1 Baud Rate Obtained when a Dedicated Baud Rate Generator (Reload Counter) Divides the Frequency of the Internal Clock 1529
41.6.1.2 Baud Rate Obtained when a Dedicated Baud Rate Generator (Reload Counter) Divides the Frequency of the External Clock 1529
41.6.1.3 Baud Rate Due from External Clock (One-to-one Mode) 1530
41.6.2 Baud Rate Setting 1531
41.6.2.1 Baud Rate Calculations 1531
41.6.2.2 Baud Rate Setting Example for Each Clock Frequency 1532
41.6.2.3 Use of the External Clock 1533
41.6.2.4 Reload Counter Operation 1534
41.6.3 Reload Counter 1535
41.6.3.1 Reload Counter Functions 1535
41.6.3.2 Count Start 1535
41.6.3.3 Restart 1535
41.6.3.4 Counter Clear 1535
41.6.3.5 Simple Timer Use 1536
41.7 Operation 1537
41.7.1 Overview 1538
41.7.1.1 Operation Mode 1538
41.7.1.2 Connection Method between CPUs 1539
41.7.1.3 Synchronization System 1539
41.7.1.4 Signaling System 1539
41.7.1.5 Transmission/Reception Start 1539
41.7.1.6 Stopping of Transmission/Reception 1539
41.7.1.7 Stopping of Transmission/Reception in Progress 1539
41.7.2 Asynchronous Mode (Operation Modes 0 and 1) 1540
41.7.2.1 Transmission/Reception Data Format 1540
41.7.2.2 Transmission Operation 1541
41.7.2.3 Reception Operation 1541
41.7.2.4 Clock Usage 1541
41.7.2.5 Stop Bit 1541
41.7.2.6 Error Detection 1542
41.7.2.7 Parity 1542
41.7.2.8 Data Signaling Method 1542
41.7.2.9 Data Transfer Method 1542
41.7.3 Synchronous Mode (Operation Mode 2) 1543
41.7.3.1 Transmission/Reception Data Format 1543
41.7.3.2 Master/Slave Setting 1543
41.7.3.3 Sampling Edge Selection 1544

41.7.3.4 Clock Supply 1544
41.7.3.5 Clock Usage 1545
41.7.3.6 Delayed Serial Clock 1545
41.7.3.7 Sequential Serial Clock 1546
41.7.3.8 Parity 1546
41.7.3.9 Data Signaling Method 1546
41.7.3.10 Stop Bit 1547
41.7.3.11 Error Detection 1547
41.7.3.12 Communication Start 1547
41.7.3.13 Communication End 1547
41.7.3.14 Data Transfer Method 1547
41.7.4 LIN Mode (Operation Mode 3) 1548
41.7.4.1 Transmission/Reception Data Format 1548
41.7.4.2 LIN Master Operation 1548
41.7.4.3 LIN Slave Operation 1549
41.7.4.4 LIN Bus Timing 1550
41.7.4.5 Baud Rate Calculation 1551
41.7.4.6 Clock Usage 1552
41.7.4.7 Data Signaling Method 1552
41.7.4.8 Stop Bit 1552
41.7.4.9 Error Detection 1552
41.7.5 Direct Access to the Serial Pin 1553
41.7.6 Bidirectional Communication Function (Normal Mode) 1554
41.7.6.1 Connection between CPUs 1555
41.7.6.2 Communication Procedure 1555
41.7.7 Master/Slave Mode Communication Function (Multi-processor Mode) 1556
41.7.7.1 Connection between CPUs 1556
41.7.7.2 Function Selection 1557
41.7.7.3 Communication Procedure 1557
41.7.8 LIN Communication Function 1559
41.7.8.1 LIN Master/Slave Communication Function 1559
41.7.8.2 LIN Device Connection 1559
41.7.9 LIN-UART Sample Flowchart in LIN Communication Mode (Operation Mode 3) 1560
41.7.9.1 LIN-UART as a Master Device 1560
41.7.9.2 LIN-UART as a Slave Device 1561
41.8 Notes on Usage 1562
41.8.1 Operation Enable 1562
41.8.2 Communication Mode Setting 1562
41.8.3 Timing of Enabling Transmission Interrupt 1562
41.8.4 Operation Setting Change 1562
41.8.5 Detection of a LIN Synch Break 1562
41.8.6 LIN Slave Setting 1563

41.8.7 Program Compatibility 1563
41.8.8 Address/Data Format Selection Bit (SCR:AD) 1563
41.8.9 LIN-UART Software Reset 1563
41.8.10 Detection of LIN Synch Field in Input Capture 1563
41.8.11 Bus Idle Detection 1563
41.9 Notes on DMAC Linkage Operation 1564
41.9.1 Transmission Operation 1564
41.9.2 Reception Operation 1564

42. CAN 1565

42.1 Overview 1566
42.2 Features 1567
42.3 Configuration 1568
42.4 Registers 1569
42.4.1 Overview 1570
42.4.2 Overall Control Registers 1579
42.4.2.1 CAN Control Register (CTRLR) 1580
42.4.2.2 CAN Status Register (STATR) 1583
42.4.2.3 CAN Error Counter (ERRCNT) 1586
42.4.2.4 CAN Bit Timing Register (BTR) 1587
42.4.2.5 CAN Interrupt Register (INTR) 1588
42.4.2.6 CAN Test Register (TESTR) 1590
42.4.2.7 CAN Prescaler Extension Register (BRPER) 1592
42.4.3 Message Interface Register 1593
42.4.3.1 IFx Command Request Register (IFxCREQ) 1594
42.4.3.2 IFx Command Mask Register (IFxCMSK) 1597
42.4.3.3 IFx Mask Registers 1, 2 (IFxMSK1, IFxMSK2) 1602
42.4.3.4 IFx Arbitration Registers 1, 2 (IFxARB1, IFxARB2) 1603
42.4.3.5 IFx Message Control Register (IFxMCTR) 1604
42.4.3.6 IFx Data Registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2) 1605
42.4.4 Message Object 1606
42.4.4.1 Configuration of Message Object 1606
42.4.4.2 Functions of Message Object 1607
42.4.5 Message Handler Registers 1613
42.4.5.1 CAN Transmission Request Registers (TREQR1, TREQR2) 1614
42.4.5.2 CAN Data Update Registers (NEWDT1, NEWDT2) 1617
42.4.5.3 CAN Interrupt Pending Registers (INTPND1, INTPND2) 1619
42.4.5.4 CAN Message Valid Registers (MSGVAL1, MSGVAL2) 1621
42.4.6 CAN Prescaler Register (CANPRE) 1623
42.5 Operation 1625
42.5.1 Message Object 1626
42.5.1.1 Message Object 1626
42.5.1.2 Data Transmission/Reception with Message RAM 1627
42.5.2 Message Transmission Operation 1628

42.5.2.1 Message Transmission 1628
42.5.2.2 Transmission Priority 1628
42.5.2.3 Transmission Message Object Setting 1629
42.5.2.4 Update of Transmission Message Object 1629
42.5.3 Message Reception Operation 1630
42.5.3.1 Reception Message Acceptance Filter 1630
42.5.3.2 Reception Priority 1630
42.5.3.3 Data Frame Reception 1630
42.5.3.4 Remote Frame 1631
42.5.3.5 Reception Message Object Setting 1631
42.5.3.6 Reception Message Processing 1632
42.5.4 FIFO Buffer Function 1633
42.5.4.1 Configuration of FIFO Buffer 1633
42.5.4.2 Message Reception by FIFO Buffer 1633
42.5.4.3 Reading from FIFO Buffer 1633
42.5.5 Interrupt Function 1635
42.5.6 Bit Timing and CAN System Clock (fsys) Generation 1636
42.5.7 Test Mode 1639
42.5.7.1 Test Mode Setting 1639
42.5.7.2 Silent Mode 1639
42.5.7.3 Loopback Mode 1640
42.5.7.4 Combination of Silent and Loopback Modes 1641
42.5.7.5 Basic Mode 1642
42.5.7.6 Software Control of the CAN_TX Pin 1643
42.5.8 Software Initialization 1644
42.6 Limitations 1645
42.6.1 INIT Bit 1646
42.6.1.1 Limitations 1647
42.6.1.2 Workaround 1648

43. D/A Converter 1649

43.1 Overview 1650
43.2 Features 1651
43.3 Configuration 1652
43.4 Registers 1653
43.4.1 DA Control Register: DACR 1654
43.4.2 DA Data Register: DADR 1655
43.5 Operation 1656
43.6 Note 1657

44. A/D Converter 1659

44.1 Overview 1660
44.2 Features 1661
44.3 Configuration 1662

44.4 Registers 1663
44.4.1 Analog Input Enable Register: ADER 1664
44.4.2 A/D Control Status Register (Upper): ADCS1 1665
44.4.3 A/D Control Status Register (Lower): ADCS0 1668
44.4.4 Data Register: ADCR0, ADCR1 1670
44.4.5 Conversion Time Setting Register: ADCT 1671
44.4.6 A/D Start/Completion Channel Setting Register: ADSCH, ADECH 1673
44.4.7 Extended Analog Input Enable Register: EADERLL 1676
44.4.8 Extended A/D Control Status Register: EADCS 1677
44.5 Operation 1679
44.5.1 Single Conversion Operation 1680
44.5.2 Scan Conversion Operation 1681
44.5.3 Conversion Mode 1682
44.6 Setting 1684
44.7 Q&A 1687
44.7.1 Conversion Mode Type and Setting Method 1688
44.7.2 How Can I Specify the Bit Length 1688
44.7.3 How Can I Select Channels 1689
44.7.4 How Can I Set the Conversion Time 1693
44.7.5 How Can I Enable the Analog Pin Input 1694
44.7.6 How Can I Select the A/D Converter Activation Method 1696
44.7.7 How Can I Activate the A/D Converter 1696
44.7.8 How Can I Check the Conversion Completion 1697
44.7.9 How Can I Read the Conversion Value 1697
44.7.10 How Can I Stop the A/D Conversion Operation Forcibly 1698
44.7.11 Interrupt-related Register 1698
44.7.12 Interrupt Type 1698
44.7.13 How Can I Enable/Disable/Clear the Interrupt 1699
44.8 Sample Program 1700
44.9 Notes 1703
44.10 Term Definition for A/D Converter 1704

45. Flash Memory 1709

45.1 Overview 1710
45.2 Features 1711
45.3 Configuration 1712
45.3.1 Block Diagram 1713
45.3.2 CRC Control Register: CRCCR 1714
45.3.3 Sector Number and Flash Macro Number Correspondence Chart 1716
45.4 Registers 1718
45.4.1 Flash Control Register: FCTLR 1719
45.4.2 Flash Status Register: FSTR (Flash Status Register) 1722
45.4.3 Flash Interface Control Register: FLIFCTLR(Flash I/F Control Register) 1724
45.4.4 Flash I/F Feature Extension Register 1: FLIFFER1 1726
45.4.5 Flash I/F Feature Extension Register 2: FLIFFER2 1727
45.5 Operation 1728
45.5.1 Access Mode Setting 1729
45.5.1.1 Configuring CPU-ROM Mode 1730

45.5.1.2 Configuring CPU Programming Mode....1730
45.5.2 Programming Flash Memory by CPU....1731
45.5.3 Automatic Algorithm....1732
45.5.3.1 Command Sequence....1733
45.5.3.2 Automatic Algorithm Execution State....1736
45.5.4 Reset Command....1740
45.5.5 Write Command....1741
45.5.6 Chip Erase Command....1743
45.5.7 Sector Erase Command....1744
45.5.8 Sector Erase Suspend Command....1747
45.5.9 Security Function....1748
45.5.9.1 Flash Security On/Off Determination When Reset Released....1748
45.5.9.2 Flash Security Setting Method....1748
45.5.9.3 Unlocking Flash Security....1748
45.5.9.4 Flash Access Restrictions When Security is ON....1749
45.5.10 Notes on Using Flash Memory....1750

46. WorkFlash Memory....1751

46.1 Overview....1752
46.2 Features....1753
46.3 Configuration....1754
46.3.1 Block Diagram....1755
46.3.2 Sector Configuration Diagram....1756
46.4 Registers....1757
46.4.1 WorkFlash Control Register: DFCTLR....1758
46.4.2 WorkFlash Status Register: DFSTR....1759
46.4.3 Flash Interface Control Register: FLIFCTLR....1761
46.5 Operation....1763
46.5.1 Access Mode Setting....1764
46.5.1.1 Configuring CPU-ROM Mode Below....1765
46.5.1.2 Configuring CPU Programming Mode....1765
46.5.2 Writing Flash Memory by CPU....1766
46.5.3 Automatic Algorithm....1767
46.5.3.1 Command Sequence....1768
46.5.3.2 Automatic Algorithm Execution State....1771
46.5.4 Reset Command....1775
46.5.5 Write Command....1776
46.5.6 Chip Erase Command....1778
46.5.7 Sector Erase Command....1779
46.5.8 Sector Erase Suspend Command....1781
46.5.9 Security Function....1782
46.5.9.1 Flash Security On/Off Determination When Reset Released....1783
46.5.9.2 Flash Security Setting Method....1783
46.5.9.3 Unlocking Flash Security....1783
46.5.9.4 Flash Access Restrictions When Security is ON....1784

46.5.10 Notes on Using Flash Memory 1785

47. HS-SPI 1787

48. On chip Debugger (OCD) 1789

48.1 Overview 1790
48.2 Features 1791
48.3 Configuration 1793
48.3.1 DEBUG I/F Clock 1795
48.3.1.1 DEBUG I/F Main Clock (M_MCLK) 1795
48.3.1.2 DEBUG I/F PLL Clock (M_PCLK) 1795
48.4 Registers 1796
48.4.1 DBG Register 1797
48.4.2 User IO Register 1798
48.5 Operation 1799
48.5.1 OCDU Operating Mode 1800
48.5.1.1 Operating Mode 1801
48.5.1.2 Operating Mode Status Transition 1802
48.5.2 Overview of DEBUG I/F 1803
48.5.2.1 Chip Reset Sequence 1804
48.5.2.2 Security Function 1806
48.5.3 Specification Restrictions at Connection to OCD Tool of this Series 1807
48.5.3.1 Clock Setting 1807
48.5.3.2 Standby Mode 1807
48.5.3.3 Clock Reset State Transitions 1808
48.5.3.4 Summary of Specification Restrictions 1810
48.5.4 OCD-DSU ID Code and Mount Type Information on this Series 1814

A. Appendix 1815

A.1 Memory Map 1816
A.2 I/O Map 1818
A.3 Lists of Interrupt Vector 1857
A.4 Pins Statuses in State of CPU 1861

Revision History 1863

Document Revision History 1863

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 CPU
1.7 Pin Assignment
1.8 Package Dimensions
1.9 Explanation of Pin Functions
1.10 Pins of Each Function
1.11 I/O Circuit Types

1.1 Overview

This section explains features and basic specification of this series.

CY91F570 Series is Cypress 32-bit microcontroller for dashboard control for automotives. The FR81S CPU that is compatible with the FR family is used.

1.2 Features

This section explains the features of this series.

1.2.1 FR81S CPU Core

1.2.2 Peripheral Functions

1.2.1 FR81S CPU Core

This section explains FR81S CPU core.

- 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-bit ×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 instruction 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 explains the peripheral functions of this series.

- Clock generation (equipped with SSCG function)
  - ☐ Main oscillation (4 MHz)
  - ☐ Sub oscillation (32 kHz) or no sub oscillation
  - ☐ PLL multiplication rate : 1 to 20 times
- Built-in Program flash memory capacity
  - ☐ CY91F575: 512 + 64KB
  - ☐ CY91F577: 1024 + 64KB
- Built-in Data flash memory (WorkFlash) capacity 64KB
- Built-in RAM capacity
  - ☐ Main RAM

    CY91F575: 40KB

    CY91F577: 64KB
  - ☐ Backup RAM 8KB
- General-purpose ports: 111 (none sub oscillation), 109 (with sub oscillation)
  - ☐ Included I^2C pseudo open drain ports: 4
  - ☐ P057: Input only
- External bus interface
  - ☐ 22-bit address, 16-bit data
  - ☐ 23 pins of 9-bit address, 8-bit data, ASX, CS0X, CS1X, RDX, WR0X, and WR1X can select 5 V/3.3 V by the VCCE power supply
- DMA Controller
  - ☐ Up to 16 channels can be started simultaneously.
  - ☐ 2 transfer factors (Internal peripheral request and software)
- A/D converter (successive approximation type)
  - ☐ 8/10-bit resolution: 40 channels
  - ☐ Conversion time: 3µs
- D/A converter (R-2R type)
  - ☐ 8-bit resolution: 2 channels
- External interrupt input: 16 channels
  - ☐ Level ("H" / "L"), or edge detection (rising or falling) enabled

- LIN-UART
  - ☐ 6 channels, ch.2 to ch.7
  - ☐ Selectable from UART, synchronous mode or LIN-UART mode
  - ☐ LIN protocol Revision 2.1 supported (LIN-UART).
  - ☐ SPI (Serial Peripheral Interface) supported (synchronous mode)
  - ☐ Full-duplex double buffering system
  - ☐ LIN synch break detection (linked to the input capture)
  - ☐ Built-in dedicated baud rate generator
  - ☐ DMA transfer support
- Multi-function serial communication (built-in transmission/reception FIFO memory): 4 channels

  < UART (Asynchronous serial interface) >
  - ☐ Full-duplex double buffering system, 16-byte transmission FIFO memory, 16-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 detect functions provided
  - ☐ DMA transfer support

  <CSIO (Synchronous serial interface) >
  - ☐ Full-duplex double buffering system, 16-byte transmission FIFO, memory, 16-byte reception FIFO memory
  - ☐ SPI supported; master and slave systems supported; 5 to 9-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

  <LIN-UART (Asynchronous Serial Interface for LIN) >
  - ☐ Full-duplex double buffering system, 16-byte transmission FIFO memory, 16-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

< I^2C >

- ☐ Full-duplex double buffering system, 16-byte transmission FIFO memory, 16-byte reception FIFO memory
- ☐ Standard mode (Max. 100 kbps) / high-speed mode (Max. 400 kbps) supported
- ☐ DMA transfer supported (for transmission only)
- ☐ I^2C supporting I/O (for ch.0 and ch.1 only)

- CAN Controller (CAN): 3 channels
  - ☐ Transfer speed: Up to 1 Mbps
  - ☐ 64-transmission/reception message buffering: 1 channel,
    32-transmission/reception message buffering: 2 channels
- PPG: 16-bit × 24 channels
- Reload timer: 16-bit × 7 channels (3 channels are for regular timer interrupt generation.)
- Free-run timer: 32-bit × 6 channels (Can select each channel for input capture, output compare.)
- Input capture: 32-bit × 12 channels (linked to the free-run timer)
- Output compare: 32-bit × 12 channels (linked to the free-run timer)
- Sound generator: 5 channels
  - ☐ Frequency and amplitude sequencers provided
- Stepping motor controller: 6 channels
  - ☐ 8/10-bit PWM
  - ☐ High current output supported (4 lines × 6 channels)
  - ☐ Can refer back electromotive force using pin-shared A/D converter
- LCD controller
  - ☐ Common output: 4 , Segment output: 32
  - ☐ Duty drive (SEG0 to SEG31) and static drive (ST0 to ST8) can be switched.
  - ☐ Each of COM0 to COM3, SEG0 to SEG31, V0, V1, V2, and V3 pins for duty drive can be switched to the general-purpose port. (The SEG23 to SEG31 pins can be switched to static driving.)
  - ☐ V0, V1, V2 and V3 pin can be used as the general-purpose port. But V3 pin cannot be used as an output pin.
  - ☐ Each of ST0 to ST8 pins for static drive can be switched to the general-purpose port, or it can be switched to the segment output of duty drive.
  - ☐ The amplitude of the SEG0 to SEG22 output is determined by the VCC5 power supply pin or by the V3 pin even if VCCE pin is supplied to 3.3 V.
- Up/Down counter: 2 channels
  - ☐ 8/16-bit up/down counter
- Real-time clock (RTC) (for day, hours, minutes, seconds)
  - ☐ Main oscillation / sub oscillation frequency can be selected for the operation clock
- Calibration: A hardware watchdog of the CR oscillation drive and real-time clock (RTC) of the subclock drive
  - ☐ The CR oscillation frequency can be trimmed
  - ☐ The main clock to sub clock ratio can be corrected by setting the real-time clock prescaler

- Clock Supervisor
  - Monitoring abnormality (damage of crystal etc.) of suboscillation (32KHz) (dual clock products) and main oscillation (4 MHz)
  - When abnormality is detected, it switches to the CR clock.
- Base timer: 2 channels
  - 16-bit timer
  - The timer mode is selected from PWM/PPG/PWC/reload.
  - In the cascaded mode, a pair of 16-bit timers can be used as one 32-bit timer.
- CRC generation
- HS-SPI

  Note: In this series, the HS-SPI function is prohibited.
  - E^2PROM and the flash device of the Single/Dual/Quad-SPI protocol can be connected.
  - The power supply of 5 V/3.3 V supplied to the VCCE power supply pin is used.
  - Maximum 16 MHz (Maximum 8 MHz at the slave.)
- Watchdog timer
  - Hardware watchdog
  - Software watchdog
- NMI
- Interrupt controller
- Interrupt request batch read
  - Multiple interrupts from peripherals can be read by a series of registers.
- I/O relocation
  - Peripheral function pins can be reassigned.
- Low-power consumption mode
  - Sleep / Stop / Watch / Sub RUN mode
  - Stop (power shutdown) / Watch (power shutdown) mode
- Power on reset / internal low-voltage detection reset
- Low-voltage detection reset
- Device Package: LQFP-144
- CMOS 90 nm Technology
- Power supplies
  - 5 V Power supply
  - The internal 1.2 V is generated from 5 V with the voltage step-down regulator.
  - I/O of P010 to P017, P020 to P027, and P030 to P036 uses the power supply of 5 V/3.3 V supplied to the VCCE power supply pin.

1.3 Product Line-up

This section explains product line-up of this series.

Table 1-1. Product Line-up

Product No.	CY91F575B/S	CY91F575BH/S
System Clock	On-chip PLL Clock multiplier method	
Minimum instruction execution time	Around 12.5 ns (80 MHz)	
Sub clock	Yes (Non-S series) No (S series)	
FLASH Capacity (Program)	512 + 64KB	
FLASH Capacity (Data)	64KB	
RAM	40KB + 8KB	
BI-ROM	4KB	
GDC	None	
External BUS I/F	Address: 22-bit Data :16-bit (Part of the External BUS I/F pins can select the power supply 5 V or 3.3 V)	
DMA Transfer	16 channels	
16-bit Base Timer	2 channels	
Free-run Timer	6 channels	
Input capture	12 channels	
Output Compare	12 channels	
16-bit Reload Timer	7 channels	
PPG	24 channels	
Up/down Counter	2 channels	
Clock Supervisor	Yes	
D/A	2 channels	
External Interrupt	16 channels	
A/D	40 channels	

Product No.	CY91F575B/S	CY91F575BH/S
LIN-UART	6 channels	
Multi-Function serial	4 channels[1]	
HS-SPI	Yes Up to 16 MHz **Note:** In this series, the HS-SPI function is prohibited.	
LCD Controller	32seg × 4com (Static drive 8seg × 1com)	
CAN	64msg × 1 channel / 32msg × 2 channels	
Stepping Motor Controller	6 channels	
Sound Generator	5 channels	
Software Watchdog	Yes	
Hardware Watchdog	Yes	
Clock supervisor	Initial value “ON”	Initial value “OFF”
CRC generation	Yes	
Low-voltage detection reset (External low-voltage detection)	Yes	
Internal low-voltage detection reset	Yes	
Package	LQFP-144	
Others	Flash Products	
On-Chip Debug	Built-in OCD	

[1]: I^2C-UART function is supported only at ch.0 and ch.1.

Product No.	CY91F577B/S	CY91F577BH/S
System Clock	On-chip PLL Clock multiplier method	
Minimum instruction execution time	Around 12.5 ns (80 MHz)	
Sub clock	Yes(Non-S series) No(S series)	
FLASH Capacity (Program)	1024 + 64KB	
FLASH Capacity (Data)	64KB	
RAM	64KB + 8KB	
BI-ROM	4KB	
GDC	None	
External BUS I/F	Address: 22-bit Data: 16-bit (Part of the External BUS I/F pins can select the power supply 5 V or 3.3 V)	
DMA Transfer	16 channels	
16-bit Base Timer	2 channels	
Free-run Timer	6 channels	
Input capture	12 channels	
Output Compare	12 channels	
16-bit Reload Timer	7 channels	
PPG	24 channels	
Up/down Counter	2 channels	
Clock Supervisor	Yes	
D/A	2 channels	
External Interrupt	16 channels	
A/D	40 channels	
LIN-UART	6 channels	
Multi-Function serial	4 channels[1]	

Product No.	CY91F577B/S	CY91F577BH/S
HS-SPI	Yes Up to 16 MHz **Note:** In this series, the HS-SPI function is prohibited.	
LCD Controller	32seg × 4com (Static drive 8seg × 1com)	
CAN	64msg × 1 channel / 32msg × 2 channels	
Stepping Motor Controller	6 channels	
Sound Generator	5 channels	
Software Watchdog	Yes	
Hardware Watchdog	Yes	
Clock supervisor	Initial value “ON”	Initial value “OFF”
CRC generation	Yes	
Low-voltage detection reset (External low-voltage detection)	Yes	
Internal low-voltage detection reset	Yes	
Package	LQFP-144	
Others	Flash Products	
On-Chip Debug	Built-in OCD	

[1]: I^2C-UART function is supported only at ch.0 and ch.1.

- CY91F578/579 Line-up

  CY91F578 and 579 is available. See the latest Product Line-up on the Datasheet of CY91570 Series.

1.4 Function Overview

This section explains function outline of this series.

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: 4MHz Sub oscillation: 32kHz or None PLL multiplication rate: Up to 20 times of multiplication Built-in CR oscillator as the count clock of hardware watchdog timer
I/O ports	Each bit can be programmed for I/O or peripheral signals Input thresholds, driving capacity, and pull-up/pull-down can be set.
External bus Interface	22-bit address, 8/16-bit Data output
Internal bus interface	On-chip bus: 32-bit, maximum operating frequency: 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 memory interface	Wild register function provided. For small sector (64KB)
DMA controller	Up to 16 channels can be started simultaneously. The transfer cause (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 demand. ■ 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. As for the functions of PWC and reload timer, 2 channels of cascade mode can be used as 32-bit timer.
Free-run timer	32-bit up counter

Function	Features
Input capture	32-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 32-bit free-run timer is latched and an interrupt request is generated. LIN synch break/synch field linkage: Input capture ch.0 → LIN-UART ch.2 Input capture ch.1 → LIN-UART ch.3 Input capture ch.2 → LIN-UART ch.4 Input capture ch.3 → LIN-UART ch.5 Input capture ch.4 → LIN-UART ch.6 Input capture ch.5 → LIN-UART ch.7 Input capture ch.6 → Multi-function serial ch.0 Input capture ch.7 → Multi-function serial ch.1 Input capture ch.8 → Multi-function serial ch.8 Input capture ch.9 → Multi-function serial ch.9 Cooperation with the free-run timer is as follows. Input capture ch.0 to ch.5 → Free-run timer ch.0 or ch.1 Input capture ch.6 to ch.11 → Free-run timer ch.2 or ch.3
Output compare	An interrupt signal is output during collating with the 32 bit free-run timer. Cooperation with the free-run timer is as follows. Output compare ch.0 to ch.3 → Free-run timer ch.0 or ch.1 Output compare ch.4, ch.5 → Free-run timer ch.4 or ch.5 Output compare ch.6 to ch.9 → Free-run timer ch.2 or ch.3 Output compare ch.10, ch.11 → Free-run timer ch.4 or ch.5
Reload timer	16-bit reload timer operation (The toggle output or one-shot output can be selected) Event count function can be selected. Ch.4-ch.6 has no external pin.
PPG	The cycle and duty used for the one-shot square wave output and PWM output can be changed by the software. Operation clock frequency: Can be selected from following 4 types : PCLK × 1, $1/2^2$, $1/2^4$, $1/2^6$
Delay interrupt	An interrupt for task switching is generated. The CPU interrupt request can be generated or canceled by the software.
External interrupt	16 channel, 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	With built-in A/D converter 1ch of resolution in 10-bit or 8-bit Able to sample the analog value from 40ch input port Conversion time: 3 µs External trigger activation Can be started by the internal timer (16-bit reload timer)

Function	Features
D/A Converter	Resolution in 8-bit × 2 channels
LIN-UART (6 channels)	Full-duplex system Asynchronous/synchronous transfer (with start/stop bits) Built-in dedicated baud rate generator LIN protocol, slave node supported, and LIN synch break/synch field detectable SPI (Serial Peripheral Interface) supported LIN protocol Ver2.1 supported
Multi-function serial (4 channels)	Any of UART/CSIO/LIN-UART/I^{2}C-UART functions can be selected and used. Transmission FIFO memory 16-byte, and reception FIFO memory 16-byte provided Reception interrupt cause (3 types) ■ Receive error detection (parity, overrun, and frame error) ■ Detects FIFO's reception of data up to an amount of its threshold. ■ Detects the idling period which is 8 × baud rate clock or more, when amount of the data received is less than FIFO's threshold. Transmission interrupt cause (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 **Note:** Only two channels of four (ch.0 and ch.1) support I^{2}C-UART function.
HS-SPI	The flash device of the Single/Dual/Quad-SPI protocol can be connected 3.3 V supported. Maximum 16 MHz at master Maximum 8 MHz at the slave Note: In this series, the HS-SPI function is prohibited
LCD controller/driver (1 channel)	<Duty drive> Segment driver and common driver that can directly drive LCD panel (liquid crystal display) : 32SEG × 4COM If the external bus interface is used, the SEG0 to SEG31, COM0 to COM3, V0, V1, V2, and V3 pins of LCD can be used by EPFR setting. Each of COM0 to COM3, SEG0 to SEG31, V0, V1, V2, and V3 pins can be switched to the general-purpose port by the software. (The SEG23 to SEG31 pins can be switched to static driving.) <Static drive> Segment driver and common driver that can directly drive LCD panel (liquid crystal display) : 8SEG/1COM If the external bus interface is used, the ST0 to ST8 pins of LCD can be used by EPFR setting. Each of ST0 to ST8 pins can be switched to the general-purpose port or to the segment output of duty drive by software.
Interrupt controller	Detects an interrupt request. Sets an interrupt level.

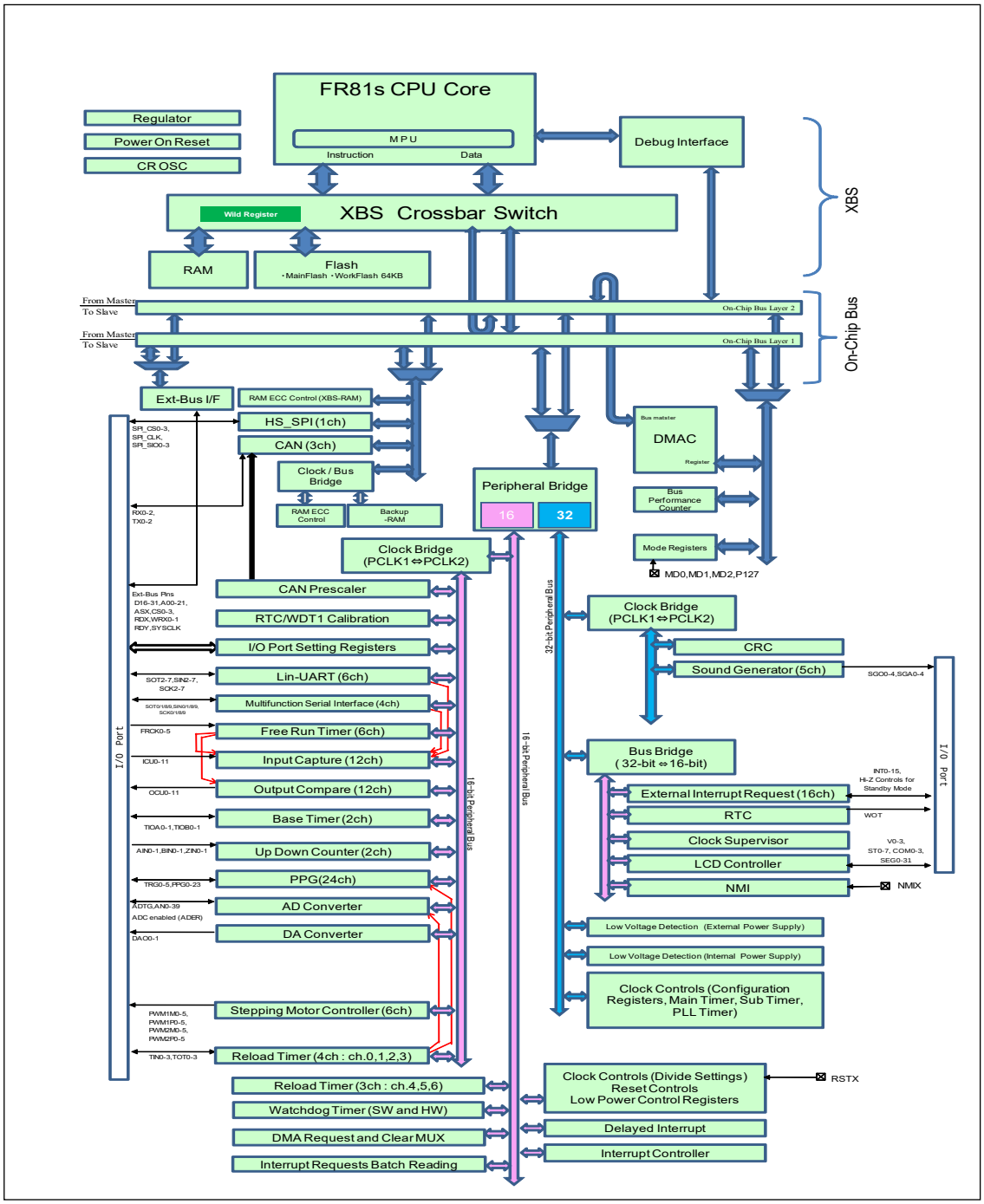
Function	Features
Interrupt request batch read	A generation of multiple interrupts from peripherals can be read by a series of registers.
CAN interface	CAN Specifications Version 2.0, Part A and Part B satisfied 64 message buffers × 1 channel, 32 message buffers × 2 channels Support plural messages Flexible composition of acceptance filter : ■ Entire bit compare ■ Entire bit Mask ■ 2 portion bit Mask Up to 1Mbps supported. CAN prescaler is mounted for the CAN operation clock
Stepping motor controller	High current output × 4 lines The PWM cycle can be set to 15.625 kHz (when the peripheral clock operates in 16 MHz). The back electromotive force from the motor can be detected by the pin sharing ADC.
U/D counter (2 channels)	8/16-bit up/counter × 2 channels
Sound generator	In addition to the frequency data and amplitude data setting, the followings can be set: ■ Decrement or increment data, and execution cycle ■ Tone output pulse count (output interval)
Real-time clock	Day/hours/minutes/seconds register Main or sub oscillation frequency can be selected for the operation clock. Sub clock correction function ■ The sub clock cycle error is monitored by the main clock. ■ The detected error is reflected on the second counter set value. An interrupt can be generated in unit of 0.5 second, seconds, minutes, hours, or day.
Calibration	The real-time clock of the sub clock drive is corrected by comparison with the main clock. The CR oscillation frequency can be corrected by the comparison with the main 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 × (2^9 to 2^{24}) cycles).
Hardware watchdog	CR-based CPU operation detection counter Used against program overrun Period: 260 ms to 416 ms (usually 328 ms, depending on the accuracy of the CR oscillation) The calibration is possible with "RTC/WDT1 correction" circuit. **Note:** As shown above, a period of the CR oscillation clock varies widely due to the production process.

Function	Features
CRC generation	When data is sequentially written in the input registers, the CRC code is displayed in the result register.
Low-voltage detection (external low-voltage detection)	Reset generation at low-voltage detection
Internal low-voltage detection	Monitors 1.2 V power supply and generate the reset.
Low-power consumption mode	Sleep mode Stop mode Watch mode Stop mode (power shutdown) Watch mode (power shutdown) Sub RUN Mode
I/O relocation	Relocation target peripheral function and number of branches ■ PPG 24 channels (4 divergences of ch.1, 3 divergences of ch.0 and ch.2 to ch.10, 2 divergences of ch.11 to ch.15, no divergence in ch.16 to ch.23) ■ Input capture 12 channels × 3 divergence ■ LIN-UART 6 channels × 2 divergence ■ Reload timer 4 channels × 3 divergence ■ Free-run timer 2 channels ×no divergence ,4 channels × 2 divergence ■ Output compare 12 channels × 2 divergence ■ Multi-function serial 4 channels (2 divergences of ch.0, ch.1,and ch.9, 3 divergences of ch.8), But only I^2C-UART cannot be relocated. ■ External interrupt 16 channels × 2 divergence ■ Up/Down counter 2 channels (3 divergences of ch.0, 2 divergences of ch.1) ■ Sound generator 5 channels (no divergences in ch.0 to ch.3, 2 divergences of ch.2) **Note:** SCK/ SOT/ SIN ports of LIN-UART and Multi-function serial must be assigned to same group (SCKn/ SOTn/ SINn, or SCKn_1/SOTn_1/ SINn_1). Settings as show below are prohibited. e.g. SCKn_0/ SOTn_1/ SINn_0 For detail of the setting, refer to "Extended Port Function Register" in "Chapter: I/O Ports".
NMI request	Non-maskable interrupt signal that is entered from NMIX pin.
Debug interface	Built-in OCD

1.5 Block Diagram

This section explains the block diagram of this series.

Figure 1-1. Block Diagram



Note:

In this series, the HS-SPI function is prohibited

1.6 CPU

This section explains the general-purpose registers and dedicated registers of CPU.

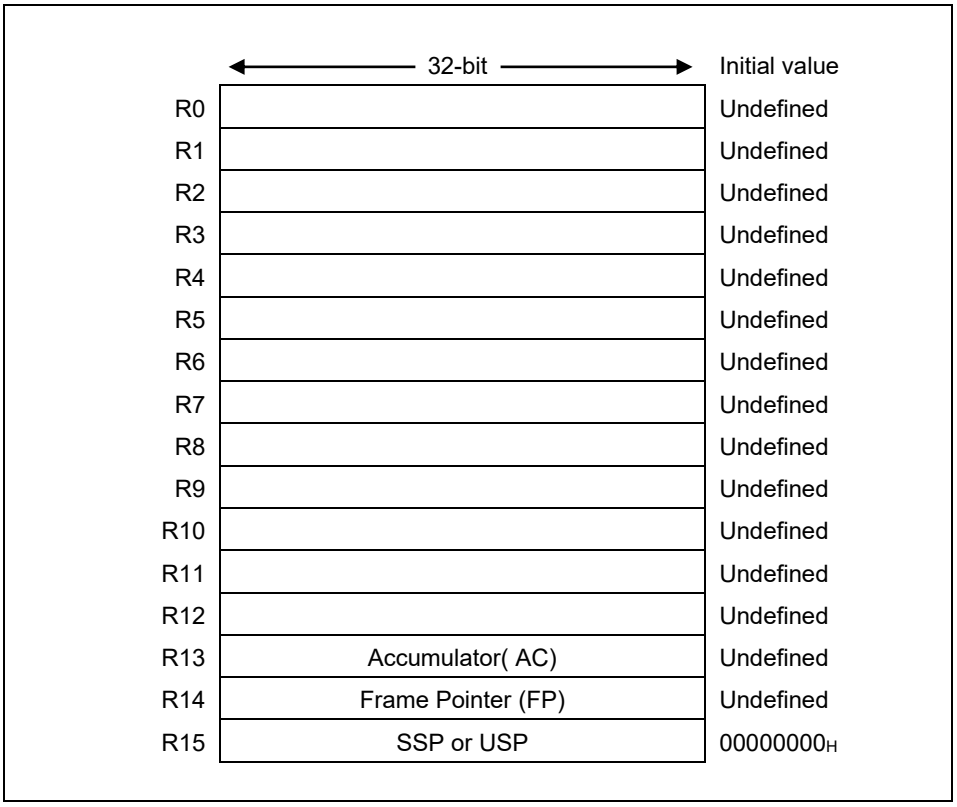
1.6.1 General-purpose Registers

1.6.2 Dedicated Registers

1.6.1 General-purpose Registers

Registers R0 to R15 are general-purpose registers. They are used as the accumulators for various operations and as pointers for memory access.

Figure 1-2. General-purpose Registers



Among these 16 registers, the following registers are assumed to be for special applications. Therefore, some instruction functions have been enhanced.

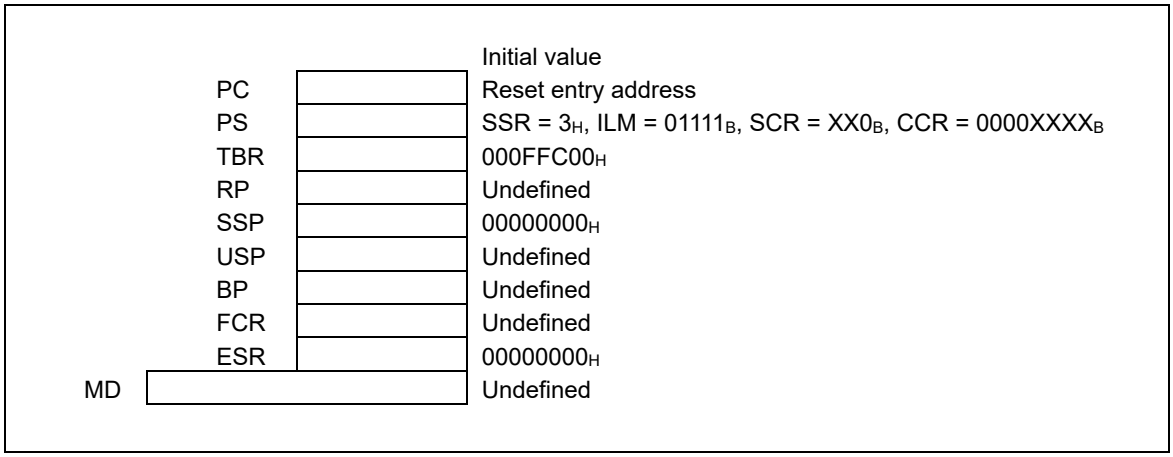
- R13: AC (Accumulator)
- R14: FP (Frame Pointer)
- R15: SP (Stack Pointer)

The initial value during reset is undefined for registers R0 to R14. Register R15 has 00000000_H (SSP value).

1.6.2 Dedicated Registers

The register dedicated for 32-bit length exclusive for various usages are ten, and there is one dedicated register for 64-bit length of the multiplication and division calculation.

Figure 1-3. List of Dedicated Registers



Dedicated register is used for a specific purpose.

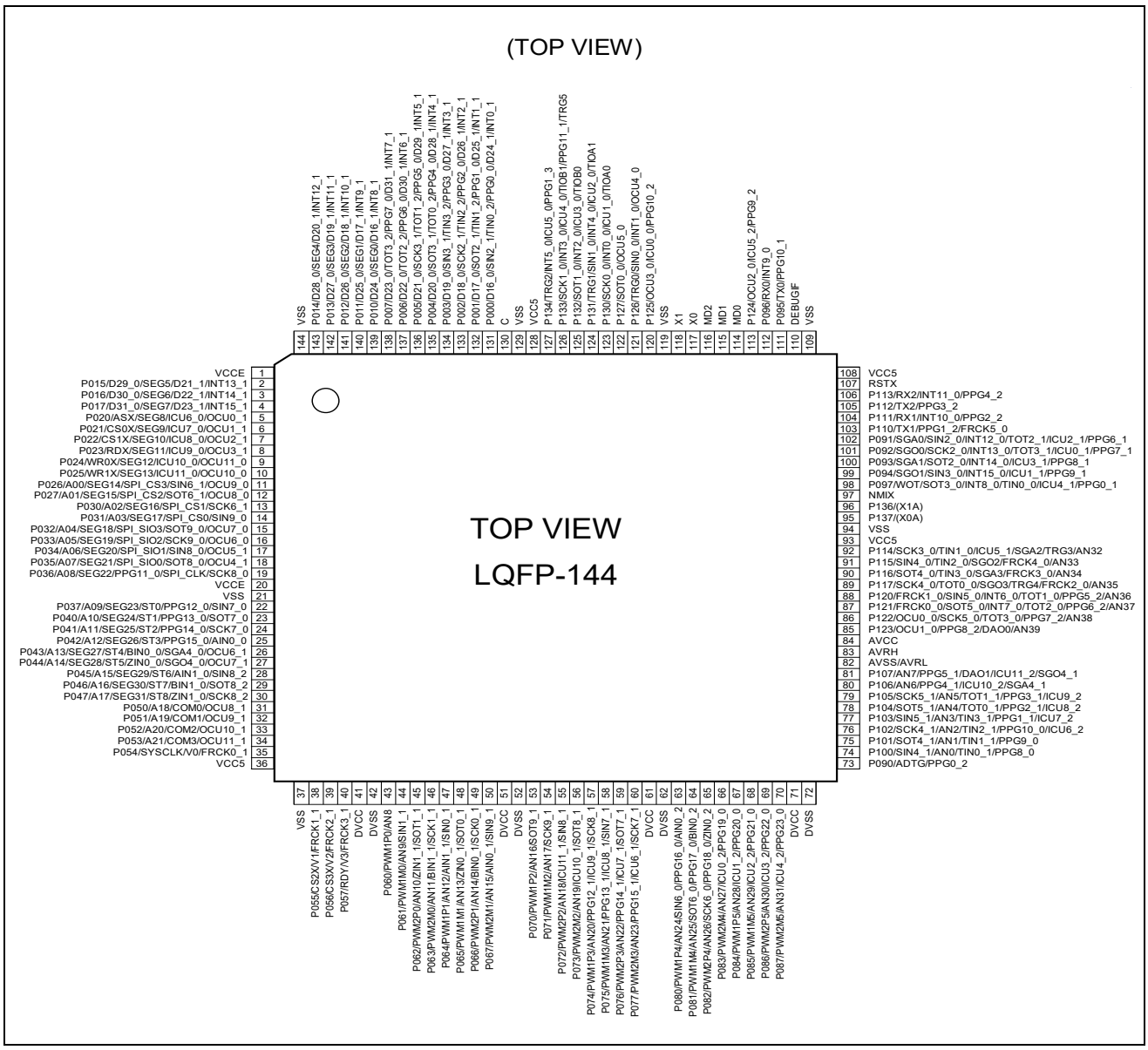
In the FR family, the following dedicated registers are prepared.

- Program counter (PC)
- Program status (PS)
- Table base register (TBR)
- Return pointer (RP)
- System stack pointer (SSP)
- User stack pointer (USP)
- Base pointer (BP)
- FPU control register (FCR)
- Exception status register (ESR)
- Multiplication and division calculation register (MD)

1.7 Pin Assignment

This section explains the pin assignment of this series.

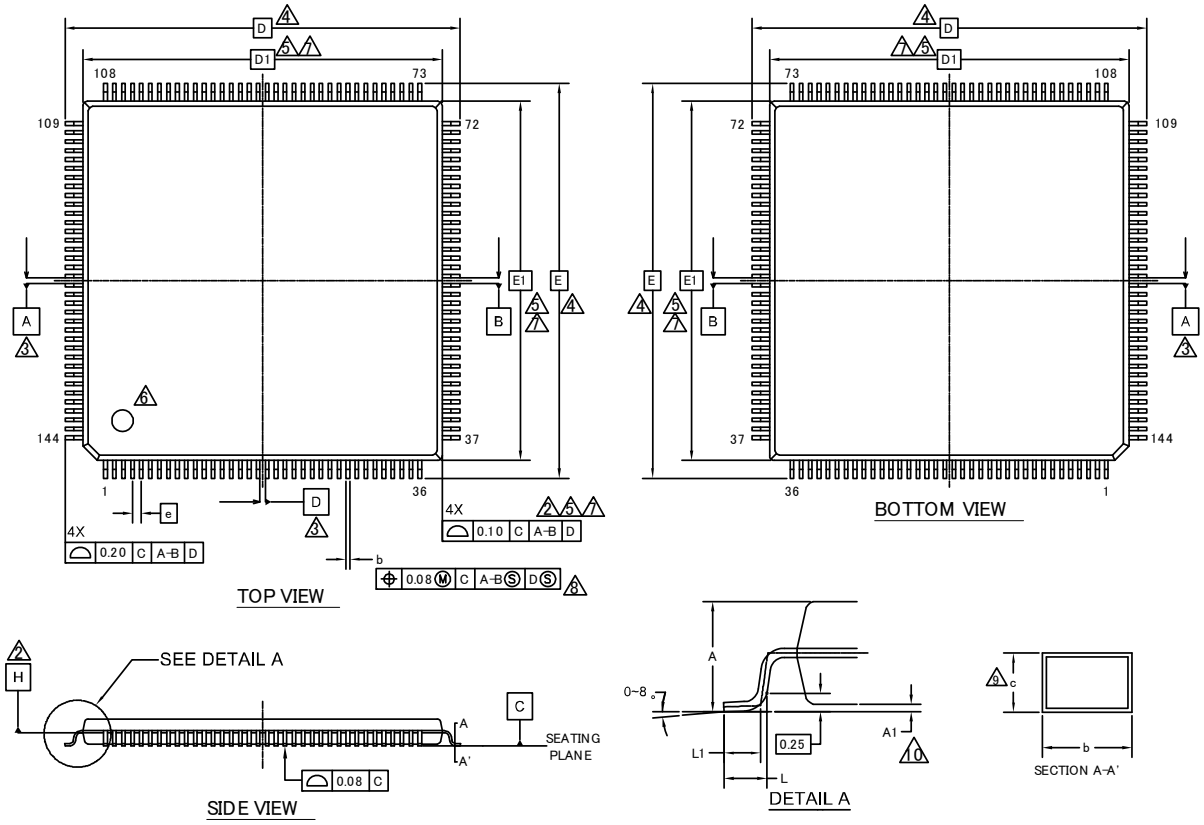
Figure 1-4. Pin Assignment



1.8 Package Dimensions

This section explains device package of this series.

Figure 1-5. LQFP-144 (LQS144) External Dimensions



SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	—	—	1.70
A1	0.05	—	0.15
b	0.17	0.22	0.27
c	0.09	—	0.20
D	22.00 BSC		
D1	20.00 BSC		
e	0.50 BSC		
E	22.00 BSC		
E1	20.00 BSC		
L	0.45	0.60	0.75
L1	0.30	0.50	0.70

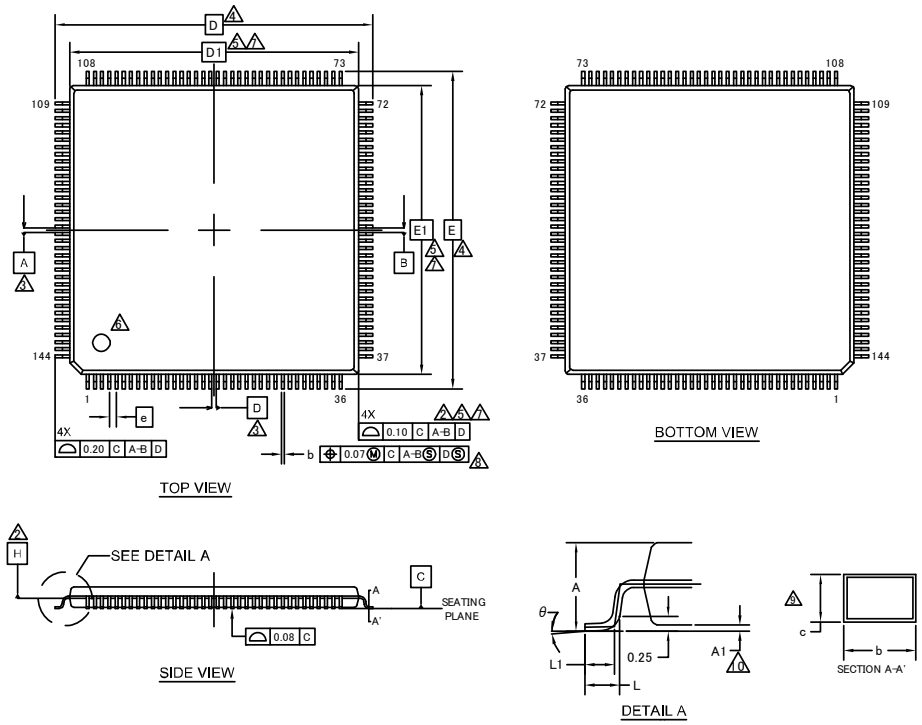
NOTES

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

002-13015 *A

PACKAGE OUTLINE, 144 LEAD LQFP
20.0X20.0X1.7 MM LQS144 REV*A

Figure 1-6. LQFP-144 (LQN144) External Dimensions



SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	——	——	1.70
A1	0.05	——	0.15
b	0.145	0.18	0.215
c	0.115	——	0.195
D	18.00 BSC		
D1	16.00 BSC		
e	0.40 BSC		
E	18.00 BSC		
E1	16.00 BSC		
L	0.45	0.60	0.75
L1	0.30	0.50	0.70
θ	0°	——	8°

NOTES

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

002-14045 **

PACKAGE OUTLINE, 144 LEAD LQFP
16.0X16.0X1.7 MM LQN144 REV**

1.9 Explanation of Pin Functions

The pin function list of this series is shown below.

Table 1-3. List of Pin Functions

Pin Number	Pin Name	I/O Circuit Type	Function Description
2	P015	H	General-Purpose I/O Port
	D29_0		External Bus Data I/O pin
	SEG5		LCDC Segment(Duty)Output pin
	D21_1		External Bus Data I/O pin
	INT13_1		External Interrupt Request Input pin ch.13 relocation 1
3	P016	H	General-Purpose I/O Port
	D30_0		External Bus Data I/O pin
	SEG6		LCDC Segment(Duty)Output pin
	D22_1		External Bus Data I/O pin
	INT14_1		External Interrupt Request Input pin ch.14 relocation 1
4	P017	H	General-Purpose I/O Port
	D31_0		External Bus Data I/O pin
	SEG7		LCDC Segment(Duty)Output pin
	D23_1		External Bus Data I/O pin
	INT15_1		External Interrupt Request Input pin ch.15 relocation 1
5	P020	H	General-Purpose I/O Port
	ASX		External Bus Address-Strobe Output pin
	SEG8		LCDC Segment(Duty)Output pin
	ICU6_0		Input Capture Input pin ch.6 relocation 0
	OCU0_1		Output Compare Output pin ch.0 relocation 1
6	P021	H	General-Purpose I/O Port
	CS0X		External Bus Chip-Select 0 Output pin
	SEG9		LCDC Segment(Duty)Output pin
	ICU7_0		Input Capture Input pin ch.7 relocation 0
	OCU1_1		Output Compare Output pin ch.1 relocation 1
7	P022	H	General-Purpose I/O Port
	CS1X		External Bus Chip-Select 1 Output pin
	SEG10		LCDC Segment(Duty)Output pin
	ICU8_0		Input Capture Input pin ch.8 relocation 0
	OCU2_1		Output Compare Output pin ch.2 relocation 1
8	P023	H	General-Purpose I/O Port
	RDX		External Bus Read-Strobe Output pin
	SEG11		LCDC Segment(Duty)Output pin
	ICU9_0		Input Capture Input pin ch.9 relocation 0
	OCU3_1		Output Compare Output pin ch.3 relocation 1

Pin Number	Pin Name	I/O Circuit Type	Function Description
9	P024	H	General-Purpose I/O Port
	WR0X		External Bus Write-Strobe 0 Output pin
	SEG12		LCDC Segment(Duty)Output pin
	ICU10_0		Input Capture Input pin ch.10 relocation 0
	OCU11_0		Output Compare Output pin ch.11 relocation 0
10	P025	H	General-Purpose I/O Port
	WR1X		External Bus Write-Strobe 1 Output pin
	SEG13		LCDC Segment(Duty)Output pin
	ICU11_0		Input Capture Input pin ch.11 relocation 0
	OCU10_0		Output Compare Output pin ch.10 relocation 0
11	P026	H	General-Purpose I/O Port
	A00		External Bus Address Output pin
	SEG14		LCDC Segment(Duty)Output pin
	SPI_CS3		HS_SPI SSEL3 Output pin
	SIN6_1		LIN_UART Serial Input pin ch.6 relocation 1
	OCU9_0		Output Compare Output pin ch.9 relocation 0
12	P027	H	General-Purpose I/O Port
	A01		External Bus Address Output pin
	SEG15		LCDC Segment(Duty)Output pin
	SPI_CS2		HS_SPI SSEL2 Output pin
	SOT6_1		LIN_UART Serial Output pin ch.6 relocation 1
	OCU8_0		Output Compare Output pin ch.8 relocation 0
13	P030	H	General-Purpose I/O Port
	A02		External Bus Address Output pin
	SEG16		LCDC Segment(Duty)Output pin
	SPI_CS1		HS_SPI SSEL1 Output pin
	SCK6_1		LIN_UART Serial Clock I/O pin ch.6 relocation 1
14	P031	H	General-Purpose I/O Port
	A03		External Bus Address Output pin
	SEG17		LCDC Segment(Duty)Output pin
	SPI_CS0		HS_SPI SSEL0 I/O pin
	SIN9_0		Multi-function Serial Input pin ch.9 relocation 0
15	P032	H	General-Purpose I/O Port
	A04		External Bus Address Output pin
	SEG18		LCDC Segment(Duty)Output pin
	SPI_SIO3		HS_SPI SDATA3 I/O pin
	SOT9_0		Multi-function Serial Output pin ch.9 relocation 0
	OCU7_0		Output Compare Output pin ch.7 relocation 0

Pin Number	Pin Name	I/O Circuit Type	Function Description
16	P033	H	General-Purpose I/O Port
	A05		External Bus Address Output pin
	SEG19		LCDC Segment(Duty)Output pin
	SPI_SIO2		HS_SPI SDATA2 I/O pin
	SCK9_0		Multi-function Serial Clock I/O pin ch.9 relocation 0
	OCU6_0		Output Compare Output pin ch.6 relocation 0
17	P034	H	General-Purpose I/O Port
	A06		External Bus Address Output pin
	SEG20		LCDC Segment(Duty)Output pin
	SPI_SIO1		HS_SPI SDATA1 I/O pin
	SIN8_0		Multi-function Serial Input pin ch.8 relocation 0
	OCU5_1		Output Compare Output pin ch.5 relocation 1
18	P035	H	General-Purpose I/O Port
	A07		External Bus Address Output pin
	SEG21		LCDC Segment(Duty)Output pin
	SPI_SIO0		HS_SPI SDATA0 I/O pin
	SOT8_0		Multi-function Serial Output pin ch.8 relocation 0
	OCU4_1		Output Compare Output pin ch.4 relocation 1
19	P036	H	General-Purpose I/O Port
	A08		External Bus Address Output pin
	SEG22		LCDC Segment(Duty)Output pin
	PPG11_0		PPG Output pin ch.11 relocation 0
	SPI_CLK		HS_SPI SCLK I/O pin
	SCK8_0		Multi-function Serial Clock I/O pin ch.8 relocation 0
22	P037	I	General-Purpose I/O Port
	A09		External Bus Address Output pin
	SEG23		LCDC Segment(Duty)Output pin
	ST0		LCDC Segment(Static)Output pin
	PPG12_0		PPG Output pin ch.12 relocation 0
	SIN7_0		LIN_UART Serial Input pin ch.7 relocation 0
23	P040	I	General-Purpose I/O Port
	A10		External Bus Address Output pin
	SEG24		LCDC Segment(Duty)Output pin
	ST1		LCDC Segment(Static)Output pin
	PPG13_0		PPG Output pin ch.13 relocation 0
	SOT7_0		LIN_UART Serial Output pin ch.7 relocation 0

Pin Number	Pin Name	I/O Circuit Type	Function Description
24	P041	I	General-Purpose I/O Port
	A11		External Bus Address Output pin
	SEG25		LCDC Segment(Duty)Output pin
	ST2		LCDC Segment(Static)Output pin
	PPG14_0		PPG Output pin ch.14 relocation 0
	SCK7_0		LIN_UART Serial Clock I/O pin ch.7 relocation 0
25	P042	I	General-Purpose I/O Port
	A12		External Bus Address Output pin
	SEG26		LCDC Segment(Duty)Output pin
	ST3		LCDC Segment(Static)Output pin
	PPG15_0		PPG Output pin ch.15 relocation 0
	AIN0_0		Up/down Counter AIN Input pin ch.0 relocation 0
26	P043	I	General-Purpose I/O Port
	A13		External Bus Address Output pin
	SEG27		LCDC Segment(Duty)Output pin
	ST4		LCDC Segment(Static)Output pin
	BIN0_0		Up/down Counter BIN Input pin ch.0 relocation 0
	SGA4_0		Sound Generator SGA Output pin ch.4 relocation 0
	OCU6_1		Output Compare Output pin ch.6 relocation 1
27	P044	I	General-Purpose I/O Port
	A14		External Bus Address Output pin
	SEG28		LCDC Segment(Duty)Output pin
	ST5		LCDC Segment(Static)Output pin
	ZIN0_0		Up/down Counter ZIN Input pin ch.0 relocation 0
	SGO4_0		Sound Generator SGO Output pin ch.4 relocation 0
	OCU7_1		Output Compare Output pin ch.7 relocation 1
28	P045	I	General-Purpose I/O Port
	A15		External Bus Address Output pin
	SEG29		LCDC Segment(Duty)Output pin
	ST6		LCDC Segment(Static)Output pin
	AIN1_0		Up/down Counter AIN Input pin ch.1 relocation 0
	SIN8_2		Multi-function Serial Input pin ch.8 relocation 2
29	P046	I	General-Purpose I/O Port
	A16		External Bus Address Output pin
	SEG30		LCDC Segment(Duty)Output pin
	ST7		LCDC Segment(Static)Output pin
	BIN1_0		Up/down Counter BIN Input pin ch.1 relocation 0
	SOT8_2		Multi-function Serial Output pin ch.8 relocation 2

Pin Number	Pin Name	I/O Circuit Type	Function Description
30	P047	I	General-Purpose I/O Port
	A17		External Bus Address Output pin
	SEG31		LCDC Segment(Duty)Output pin
	ST8		LCDC Segment(Static)Output pin
	ZIN1_0		Up/down Counter ZIN Input pin ch.1 relocation 0
	SCK8_2		Multi-function Serial Clock I/O pin ch.8 relocation 2
31	P050	I	General-Purpose I/O Port
	A18		External Bus Address Output pin
	COM0		LCDC Segment(Duty)Common Output pin
	OCU8_1		Output Compare Output pin ch.8 relocation 1
32	P051	I	General-Purpose I/O Port
	A19		External Bus Address Output pin
	COM1		LCDC Segment(Duty)Common Output pin
	OCU9_1		Output Compare Output pin ch.9 relocation 1
33	P052	I	General-Purpose I/O Port
	A20		External Bus Address Output pin
	COM2		LCDC Segment(Duty)Common Output pin
	OCU10_1		Output Compare Output pin ch.10 relocation 1
34	P053	I	General-Purpose I/O Port
	A21		External Bus Address Output pin
	COM3		LCDC Segment(Duty)Common Output pin
	OCU11_1		Output Compare Output pin ch.11 relocation 1
35	P054	I2	General-Purpose I/O Port
	SYSCLK		External Bus Clock Output pin
	V0		LCDC Reference Voltage V0 Input pin
	FRCK0_1		Free-Run Timer Clock Input pin ch.0 relocation 1
38	P055	I2	General-Purpose I/O Port
	CS2X		External Bus Chip-Select 2 Output pin
	V1		LCDC Reference Voltage V1 Input pin
	FRCK1_1		Free-Run Timer Clock Input pin ch.1 relocation 1
39	P056	I2	General-Purpose I/O Port
	CS3X		External Bus Chip-Select 3 Output pin
	V2		LCDC Reference Voltage V2 Input pin
	FRCK2_1		Free-Run Timer Clock Input pin ch.2 relocation 1
40	P057	I3	General-Purpose I/O Port (Input only. No output.)
	RDY		External Bus RDY Input pin
	V3		LCDC Reference Voltage V3 Input pin
	FRCK3_1		Free-Run Timer Clock Input pin ch.3 relocation 1
43	P060	K	General-Purpose I/O Port
	PWM1P0		SMC Output pin ch.0
	AN8		ADC Analog Input pin ch.8

Pin Number	Pin Name	I/O Circuit Type	Function Description
44	P061	K	General-Purpose I/O Port
	PWM1M0		SMC Output pin ch.0
	AN9		ADC Analog Input pin ch.9
	SIN1_1		Multi-function Serial Input pin ch.1 relocation 1
45	P062	K	General-Purpose I/O Port
	PWM2P0		SMC Output pin ch.0
	AN10		ADC Analog Input pin ch.10
	ZIN1_1		Up/down Counter ZIN Input pin ch.1 relocation 1
	SOT1_1		Multi-function Serial Output pin ch.1 relocation 1
46	P063	K	General-Purpose I/O Port
	PWM2M0		SMC Output pin ch.0
	AN11		ADC Analog Input pin ch.11
	BIN1_1		Up/down Counter BIN Input pin ch.1 relocation 1
	SCK1_1		Multi-function Serial Clock I/O pin ch.1 relocation 1
47	P064	K	General-Purpose I/O Port
	PWM1P1		SMC Output pin ch.1
	AN12		ADC Analog Input pin ch.12
	AIN1_1		Up/down Counter AIN Input pin ch.1 relocation 1
	SIN0_1		Multi-function Serial Input pin ch.0 relocation 1
48	P065	K	General-Purpose I/O Port
	PWM1M1		SMC Output pin ch.1
	AN13		ADC Analog Input pin ch.13
	ZIN0_1		Up/down Counter ZIN Input pin ch.0 relocation 1
	SOT0_1		Multi-function Serial Output pin ch.0 relocation 1
49	P066	K	General-Purpose I/O Port
	PWM2P1		SMC Output pin ch.1
	AN14		ADC Analog Input pin ch.14
	BIN0_1		Up/down Counter BIN Input pin ch.0 relocation 1
	SCK0_1		Multi-function Serial Clock I/O pin ch.0 relocation 1
50	P067	K	General-Purpose I/O Port
	PWM2M1		SMC Output pin ch.1
	AN15		ADC Analog Input pin ch.15
	AIN0_1		Up/down Counter AIN Input pin ch.0 relocation 1
	SIN9_1		Multi-function Serial Input pin ch.9 relocation 1
53	P070	K	General-Purpose I/O Port
	PWM1P2		SMC Output pin ch.2
	AN16		ADC Analog Input pin ch.16
	SOT9_1		Multi-function Serial Output pin ch.9 relocation 1

Pin Number	Pin Name	I/O Circuit Type	Function Description
54	P071	K	General-Purpose I/O Port
	PWM1M2		SMC Output pin ch.2
	AN17		ADC Analog Input pin ch.17
	SCK9_1		Multi-function Serial Clock I/O pin ch.9 relocation 1
55	P072	K	General-Purpose I/O Port
	PWM2P2		SMC Output pin ch.2
	AN18		ADC Analog Input pin ch.18
	ICU11_1		Input Capture Input pin ch.11 relocation 1
	SIN8_1		Multi-function Serial Input pin ch.8 relocation 1
56	P073	K	General-Purpose I/O Port
	PWM2M2		SMC Output pin ch.2
	AN19		ADC Analog Input pin ch.19
	ICU10_1		Input Capture Input pin ch.10 relocation 1
	SOT8_1		Multi-function Serial Output pin ch.8 relocation 1
57	P074	K	General-Purpose I/O Port
	PWM1P3		SMC Output pin ch.3
	AN20		ADC Analog Input pin ch.20
	PPG12_1		PPG Output pin ch.12 relocation 1
	ICU9_1		Input Capture Input pin ch.9 relocation 1
	SCK8_1		Multi-function Serial Clock I/O pin ch.8 relocation 1
58	P075	K	General-Purpose I/O Port
	PWM1M3		SMC Output pin ch.3
	AN21		ADC Analog Input pin ch.21
	PPG13_1		PPG Output pin ch.13 relocation 1
	ICU8_1		Input Capture Input pin ch.8 relocation 1
	SIN7_1		LIN_UART Serial Input pin ch.7 relocation 1
59	P076	K	General-Purpose I/O Port
	PWM2P3		SMC Output pin ch.3
	AN22		ADC Analog Input pin ch.22
	PPG14_1		PPG Output pin ch.14 relocation 1
	ICU7_1		Input Capture Input pin ch.7 relocation 1
	SOT7_1		LIN_UART Serial Output pin ch.7 relocation 1
60	P077	K	General-Purpose I/O Port
	PWM2M3		SMC Output pin ch.3
	AN23		ADC Analog Input pin ch.23
	PPG15_1		PPG Output pin ch.15 relocation 1
	ICU6_1		Input Capture Input pin ch.6 relocation 1
	SCK7_1		LIN_UART Serial Clock I/O pin ch.7 relocation 1

Pin Number	Pin Name	I/O Circuit Type	Function Description
63	P080	K	General-Purpose I/O Port
	PWM1P4		SMC Output pin ch.4
	AN24		ADC Analog Input pin ch.24
	SIN6_0		LIN_UART Serial Input pin ch.6 relocation 0
	PPG16_0		PPG Output pin ch.16 relocation 0
	AIN0_2		Up/down Counter AIN Input pin ch.0 relocation 2
64	P081	K	General-Purpose I/O Port
	PWM1M4		SMC Output pin ch.4
	AN25		ADC Analog Input pin ch.25
	SOT6_0		LIN_UART Serial Output pin ch.6 relocation 0
	PPG17_0		PPG Output pin ch.17 relocation 0
	BIN0_2		Up/down Counter BIN Input pin ch.0 relocation 2
65	P082	K	General-Purpose I/O Port
	PWM2P4		SMC Output pin ch.4
	AN26		ADC Analog Input pin ch.26
	SCK6_0		LIN_UART Serial Clock I/O pin ch.6 relocation 0
	PPG18_0		PPG Output pin ch.18 relocation 0
	ZIN0_2		Up/down Counter ZIN Input pin ch.0 relocation 2
66	P083	K	General-Purpose I/O Port
	PWM2M4		SMC Output pin ch.4
	AN27		ADC Analog Input pin ch.27
	ICU0_2		Input Capture Input pin ch.0 relocation 2
	PPG19_0		PPG Output pin ch.19 relocation 0
67	P084	K	General-Purpose I/O Port
	PWM1P5		SMC Output pin ch.5
	AN28		ADC Analog Input pin ch.28
	ICU1_2		Input Capture Input pin ch.1 relocation 2
	PPG20_0		PPG Output pin ch.20 relocation 0
68	P085	K	General-Purpose I/O Port
	PWM1M5		SMC Output pin ch.5
	AN29		ADC Analog Input pin ch.29
	ICU2_2		Input Capture Input pin ch.2 relocation 2
	PPG21_0		PPG Output pin ch.21 relocation 0
69	P086	K	General-Purpose I/O Port
	PWM2P5		SMC Output pin ch.5
	AN30		ADC Analog Input pin ch.30
	ICU3_2		Input Capture Input pin ch.3 relocation 2
	PPG22_0		PPG Output pin ch.22 relocation 0

Pin Number	Pin Name	I/O Circuit Type	Function Description
70	P087	K	General-Purpose I/O Port
	PWM2M5		SMC Output pin ch.5
	AN31		ADC Analog Input pin ch.31
	ICU4_2		Input Capture Input pin ch.4 relocation 2
	PPG23_0		PPG Output pin ch.23 relocation 0
73	P090	M	General-Purpose I/O Port
	ADTG		ADC External Trigger Input pin
	PPG0_2		PPG Output pin ch.0 relocation 2
74	P100	J	General-Purpose I/O Port
	SIN4_1		LIN_UART Serial Input pin ch.4 relocation 1
	AN0		ADC Analog Input pin ch.0
	TIN0_1		Reload Timer Event Input pin ch.0 relocation 1
	PPG8_0		PPG Output pin ch.8 relocation 0
75	P101	J	General-Purpose I/O Port
	SOT4_1		LIN_UART Serial Output pin ch.4 relocation 1
	AN1		ADC Analog Input pin ch.1
	TIN1_1		Reload Timer Event Input pin ch.1 relocation 1
	PPG9_0		PPG Output pin ch.9 relocation 0
76	P102	J	General-Purpose I/O Port
	SCK4_1		LIN_UART Serial Clock I/O pin ch.4 relocation 1
	AN2		ADC Analog Input pin ch.2
	TIN2_1		Reload Timer Event Input pin ch.2 relocation 1
	PPG10_0		PPG Output pin ch.10 relocation 0
	ICU6_2		Input Capture Input pin ch.6 relocation 2
77	P103	J	General-Purpose I/O Port
	SIN5_1		LIN_UART Serial Input pin ch.5 relocation 1
	AN3		ADC Analog Input pin ch.3
	TIN3_1		Reload Timer Event Input pin ch.3 relocation 1
	PPG1_1		PPG Output pin ch.1 relocation 1
	ICU7_2		Input Capture Input pin ch.7 relocation 2
78	P104	J	General-Purpose I/O Port
	SOT5_1		LIN_UART Serial Output pin ch.5 relocation 1
	AN4		ADC Analog Input pin ch.4
	TOT0_1		Reload Timer Output pin ch.0 relocation 1
	PPG2_1		PPG Output pin ch.2 relocation 1
	ICU8_2		Input Capture Input pin ch.8 relocation 2

Pin Number	Pin Name	I/O Circuit Type	Function Description
79	P105	J	General-Purpose I/O Port
	SCK5_1		LIN_UART Serial Clock I/O pin ch.5 relocation 1
	AN5		ADC Analog Input pin ch.5
	TOT1_1		Reload Timer Output pin ch.1 relocation 1
	PPG3_1		PPG Output pin ch.3 relocation 1
	ICU9_2		Input Capture Input pin ch.9 relocation 2
80	P106	J	General-Purpose I/O Port
	AN6		ADC Analog Input pin ch.6
	PPG4_1		PPG Output pin ch.4 relocation 1
	ICU10_2		Input Capture Input pin ch.10 relocation 2
	SGA4_1		Sound Generator SGA Output pin ch.4 relocation 1
81	P107	L	General-Purpose I/O Port
	AN7		ADC Analog Input pin ch.7
	PPG5_1		PPG Output pin ch.5 relocation 1
	DAO1		DAC Output pin ch.1
	ICU11_2		Input Capture Input pin ch.11 relocation 2
	SGO4_1		Sound Generator SGO Output pin ch.4 relocation 1
85	P123	L	General-Purpose I/O Port
	OCU1_0		Output Compare Output pin ch.1 relocation 0
	PPG8_2		PPG Output pin ch.8 relocation 2
	DAO0		DAC Output pin ch.0
	AN39		ADC Analog Input pin ch.39
86	P122	J	General-Purpose I/O Port
	OCU0_0		Output Compare Output pin ch.0 relocation 0
	SCK5_0		LIN_UART Serial Clock I/O pin ch.5 relocation 0
	TOT3_0		Reload Timer Output pin ch.3 relocation 0
	PPG7_2		PPG Output pin ch.7 relocation 2
	AN38		ADC Analog Input pin ch.38
87	P121	J	General-Purpose I/O Port
	FRCK0_0		Free-Run Timer Clock Input pin ch.0 relocation 0
	SOT5_0		LIN_UART Serial Output pin ch.5 relocation 0
	INT7_0		External Interrupt Request Input pin ch.7 relocation 0
	TOT2_0		Reload Timer Output pin ch.2 relocation 0
	PPG6_2		PPG Output pin ch.6 relocation 2
	AN37		ADC Analog Input pin ch.37

Pin Number	Pin Name	I/O Circuit Type	Function Description
88	P120	J	General-Purpose I/O Port
	FRCK1_0		Free-Run Timer Clock Input pin ch.1 relocation 0
	SIN5_0		LIN_UART Serial Input pin ch.5 relocation 0
	INT6_0		External Interrupt Request Input pin ch.6 relocation 0
	TOT1_0		Reload Timer Output pin ch.1 relocation 0
	PPG5_2		PPG Output pin ch.5 relocation 2
	AN36		ADC Analog Input pin ch.36
89	P117	J	General-Purpose I/O Port
	SCK4_0		LIN_UART Serial Clock I/O pin ch.4 relocation 0
	TOT0_0		Reload Timer Output pin ch.0 relocation 0
	SGO3		Sound Generator SGO Output pin ch.3
	TRG4		PPG Trigger Input pin 4 (ch.16-ch.19)
	FRCK2_0		Free-Run Timer Clock Input pin ch.2 relocation 0
	AN35		ADC Analog Input pin ch.35
90	P116	J	General-Purpose I/O Port
	SOT4_0		LIN_UART Serial Output pin ch.4 relocation 0
	TIN3_0		Reload Timer Event Input pin ch.3 relocation 0
	SGA3		Sound Generator SGA Output pin ch.3
	FRCK3_0		Free-Run Timer Clock Input pin ch.3 relocation 0
	AN34		ADC Analog Input pin ch.34
91	P115	J	General-Purpose I/O Port
	SIN4_0		LIN_UART Serial Input pin ch.4 relocation 0
	TIN2_0		Reload Timer Event Input pin ch.2 relocation 0
	SGO2		Sound Generator SGO Output pin ch.2
	FRCK4_0		Free-Run Timer Clock Input pin ch.4 relocation 0
	AN33		ADC Analog Input pin ch.33
92	P114	J	General-Purpose I/O Port
	SCK3_0		LIN_UART Serial Clock I/O pin ch.3 relocation 0
	TIN1_0		Reload Timer Event Input pin ch.1 relocation 0
	ICU5_1		Input Capture Input pin ch.5 relocation 1
	SGA2		Sound Generator SGA Output pin ch.2
	TRG3		PPG Trigger Input pin 3 (ch.12-ch.15)
	AN32		ADC Analog Input pin ch.32
95	P137	M (Y)	General-Purpose I/O Port
	(X0A)		Sub Clock OSC Input pin (only dual clock product)
96	P136	M (Y)	General-Purpose I/O Port
	(X1A)		Sub Clock OSC Output pin (only dual clock product)
97	NMIX	R	NMI Pin

Pin Number	Pin Name	I/O Circuit Type	Function Description
98	P097	M	General-Purpose I/O Port
	WOT		RTC Overflow Output pin
	SOT3_0		LIN_UART Serial Output pin ch.3 relocation 0
	INT8_0		External Interrupt Request Input pin ch.8 relocation 0
	TIN0_0		Reload Timer Event Input pin ch.0 relocation 0
	ICU4_1		Input Capture Input pin ch.4 relocation 1
	PPG0_1		PPG Output pin ch.0 relocation 1
99	P094	M	General-Purpose I/O Port
	SGO1		Sound Generator SGO Output pin ch.1
	SIN3_0		LIN_UART Serial Input pin ch.3 relocation 0
	INT15_0		External Interrupt Request Input pin ch.15 relocation 0
	ICU1_1		Input Capture Input pin ch.1 relocation 1
	PPG9_1		PPG Output pin ch.9 relocation 1
100	P093	M	General-Purpose I/O Port
	SGA1		Sound Generator SGA Output pin ch.1
	SOT2_0		LIN_UART Serial Output pin ch.2 relocation 0
	INT14_0		External Interrupt Request Input pin ch.14 relocation 0
	ICU3_1		Input Capture Input pin ch.3 relocation 1
	PPG8_1		PPG Output pin ch.8 relocation 1
101	P092	M	General-Purpose I/O Port
	SGO0		Sound Generator SGO Output pin ch.0
	SCK2_0		LIN_UART Serial Clock I/O pin ch.2 relocation 0
	INT13_0		External Interrupt Request Input pin ch.13 relocation 0
	TOT3_1		Reload Timer Output pin ch.3 relocation 1
	ICU0_1		Input Capture Input pin ch.0 relocation 1
	PPG7_1		PPG Output pin ch.7 relocation 1
102	P091	M	General-Purpose I/O Port
	SGA0		Sound Generator SGA Output pin ch.0
	SIN2_0		LIN_UART Serial Input pin ch.2 relocation 0
	INT12_0		External Interrupt Request Input pin ch.12 relocation 0
	TOT2_1		Reload Timer Output pin ch.2 relocation 1
	ICU2_1		Input Capture Input pin ch.2 relocation 1
	PPG6_1		PPG Output pin ch.6 relocation 1
103	P110	M	General-Purpose I/O Port
	TX1		CAN TX Data Output pin ch.1
	PPG1_2		PPG Output pin ch.1 relocation 2
	FRCK5_0		Free-Run Timer Clock Input pin ch.5 relocation 0

Pin Number	Pin Name	I/O Circuit Type	Function Description
104	P111	M	General-Purpose I/O Port
	RX1		CAN RX Data Input pin ch.1
	INT10_0		External Interrupt Request Input pin ch.10 relocation 0
	PPG2_2		PPG Output pin ch.2 relocation 2
105	P112	M	General-Purpose I/O Port
	TX2		CAN TX Data Output pin ch.2
	PPG3_2		PPG Output pin ch.3 relocation 2
106	P113	M	General-Purpose I/O Port
	RX2		CAN RX Data Input pin ch.2
	INT11_0		External Interrupt Request Input pin ch.11 relocation 0
	PPG4_2		PPG Output pin ch.4 relocation 2
107	RSTX	R	Reset Pin
110	DEBUGIF	B	DEBUG I/F pin
111	P095	M	General-Purpose I/O Port
	TX0		CAN TX Data Output pin ch.0
	PPG10_1		PPG Output pin ch.10 relocation 1
112	P096	M	General-Purpose I/O Port
	RX0		CAN RX Data Input pin ch.0
	INT9_0		External Interrupt Request Input pin ch.9 relocation 0
113	P124	M	General-Purpose I/O Port
	OCU2_0		Output Compare Output pin ch.2 relocation 0
	ICU5_2		Input Capture Input pin ch.5 relocation 2
	PPG9_2		PPG Output pin ch.9 relocation 2
114	MD0	A	Mode Pin
115	MD1	A	Mode Pin
116	MD2	R2	Mode Pin
117	X0	X	Main Clock OSC Input pin
118	X1	X	Main Clock OSC Output pin
120	P125	M	General-Purpose I/O Port
	OCU3_0		Output Compare Output pin ch.3 relocation 0
	ICU0_0		Input Capture Input pin ch.0 relocation 0
	PPG10_2		PPG Output pin ch.10 relocation 2
121	P126	M	General-Purpose I/O Port
	TRG0		PPG Trigger Input pin 0 (ch.0-ch.3)
	SIN0_0		Multi-function Serial Input pin ch.0 relocation 0
	INT1_0		External Interrupt Request Input pin ch.1 relocation 0
	OCU4_0		Output Compare Output pin ch.4 relocation 0
122	P127	N	General-Purpose I/O Port
	SOT0_0		Multi-function Serial Output pin ch.0 relocation 0
	OCU5_0		Output Compare Output pin ch.5 relocation 0

Pin Number	Pin Name	I/O Circuit Type	Function Description
123	P130	N	General-Purpose I/O Port
	SCK0_0		Multi-function Serial Clock I/O pin ch.0 relocation 0
	INT0_0		External Interrupt Request Input pin ch.0 relocation 0
	ICU1_0		Input Capture Input pin ch.1 relocation 0
	TIOA0		Base Timer Output pin ch.0
124	P131	M	General-Purpose I/O Port
	TRG1		PPG Trigger Input pin 1 (ch.4-ch.7)
	SIN1_0		Multi-function Serial Input pin ch.1 relocation 0
	INT4_0		External Interrupt Request Input pin ch.4 relocation 0
	ICU2_0		Input Capture Input pin ch.2 relocation 0
	TIOA1		Base Timer I/O pin ch.1
125	P132	N	General-Purpose I/O Port
	SOT1_0		Multi-function Serial Output pin ch.1 relocation 0
	INT2_0		External Interrupt Request Input pin ch.2 relocation 0
	ICU3_0		Input Capture Input pin ch.3 relocation 0
	TIOB0		Base Timer Input pin ch.0
126	P133	N	General-Purpose I/O Port
	SCK1_0		Multi-function Serial Clock I/O pin ch.1 relocation 0
	INT3_0		External Interrupt Request Input pin ch.3 relocation 0
	ICU4_0		Input Capture Input pin ch.4 relocation 0
	TIOB1		Base Timer Input pin ch.1
	PPG11_1		PPG Output pin ch.11 relocation 1
	TRG5		PPG Trigger Input pin 5 (ch.20-ch.23)
127	P134	M	General-Purpose I/O Port
	TRG2		PPG Trigger Input pin 2 (ch.8-ch.11)
	INT5_0		External Interrupt Request Input pin ch.5 relocation 0
	ICU5_0		Input Capture Input pin ch.5 relocation 0
	PPG1_3		PPG Output pin ch.1 relocation 3
131	P000	M	General-Purpose I/O Port
	D16_0		External Bus Data I/O pin
	SIN2_1		LIN_UART Serial Input pin ch.2 relocation 1
	TIN0_2		Reload Timer Event Input pin ch.0 relocation 2
	PPG0_0		PPG Output pin ch.0 relocation 0
	D24_1		External Bus Data I/O pin
	INT0_1		External Interrupt Request Input pin ch.0 relocation 1

Pin Number	Pin Name	I/O Circuit Type	Function Description
132	P001	M	General-Purpose I/O Port
	D17_0		External Bus Data I/O pin
	SOT2_1		LIN_UART Serial Output pin ch.2 relocation 1
	TIN1_2		Reload Timer Event Input pin ch.1 relocation 2
	PPG1_0		PPG Output pin ch.1 relocation 0
	D25_1		External Bus Data I/O pin
	INT1_1		External Interrupt Request Input pin ch.1 relocation 1
133	P002	M	General-Purpose I/O Port
	D18_0		External Bus Data I/O pin
	SCK2_1		LIN_UART Serial Clock I/O pin ch.2 relocation 1
	TIN2_2		Reload Timer Event Input pin ch.2 relocation 2
	PPG2_0		PPG Output pin ch.2 relocation 0
	D26_1		External Bus Data I/O pin
	INT2_1		External Interrupt Request Input pin ch.2 relocation 1
134	P003	M	General-Purpose I/O Port
	D19_0		External Bus Data I/O pin
	SIN3_1		LIN_UART Serial Input pin ch.3 relocation 1
	TIN3_2		Reload Timer Event Input pin ch.3 relocation 2
	PPG3_0		PPG Output pin ch.3 relocation 0
	D27_1		External Bus Data I/O pin
	INT3_1		External Interrupt Request Input pin ch.3 relocation 1
135	P004	M	General-Purpose I/O Port
	D20_0		External Bus Data I/O pin
	SOT3_1		LIN_UART Serial Output pin ch.3 relocation 1
	TOT0_2		Reload Timer Output pin ch.0 relocation 2
	PPG4_0		PPG Output pin ch.4 relocation 0
	D28_1		External Bus Data I/O pin
	INT4_1		External Interrupt Request Input pin ch.4 relocation 1
136	P005	M	General-Purpose I/O Port
	D21_0		External Bus Data I/O pin
	SCK3_1		LIN_UART Serial Clock I/O pin ch.3 relocation 1
	TOT1_2		Reload Timer Output pin ch.1 relocation 2
	PPG5_0		PPG Output pin ch.5 relocation 0
	D29_1		External Bus Data I/O pin
	INT5_1		External Interrupt Request Input pin ch.5 relocation 1
137	P006	M	General-Purpose I/O Port
	D22_0		External Bus Data I/O pin
	TOT2_2		Reload Timer Output pin ch.2 relocation 2
	PPG6_0		PPG Output pin ch.6 relocation 0
	D30_1		External Bus Data I/O pin
	INT6_1		External Interrupt Request Input pin ch.6 relocation 1

Pin Number	Pin Name	I/O Circuit Type	Function Description
138	P007	M	General-Purpose I/O Port
	D23_0		External Bus Data I/O pin
	TOT3_2		Reload Timer Output pin ch.3 relocation 2
	PPG7_0		PPG Output pin ch.7 relocation 0
	D31_1		External Bus Data I/O pin
	INT7_1		External Interrupt Request Input pin ch.7 relocation 1
139	P010	H	General-Purpose I/O Port
	D24_0		External Bus Data I/O pin
	SEG0		LCDC Segment(Duty)Output pin
	D16_1		External Bus Data I/O pin
	INT8_1		External Interrupt Request Input pin ch.8 relocation 1
140	P011	H	General-Purpose I/O Port
	D25_0		External Bus Data I/O pin
	SEG1		LCDC Segment(Duty)Output pin
	D17_1		External Bus Data I/O pin
	INT9_1		External Interrupt Request Input pin ch.9 relocation 1
141	P012	H	General-Purpose I/O Port
	D26_0		External Bus Data I/O pin
	SEG2		LCDC Segment(Duty)Output pin
	D18_1		External Bus Data I/O pin
	INT10_1		External Interrupt Request Input pin ch.10 relocation 1
142	P013	H	General-Purpose I/O Port
	D27_0		External Bus Data I/O pin
	SEG3		LCDC Segment(Duty)Output pin
	D19_1		External Bus Data I/O pin
	INT11_1		External Interrupt Request Input pin ch.11 relocation 1
143	P014	H	General-Purpose I/O Port
	D28_0		External Bus Data I/O pin
	SEG4		LCDC Segment(Duty)Output pin
	D20_1		External Bus Data I/O pin
	INT12_1		External Interrupt Request Input pin ch.12 relocation 1
1	VCCE	-	+3.3 V/+5.0 V Power Supply
20	VCCE	-	+3.3 V/+5.0 V Power Supply
21	VSS	-	GND
36	VCC5	-	+5.0 V Power Supply
37	VSS	-	GND
41	DVCC	-	Power Supply for SMC high current
42	DVSS	-	GND for SMC high current
51	DVCC	-	Power Supply for SMC high current
52	DVSS	-	GND for SMC high current

Pin Number	Pin Name	I/O Circuit Type	Function Description
61	DVCC	-	Power Supply for SMC high current
62	DVSS	-	GND for SMC high current
71	DVCC	-	Power Supply for SMC high current
72	DVSS	-	GND for SMC high current
82	AVSS/AVRL	-	ADC, DAC GND / Low Reference Voltage
83	AVRH	-	ADC High Reference Voltage
84	AVCC	-	ADC,DAC Analog Power Supply
93	VCC5	-	+5.0 V Power Supply
94	VSS	-	GND
108	VCC5	-	+5.0 V Power Supply
109	VSS	-	GND
119	VSS	-	GND
128	VCC5	-	+5.0 V Power Supply
129	VSS	-	GND
130	C	-	External Capacitance Connection Pin
144	VSS	-	GND

1.10 Pins of Each Function

Pins of each function of the CY91570 series are shown below.

1.10.1 Pins of A/D convertor
1.10.2 Pins of Base Timer (ch.0, ch.1)
1.10.3 Pins of CAN (ch.0 to ch.2)
1.10.4 Pins of DAC (ch.0, ch.1)
1.10.5 Pins of External Bus Interface
1.10.6 Pins of External Interrupt Request Input (ch.0 to ch.15)
1.10.7 Pins of Free-run Timer (ch.0 to ch.5)
1.10.8 Pins of General-purpose I/O Port
1.10.9 Pins of HS_SPI
1.10.10 Pins of Input Capture (ch.0 to ch.11)
1.10.11 Pins of LCDC
1.10.12 Pins of LIN-UART (ch.2 to ch.7)
1.10.13 Pins of Multi-function Serial Interface (ch.0 to ch.1, ch.8 to ch.9)
1.10.14 Pins of Output Compare (ch.0 to ch.11)
1.10.15 Pins of PPG (ch.0 to ch.23)
1.10.16 Pins of Reload Timer (ch.0 to ch.3)
1.10.17 Pins of Real Time Clock
1.10.18 Pins of Stepping Motor Controller (ch.0 to ch.5)
1.10.19 Pins of Sound Generator (ch.0 to ch.4)
1.10.20 Pins of Up/Down Counter (ch.0, ch.1)
1.10.21 Pins of Clock
1.10.22 Pins of Mode
1.10.23 Pins of Reset
1.10.24 Pins of Power Supply

1.10.1 Pins of A/D convertor

Pins of the A/D convertor are shown below.

- ADC external trigger input (pin name) ADTG (pin no.) 73
- ADC Analog input ch.0 (pin name) AN0 (pin no.) 74
- ADC Analog input ch.1 (pin name) AN1 (pin no.) 75
- ADC Analog input ch.2 (pin name) AN2 (pin no.) 76
- ADC Analog input ch.3 (pin name) AN3 (pin no.) 77
- ADC Analog input ch.4 (pin name) AN4 (pin no.) 78
- ADC Analog input ch.5 (pin name) AN5 (pin no.) 79
- ADC Analog input ch.6 (pin name) AN6 (pin no.) 80
- ADC Analog input ch.7 (pin name) AN7 (pin no.) 81
- ADC Analog input ch.8 (pin name) AN8 (pin no.) 43
- ADC Analog input ch.9 (pin name) AN9 (pin no.) 44
- ADC Analog input ch.10 (pin name) AN10 (pin no.) 45
- ADC Analog input ch.11 (pin name) AN11 (pin no.) 46
- ADC Analog input ch.12 (pin name) AN12 (pin no.) 47
- ADC Analog input ch.13 (pin name) AN13 (pin no.) 48
- ADC Analog input ch.14 (pin name) AN14 (pin no.) 49
- ADC Analog input ch.15 (pin name) AN15 (pin no.) 50
- ADC Analog input ch.16 (pin name) AN16 (pin no.) 53
- ADC Analog input ch.17 (pin name) AN17 (pin no.) 54
- ADC Analog input ch.18 (pin name) AN18 (pin no.) 55
- ADC Analog input ch.19 (pin name) AN19 (pin no.) 56
- ADC Analog input ch.20 (pin name) AN20 (pin no.) 57
- ADC Analog input ch.21 (pin name) AN21 (pin no.) 58
- ADC Analog input ch.22 (pin name) AN22 (pin no.) 59
- ADC Analog input ch.23 (pin name) AN23 (pin no.) 60
- ADC Analog input ch.24 (pin name) AN24 (pin no.) 63
- ADC Analog input ch.25 (pin name) AN25 (pin no.) 64
- ADC Analog input ch.26 (pin name) AN26 (pin no.) 65
- ADC Analog input ch.27 (pin name) AN27 (pin no.) 66
- ADC Analog input ch.28 (pin name) AN28 (pin no.) 67
- ADC Analog input ch.29 (pin name) AN29 (pin no.) 68
- ADC Analog input ch.30 (pin name) AN30 (pin no.) 69
- ADC Analog input ch.31 (pin name) AN31 (pin no.) 70
- ADC Analog input ch.32 (pin name) AN32 (pin no.) 92
- ADC Analog input ch.33 (pin name) AN33 (pin no.) 91
- ADC Analog input ch.34 (pin name) AN34 (pin no.) 90
- ADC Analog input ch.35 (pin name) AN35 (pin no.) 89

- ADC Analog input ch.36 (pin name) AN36 (pin no.) 88
- ADC Analog input ch.37 (pin name) AN37 (pin no.) 87
- ADC Analog input ch.38 (pin name) AN38 (pin no.) 86
- ADC Analog input ch.39 (pin name) AN39 (pin no.) 85

1.10.2 Pins of Base Timer (ch.0, ch.1)

Pins of the base timer are shown below.

- Base Timer output ch.0 (pin name) TIOA0 (pin no.) 123
- Base Timer input ch.0 (pin name) TIOB0 (pin no.) 125
- Base Timer input/output ch.1 (pin name) TIOA1 (pin no.) 124
- Base Timer input ch.1 (pin name) TIOB1 (pin no.) 126

1.10.3 Pins of CAN (ch.0 to ch.2)

Pins of CAN are shown below.

- CAN reception data input ch.0 (pin name) RX0 (pin no.) 112
- CAN reception data input ch.1 (pin name) RX1 (pin no.) 104
- CAN reception data input ch.2 (pin name) RX2 (pin no.) 106
- CAN transmission data output ch.0 (pin name) TX0 (pin no.) 111
- CAN transmission data output ch.1 (pin name) TX1 (pin no.) 103
- CAN transmission data output ch.2 (pin name) TX2 (pin no.) 105

1.10.4 Pins of DAC (ch.0, ch.1)

Pins of DAC are shown below.

- DAC output ch.0 (pin name) DAO0 (pin no.) 85
- DAC output ch.1 (pin name) DAO1 (pin no.) 81

1.10.5 Pins of External Bus Interface

Pins of the external bus interface are shown below.

- External Bus Address output (pin name) A00 (pin no.) 11
- External Bus Address output (pin name) A01 (pin no.) 12
- External Bus Address output (pin name) A02 (pin no.) 13
- External Bus Address output (pin name) A03 (pin no.) 14
- External Bus Address output (pin name) A04 (pin no.) 15
- External Bus Address output (pin name) A05 (pin no.) 16
- External Bus Address output (pin name) A06 (pin no.) 17
- External Bus Address output (pin name) A07 (pin no.) 18
- External Bus Address output (pin name) A08 (pin no.) 19
- External Bus Address output (pin name) A09 (pin no.) 22
- External Bus Address output (pin name) A10 (pin no.) 23
- External Bus Address output (pin name) A11 (pin no.) 24
- External Bus Address output (pin name) A12 (pin no.) 25
- External Bus Address output (pin name) A13 (pin no.) 26
- External Bus Address output (pin name) A14 (pin no.) 27
- External Bus Address output (pin name) A15 (pin no.) 28
- External Bus Address output (pin name) A16 (pin no.) 29
- External Bus Address output (pin name) A17 (pin no.) 30
- External Bus Address output (pin name) A18 (pin no.) 31
- External Bus Address output (pin name) A19 (pin no.) 32
- External Bus Address output (pin name) A20 (pin no.) 33
- External Bus Address output (pin name) A21 (pin no.) 34
- External Bus Address Strobe output (pin name) ASX (pin no.) 5
- External Bus Chip Select 0 output (pin name) CS0X (pin no.) 6
- External Bus Chip Select 1 output (pin name) CS1X (pin no.) 7
- External Bus Chip Select 2 output (pin name) CS2X (pin no.) 38
- External Bus Chip Select 3 output (pin name) CS3X (pin no.) 39
- External Bus Clock output (pin name) SYSCLK (pin no.) 35
- External Bus Data input/output (pin name) D16_0 (pin no.) 131
- External Bus Data input/output (pin name) D16_1 (pin no.) 139
- External Bus Data input/output (pin name) D17_0 (pin no.) 132
- External Bus Data input/output (pin name) D17_1 (pin no.) 140
- External Bus Data input/output (pin name) D18_0 (pin no.) 133
- External Bus Data input/output (pin name) D18_1 (pin no.) 141
- External Bus Data input/output (pin name) D19_0 (pin no.) 134
- External Bus Data input/output (pin name) D19_1 (pin no.) 142
- External Bus Data input/output (pin name) D20_0 (pin no.) 135

- External Bus Data input/output (pin name) D20_1 (pin no.) 143
- External Bus Data input/output (pin name) D21_0 (pin no.) 136
- External Bus Data input/output (pin name) D21_1 (pin no.) 2
- External Bus Data input/output (pin name) D22_0 (pin no.) 137
- External Bus Data input/output (pin name) D22_1 (pin no.) 3
- External Bus Data input/output (pin name) D23_0 (pin no.) 138
- External Bus Data input/output (pin name) D23_1 (pin no.) 4
- External Bus Data input/output (pin name) D24_0 (pin no.) 139
- External Bus Data input/output (pin name) D24_1 (pin no.) 131
- External Bus Data input/output (pin name) D25_0 (pin no.) 140
- External Bus Data input/output (pin name) D25_1 (pin no.) 132
- External Bus Data input/output (pin name) D26_0 (pin no.) 141
- External Bus Data input/output (pin name) D26_1 (pin no.) 133
- External Bus Data input/output (pin name) D27_0 (pin no.) 142
- External Bus Data input/output (pin name) D27_1 (pin no.) 134
- External Bus Data input/output (pin name) D28_0 (pin no.) 143
- External Bus Data input/output (pin name) D28_1 (pin no.) 135
- External Bus Data input/output (pin name) D29_0 (pin no.) 2
- External Bus Data input/output (pin name) D29_1 (pin no.) 136
- External Bus Data input/output (pin name) D30_0 (pin no.) 3
- External Bus Data input/output (pin name) D30_1 (pin no.) 137
- External Bus Data input/output (pin name) D31_0 (pin no.) 4
- External Bus Data input/output (pin name) D31_1 (pin no.) 138
- External Bus RDY input (pin name) RDY (pin no.) 40
- External Bus Read Strobe output (pin name) RDX (pin no.) 8
- External Bus Write Strobe 0 output (pin name) WR0X (pin no.) 9
- External Bus Write Strobe 1 output (pin name) WR1X (pin no.) 10

1.10.6 Pins of External Interrupt Request Input (ch.0 to ch.15)

Pins of external interrupt request input are shown below.

- External Interrupt Request Input ch.0 relocation 0 (pin name) INT0_0 (pin no.) 123
- External Interrupt Request Input ch.0 relocation 1 (pin name) INT0_1 (pin no.) 131
- External Interrupt Request Input ch.1 relocation 0 (pin name) INT1_0 (pin no.) 121
- External Interrupt Request Input ch.1 relocation 1 (pin name) INT1_1 (pin no.) 132
- External Interrupt Request Input ch.2 relocation 0 (pin name) INT2_0 (pin no.) 125
- External Interrupt Request Input ch.2 relocation 1 (pin name) INT2_1 (pin no.) 133
- External Interrupt Request Input ch.3 relocation 0 (pin name) INT3_0 (pin no.) 126
- External Interrupt Request Input ch.3 relocation 1 (pin name) INT3_1 (pin no.) 134
- External Interrupt Request Input ch.4 relocation 0 (pin name) INT4_0 (pin no.) 124
- External Interrupt Request Input ch.4 relocation 1 (pin name) INT4_1 (pin no.) 135
- External Interrupt Request Input ch.5 relocation 0 (pin name) INT5_0 (pin no.) 127
- External Interrupt Request Input ch.5 relocation 1 (pin name) INT5_1 (pin no.) 136
- External Interrupt Request Input ch.6 relocation 0 (pin name) INT6_0 (pin no.) 88
- External Interrupt Request Input ch.6 relocation 1 (pin name) INT6_1 (pin no.) 137
- External Interrupt Request Input ch.7 relocation 0 (pin name) INT7_0 (pin no.) 87
- External Interrupt Request Input ch.7 relocation 1 (pin name) INT7_1 (pin no.) 138
- External Interrupt Request Input ch.8 relocation 0 (pin name) INT8_0 (pin no.) 98
- External Interrupt Request Input ch.8 relocation 1 (pin name) INT8_1 (pin no.) 139
- External Interrupt Request Input ch.9 relocation 0 (pin name) INT9_0 (pin no.) 112
- External Interrupt Request Input ch.9 relocation 1 (pin name) INT9_1 (pin no.) 140
- External Interrupt Request Input ch.10 relocation 0 (pin name) INT10_0 (pin no.) 104
- External Interrupt Request Input ch.10 relocation 1 (pin name) INT10_1 (pin no.) 141
- External Interrupt Request Input ch.11 relocation 0 (pin name) INT11_0 (pin no.) 106
- External Interrupt Request Input ch.11 relocation 1 (pin name) INT11_1 (pin no.) 142
- External Interrupt Request Input ch.12 relocation 0 (pin name) INT12_0 (pin no.) 102
- External Interrupt Request Input ch.12 relocation 1 (pin name) INT12_1 (pin no.) 143
- External Interrupt Request Input ch.13 relocation 0 (pin name) INT13_0 (pin no.) 101
- External Interrupt Request Input ch.13 relocation 1 (pin name) INT13_1 (pin no.) 2
- External Interrupt Request Input ch.14 relocation 0 (pin name) INT14_0 (pin no.) 100
- External Interrupt Request Input ch.14 relocation 1 (pin name) INT14_1 (pin no.) 3
- External Interrupt Request Input ch.15 relocation 0 (pin name) INT15_0 (pin no.) 99
- External Interrupt Request Input ch.15 relocation 1 (pin name) INT15_1 (pin no.) 4

1.10.7 Pins of Free-run Timer (ch.0 to ch.5)

Pins of the Free-run timer are shown below.

- Free-run timer clock input ch.0 relocation 0 (pin name) FRCK0_0 (pin no.) 87
- Free-run timer clock input ch.0 relocation 1 (pin name) FRCK0_1 (pin no.) 35
- Free-run timer clock input ch.1 relocation 0 (pin name) FRCK1_0 (pin no.) 88
- Free-run timer clock input ch.1 relocation 1 (pin name) FRCK1_1 (pin no.) 38
- Free-run timer clock input ch.2 relocation 0 (pin name) FRCK2_0 (pin no.) 89
- Free-run timer clock input ch.2 relocation 1 (pin name) FRCK2_1 (pin no.) 39
- Free-run timer clock input ch.3 relocation 0 (pin name) FRCK3_0 (pin no.) 90
- Free-run timer clock input ch.3 relocation 1 (pin name) FRCK3_1 (pin no.) 40
- Free-run timer clock input ch.4 relocation 0 (pin name) FRCK4_0 (pin no.) 91
- Free-run timer clock input ch.5 relocation 0 (pin name) FRCK5_0 (pin no.) 103

1.10.8 Pins of General-purpose I/O Port

Pins of the general-purpose I/O port are shown below.

- General-purpose I/O port (pin name) P000 (pin no.) 131
- General-purpose I/O port (pin name) P001 (pin no.) 132
- General-purpose I/O port (pin name) P002 (pin no.) 133
- General-purpose I/O port (pin name) P003 (pin no.) 134
- General-purpose I/O port (pin name) P004 (pin no.) 135
- General-purpose I/O port (pin name) P005 (pin no.) 136
- General-purpose I/O port (pin name) P006 (pin no.) 137
- General-purpose I/O port (pin name) P007 (pin no.) 138
- General-purpose I/O port (pin name) P010 (pin no.) 139
- General-purpose I/O port (pin name) P011 (pin no.) 140
- General-purpose I/O port (pin name) P012 (pin no.) 141
- General-purpose I/O port (pin name) P013 (pin no.) 142
- General-purpose I/O port (pin name) P014 (pin no.) 143
- General-purpose I/O port (pin name) P015 (pin no.) 2
- General-purpose I/O port (pin name) P016 (pin no.) 3
- General-purpose I/O port (pin name) P017 (pin no.) 4
- General-purpose I/O port (pin name) P020 (pin no.) 5
- General-purpose I/O port (pin name) P021 (pin no.) 6
- General-purpose I/O port (pin name) P022 (pin no.) 7
- General-purpose I/O port (pin name) P023 (pin no.) 8
- General-purpose I/O port (pin name) P024 (pin no.) 9
- General-purpose I/O port (pin name) P025 (pin no.) 10
- General-purpose I/O port (pin name) P026 (pin no.) 11
- General-purpose I/O port (pin name) P027 (pin no.) 12
- General-purpose I/O port (pin name) P030 (pin no.) 13
- General-purpose I/O port (pin name) P031 (pin no.) 14
- General-purpose I/O port (pin name) P032 (pin no.) 15
- General-purpose I/O port (pin name) P033 (pin no.) 16
- General-purpose I/O port (pin name) P034 (pin no.) 17
- General-purpose I/O port (pin name) P035 (pin no.) 18
- General-purpose I/O port (pin name) P036 (pin no.) 19
- General-purpose I/O port (pin name) P037 (pin no.) 22
- General-purpose I/O port (pin name) P040 (pin no.) 23
- General-purpose I/O port (pin name) P041 (pin no.) 24
- General-purpose I/O port (pin name) P042 (pin no.) 25
- General-purpose I/O port (pin name) P043 (pin no.) 26
- General-purpose I/O port (pin name) P044 (pin no.) 27

- General-purpose I/O port (pin name) P045 (pin no.) 28
- General-purpose I/O port (pin name) P046 (pin no.) 29
- General-purpose I/O port (pin name) P047 (pin no.) 30
- General-purpose I/O port (pin name) P050 (pin no.) 31
- General-purpose I/O port (pin name) P051 (pin no.) 32
- General-purpose I/O port (pin name) P052 (pin no.) 33
- General-purpose I/O port (pin name) P053 (pin no.) 34
- General-purpose I/O port (pin name) P054 (pin no.) 35
- General-purpose I/O port (pin name) P055 (pin no.) 38
- General-purpose I/O port (pin name) P056 (pin no.) 39
- General-purpose I/O port (input only no-output) (pin name) P057 (pin no.) 40
- General-purpose I/O port (pin name) P060 (pin no.) 43
- General-purpose I/O port (pin name) P061 (pin no.) 44
- General-purpose I/O port (pin name) P062 (pin no.) 45
- General-purpose I/O port (pin name) P063 (pin no.) 46
- General-purpose I/O port (pin name) P064 (pin no.) 47
- General-purpose I/O port (pin name) P065 (pin no.) 48
- General-purpose I/O port (pin name) P066 (pin no.) 49
- General-purpose I/O port (pin name) P067 (pin no.) 50
- General-purpose I/O port (pin name) P070 (pin no.) 53
- General-purpose I/O port (pin name) P071 (pin no.) 54
- General-purpose I/O port (pin name) P072 (pin no.) 55
- General-purpose I/O port (pin name) P073 (pin no.) 56
- General-purpose I/O port (pin name) P074 (pin no.) 57
- General-purpose I/O port (pin name) P075 (pin no.) 58
- General-purpose I/O port (pin name) P076 (pin no.) 59
- General-purpose I/O port (pin name) P077 (pin no.) 60
- General-purpose I/O port (pin name) P080 (pin no.) 63
- General-purpose I/O port (pin name) P081 (pin no.) 64
- General-purpose I/O port (pin name) P082 (pin no.) 65
- General-purpose I/O port (pin name) P083 (pin no.) 66
- General-purpose I/O port (pin name) P084 (pin no.) 67
- General-purpose I/O port (pin name) P085 (pin no.) 68
- General-purpose I/O port (pin name) P086 (pin no.) 69
- General-purpose I/O port (pin name) P087 (pin no.) 70
- General-purpose I/O port (pin name) P090 (pin no.) 73
- General-purpose I/O port (pin name) P091 (pin no.) 102
- General-purpose I/O port (pin name) P092 (pin no.) 101
- General-purpose I/O port (pin name) P093 (pin no.) 100
- General-purpose I/O port (pin name) P094 (pin no.) 99

- General-purpose I/O port (pin name) P095 (pin no.) 111
- General-purpose I/O port (pin name) P096 (pin no.) 112
- General-purpose I/O port (pin name) P097 (pin no.) 98
- General-purpose I/O port (pin name) P100 (pin no.) 74
- General-purpose I/O port (pin name) P101 (pin no.) 75
- General-purpose I/O port (pin name) P102 (pin no.) 76
- General-purpose I/O port (pin name) P103 (pin no.) 77
- General-purpose I/O port (pin name) P104 (pin no.) 78
- General-purpose I/O port (pin name) P105 (pin no.) 79
- General-purpose I/O port (pin name) P106 (pin no.) 80
- General-purpose I/O port (pin name) P107 (pin no.) 81
- General-purpose I/O port (pin name) P110 (pin no.) 103
- General-purpose I/O port (pin name) P111 (pin no.) 104
- General-purpose I/O port (pin name) P112 (pin no.) 105
- General-purpose I/O port (pin name) P113 (pin no.) 106
- General-purpose I/O port (pin name) P114 (pin no.) 92
- General-purpose I/O port (pin name) P115 (pin no.) 91
- General-purpose I/O port (pin name) P116 (pin no.) 90
- General-purpose I/O port (pin name) P117 (pin no.) 89
- General-purpose I/O port (pin name) P120 (pin no.) 88
- General-purpose I/O port (pin name) P121 (pin no.) 87
- General-purpose I/O port (pin name) P122 (pin no.) 86
- General-purpose I/O port (pin name) P123 (pin no.) 85
- General-purpose I/O port (pin name) P124 (pin no.) 113
- General-purpose I/O port (pin name) P125 (pin no.) 120
- General-purpose I/O port (pin name) P126 (pin no.) 121
- General-purpose I/O port (pin name) P127 (pin no.) 122
- General-purpose I/O port (pin name) P130 (pin no.) 123
- General-purpose I/O port (pin name) P131 (pin no.) 124
- General-purpose I/O port (pin name) P132 (pin no.) 125
- General-purpose I/O port (pin name) P133 (pin no.) 126
- General-purpose I/O port (pin name) P134 (pin no.) 127
- General-purpose I/O port (pin name) P136 (pin no.) 96
- General-purpose I/O port (pin name) P137 (pin no.) 95

1.10.9 Pins of HS_SPI

Pins of HS_SPI are shown below.

- HS_SPI SCLK input/output (pin name) SPI_CLK (pin no.) 19
- HS_SPI SDATA0 input/output (pin name) SPI_SIO0 (pin no.) 18
- HS_SPI SDATA1 input/output (pin name) SPI_SIO1 (pin no.) 17
- HS_SPI SDATA2 input/output (pin name) SPI_SIO2 (pin no.) 16
- HS_SPI SDATA3 input/output (pin name) SPI_SIO3 (pin no.) 15
- HS_SPI SSEL0 input/output (pin name) SPI_CS0 (pin no.) 14
- HS_SPI SSEL1 output (pin name) SPI_CS1 (pin no.) 13
- HS_SPI SSEL2 output (pin name) SPI_CS2 (pin no.) 12
- HS_SPI SSEL3 output (pin name) SPI_CS3 (pin no.) 11

Note: In this series, the HS-SPI function is prohibited.

1.10.10 Pins of Input Capture (ch.0 to ch.11)

Pins of the Input capture are shown below.

- Input capture input ch.0 relocation 0 (pin name) ICU0_0 (pin no.) 120
- Input capture input ch.0 relocation 1 (pin name) ICU0_1 (pin no.) 101
- Input capture input ch.0 relocation 2 (pin name) ICU0_2 (pin no.) 66
- Input capture input ch.1 relocation 0 (pin name) ICU1_0 (pin no.) 123
- Input capture input ch.1 relocation 1 (pin name) ICU1_1 (pin no.) 99
- Input capture input ch.1 relocation 2 (pin name) ICU1_2 (pin no.) 67
- Input capture input ch.2 relocation 0 (pin name) ICU2_0 (pin no.) 124
- Input capture input ch.2 relocation 1 (pin name) ICU2_1 (pin no.) 102
- Input capture input ch.2 relocation 2 (pin name) ICU2_2 (pin no.) 68
- Input capture input ch.3 relocation 0 (pin name) ICU3_0 (pin no.) 125
- Input capture input ch.3 relocation 1 (pin name) ICU3_1 (pin no.) 100
- Input capture input ch.3 relocation 2 (pin name) ICU3_2 (pin no.) 69
- Input capture input ch.4 relocation 0 (pin name) ICU4_0 (pin no.) 126
- Input capture input ch.4 relocation 1 (pin name) ICU4_1 (pin no.) 98
- Input capture input ch.4 relocation 2 (pin name) ICU4_2 (pin no.) 70
- Input capture input ch.5 relocation 0 (pin name) ICU5_0 (pin no.) 127
- Input capture input ch.5 relocation 1 (pin name) ICU5_1 (pin no.) 92
- Input capture input ch.5 relocation 2 (pin name) ICU5_2 (pin no.) 113
- Input capture input ch.6 relocation 0 (pin name) ICU6_0 (pin no.) 5
- Input capture input ch.6 relocation 1 (pin name) ICU6_1 (pin no.) 60
- Input capture input ch.6 relocation 2 (pin name) ICU6_2 (pin no.) 76
- Input capture input ch.7 relocation 0 (pin name) ICU7_0 (pin no.) 6
- Input capture input ch.7 relocation 1 (pin name) ICU7_1 (pin no.) 59
- Input capture input ch.7 relocation 2 (pin name) ICU7_2 (pin no.) 77
- Input capture input ch.8 relocation 0 (pin name) ICU8_0 (pin no.) 7
- Input capture input ch.8 relocation 1 (pin name) ICU8_1 (pin no.) 58
- Input capture input ch.8 relocation 2 (pin name) ICU8_2 (pin no.) 78
- Input capture input ch.9 relocation 0 (pin name) ICU9_0 (pin no.) 8
- Input capture input ch.9 relocation 1 (pin name) ICU9_1 (pin no.) 57
- Input capture input ch.9 relocation 2 (pin name) ICU9_2 (pin no.) 79
- Input capture input ch.10 relocation 0 (pin name) ICU10_0 (pin no.) 9
- Input capture input ch.10 relocation 1 (pin name) ICU10_1 (pin no.) 56
- Input capture input ch.10 relocation 2 (pin name) ICU10_2 (pin no.) 80
- Input capture input ch.11 relocation 0 (pin name) ICU11_0 (pin no.) 10
- Input capture input ch.11 relocation 1 (pin name) ICU11_1 (pin no.) 55
- Input capture input ch.11 relocation 2 (pin name) ICU11_2 (pin no.) 81

1.10.11 Pins of LCDC

Pins of LCDC are shown below.

- LCDC Reference voltage V0 input (pin name) V0 (pin no.) 35
- LCDC Reference voltage V1 input (pin name) V1 (pin no.) 38
- LCDC Reference voltage V2 input (pin name) V2 (pin no.) 39
- LCDC Reference voltage V3 input (pin name) V3 (pin no.) 40
- LCDC Segment (Duty) Common output (pin name) COM0 (pin no.) 31
- LCDC Segment (Duty) Common output (pin name) COM1 (pin no.) 32
- LCDC Segment (Duty) Common output (pin name) COM2 (pin no.) 33
- LCDC Segment (Duty) Common output (pin name) COM3 (pin no.) 34
- LCDC Segment (Duty) output (pin name) SEG0 (pin no.) 139
- LCDC Segment (Duty) output (pin name) SEG1 (pin no.) 140
- LCDC Segment (Duty) output (pin name) SEG2 (pin no.) 141
- LCDC Segment (Duty) output (pin name) SEG3 (pin no.) 142
- LCDC Segment (Duty) output (pin name) SEG4 (pin no.) 143
- LCDC Segment (Duty) output (pin name) SEG5 (pin no.) 2
- LCDC Segment (Duty) output (pin name) SEG6 (pin no.) 3
- LCDC Segment (Duty) output (pin name) SEG7 (pin no.) 4
- LCDC Segment (Duty) output (pin name) SEG8 (pin no.) 5
- LCDC Segment (Duty) output (pin name) SEG9 (pin no.) 6
- LCDC Segment (Duty) output (pin name) SEG10 (pin no.) 7
- LCDC Segment (Duty) output (pin name) SEG11 (pin no.) 8
- LCDC Segment (Duty) output (pin name) SEG12 (pin no.) 9
- LCDC Segment (Duty) output (pin name) SEG13 (pin no.) 10
- LCDC Segment (Duty) output (pin name) SEG14 (pin no.) 11
- LCDC Segment (Duty) output (pin name) SEG15 (pin no.) 12
- LCDC Segment (Duty) output (pin name) SEG16 (pin no.) 13
- LCDC Segment (Duty) output (pin name) SEG17 (pin no.) 14
- LCDC Segment (Duty) output (pin name) SEG18 (pin no.) 15
- LCDC Segment (Duty) output (pin name) SEG19 (pin no.) 16
- LCDC Segment (Duty) output (pin name) SEG20 (pin no.) 17
- LCDC Segment (Duty) output (pin name) SEG21 (pin no.) 18
- LCDC Segment (Duty) output (pin name) SEG22 (pin no.) 19
- LCDC Segment (Duty) output (pin name) SEG23 (pin no.) 22
- LCDC Segment (Duty) output (pin name) SEG24 (pin no.) 23
- LCDC Segment (Duty) output (pin name) SEG25 (pin no.) 24
- LCDC Segment (Duty) output (pin name) SEG26 (pin no.) 25
- LCDC Segment (Duty) output (pin name) SEG27 (pin no.) 26
- LCDC Segment (Duty) output (pin name) SEG28 (pin no.) 27

- LCDC Segment (Duty) output (pin name) SEG29 (pin no.) 28
- LCDC Segment (Duty) output (pin name) SEG30 (pin no.) 29
- LCDC Segment (Duty) output (pin name) SEG31 (pin no.) 30
- LCDC Segment (Static) output (pin name) ST0 (pin no.) 22
- LCDC Segment (Static) output (pin name) ST1 (pin no.) 23
- LCDC Segment (Static) output (pin name) ST2 (pin no.) 24
- LCDC Segment (Static) output (pin name) ST3 (pin no.) 25
- LCDC Segment (Static) output (pin name) ST4 (pin no.) 26
- LCDC Segment (Static) output (pin name) ST5 (pin no.) 27
- LCDC Segment (Static) output (pin name) ST6 (pin no.) 28
- LCDC Segment (Static) output (pin name) ST7 (pin no.) 29
- LCDC Segment (Static) output (pin name) ST8 (pin no.) 30

1.10.12 Pins of LIN-UART (ch.2 to ch.7)

Pins of LIN-UART are shown below.

- LIN-UART serial clock I/O ch.2 relocation 0 (pin name) SCK2_0 (pin no.) 101
- LIN-UART serial clock I/O ch.2 relocation 1 (pin name) SCK2_1 (pin no.) 133
- LIN-UART serial clock I/O ch.3 relocation 0 (pin name) SCK3_0 (pin no.) 92
- LIN-UART serial clock I/O ch.3 relocation 1 (pin name) SCK3_1 (pin no.) 136
- LIN-UART serial clock I/O ch.4 relocation 0 (pin name) SCK4_0 (pin no.) 89
- LIN-UART serial clock I/O ch.4 relocation 1 (pin name) SCK4_1 (pin no.) 76
- LIN-UART serial clock I/O ch.5 relocation 0 (pin name) SCK5_0 (pin no.) 86
- LIN-UART serial clock I/O ch.5 relocation 1 (pin name) SCK5_1 (pin no.) 79
- LIN-UART serial clock I/O ch.6 relocation 0 (pin name) SCK6_0 (pin no.) 65
- LIN-UART serial clock I/O ch.6 relocation 1 (pin name) SCK6_1 (pin no.) 13
- LIN-UART serial clock I/O ch.7 relocation 0 (pin name) SCK7_0 (pin no.) 24
- LIN-UART serial clock I/O ch.7 relocation 1 (pin name) SCK7_1 (pin no.) 60
- LIN-UART serial data input ch.2 relocation 0 (pin name) SIN2_0 (pin no.) 102
- LIN-UART serial data input ch.2 relocation 1 (pin name) SIN2_1 (pin no.) 131
- LIN-UART serial data input ch.3 relocation 0 (pin name) SIN3_0 (pin no.) 99
- LIN-UART serial data input ch.3 relocation 1 (pin name) SIN3_1 (pin no.) 134
- LIN-UART serial data input ch.4 relocation 0 (pin name) SIN4_0 (pin no.) 91
- LIN-UART serial data input ch.4 relocation 1 (pin name) SIN4_1 (pin no.) 74
- LIN-UART serial data input ch.5 relocation 0 (pin name) SIN5_0 (pin no.) 88
- LIN-UART serial data input ch.5 relocation 1 (pin name) SIN5_1 (pin no.) 77
- LIN-UART serial data input ch.6 relocation 0 (pin name) SIN6_0 (pin no.) 63
- LIN-UART serial data input ch.6 relocation 1 (pin name) SIN6_1 (pin no.) 11
- LIN-UART serial data input ch.7 relocation 0 (pin name) SIN7_0 (pin no.) 22
- LIN-UART serial data input ch.7 relocation 1 (pin name) SIN7_1 (pin no.) 58
- LIN-UART serial data output ch.2 relocation 0 (pin name) SOT2_0 (pin no.) 100
- LIN-UART serial data output ch.2 relocation 1 (pin name) SOT2_1 (pin no.) 132
- LIN-UART serial data output ch.3 relocation 0 (pin name) SOT3_0 (pin no.) 98
- LIN-UART serial data output ch.3 relocation 1 (pin name) SOT3_1 (pin no.) 135
- LIN-UART serial data output ch.4 relocation 0 (pin name) SOT4_0 (pin no.) 90
- LIN-UART serial data output ch.4 relocation 1 (pin name) SOT4_1 (pin no.) 75
- LIN-UART serial data output ch.5 relocation 0 (pin name) SOT5_0 (pin no.) 87
- LIN-UART serial data output ch.5 relocation 1 (pin name) SOT5_1 (pin no.) 78
- LIN-UART serial data output ch.6 relocation 0 pin name) SOT6_0 (pin no.) 64
- LIN-UART serial data output ch.6 relocation 1 (pin name) SOT6_1 (pin no.) 12
- LIN-UART serial data output ch.7 relocation 0 (pin name) SOT7_0 (pin no.) 23
- LIN-UART serial data output ch.7 relocation 1 (pin name) SOT7_1 (pin no.) 59

1.10.13 Pins of Multi-function Serial Interface (ch.0 to ch.1, ch.8 to ch.9)

Pins of the multi-function serial interface are shown below.

- Multi-function Serial Clock I/O pin ch.0 relocation0 (pin name) SCK0_0 (pin no.) 123
- Multi-function Serial Clock I/O pin ch.0 relocation1 (pin name) SCK0_1 (pin no.) 49
- Multi-function Serial Clock I/O pin ch.1 relocation0 (pin name) SCK1_0 (pin no.) 126
- Multi-function Serial Clock I/O pin ch.1 relocation1 (pin name) SCK1_1 (pin no.) 46
- Multi-function Serial Clock I/O pin ch.8 relocation0 (pin name) SCK8_0 (pin no.) 19
- Multi-function Serial Clock I/O pin ch.8 relocation 1 (pin name) SCK8_1 (pin no.) 57
- Multi-function Serial Clock I/O pin ch.8 relocation 2 (pin name) SCK8_2 (pin no.) 30
- Multi-function Serial Clock I/O pin ch.9 relocation 0 (pin name) SCK9_0 (pin no.) 16
- Multi-function Serial Clock I/O pin ch.9 relocation 1 (pin name) SCK9_1 (pin no.) 54
- Multi-function Serial Input pin ch.0 relocation 0 (pin name) SIN0_0 (pin no.) 121
- Multi-function Serial Input pin ch.0 relocation 1 (pin name) SIN0_1 (pin no.) 47
- Multi-function Serial Input pin ch.1 relocation 0 (pin name) SIN1_0 (pin no.) 124
- Multi-function Serial Input pin ch.1 relocation 1 (pin name) SIN1_1 (pin no.) 44
- Multi-function Serial Input pin ch.8 relocation 0 (pin name) SIN8_0 (pin no.) 17
- Multi-function Serial Input pin ch.8 relocation 1 (pin name) SIN8_1 (pin no.) 55
- Multi-function Serial Input pin ch.8 relocation 2 (pin name) SIN8_2 (pin no.) 28
- Multi-function Serial Input pin ch.9 relocation 0 (pin name) SIN9_0 (pin no.) 14
- Multi-function Serial Input pin ch.9 relocation 1 (pin name) SIN9_1 (pin no.) 50
- Multi-function Serial Output pin ch.0 relocation 0 (pin name) SOT0_0 (pin no.) 122
- Multi-function Serial Output pin ch.0 relocation 1 (pin name) SOT0_1 (pin no.) 48
- Multi-function Serial Output pin ch.1 relocation 0 (pin name) SOT1_0 (pin no.) 125
- Multi-function Serial Output pin ch.1 relocation 1 (pin name) SOT1_1 (pin no.) 45
- Multi-function Serial Output pin ch.8 relocation 0 (pin name) SOT8_0 (pin no.) 18
- Multi-function Serial Output pin ch.8 relocation 1 (pin name) SOT8_1 (pin no.) 56
- Multi-function Serial Output pin ch.8 relocation 2 (pin name) SOT8_2 (pin no.) 29
- Multi-function Serial Output pin ch.9 relocation 0 (pin name) SOT9_0 (pin no.) 15
- Multi-function Serial Output pin ch.9 relocation 1 (pin name) SOT9_1 (pin no.) 53

1.10.14 Pins of Output Compare (ch.0 to ch.11)

Pins of the Output compare are shown below.

- Output compare output ch.0 relocation 0 (pin name) OCU0_0 (pin no.) 86
- Output compare output ch.0 relocation 1 (pin name) OCU0_1 (pin no.) 5
- Output compare output ch.1 relocation 0 (pin name) OCU1_0 (pin no.) 85
- Output compare output ch.1 relocation 1 (pin name) OCU1_1 (pin no.) 6
- Output compare output ch.2 relocation 0 (pin name) OCU2_0 (pin no.) 113
- Output compare output ch.2 relocation 1 (pin name) OCU2_1 (pin no.) 7
- Output compare output ch.3 relocation 0 (pin name) OCU3_0 (pin no.) 120
- Output compare output ch.3 relocation 1 (pin name) OCU3_1 (pin no.) 8
- Output compare output ch.4 relocation 0 (pin name) OCU4_0 (pin no.) 121
- Output compare output ch.4 relocation 1 (pin name) OCU4_1 (pin no.) 18
- Output compare output ch.5 relocation 0 (pin name) OCU5_0 (pin no.) 122
- Output compare output ch.5 relocation 1 (pin name) OCU5_1 (pin no.) 17
- Output compare output ch.6 relocation 0 (pin name) OCU6_0 (pin no.) 16
- Output compare output ch.6 relocation 1 (pin name) OCU6_1 (pin no.) 26
- Output compare output ch.7 relocation 0 (pin name) OCU7_0 (pin no.) 15
- Output compare output ch.7 relocation 1 (pin name) OCU7_1 (pin no.) 27
- Output compare output ch.8 relocation 0 (pin name) OCU8_0 (pin no.) 12
- Output compare output ch.8 relocation 1 (pin name) OCU8_1 (pin no.) 31
- Output compare output ch.9 relocation 0 (pin name) OCU9_0 (pin no.) 11
- Output compare output ch.9 relocation 1 (pin name) OCU9_1 (pin no.) 32
- Output compare output ch.10 relocation 0 (pin name) OCU10_0 (pin no.) 10
- Output compare output ch.10 relocation 1 (pin name) OCU10_1 (pin no.) 33
- Output compare output ch.11 relocation 0 (pin name) OCU11_0 (pin no.) 9
- Output compare output ch.11 relocation 1 (pin name) OCU11_1 (pin no.) 34

1.10.15 Pins of PPG (ch.0 to ch.23)

Pins of PPG are shown below.

- PPG output ch.0 relocation 0 (pin name) PPG0_0 (pin no.) 131
- PPG output ch.0 relocation 1 (pin name) PPG0_1 (pin no.) 98
- PPG output ch.0 relocation 2 (pin name) PPG0_2 (pin no.) 73
- PPG output ch.1 relocation 0 (pin name) PPG1_0 (pin no.) 132
- PPG output ch.1 relocation 1 (pin name) PPG1_1 (pin no.) 77
- PPG output ch.1 relocation 2 (pin name) PPG1_2 (pin no.) 103
- PPG output ch.1 relocation 3 (pin name) PPG1_3 (pin no.) 127
- PPG output ch.2 relocation 0 (pin name) PPG2_0 (pin no.) 133
- PPG output ch.2 relocation 1 (pin name) PPG2_1 (pin no.) 78
- PPG output ch.2 relocation 2 (pin name) PPG2_2 (pin no.) 104
- PPG output ch.3 relocation 0 (pin name) PPG3_0 (pin no.) 134
- PPG output ch.3 relocation 1 (pin name) PPG3_1 (pin no.) 79
- PPG output ch.3 relocation 2 (pin name) PPG3_2 (pin no.) 105
- PPG output ch.4 relocation 0 (pin name) PPG4_0 (pin no.) 135
- PPG output ch.4 relocation 1 (pin name) PPG4_1 (pin no.) 80
- PPG output ch.4 relocation 2 (pin name) PPG4_2 (pin no.) 106
- PPG output ch.5 relocation 0 (pin name) PPG5_0 (pin no.) 136
- PPG output ch.5 relocation 1 (pin name) PPG5_1 (pin no.) 81
- PPG output ch.5 relocation 2 (pin name) PPG5_2 (pin no.) 88
- PPG output ch.6 relocation 0 (pin name) PPG6_0 (pin no.) 137
- PPG output ch.6 relocation 1 (pin name) PPG6_1 (pin no.) 102
- PPG output ch.6 relocation 2 (pin name) PPG6_2 (pin no.) 87
- PPG output ch.7 relocation 0 (pin name) PPG7_0 (pin no.) 138
- PPG output ch.7 relocation 1 (pin name) PPG7_1 (pin no.) 101
- PPG output ch.7 relocation 2 (pin name) PPG7_2 (pin no.) 86
- PPG output ch.8 relocation 0 (pin name) PPG8_0 (pin no.) 74
- PPG output ch.8 relocation 1 (pin name) PPG8_1 (pin no.) 100
- PPG output ch.8 relocation 2 (pin name) PPG8_2 (pin no.) 85
- PPG output ch.9 relocation 0 (pin name) PPG9_0 (pin no.) 75
- PPG output ch.9 relocation 1 (pin name) PPG9_1 (pin no.) 99
- PPG output ch.9 relocation 2 (pin name) PPG9_2 (pin no.) 113
- PPG output ch.10 relocation 0 (pin name) PPG10_0 (pin no.) 76
- PPG output ch.10 relocation 1 (pin name) PPG10_1 (pin no.) 111
- PPG output ch.10 relocation 2 (pin name) PPG10_2 (pin no.) 120
- PPG output ch.11 relocation 0 (pin name) PPG11_0 (pin no.) 19
- PPG output ch.11 relocation 1 (pin name) PPG11_1 (pin no.) 126
- PPG output ch.12 relocation 0 (pin name) PPG12_0 (pin no.) 22

- PPG output ch.12 relocation 1 (pin name) PPG12_1 (pin no.) 57
- PPG output ch.13 relocation 0 (pin name) PPG13_0 (pin no.) 23
- PPG output ch.13 relocation 1 (pin name) PPG13_1 (pin no.) 58
- PPG output ch.14 relocation 0 (pin name) PPG14_0 (pin no.) 24
- PPG output ch.14 relocation 1 (pin name) PPG14_1 (pin no.) 59
- PPG output ch.15 relocation 0 (pin name) PPG15_0 (pin no.) 25
- PPG output ch.15 relocation 1 (pin name) PPG15_1 (pin no.) 60
- PPG output ch.16 relocation 0 (pin name) PPG16_0 (pin no.) 63
- PPG output ch.17 relocation 0 (pin name) PPG17_0 (pin no.) 64
- PPG output ch.18 relocation 0 (pin name) PPG18_0 (pin no.) 65
- PPG output ch.19 relocation 0 (pin name) PPG19_0 (pin no.) 66
- PPG output ch.20 relocation 0 (pin name) PPG20_0 (pin no.) 67
- PPG output ch.21 relocation 0 (pin name) PPG21_0 (pin no.) 68
- PPG output ch.22 relocation 0 (pin name) PPG22_0 (pin no.) 69
- PPG output ch.23 relocation 0 (pin name) PPG23_0 (pin no.) 70
- PPG trigger input 0 (ch.0 to ch.3) (pin name) TRG0 (pin no.) 121
- PPG trigger input 1 (ch.4 to ch.7) (pin name) TRG1 (pin no.) 124
- PPG trigger input 2 (ch.8 to ch.11) (pin name) TRG2 (pin no.) 127
- PPG trigger input 3 (ch.12 to ch.15) (pin name) TRG3 (pin no.) 92
- PPG trigger input 4 (ch.16 to ch.19) (pin name) TRG4 (pin no.) 89
- PPG trigger input 5 (ch.20 to ch.23) (pin name) TRG5 (pin no.) 126

1.10.16 Pins of Reload Timer (ch.0 to ch.3)

Pins of the Reload timer are shown below.

- Reload timer event input ch.0 relocation 0 (pin name) TIN0_0 (pin no.) 98
- Reload timer event input ch.0 relocation 1 (pin name) TIN0_1 (pin no.) 74
- Reload timer event input ch.0 relocation 2 (pin name) TIN0_2 (pin no.) 131
- Reload timer event input ch.1 relocation 0 (pin name) TIN1_0 (pin no.) 92
- Reload timer event input ch.1 relocation 1 (pin name) TIN1_1 (pin no.) 75
- Reload timer event input ch.1 relocation 2 (pin name) TIN1_2 (pin no.) 132
- Reload timer event input ch.2 relocation 0 (pin name) TIN2_0 (pin no.) 91
- Reload timer event input ch.2 relocation 1 (pin name) TIN2_1 (pin no.) 76
- Reload timer event input ch.2 relocation 2 (pin name) TIN2_2 (pin no.) 133
- Reload timer event input ch.3 relocation 0 (pin name) TIN3_0 (pin no.) 90
- Reload timer event input ch.3 relocation 1 (pin name) TIN3_1 (pin no.) 77
- Reload timer event input ch.3 relocation 2 (pin name) TIN3_2 (pin no.) 134
- Reload timer output ch.0 relocation 0 (pin name) TOT0_0 (pin no.) 89
- Reload timer output ch.0 relocation 1 (pin name) TOT0_1 (pin no.) 78
- Reload timer output ch.0 relocation 2 (pin name) TOT0_2 (pin no.) 135
- Reload timer output ch.1 relocation 0 (pin name) TOT1_0 (pin no.) 88
- Reload timer output ch.1 relocation 1 (pin name) TOT1_1 (pin no.) 79
- Reload timer output ch.1 relocation 2 (pin name) TOT1_2 (pin no.) 136
- Reload timer output ch.2 relocation 0 (pin name) TOT2_0 (pin no.) 87
- Reload timer output ch.2 relocation 1 (pin name) TOT2_1 (pin no.) 102
- Reload timer output ch.2 relocation 2 (pin name) TOT2_2 (pin no.) 137
- Reload timer output ch.3 relocation 0 (pin name) TOT3_0 (pin no.) 86
- Reload timer output ch.3 relocation 1 (pin name) TOT3_1 (pin no.) 101
- Reload timer output ch.3 relocation 2 (pin name) TOT3_2 (pin no.) 138

1.10.17 Pins of Real Time Clock

Pins of the real time clock are shown below.

- RTC overflow output (pin name) WOT (pin no.) 98

1.10.18 Pins of Stepping Motor Controller (ch.0 to ch.5)

Pins of the stepping motor controller are shown below.

- SMC output ch.0 (pin name) PWM1M0(pin no.) 44
- SMC output ch.0 (pin name) PWM1P0 (pin no.) 43
- SMC output ch.0 (pin name) PWM2M0(pin no.) 46
- SMC output ch.0 (pin name) PWM2P0 (pin no.) 45
- SMC output ch.1 (pin name) PWM1M1(pin no.) 48
- SMC output ch.1 (pin name) PWM1P1 (pin no.) 47
- SMC output ch.1 (pin name) PWM2M1(pin no.) 50
- SMC output ch.1 (pin name) PWM2P1 (pin no.) 49
- SMC output ch.2 (pin name) PWM1M2(pin no.) 54
- SMC output ch.2 (pin name) PWM1P2 (pin no.) 53
- SMC output ch.2 (pin name) PWM2M2(pin no.) 56
- SMC output ch.2 (pin name) PWM2P2 (pin no.) 55
- SMC output ch.3 (pin name) PWM1M3(pin no.) 58
- SMC output ch.3 (pin name) PWM1P3 (pin no.) 57
- SMC output ch.3 (pin name) PWM2M3(pin no.) 60
- SMC output ch.3 (pin name) PWM2P3 (pin no.) 59
- SMC output ch.4 (pin name) PWM1M4(pin no.) 64
- SMC output ch.4 (pin name) PWM1P4 (pin no.) 63
- SMC output ch.4 (pin name) PWM2M4(pin no.) 66
- SMC output ch.4 (pin name) PWM2P4 (pin no.) 65
- SMC output ch.5 (pin name) PWM1M5(pin no.) 68
- SMC output ch.5 (pin name) PWM1P5 (pin no.) 67
- SMC output ch.5 (pin name) PWM2M5(pin no.) 70
- SMC output ch.5 (pin name) PWM2P5 (pin no.) 69

1.10.19 Pins of Sound Generator (ch.0 to ch.4)

Pins of the sound generator are shown below.

- Sound generator SGA output ch.0 (pin name) SGA0 (pin no.) 102
- Sound generator SGA output ch.1 (pin name) SGA1 (pin no.) 100
- Sound generator SGA output ch.2 (pin name) SGA2 (pin no.) 92
- Sound generator SGA output ch.3 (pin name) SGA3 (pin no.) 90
- Sound generator SGA output ch.4 relocation 0 (pin name) SGA4_0 (pin no.) 26
- Sound generator SGA output ch.4 relocation 1 (pin name) SGA4_1 (pin no.) 80
- Sound generator SGO output ch.0 (pin name) SGO0 (pin no.) 101
- Sound generator SGO output ch.1 (pin name) SGO1 (pin no.) 99
- Sound generator SGO output ch.2 (pin name) SGO2 (pin no.) 91
- Sound generator SGO output ch.3 (pin name) SGO3 (pin no.) 89
- Sound generator SGO output ch.4 relocation 0 (pin name) SGO4_0 (pin no.) 27
- Sound generator SGO output ch.4 relocation 1 (pin name) SGO4_1 (pin no.) 81

1.10.20 Pins of Up/Down Counter (ch.0, ch.1)

Pins of the Up/down counter are shown below.

- Up/down Counter AIN input ch.0 relocation 0 (pin name) AIN0_0 (pin no.) 25
- Up/down Counter AIN input ch.0 relocation 1 (pin name) AIN0_1 (pin no.) 50
- Up/down Counter AIN input ch.0 relocation 2 (pin name) AIN0_2 (pin no.) 63
- Up/down Counter AIN input ch.1 relocation 0 (pin name) AIN1_0 (pin no.) 28
- Up/down Counter AIN input ch.1 relocation 1 (pin name) AIN1_1 (pin no.) 47
- Up/down Counter BIN input ch.0 relocation 0 (pin name) BIN0_0 (pin no.) 26
- Up/down Counter BIN input ch.0 relocation 1 (pin name) BIN0_1 (pin no.) 49
- Up/down Counter BIN input ch.0 relocation 2 (pin name) BIN0_2 (pin no.) 64
- Up/down Counter BIN input ch.1 relocation 0 (pin name) BIN1_0 (pin no.) 29
- Up/down Counter BIN input ch.1 relocation 1 (pin name) BIN1_1 (pin no.) 46
- Up/down Counter ZIN input ch.0 relocation 0 (pin name) ZIN0_0 (pin no.) 27
- Up/down Counter ZIN input ch.0 relocation 1 (pin name) ZIN0_1 (pin no.) 48
- Up/down Counter ZIN input ch.0 relocation 2 (pin name) ZIN0_2 (pin no.) 65
- Up/down Counter ZIN input ch.1 relocation 0 (pin name) ZIN1_0 (pin no.) 30
- Up/down Counter ZIN input ch.1 relocation 1 (pin name) ZIN1_1 (pin no.) 45

1.10.21 Pins of Clock

Pins of the clock are shown below.

- Main Clock OSC input (pin name) X0 (pin no.) 117
- Main Clock OSC output (pin name) X1 (pin no.) 118
- Sub Clock OSC input (Dual clock product) (pin name) (X0A) (pin no.) 95
- Sub Clock OSC output (Dual clock product) (pin name) (X1A) (pin no.) 96

1.10.22 Pins of Mode

Pins of the mode are shown below.

- Mode pin (pin name) MD0 (pin no.) 114
- Mode pin (pin name) MD1 (pin no.) 115
- Mode pin (pin name) MD2 (pin no.) 116

1.10.23 Pins of Reset

Pins of reset are shown below.

- Reset pin (pin name) RSTX (pin no.) 107
- DEBUG I/F (pin name) DEBUGIF (pin no.) 110
- NMI pin (pin name) NMIX (pin no.) 97

1.10.24 Pins of Power Supply

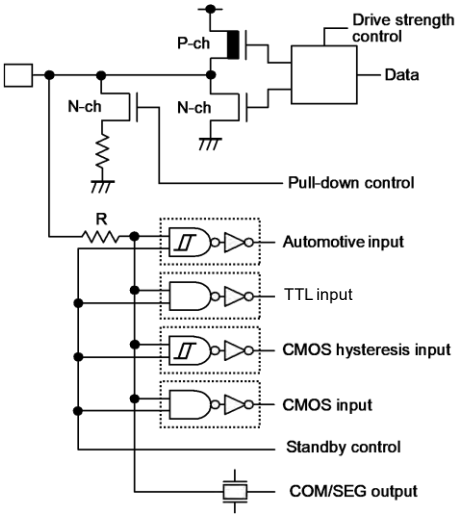
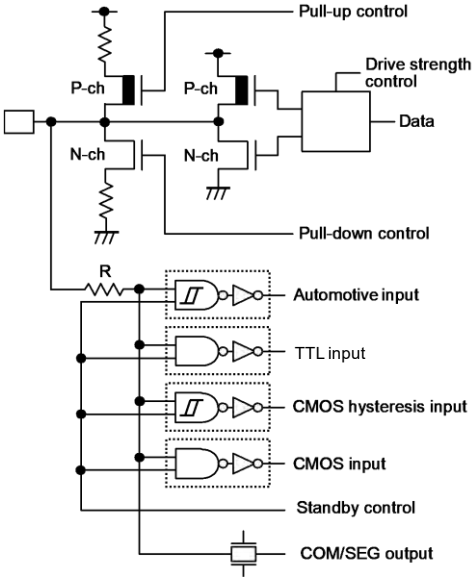
Pins of power supply are shown below.

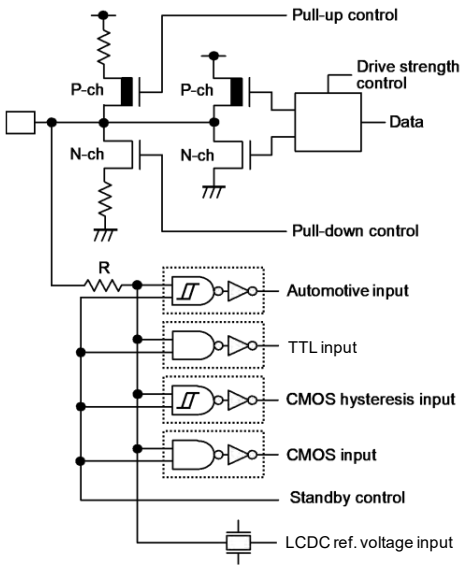
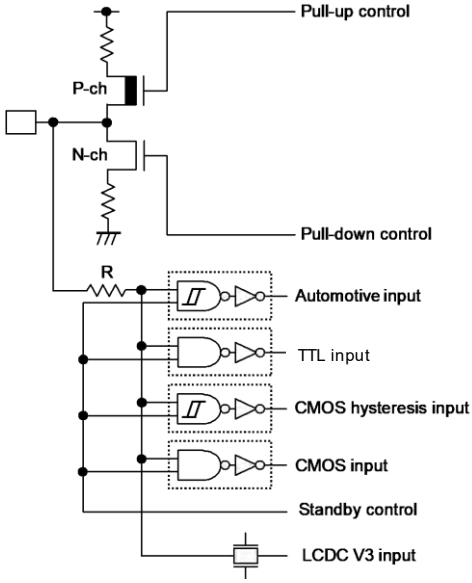
- +3.3 V/+5.0 V Power supply (pin name) VCCE (pin no.) 1
- +3.3 V/+5.0 V Power supply (pin name) VCCE (pin no.) 20
- +5.0 V Power supply (pin name) VCC5 (pin no.) 108
- +5.0 V Power supply (pin name) VCC5 (pin no.) 128
- +5.0 V Power supply (pin name) VCC5 (pin no.) 36
- +5.0 V Power supply (pin name) VCC5 (pin no.) 93
- ADC, DAC Analog Power supply (pin name) AVCC (pin no.) 84
- ADC, DAC GND / Reference low-voltage (pin name) AVSS/AVRL (pin no.) 82
- ADC Reference high-voltage (pin name) AVRH (pin no.) 83
- External Capacitance connection pin (pin name) C (pin no.) 130
- GND (pin name) VSS (pin no.) 109, 119, 129, 144, 21, 37, 94
- GND for SMC high current (pin name) DVSS (pin no.) 42, 52, 62, 72
- Power supply for SMC high current (pin name) DVCC (pin no.) 41, 51, 61, 71

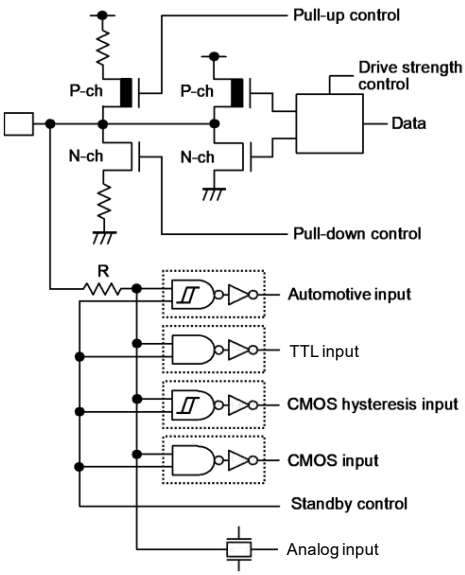
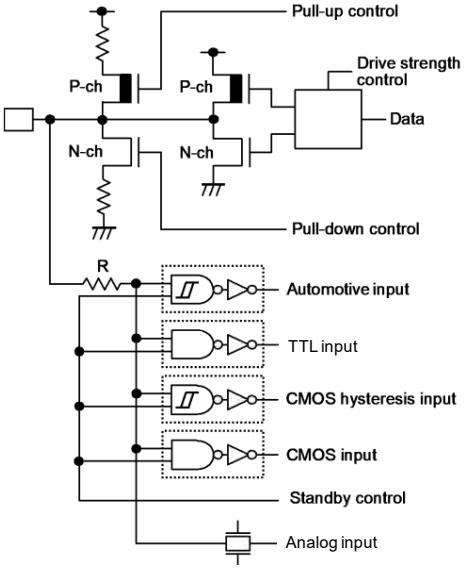
1.11 I/O Circuit Types

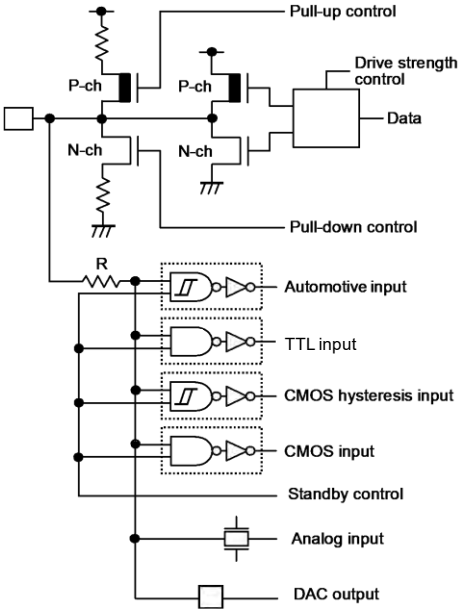
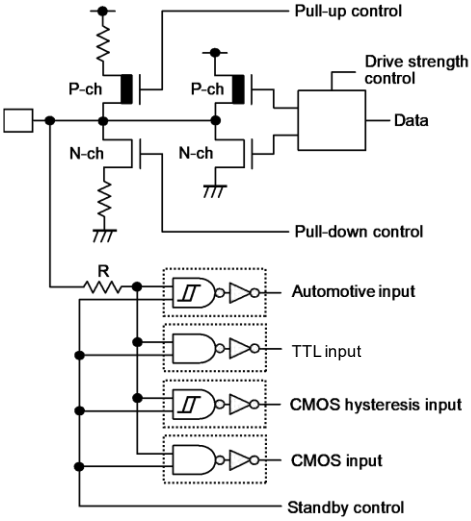
This section explains the I/O Circuit Types.

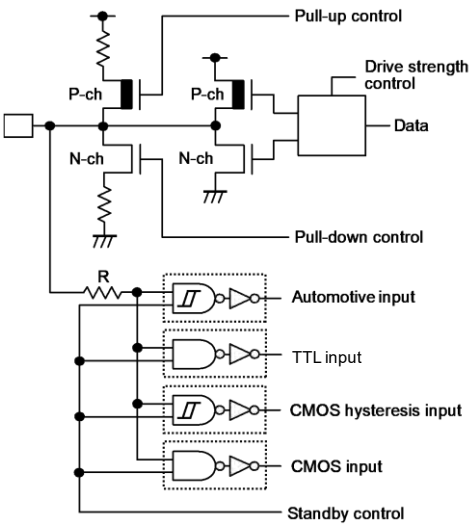
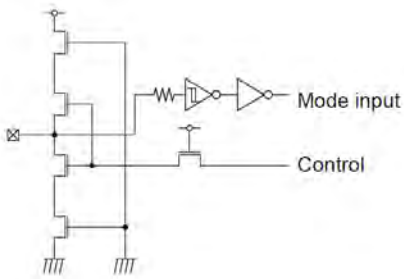
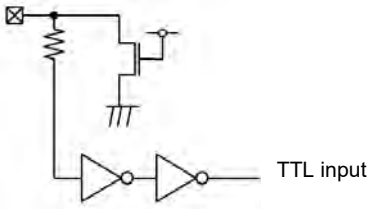
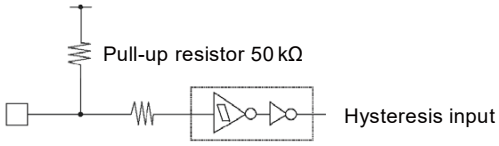
Table 1-4. I/O Circuit Types

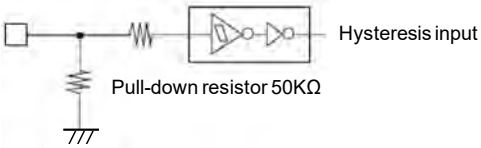
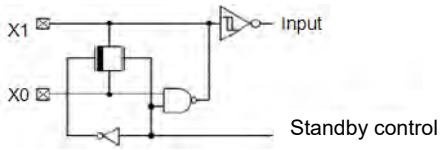
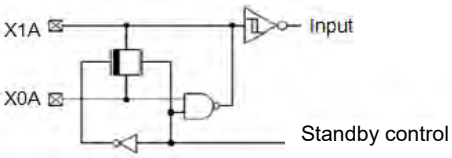
Type	Circuit	Remarks
H		General-purpose I/O port with COM/SEG output and with against 3V pad power supply (5 V tolerant). IOH = -1/-2 mA (@VCCE = 5 V), IOH = -0.5/-1/-2 mA (@VCCE = 3.3 V), IOL = ½ mA (@VCCE = 5 V), IOL = 0.5/1/2 mA (@VCCE = 3.3 V) Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input
I		General-purpose I/O port with COM/SEG output. IOH = -1/-2 mA, IOL = ½ mA Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input

Type	Circuit	Remarks
I2		General-purpose I/O port with LCDC reference voltage input IOH = -1/-2 mA, IOL = ½ mA Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input
I3		General-purpose input port with LCDC V3 input Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input

Type	Circuit	Remarks
J		General-purpose I/O port with analog input. IOH = -1/-2 mA, IOL = ½ mA Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input
K		General-purpose I/O port with analog input and with high current capable for SMC. IOH = -1/-2/-30 mA, IOL = 1/2/30 mA Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input

Type	Circuit	Remarks
L		General-purpose I/O port with analog input and with DAC output IOH = -1/-2 mA, IOL = ½ mA Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input
M		General-purpose I/O port. IOH = -1/-2 mA, IOL = ½ mA Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input

Type	Circuit	Remarks
N		General-purpose I/O port with I^2C output IOH = -1/-2/-3 mA, IOL = 1/2/3 mA Pull-up resistor control Pull-down resistor control Automotive level input TTL level input CMOS level hysteresis input CMOS level input
A		Mode pin
B		DEBUG I/F pin
R		CMOS level hysteresis input

Type	Circuit	Remarks
R2		CMOS level hysteresis input
X		Main clock
Y		Sub clock

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.

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 rating. 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.

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's 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% RH.
  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.

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 use 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.

Please check the latest handling precautions at the following URL.

http://www.cypress.com/documentation/datasheets/handling-precautions

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), analog input, and power supply to high-current output buffer pins (DVCC) and 5 V/3.3 V power supply(VCCE) must not be exceed the digital power supply (VCC5) when the power supply to the analog system, high-current output buffer pins and 5 V/3.3 V power supply is turned on or off.

  In the correct power-on sequence, turn on the digital power supply (VCC5), the power supply of high-current output buffer pins (DVCC) and 5 V/3.3 V power supply (VCCE) simultaneously. Or, turn on the digital power supply (VCC5), and then turn on analog power supplies (AVCC, AVRH), the power supply of high-current output buffer pins (DVCC) and 5 V/3.3 V power supply (VCCE) .

- Treatment of Unused Pins

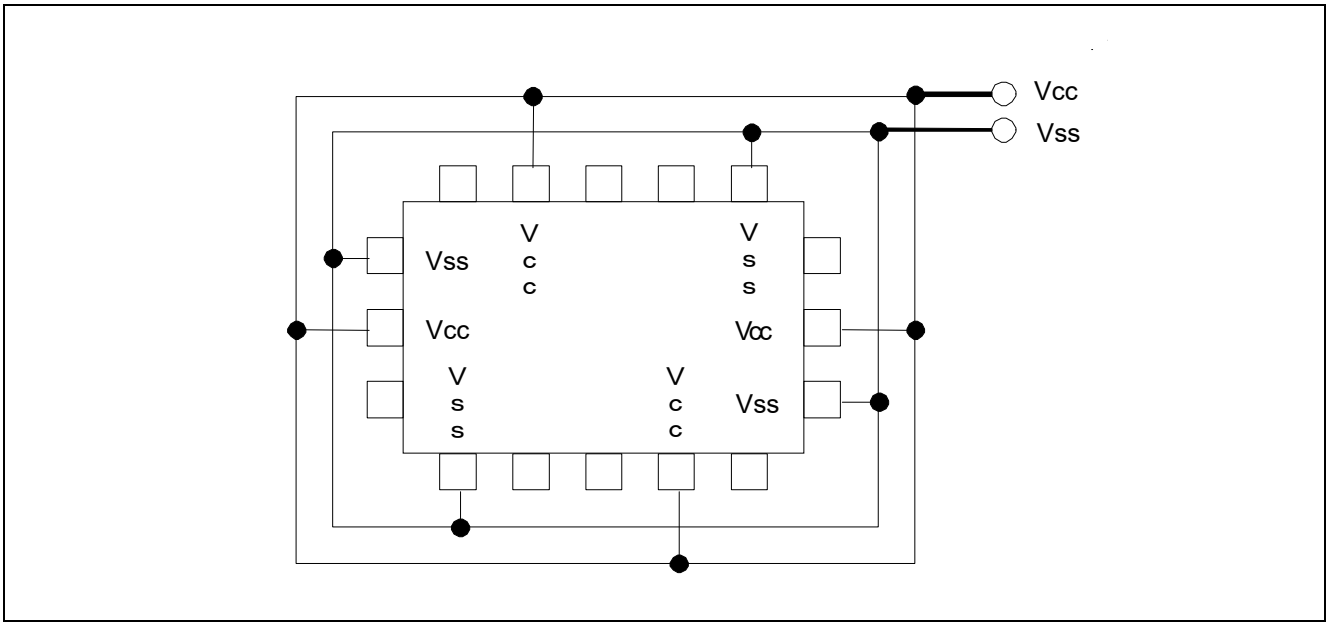
  If unused input pins are left open, they may cause a permanent damage to the device due to malfunction or latch-up. Connect a 2 kΩ resistor or more to each of unused pins for pull-up or pill-down processing.

  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 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 the 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 resonator (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 (MD2, MD1, MD0)

  Connect the MD2, MD1and 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 resonator 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=VCC5 and AVSS/AVRL=VSS even if the A/D converter is not used.

- Notes on Using External Clock

  An external clock is not supported.

  None of the external direct clock input can be used for both main clock and sub clock.

- Power-on Sequence of A/D Converter Power Supplies and Analog Inputs

  Be sure to turn on the digital power supply (Vcc5) first, and then turn on the A/D converter power supplies (AVcc, AVRH, AVRL) and analog inputs (AN0 to AN39). Also, turn off the A/D converter power supplies and analog inputs first, and then turn off the digital power supply (Vcc5). When the AVRH 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.)

- Treatment of Power Supplies for High Current Output Buffer Pins (DVcc, DVss)

  Be sure to turn on the digital power supply (Vcc) first, and then turn on the power supplies for high current output buffer pins (DVcc, DVss). Also, turn off the power supplies for high current output buffer pins first, and then turn off the digital power supply (Vcc).

  Even if the high current output buffer pins are used as general-purpose ports, the power supplies of high current output buffer pins (DVcc, DVss) must be powered. (The power supplies of high current output buffer pins and the digital power supplies 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 datasheet.

Note:

For the detailed specifications of operating voltages, see the latest datasheet.

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 Having 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). However, if a pin is also used for an external bus, its function is switched by the external bus setting. 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, stop mode, watch mode(power shutdown) or stop mode(power shutdown), follow the procedure explained in the "Activating the sleep mode, watch mode, or stop mode" or the "Activating the watch mode (power shutdown) or stop mode(power shutdown)" of "Chapter: Power Consumption Control".

Take the following notes when using a monitor debugger.

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

2.3.3 Notes When Writing Data in a Register Having the Status Flag

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

When writing data in the register that has a status flag (especially, an interrupt request flag) to control function, taking care not to clear its status flag erroneously must be followed.

The program must be written not to clear the flag to the status bit, and then to set the control bits to have the desired value.

Especially, if multiple control bits are used, the bit instruction cannot be used. (The bit instruction can access to a single bit only.) By the Byte, Half-word, or Word access, data is written to the control bits and status flag simultaneously. During this time, take care not to clear other bits (in this case, the bits of status flag) erroneously.

Note:

It is not necessary to note that the bit instruction considers this respect compared with the register to which read-modify-write (RMW) is supported. When the bit instruction is used for the register to which read-modify-write (RMW) is not supported, it is necessary to note it.

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.

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 × 16)
- 32-bit address space (4GB)
- 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 × 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
  - Protection of some address-mapped registers as system registers during user mode
  - Protection of some instructions as privilege instructions during 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 unnormalized number
- The only rounding mode supported is nearest value
- Denormalized numbers are truncated to 0 or generate an exception
- Floating-point register: 32-bit × 16 sets
- Multiply and Add, Multiply and Sub instructions supported
- Division and square root operations supported

3.3 CPU Operating Description

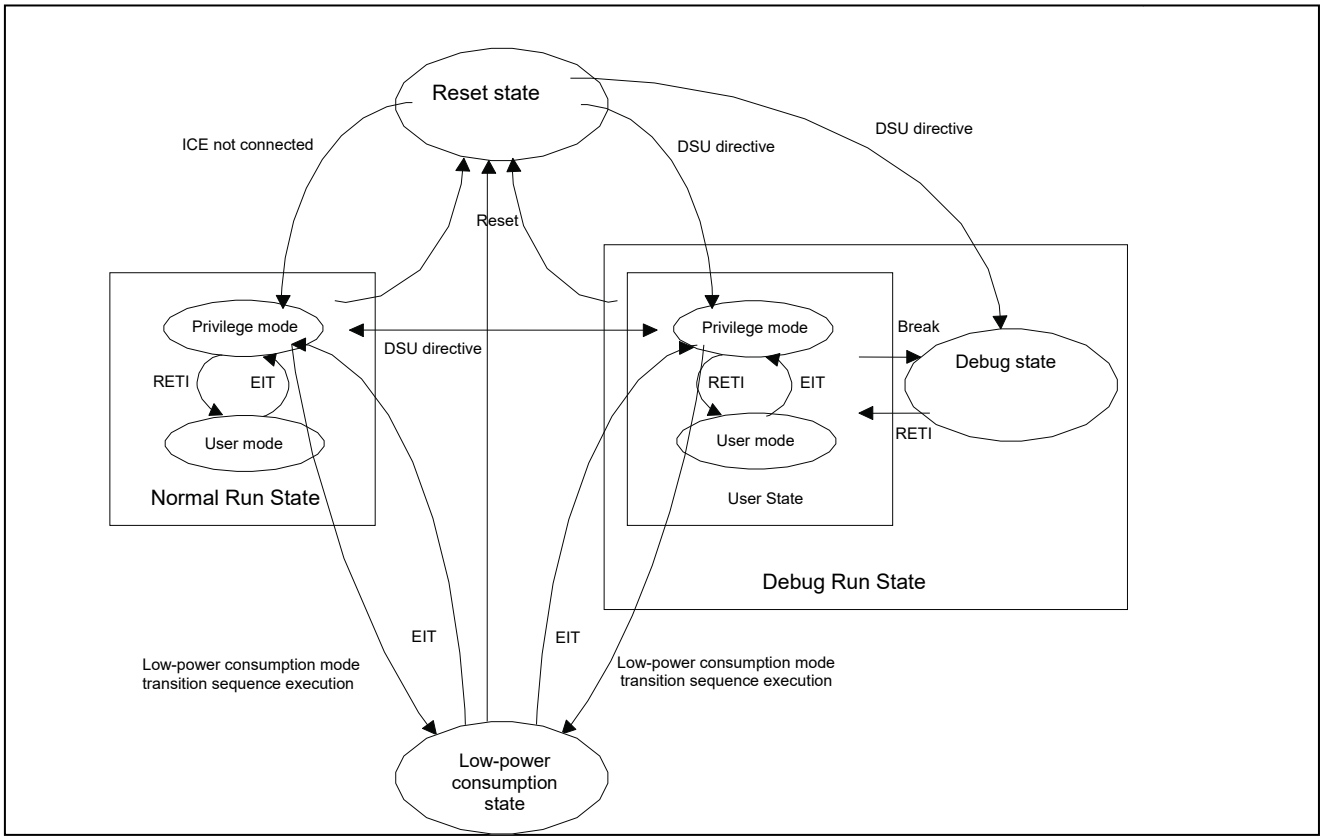
This section explains the operation of the CPU.

CPU Operating Status

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



- Reset State

  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.

- Normal Run State

  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.

- Low-power Consumption State

  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 mode. Recovery from the low-power consumption state is carried out by interrupts.

- Debug Run State

  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 of the floating point operation processing, see "FR Family FR81 32-bit Microcontroller Programming Manual".

3.6 Data Structure

This section explains the data structure.

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".

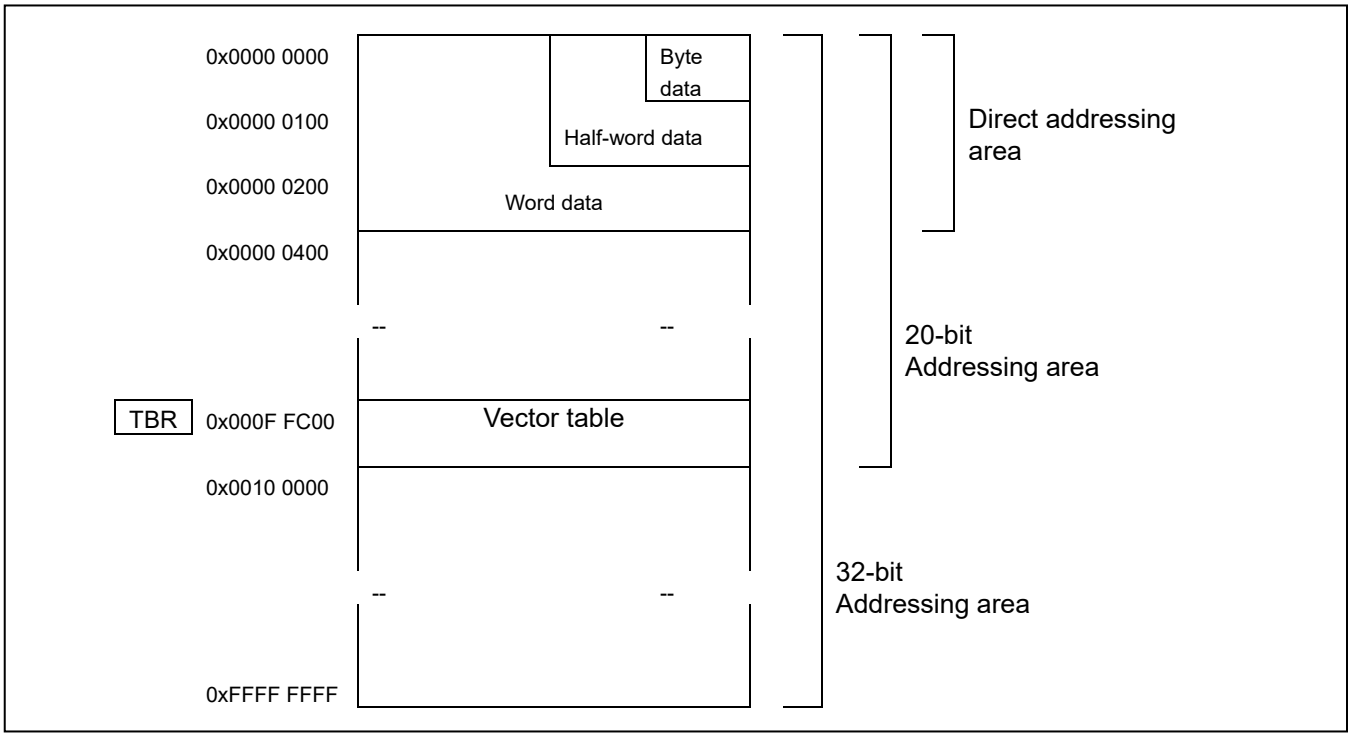
3.7 Addressing

This section explains addressing of the CPU.

A memory space is 32-bit linear.

The CPU manages the address space in bytes. 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.

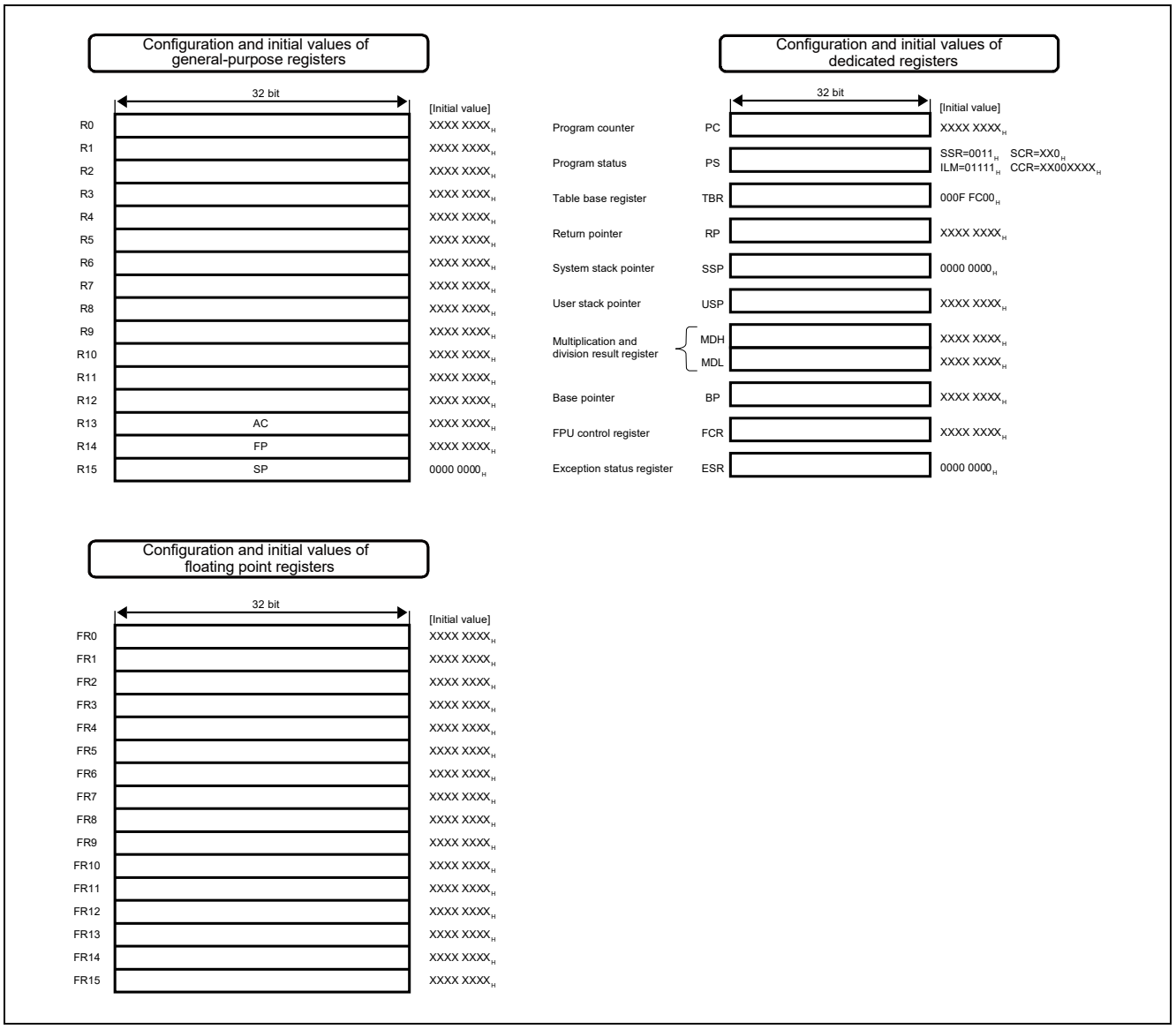
3.8.1 General-purpose Registers, Dedicated Registers, and Floating Point Registers

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

This section explains System register.

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
- Wild register control register
- FLASH control 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.

3.9.1 Reset

This section explains Reset.

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 "Chapter: Fixed Vector Function" for details.

3.9.2 EIT Processing

This section explains the EIT processing.

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

Interrupt 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
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

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

3.10.1 Overview

The overview of the memory protection function (MPU) is shown.

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 addresses 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
- 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. Registers 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				PAN1	PAN0	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 at User Mode (Initial value)
1	Instruction Fetch enable at 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	Buffer Enable 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[1:0] (Protection Area Number)

Indicates the number of configurable 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	• • •	bit0
	IPVA[31:0]		
Initial value	X	• • •	X
Attribute	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:

This register is a prohibition of use.

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:

This register is a prohibition of use.

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 violation of the data access protection occurs is saved.

DPVAR: Address 0320$_H$ (Access: Word)

	bit31	• • •	bit0
	DPVA[31:0]		
Initial value	X	• • •	X
Attribute	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 0328H (Access: Word)

	bit31	• • •	bit0
	DEA[31:0]		
Initial value	X	• • •	X
Attribute	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

This bit 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 effectively done. 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 PABR7: Address 0330_H, 0338_H, 0340_H•••(Access: Word)

	bit31		• • •					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 PACR7) 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 at "0000_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 PACR7: 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 permitted in user mode (initial value)
1	Instruction fetch permitted 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" to this bit. 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	Buffer Enable Specification for the Specified Protection Area
0	Buffer Disable (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 Protectorate Area
00000	Reserved
00001	Reserved
00010	Reserved
00011	16B
00100	32B
00101	64B
00110	128B
00111	256B
01000	512B
01001	1KB
01010	2KB
01011	4KB
01100	8KB
01101	16KB
01110	32KB
01111	64KB
10000	128KB

ASZ[4:0]	Size of the Specified Protectorate Area
10001	256KB
10010	512KB
10011	1MB
10100	2MB
10101	4MB
10110	8MB
10111	16MB
11000	32MB
11001	64MB
11010	128MB
11011	256MB
11100	512MB
11101	1GB
11110	2GB
11111	4GB

[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 (MPU)

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

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 Errors
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

This section explains setting up memory protection areas of the CPU.

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 eight 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

This section explains instruction access protection violation of the CPU.

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

This section explains data access protection violation of the CPU.

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 cancelled. 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 Errors

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 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 cancelled. 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.

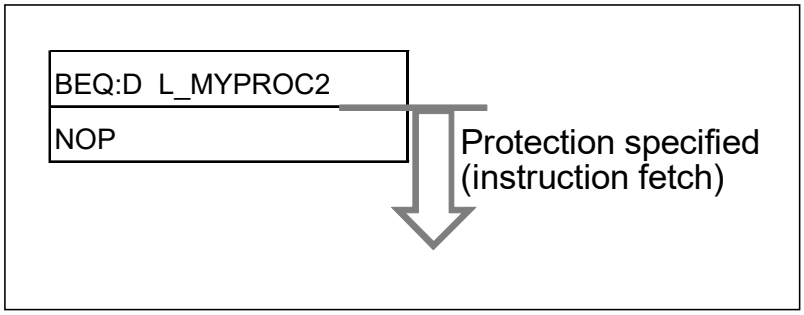
- System register access in user mode (illegal instruction exception)
- Bus error in buffer prohibition area access (illegal instruction exception)
- Bus error in buffer permission area access (data access error interrupt)

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 is different from that of instruction access protection area, 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 external bus can be used.

  The program starts from the built-in FLASH.

- Serial Writer Mode

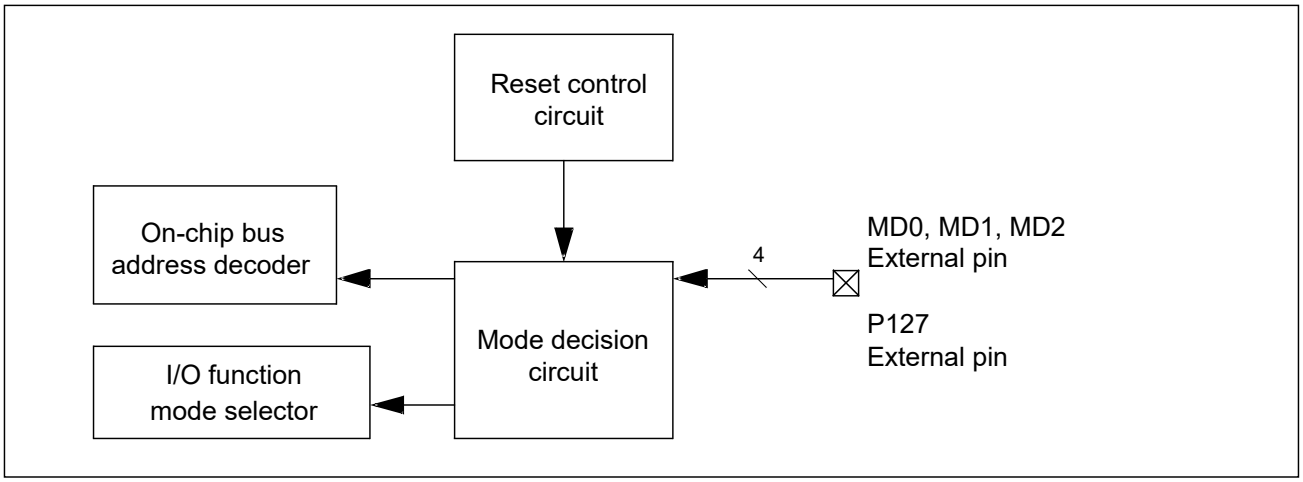
  It is a mode to which the built-in FLASH is programmed by using the serial writer.

  The program starts from the built-in Boot-ROM.

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 data register

Bus Mode Register: BMODR (Bus Mode Register)

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	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[1]: 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 shows operations of the operation mode.

4.5.1 MD0, MD1, MD2, P127 Pins Settings

4.5.2 Fetching the Operation Mode

4.5.3 Explanation of Each Operation Mode

4.5.1 MD0, MD1, MD2, P127 Pins Settings

MD0, MD1, MD2 and P127 pins settings are shown.

Table 4-1. Pin Settings

Operation Mode	MD2	MD1	MD0	P127
User mode	0	0	0	-
Serial writer mode	0	0	1	1

Other combination setting is prohibited except the above-mentioned combination.

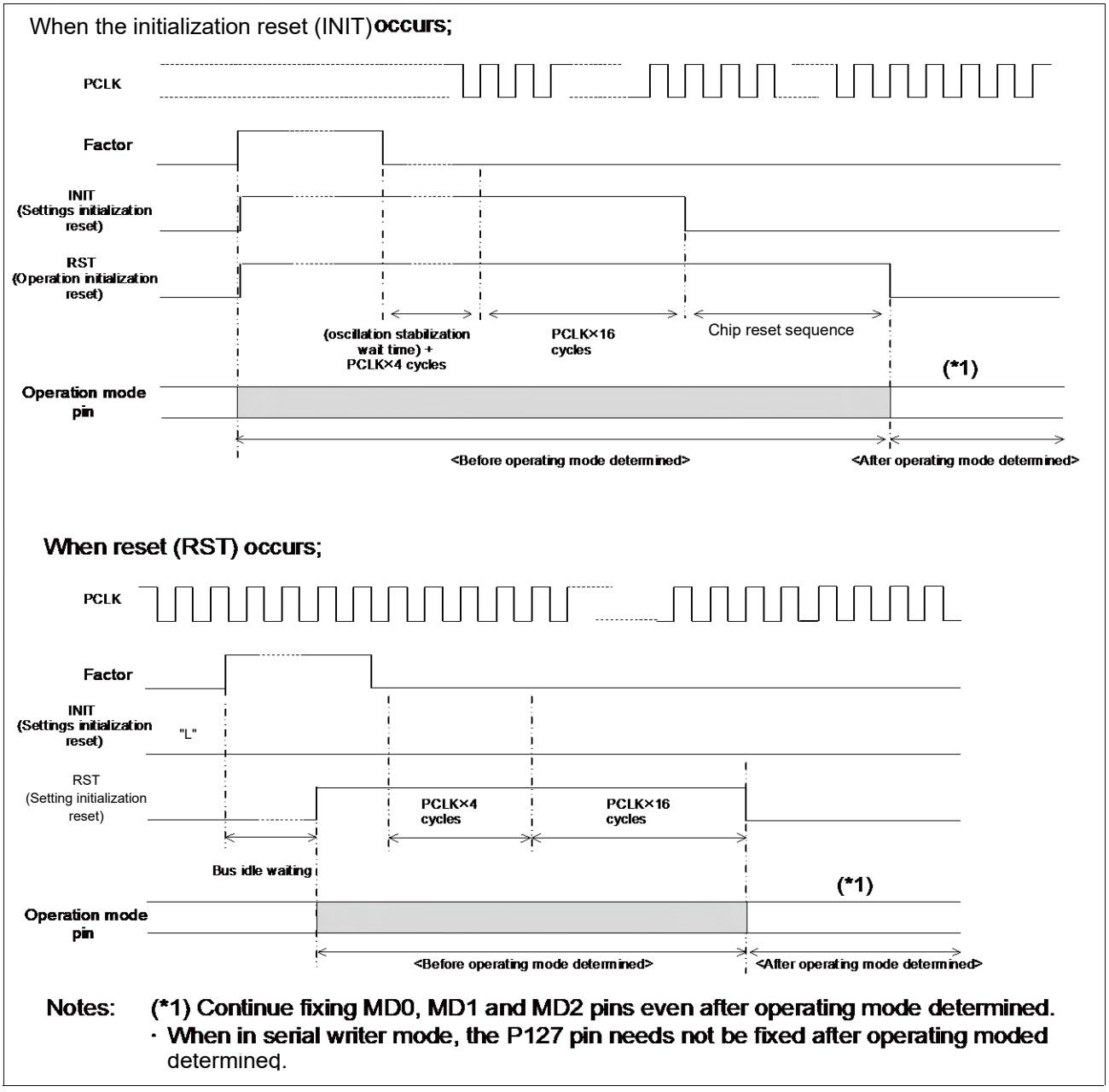
4.5.2 Fetching the Operation Mode

This section explains fetching the operation mode.

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, MD2 and P127 pin inputs must be determined. (The P127 pin needs not be determined in the User mode.)

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



4.5.3 Explanation of Each Operation Mode

This section explains each operation mode.

The following details each operation mode.

- User Mode

  An external bus pin is reset immediately when a reset is entered for the external reset pin. For details, see "A.4. Pins Statuses in State of CPU" in "Appendix".

- Serial Writer Mode

  Contact their representatives.

5. Clock

This chapter explains the clock.

5.1 Overview
5.2 Features
5.3 Configuration
5.4 Register
5.5 Operation

5.1 Overview

This section explains the overview of the clock.

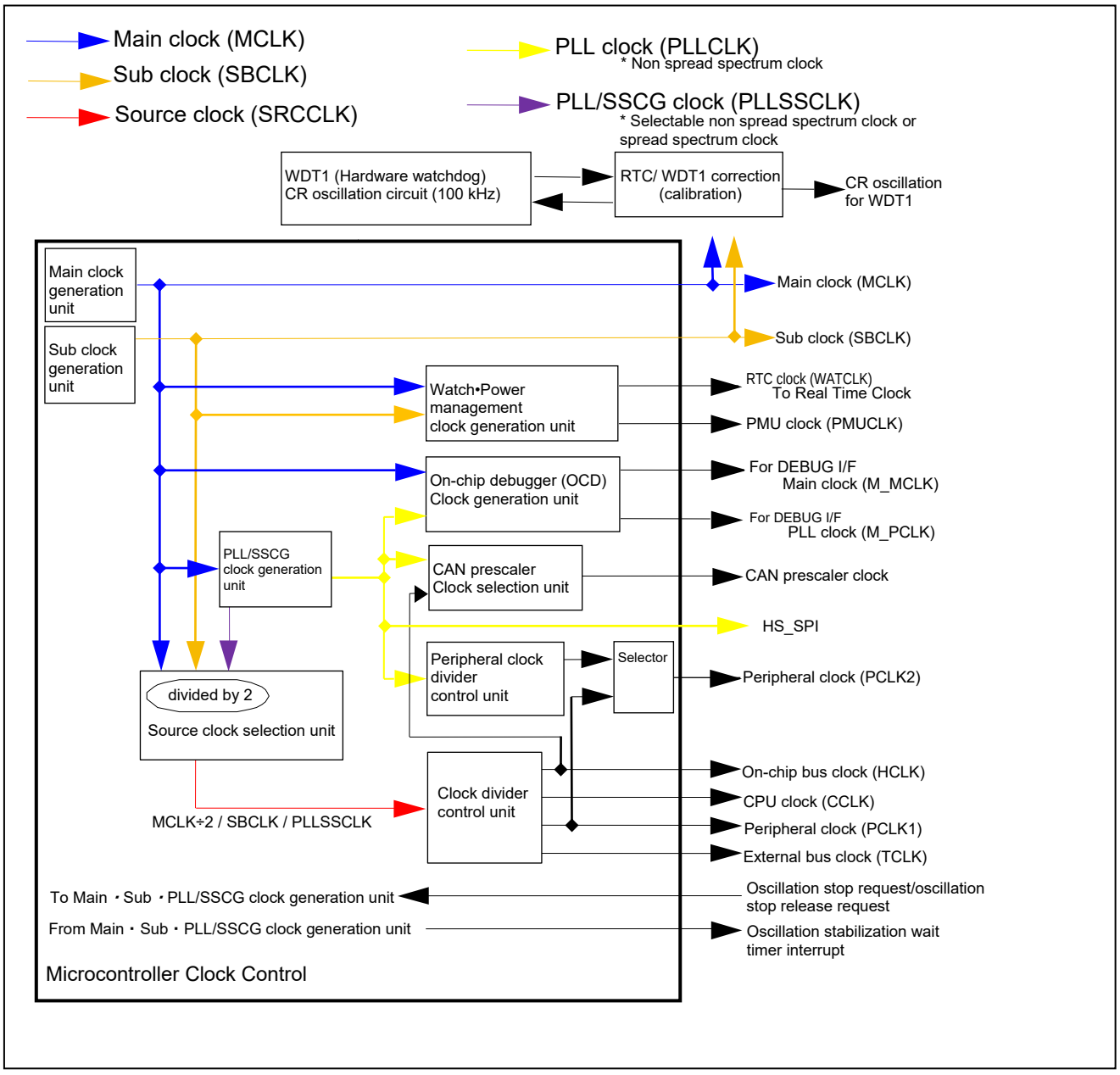
The built-in oscillation circuit generates a dual clock product, which generates individual clock systems on the chip. This product also implements the CR oscillation circuit for watchdog timer 1.

- External pins for the built-in oscillation circuit:

  Main clock: Connects to the crystal resonator
  Sub 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, or sub clock (SBCLK).
- 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.

- 2 system on-chip oscillators is implemented.
- The main clock (MCLK) is multiplied by on-chip PLL/SSCG.
- Multiplied clock is supplied by independent 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) and sub clock oscillation stabilization wait timer (sub timer) can be used as a general-purpose interrupt interval timer after the oscillation stabilization of each clock for main, and sub takes place.
- The clock for the real time clock can be selected from the main clock (MCLK) and the sub clock (SBCLK).
- Implements a CR oscillation circuit for 100 kHz WDT1 clock. See "Chapter: RTC/WDT1 Calibration" for configuration (calibration) of this oscillation circuit.
- Generates the clock for CAN prescaler. Use the PLL clock (PLLCLK) [non spread spectrum clock] when using a PLL, otherwise use the on-chip bus clock (HCLK).
- For the noise decrement, the SSCG clock [spread spectrum clock] can be selected as CPU and a clock of the resource.

5.3 Configuration

This section explains the configuration of the clock.

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

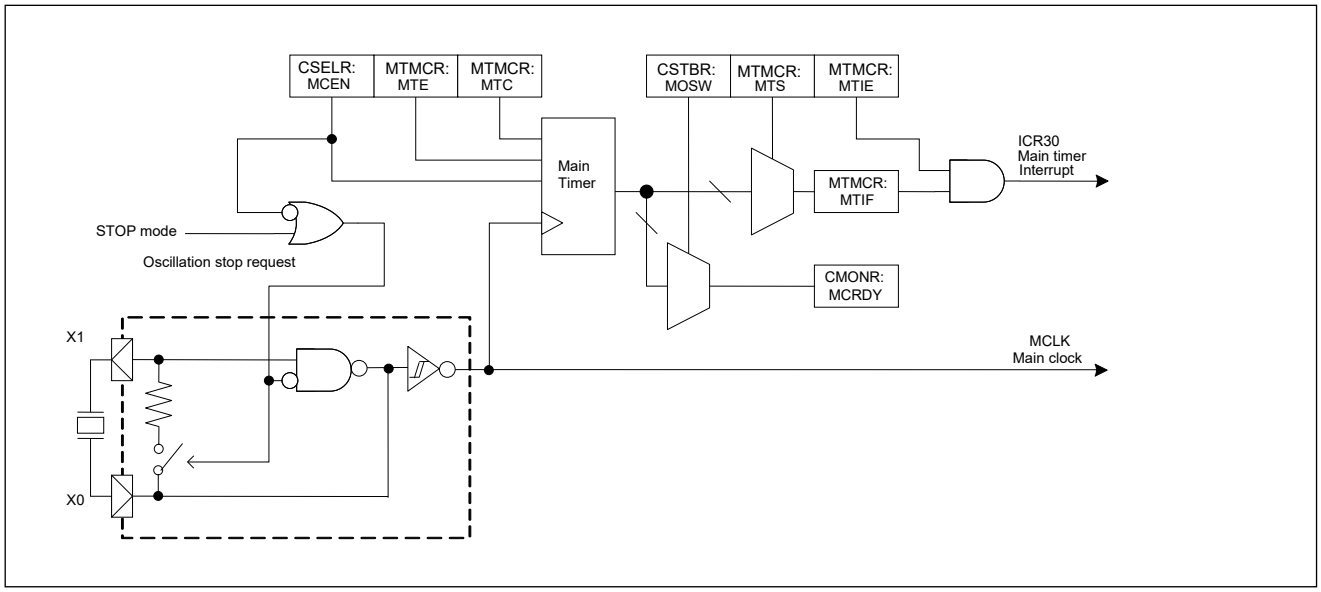


Figure 5-3. Connection Diagram of Clock (1)-2 Sub Clock Generation Unit

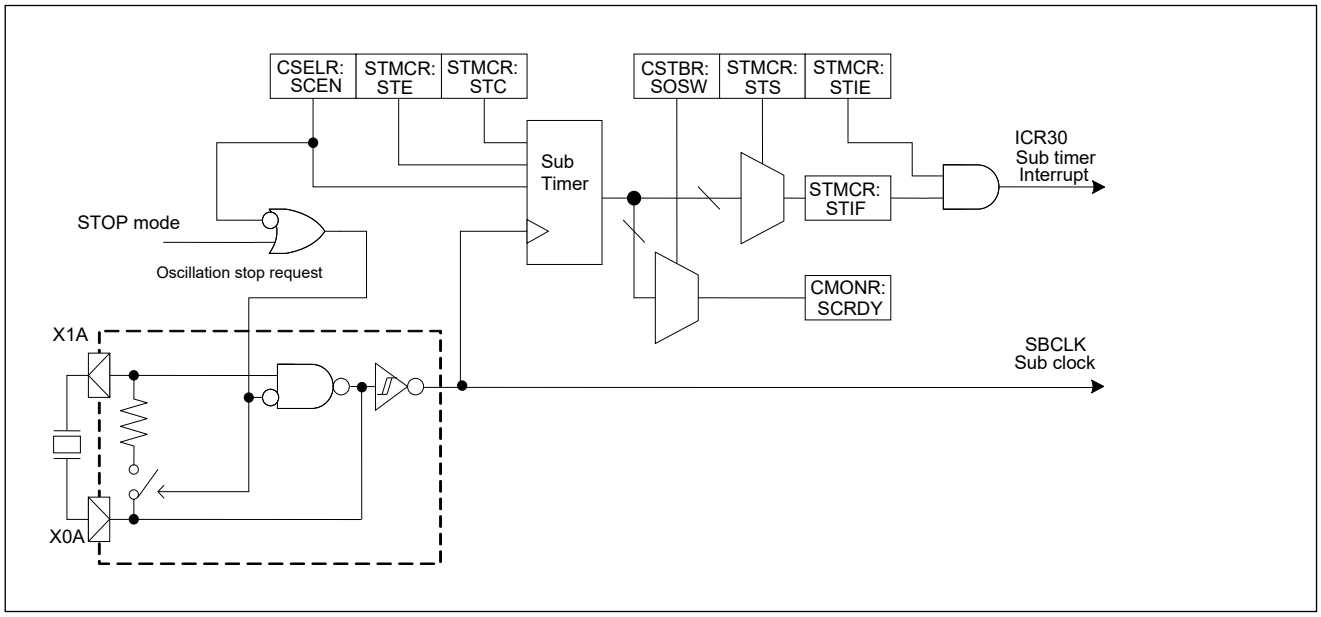


Figure 5-4. Connection Diagram of Clock (1)-3 PLL Clock Generation Unit

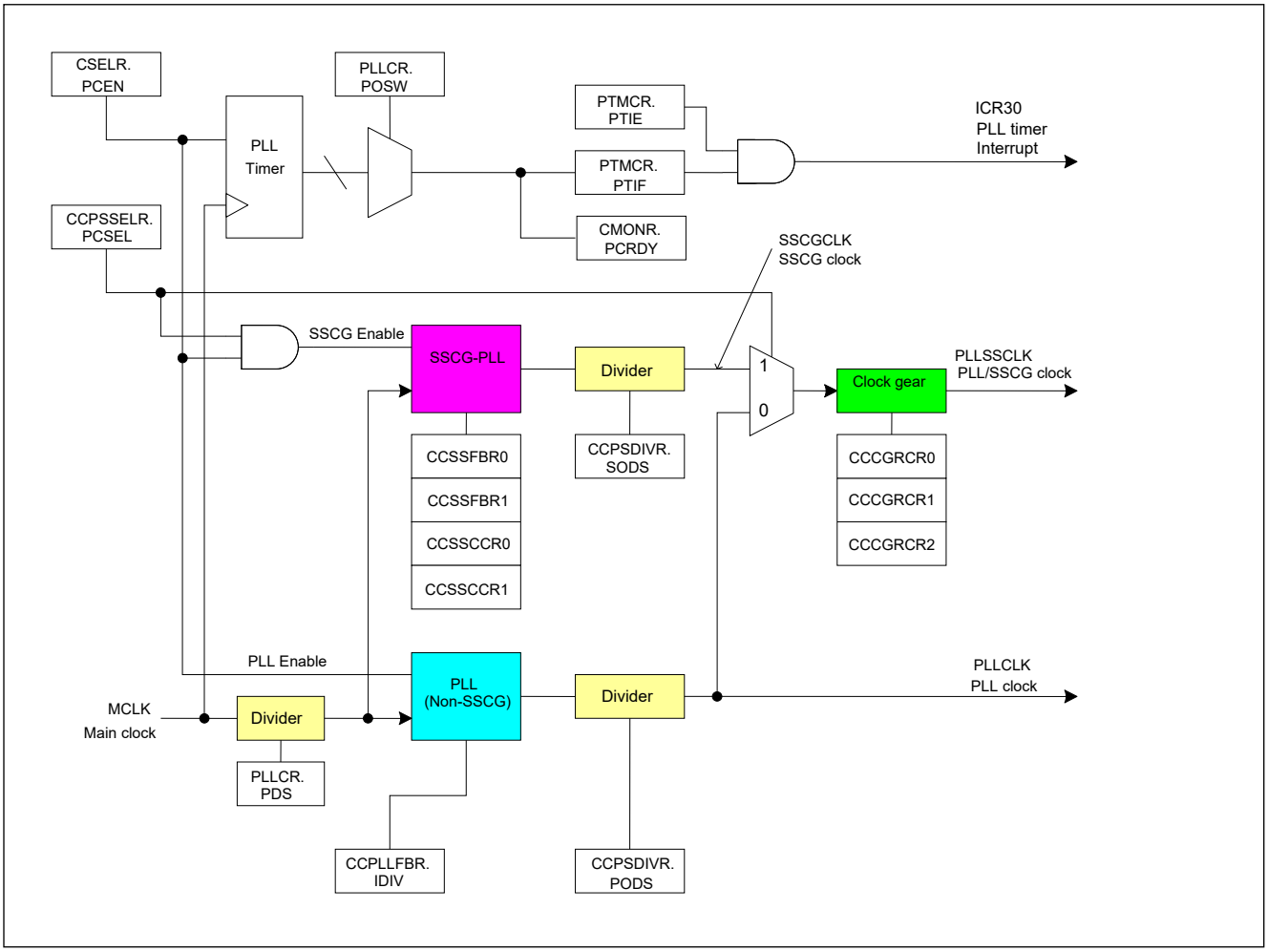


Figure 5-5. Connection Diagram of Clock (2) Source Clock Selection Unit

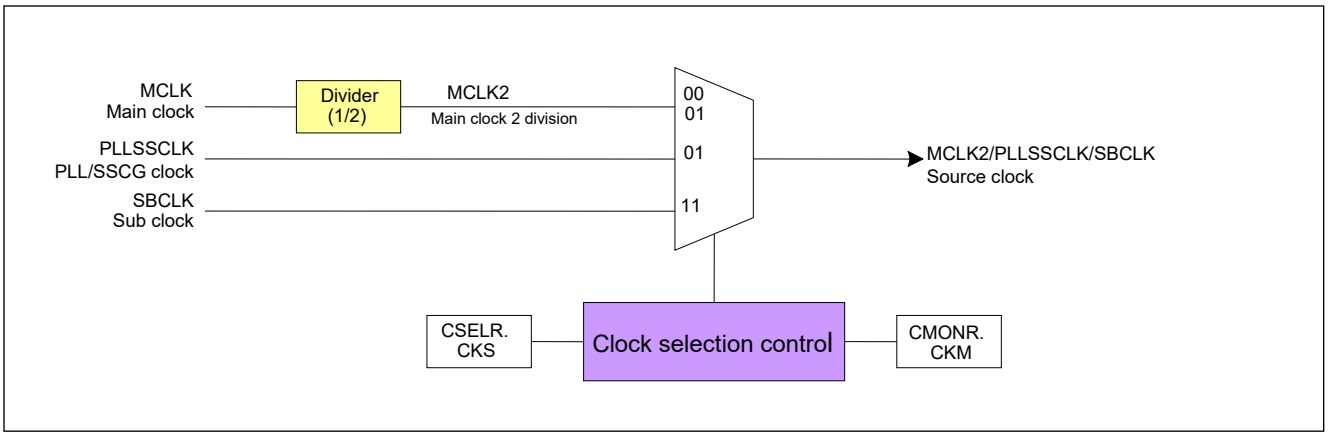


Figure 5-6. Connection Diagram of Clock (3) Divider Control

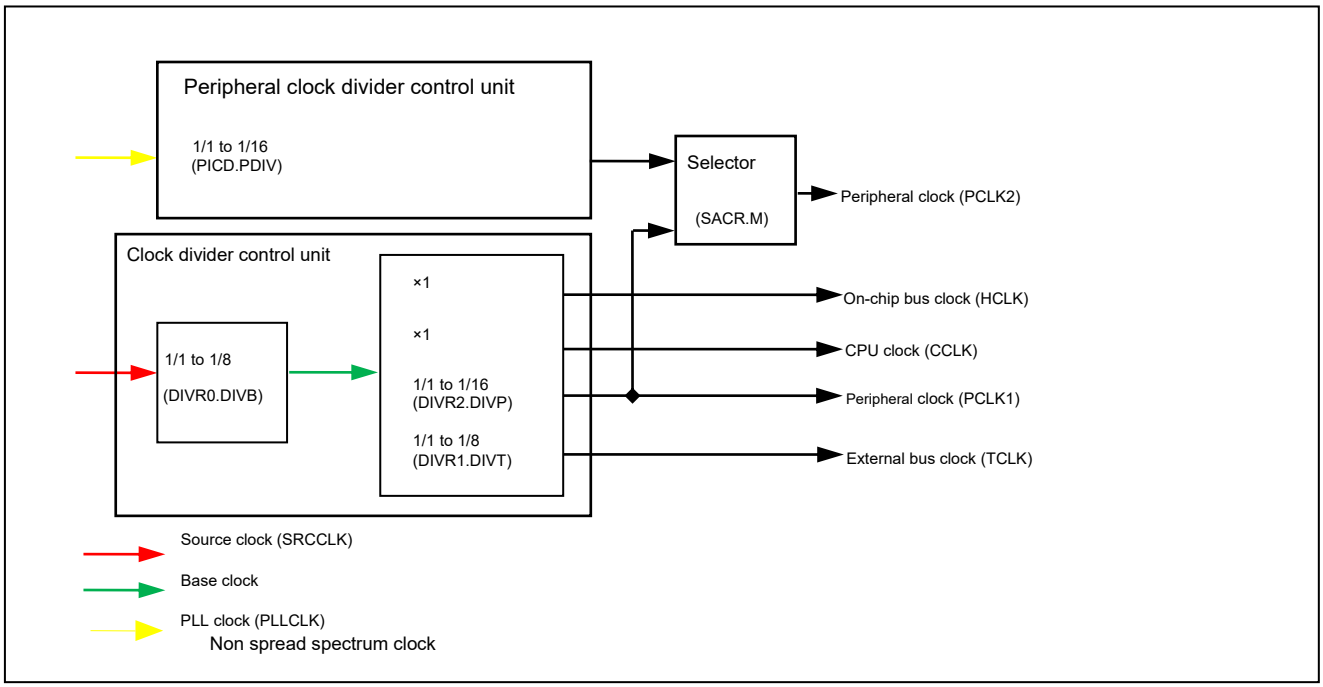


Figure 5-7. Connection Diagram of Clock (4) CAN Prescaler Clock Generation

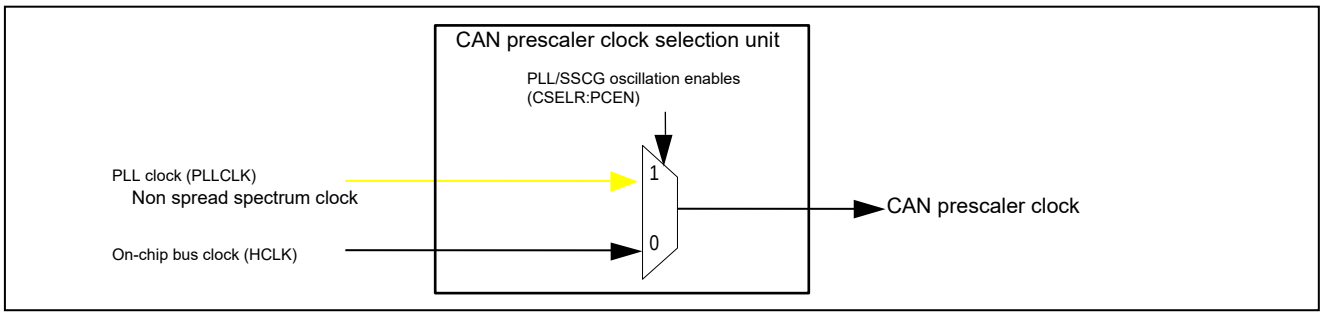


Figure 5-8. Connection Diagram of Clock (6) Watch/Power Management Clock Generation

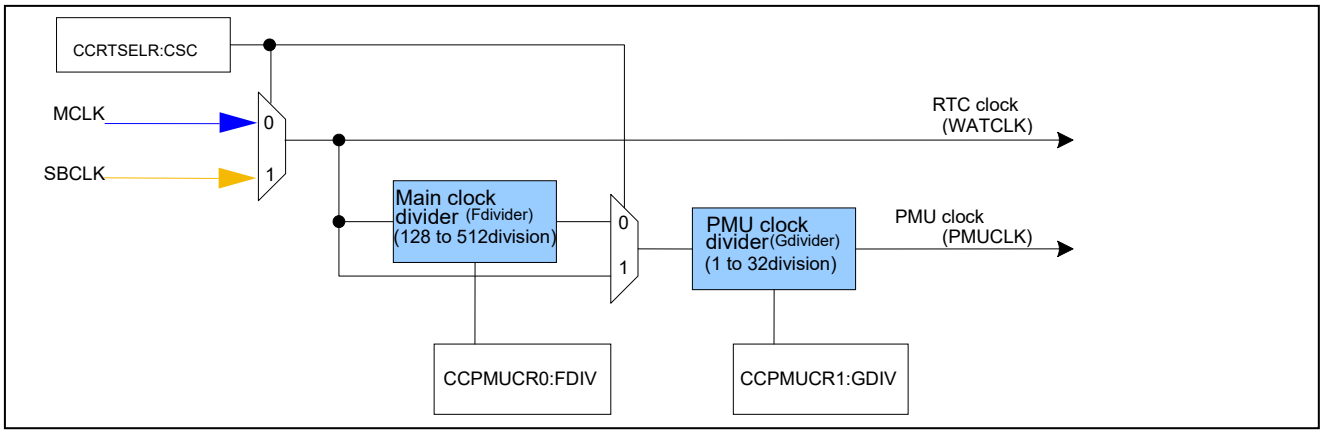
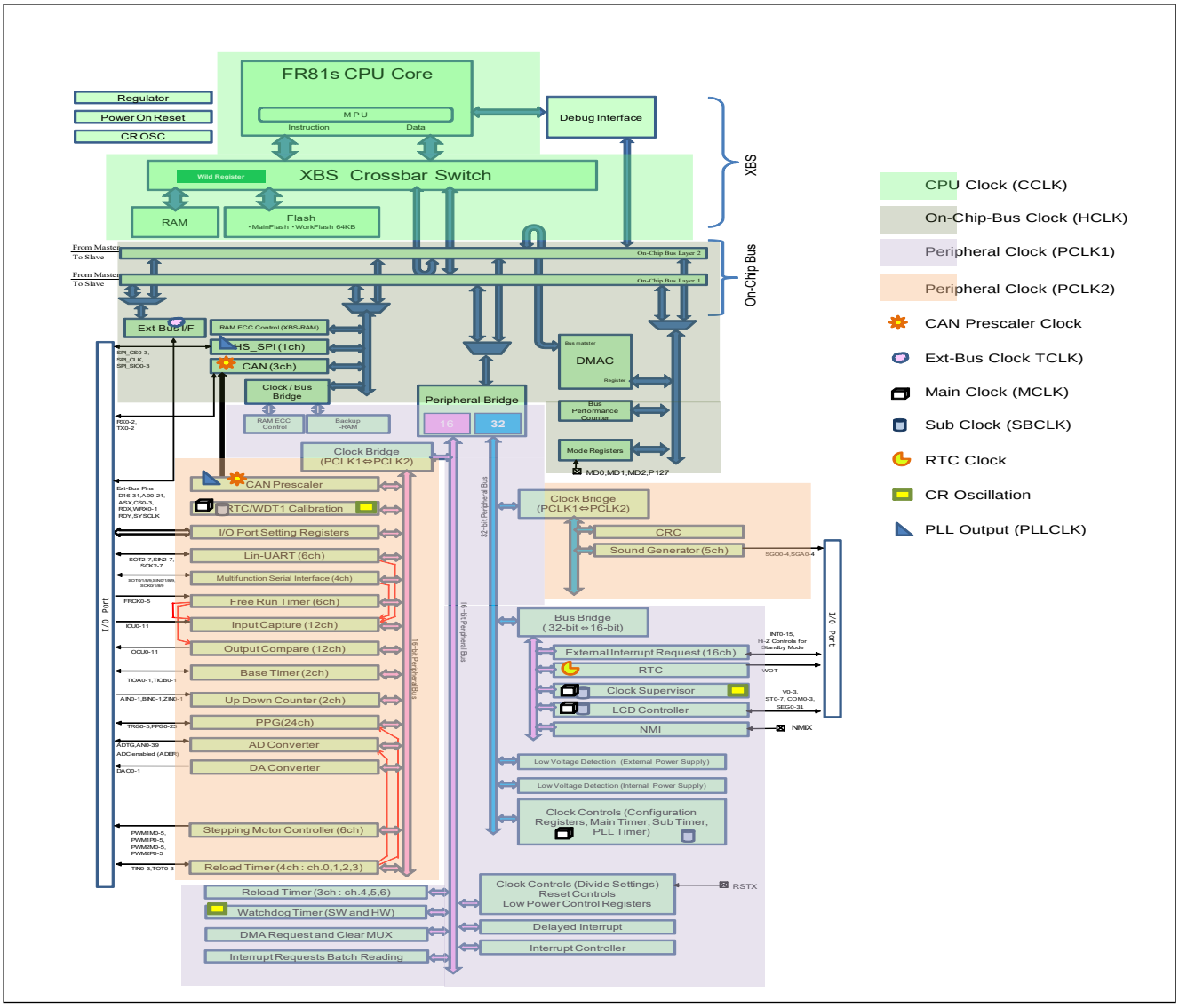


Figure 5-9. Diagram of the Clock System



5.4 Register

This section explains registers of the clock.

Table 5-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0488	DIVR0	DIVR1	DIVR2	Reserved	Division Configuration Register 0 Division Configuration Register 1 Division Configuration Register 2
0x0510	CSELR	CMONR	MTMCR	STMCR	Clock Source Configuration Register Clock Source Monitor Register Main Timer Control Register Sub Timer Control Register
0x0514	PLLCR		CSTBR	PTMCR	PLL Setting Register Oscillation Stabilization Wait Setting Register PLL clock Oscillation Stabilization Wait Timer Control Register
0x0520	CCPSSELR	Reserved	Reserved	CCPSDIVR	PLL/SSCG Clock Selection Register PLL/SSCG Output Clock Division Setting Register
0x0524	Reserved	CCPLLFBR	CCSSFBR0	CCSSFBR1	PLL Feedback Division Setting register SSCG Feedback Division Setting register 0 SSCG Feedback Division 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	CCRTSELR	Reserved	CCPMUCR0	CCPMUCR1	RTC/PMU Clock Selection Register PMU Clock Division Register 0 PMU Clock Division Register 1
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 Division Configuration Register 0: DIVR0 (Division clock configuration Register 0)

The bit configuration of the division configuration 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] (Division ratio of Baseclock): Base clock division setting

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

DIVB[2:0]	Division Ratio
000	Do not 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 Division Configuration Register 1: DIVR1 (Division clock configuration Register 1)

The bit configuration of the division configuration register 1 is shown.

This register controls division of clocks.

DIVR1: Address 0489H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TSTP	DIVT[2:0]			Reserved			
Initial value	0	0	0	1	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7] TSTP (TCLK Stop): External bus clock stop enable

This bit configures whether to stop the external bus clock (TCLK) when going into sleep mode.

TSTP	TCLK in Sleep Mode
0	Do not stop (Initial value)
1	Stop

[bit6 to bit4] DIVT[2:0] (Divide ratio of TCLK): External bus clock division setting

These bits configure the division ratio when generating the external bus clock (TCLK) from the base clock.

DIVT[2:0]	Base Clock → TCLK Division Ratio
000	Do not divide
001	2 division (Initial value)
010	3 division
011	4 division
100	5 division
101	6 division
110	7 division
111	8 division

Note:

Set this register so that the external bus clock (TCLK) definitely becomes 40 MHz or less.

[bit3 to bit0] (Reserved)

5.4.3 Division Configuration Register 2: DIVR2 (Division clock configuration Register 2)

The bit configuration of the division configuration 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] (Division ratio of PCLK): Peripheral clock division setting

These bits configure the division ratio when generating the peripheral clock (PCLK1) from the base clock.

DIVP[3:0]	Base Clock → PCLK1 Division Ratio
0000	Do not 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 (PCLK1) to be sure to become 40 MHz or less.

[bit3 to bit0] (Reserved)

5.4.4 Clock Source Selection Register: CSELR

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

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

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.

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

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

[1]: This bit is initialized to “0”. But this bit is not initialized by the return from the watch mode (power-shutdown).

[bit7] SCEN (Sub Clock Enable): Sub clock oscillation enable

This bit controls an oscillation circuit for sub clock (SBCLK) as follows.

SCEN	Oscillation Control for Sub Clock
0	Stop oscillation (Initial value)
1	Oscillate

This bit cannot be rewritten when a sub clock (SBCLK) is selected as the source clock (SRCCLK).
The oscillation circuit for sub clock always stops in stop mode regardless of the value of this bit.
The sub timer is cleared when this bit is set to "0".
For a single clock product, this bit always reads "0" and therefore a write operation would not be affected.

Note:

The SCEN bit is not initialized by the return from the standby watch mode (power-shutdown). Moreover, any reset factors other than those, caused by power on reset/internal low-voltage reset/RSTX-NMIX simultaneous assertion, can not be accepted because an internal reset signal is generated while returning from the standby watch mode (power-shutdown). At this time the SCEN bit is not initialized. Initialize this bit in case of need, when the reset signal comes from RSTX terminal input or external low-voltage detection is flagged after the return from power-shutdown.

[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 (SRCCLK). Also, this bit cannot be rewritten when the main clock oscillation is stopped or during the main clock 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 (SRCCLK).
The oscillation circuit for main clock always stops in stop mode regardless of the value of this bit.
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[1:0]	Source Selection
00	Division of the main clock (MCLK) by 2(Initial value)
01	Division of the main clock (MCLK) by 2
10	PLL/SSCG clock (PLLSSCLK)
11	Sub clock (SBCLK)

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), this bit cannot also be rewritten. A direct switch from PLL/SSCG clock (PLLSSCLK) to the sub clock (SBCLK) or vice versa cannot be performed.

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	SCRDY=1	
10	00	MCRDY=1	01,11
	10	PCRDY=1	
11	01	MCRDY=1	00,10
	11	SCRDY=1	

Do not write the values which cannot be rewritten.

5.4.5 Clock Source Monitor Register: CMONR

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

This register displays a status and a source clock (SRCCLK) for each clock source.

You can confirm that the value set at CSELR is really reflected in the actual status by reading this register.

Note:

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

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

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

[1]: This bit is initialized to “0”. But this bit is not initialized by the return from the watch mode (power-shutdown).

[bit7] SCRDY (Sub Clock Ready)

This bit shows the sub clock (SBCLK) status as follows.

SCRDY	Sub Clock (SBCLK) 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.

This bit cannot select a sub clock (SBCLK) as the source clock (SRCCLK) when this bit is set to "0".

Note:

SCRDY=1 may be read immediately after changing SCEN = 1 to 0.

[bit6] PCRDY (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.

This bit cannot select a PLL/SSCG clock (PLLSSCLK) as the source clock (SRCCLK) 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)

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.

This bit cannot select a main clock (MCLK) or a PLL/SSCG clock (PLLSSCLK) as the source clock (SRCCLK) 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	Sub clock (SBCLK)

5.4.6 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 (MTIF) 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. The MTIF bit is not set during return from the standby mode (shut-down) because the internal reset is generated.

[bit6] MTIE (Main Clock Timer Interrupt Enable): Main timer interrupt enabled

This bit controls interrupts by main timer overflow as follows.

MTIE	Main Timer Interrupt
0	Interrupt disabled (Initial value)
1	Interrupt enabled (outputs the interrupt request at the time when the MTIF bit is "1")

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

This bit clears the main timer.

MTC	Write
0	Does nothing.
1	Clear the main timer.

MTC	Read
0	Operating normally
1	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 selection): Main timer interval selection

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

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

The MTS3 bit always reads "1".
Change MTS[3:0] at the time when the main timer stops (MTE=0).

5.4.7 Sub Timer Control Register: STMCR (Sub Clock Timer Control Register)

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

This register controls the sub timer which runs with the sub clock.

STMCR: Address 0513$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	STIF	STIE	STC	STE	Reserved	STS[2:0]		
Initial value	0	0	0	0	0	1	1	1
Attribute	R(RM1),W	R/W	R(RM0),W	R/W	R0,WX	R/W	R/W	R/W

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

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

When the operation of the sub timer is not allowed (STE=0), the sub timer stops except that it is waiting for a sub clock oscillation stabilization. The write operation to this register becomes enabled only when SCRDY=1 except for STIE. Thus a sub timer clear executed by STC=1 in sub clock oscillation stabilization wait status (SCEN=1 and SCRDY=0) is not effective.

When the sub timer stops (STE=0) it will be cleared and while being cleared STC=1 will be read out. At that time the sub timer interrupt flag is not set. The sub timer overflow period (STS[2:0]) should be changed at the time when the sub timer stops (STE=0).

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

[bit7] STIF (Sub Clock Timer Interrupt Flag): Sub timer interrupt flag

The flag to indicate that an overflow happens in the interval for which the sub timer has selected.
When the STIE bit is "1" and this bit is set, a sub timer interrupt request is generated.

Clear Factor	■ "0" write ■ A DMA transfer is generated by the sub timer interrupt.
Set Factor	■ An overflow occurred in the interval set by STS[2:0]. ■ The end of oscillation stabilization wait time of the sub clock after setting SCEN=0 to 1. ■ The ends of oscillation stabilization wait time of the sub clock after exiting the stop mode.

Writing "1" to this bit is ineffective.
When the STIE 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. The STIF bit is not set during return from the standby mode (shut-down) because the internal reset is generated.

[bit6] STIE (Sub Clock Timer Interrupt Enable): Sub timer interrupt enable

This bit controls interrupts by sub timer overflow as follows.

STIE	Sub Timer Interrupt
0	Interrupt disabled (Initial value)
1	Interrupt enabled (output the interrupt request at the time STIF bit is "1")

[bit5] STC (Sub clock Timer Clear): Sub timer clear

This bit clears the sub timer.

STC	Write
0	Does nothing.
1	Clear the sub timer.

STC	Read
0	Operating normally
1	Clearing the sub timer

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

[bit4] STE (Sub Clock Timer Enable): Sub timer operation enabled

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

STE	Sub Timer Operation
0	Operation disabled (Initial value)
1	Operation enabled

At the time of STC=1, STE=1 write is prohibited.

[bit3] (Reserved)

[bit2 to bit0] STS[2:0] (Sub Clock Timer interval selection): Sub timer interval selection

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

STS[2:0]	Sub Timer Overflow Interval	At 32kHz
000	2^8× sub clock cycle	8[ms]
001	2^9× sub clock cycle	16[ms]
010	2^{10}× sub clock cycle	32[ms]
011	2^{11}× sub clock cycle	64[ms]
100	2^{12}× sub clock cycle	128[ms]
101	2^{13}× sub clock cycle	0.256[s]
110	2^{14}× sub clock cycle	0.512[s]
111	2^{15}× sub clock cycle (Initial value)	1.024[s]

5.4.8 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 0514H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved		Reserved	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 parameters 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 will be disabled.

[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 clock 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 4MHz	At 8MHz
1000	2^9 × main clock cycle	128.0[µs]	64.0[µs]
1001	2^{10} × main clock cycle	256.0[µs]	128.0[µs]
1010	2^{11} × main clock cycle	512.0[µs]	256.0[µs]
1011	2^{12} × main clock cycle	1024.0[µs]	512.0[µs]
1100	2^{13} × main clock cycle	2048.0[µs]	1024.0[µs]
1101	2^{14} × main clock cycle	4096.0[µs]	2048.0[µs]
1110	2^{15} × main clock cycle	8192.0[µs]	4096.0[µs]
1111	2^{16} × main clock cycle (Initial value)	16384.0[µs]	8192.0[µs]

POSW3 always reads "1".

Note:

The PLL/SSCG clock oscillation stabilization wait time specification in this model 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 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 Select
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.
See "5.5.1.3 PLL/SSCG Clock (PLLSSCLK)" for configuration samples.

A set value is limited. See "5.5.1.4 Limitations When PLL/SSCG Clock is Used" when you set it.

5.4.9 Clock Stabilization Selection Register: CSTBR

The bit configuration of the oscillation stabilization selection 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). Write operations to SOSW[2:0] will not be effective at the sub clock oscillation stabilization wait time (SCEN=1 and SCRDY=0).

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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	SOSW[2:0]			MOSW[3:0]			
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 bit4] : SOSW[2:0] (Sub clock Osc Wait): Sub clock oscillation stabilization wait selection

These bits select the oscillation stabilization wait time for the sub clock (SBCLK) as follows.

SOSW[2:0]	Sub Clock Oscillation Stabilization Wait Time	At 32kHz
000	2^8 × sub clock cycle (Initial value)	8[ms]
001	2^9 × sub clock cycle	16[ms]
010	2^{10} × sub clock cycle	32[ms]
011	2^{11} × sub clock cycle	64[ms]
100	2^{12} × sub clock cycle	128[ms]
101	2^{13} × sub clock cycle	0.256[s]
110	2^{14} × sub clock cycle	0.512[s]
111	2^{15} × sub clock cycle	1.024[s]

[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} ×main clock cycle (Initial value)	8[ms]
0001	2^{1} ×main clock cycle	500[ns]
0010	2^{5} ×main clock cycle	8[µs]
0011	2^{6} ×main clock cycle	16[µs]
0100	2^{7} ×main clock cycle	32[µs]
0101	2^{8} ×main clock cycle	64[µs]
0110	2^{9} ×main clock cycle	128[µs]
0111	2^{10} ×main clock cycle	256[µs]
1000	2^{11} ×main clock cycle	512[µs]
1001	2^{12} ×main clock cycle	1[ms]
1010	2^{13} ×main clock cycle	2[ms]
1011	2^{14} ×main clock cycle	4[ms]
1100	2^{17} ×main clock cycle	33[ms]
1101	2^{19} ×main clock cycle	131[ms]
1110	2^{21} ×main clock cycle	524[ms]
1111	2^{23} ×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.

5.4.10 PLL Clock Oscillation Timer Control Register: PTMCR

The bit configuration of the PLL clock Oscillation timer control register is shown.

The timer that works with the main clock that does PLL/SSCG clock oscillation stabilization wait is controlled.
The PLL/SSCG clock oscillation stabilization wait timer is used only at the oscillation stabilization wait time of the PLL/SSCG clock (PLLSSCLK).
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 clock oscillation stabilization wait timer interrupt flag

It is a flag that shows that the overflow at the time set by PLL clock 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→1

The "1" writing in this bit is invalid.

When the PTIE bit is "0", the clearness of this bit by the DMA forwarding is not done.

In the read modify write instruction, "1" is read.

The set factor is given priority when a set factor and a clear factor are generated at the same time.

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

The interrupt by the overflow of PLL/SSCG clock oscillation stabilization wait timer is controlled 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)

5.4.11 PLL/SSCG Clock Selection Register: CCPSSELR (CCTL PLL/SSCG clock Selection Register)

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

It is a register that selects either PLL or SSCG to be used. 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 Selection): PLL/SSCG Clock source selection

It selects the PLL/SSCG clock source.

PCSEL	PLL or SSCG
0	Selects PLL
1	Selects 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.

5.4.12 PLL/SSCG Output Clock Division Setting Register: CCPSDIVR (CCTL PLL/SSCG Clock Division Register)

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

It is a register that sets the ratio of dividing frequency 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[2:0] (PLL Oscillator Divider selection): Selection of PLL macro oscillation clock dividing frequency ratio

The ratio of dividing frequency of the PLL clock is set.

PODS[2:0]	Dividing Frequency 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:

It is only dividing of the even number in the setting by this bit. The odd number dividing frequency cannot be set. Duty of the output clock becomes 50%.
Please set for the PLL clock to become 80 MHz or less. (The operation guarantee that exceeds 80 MHz is not done.)

[bit3] (Reserved)

[bit2 to bit0] SODS[2:0] (SSCG Oscillator Divider selection): SSCG Selection of SSCG macro oscillation clock dividing frequency Ratio

The ratio of dividing frequency of the SSCG clock is set.

SODS[2:0]	Dividing Frequency 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:

It is only dividing of the even number in the setting by this bit. The odd number dividing frequency cannot be set. Duty of the output clock becomes 50%.
Please set for the SSCG clock to become 80 MHz or less. (The operation guarantee that exceeds 80 MHz is not done.)

A set value is limited. See "5.5.1.4 Limitations When PLL/SSCG Clock is Used" when you set it.

5.4.13 PLL Feedback Division Setting Register: CCPLLFBR (CCTL PLL FB clock division Register)

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

It is a register that sets the multiple ratio 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): Setting of PLL macro FB input dividing frequency ratio

PLL multiple ratio is set.

IDIV[6:0]	Dividing Frequency 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 "5.5.1.4 Limitations When PLL/SSCG Clock is Used" when you set it.

5.4.14 SSCG Feedback Division Setting Register 0: CCSSFBR0 (CCTL SSCG FB clock division Register 0)

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

It is a register that sets multiple ratio of SSCG.

The multiple ratio of SSCG becomes P × 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 dividing frequency ratio setting

It sets the SSCG multiple ratio N.

NDIV[5:0]	Dividing Frequency Ratio Setting
000000	Setting is prohibited
000001	2
000010	3
000011	4
…	……
111101	62
111110	63
111111	Setting is prohibited

A set value is limited. See "5.5.1.4 Limitations When PLL/SSCG Clock is Used" when you set it.

5.4.15 SSCG Feedback Division Setting Register 1: CCSSFBR1 (CCTL SSCG FB clock division Register 1)

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

It is a register that sets the multiple ratio P of SSCG. The multiplication ratio of SSCG becomes P × 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 divider frequency ratio setting

It sets the SSCG multiple ratio P.

PDIV[4:0]	Dividing Frequency Ratio Setting
00000	1
00001	2
00010	3
00011	4
…	……
11101	30
11110	31
11111	Setting is prohibited

A set value is limited. See "5.5.1.4 Limitations When PLL/SSCG Clock is Used" when you set it.

5.4.16 SSCG Configuration Setting Register 0: CCSSCCR0 (CCTL SSCG Config. Register 0)

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

SSCG is variously set.

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 bit4] (Reserved)

[bit3, bit2] SFREQ[1:0] (Spread Spectrum Modulation Frequency Settings)

The spread spectrum modulation frequency of SSCG is set.

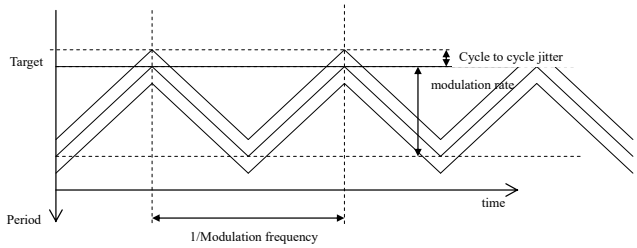
SFREQ[1:0]	Modulation Frequency
00	1/1024
01	1/2048
1x	1/4096

[bit1] SMODE (Spread Spectrum Modulation Mode settings)

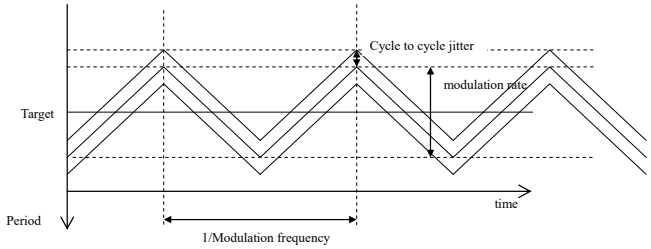
Sets spread spectrum modulation mode of SSCG.

SMODE	Modulation Mode
0	Down Spread
1	Center Spread

- Down Spread



- Center Spread



[bit0] SSEN (Spread Spectrum Enable)

This bit enables spread spectrum of SSCG.

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

Note:

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

5.4.17 SSCG Configuration Setting Register 1: CCSSCCR1 (CCTL SSCG Config. Register 1)

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

Sets various settings of SSCG.

This register can be written only when PLL/SSCG clock oscillation stops. (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)

Sets 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 has no effect.

[bit9 to bit0] (Reserved)

Always write "0" to these bits.

5.4.18 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.

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)

Displays 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)

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[1:0]=10 (PLL/SSCG clock (PLLSSCLK) selection) and CCCGRCR0:GREN=1 (clock gear enabled).
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).
In the instruction of read modify write "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)

This bit enables the operation of clock gear.

GREN	Operation
0	No use of clock gear
1	Use of clock gear

Note:

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

5.4.19 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.

Sets various settings of clock gear.

This register can be written only when PLL/SSCG clock oscillation is stopped (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 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 selection)

These bits select the step at the start of clock gear operation and select the step 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.

5.4.20 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.

Sets various settings of clock gear.

This register can be written only when PLL/SSCG clock oscillation is stopped. (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 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

5.4.21 RTC/PMU Clock Selection Register: CCRTSELR (CCTL RTC PMU clock Selection Register)

The bit configuration of the RTC/PMU clock selection register is shown.

Selects RTC/PMU clock source.

CCRTSELR: Address 0530H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CST	Reserved						CSC
Initial value	[1]	0	0	0	0	0	0	[1]
Attribute	R,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R/W

[1]: These bits are initialized to “0”. But these bits are not initialized by the return from the watch mode (power-shutdown).

[bit7] CST (Clock Source Selection Status Monitor)

A time lag by clock switch occurs until the CSC bit is written and then the clock switch completes. Whether the switch completes or not is monitored by this bit.

CST	Monitor
0	The completion of clock switch
1	During clock switch

Note:

When single clock products (SUBDIS=1), this bit is always fixed with "0". Normally, switch completes by main clock × about 3 cycles + sub clock × about 3 cycles.

[bit6 to bit1] (Reserved)

[bit0] CSC (Clock Source Selection)

Selects clock source of RTC/PMU clock.

CSC	Clock Source
0	Main oscillation clock
1	Sub oscillation clock

Note:

- The CSC bit can be rewritten only when SCRDY=1 and MCRDY=1. When single clock products, this bit is always fixed with "0" in spite of the written value.
- It takes main clock × about 3 cycles + sub clock × about 3 cycles until the switch operation of RTC and PMU clock completes after rewriting the CSC bit. When main clock and sub clock oscillation are stopped during the switching operation, the switching operation does not complete correctly. The oscillation must always be stooped in the status that the CST bit is "0" (the status of the completion of switching).

  The CSC bit is not initialized by the return from the standby watch mode (power-shutdown). Moreover, any reset factors other than those, caused by power on reset/internal low-voltage reset/RSTX-NMIX simultaneous assertion, cannot be accepted because an internal reset signal is generated while returning from the standby watch mode (power-shutdown). At this time the CSC bit is not initialized. Initialize this bit in case of need, when the reset signal comes from RSTX terminal input or external low-voltage detection is flagged after the return from power-shutdown.

5.4.22 PMU Clock Division Setting Register 0: CCPMUCR0 (CCTL PMU Clock division Register 0)

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

This register does the setting of dividing frequency of the PMU clock.

CCPMUCR0: Address 0532_H (Access: Byte, Half-word, Word)

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

[bit7] FST (F-divider Status Monitor)

A time lag by clock switch occurs until FDIV[1:0] register is written and the written value is reflected. Whether the setting value is reflected can be monitored by this bit.

Normally, it takes RTC clock × about 4 cycles + PCLK1 × about 4 cycles to reflect the setting value of the register.

FST	Monitor
0	Completion of reflecting the written value
1	During reflecting the written value

[bit6 to bit2] (Reserved)

[bit1 to bit0] FDIV[1:0] (F-divide Ratio Setting)

Sets the division rate of F-divider. The clock less than 32 kHz must be provided with PMU. When CCRTSELR:CSC=0 (selection of main oscillation clock), this bit is set to be less than 32kHz by F divider.

FDIV[1:0]	Division Rate	Target Main Oscillation Frequency
00	Divided by 128 (Initial value)	4 MHz
01	Divided by 256	8 MHz
10	Divided by 384	12 MHz
11	Divided by 512	16 MHz

Note:

Writing to this bit is ignored while the CCPMUCR0:FST bit is "1".
When CCRTSELR:CSC=1 (selection of sub oscillation clock), F-division rate become undivided in spite of the value of this bit.

5.4.23 PMU Clock Division Setting Register 1: CCPMUCR1 (CCTL PMU Clock division Register 1)

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

This register does the setting of dividing frequency of the PMU clock.

CCPMUCR1: Address 0533$_H$ (Access: Byte, Half-word, Word)

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

[bit7] GST (G-divider Status monitor)

A time lag by clock switch occurs until GDIV[4:0] register is written and the written value is reflected. Whether the setting value is reflected can be monitored by this bit.

Normally, it takes RTC clock × about 4 cycles + PCLK1 × about 4 cycles to reflect the setting value of the register.

GST	Monitor
0	Completion of reflecting the written value
1	During reflecting the written value

Note:

Writing to CCPMUCR1:GDIV[4:0] is ignored while this bit is "1".

[bit6, bit5] (Reserved)

[bit4 to bit0] GDIV[4:0] (G-divide ratio setting)

These bits set the division rate of G-divider. The period of the PMU clock must be more than four times the period of the bus clock (APB) which is provided with PMU. The division rate of the PMU clock is set by this divider to meet the above relation.

GDIV[4:0]	Division Rate
00000	Do not divide (Initial value)
00001	2
00010	3
…	……
11101	30
11110	31
11111	32

Note:

Writing to this bit is ignored while CCPMUCR1:GST bit is "1".

5.4.24 Sync/Async Control Register: SACR

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

Selects the peripheral clock (PCLK2).

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 (PCLK2)

The peripheral clock(PCLK2) is switched 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)

5.4.25 Peripheral Interface Clock Divider: PICD (Peripheral Interface Clock Divider)

The bit configuration of peripheral interface clock divider is shown.

The setting of dividing frequency of the peripheral clock made from the PLL clock (PLLCLK) is done.

- 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]: Sets peripheral clock division rate

The ratio of dividing frequency of the peripheral clock (PCLK2) is set 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 Rate
0000	Do not 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
5.5.8 About PMU Clock (PMUCLK)

5.5.1 Oscillation Control

This section explains oscillation control.

5.5.1.1 Main Clock (MCLK)

5.5.1.2 Sub Clock (SBCLK)

5.5.1.3 PLL/SSCG Clock (PLLSSCLK)

5.5.1.4 Limitations When PLL/SSCG Clock is Used

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
- While the sub clock (SBCLK) are selected as the source clock (SRCCLK) and "0" is set to CSELR:MCEN

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.

Note:

For the single clock products, the main clock oscillation enable is always enabled (MCEN=1).

5.5.1.2 Sub Clock (SBCLK)

The sub clock (SBCLK) is shown.

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

- After the occurrence of reset (the bus idle wait time before stop is required. See "Chapter: Reset").
- During the stop mode.
- While a clock other than the sub clock (SBCLK) are selected as the source clock (SRCCLK) and "0" is set to CSELR:SCEN.
- When the clock is used as a port because the clock is used for sub oscillation and port (metal option).

After all the above conditions of the oscillation stop are cancelled and then the oscillation stabilization wait time which is set to CSTBR:SOSW[2:0] goes by, supplying the clock starts. The sub clock oscillation stops until "1" is set to because CSELR:SCEN is initialized to "0" at the time of return from the reset input or the INIT status.

Notes:

- For the single clock products, the sub clock oscillation enable is always disabled (SCEN=0).
- For the single clock product, the sub timer cannot be used.

5.5.1.3 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 5-2. 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 SSCG and the peripherals are operated by PLL, because the asynchronization transfer enters between CPU/ Peripheral, the penalty of 5 ×PCLK2 to 8×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/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 (SRCCLK) 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)

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

  SSCG macro oscillation clock frequency=(PLL/SSCG input clock frequency)×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 5-10. PLL Peripheral Block Diagram

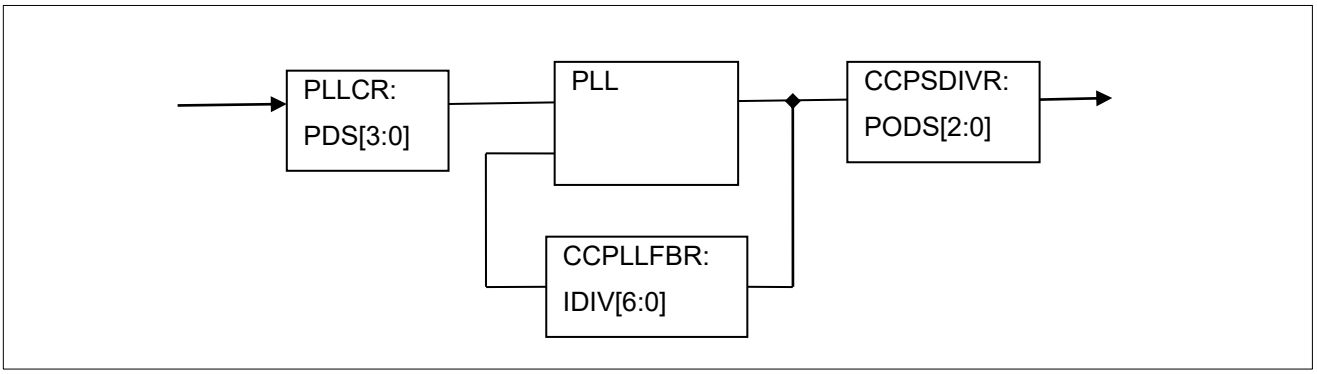
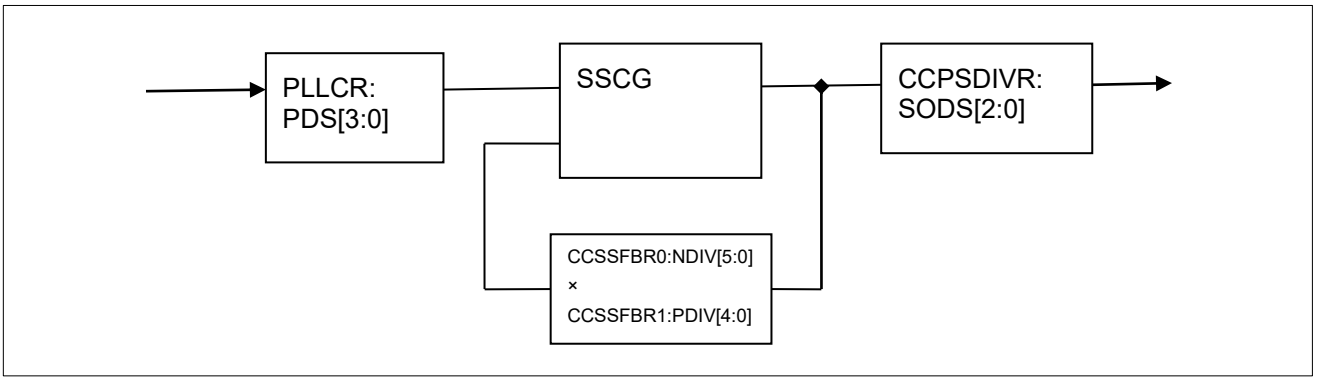


Figure 5-11. SSCG Peripheral Block Diagram



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 rages of PLL/SSCG, see the below notes.

Notes:

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 resource.
- 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:

  - ☐ 200 MHz ≤ PLL macro oscillation clock frequency ≤ 320 MHz
  - ☐ 200 MHz ≤ SSCG macro oscillation clock frequency ≤ 320 MHz (Down Spread)

5.5.1.4 Limitations When PLL/SSCG Clock is Used

The limitations of the PLL/SSCG clock are shown.

Use it according to the following limitations when you use the PLL/SSCG clock.

Clock Control PLL Clock Frequency

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

Note:

Set PLLCR or CCPSDIVR and CCPLLFBR so as not to exceed frequency (max).

Clock Control SSCG Clock Frequency

Frequency (max)	FCTLR: FAW	CCPSSELR: PCSEL	CCSSCCR0: SSEN	CCSSCCR0: SMODE	CCSSCCR1: RATESEL	Remarks
80 MHz	01	1	1	0 / 1	000 to 110	
72 MHz	00	1	1	0	000 to 110	DownSpread
72 MHz	00	1	1	1	000	CenterSpread (0.5%)
72 MHz	00	1	1	1	010	CenterSpread (1%)
72 MHz	00	1	1	1	011	CenterSpread (2%)
71 MHz	00	1	1	1	100	CenterSpread (3%)
71 MHz	00	1	1	1	101	CenterSpread (4%)
70 MHz	00	1	1	1	110	CenterSpread (5%)
80 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 for the peripheral clock so as not to exceed 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	7h	3Bh
1.00%	010	8 - 60	7h	3Bh
2.00%	011	8 - 48	7h	2Fh
3.00%	100	8 - 31	7h	1Eh
4.00%	101	8 - 23	7h	16h
5.00%	110	8 - 18	7h	11h

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 the 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 (MCLK) clock 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}× main clock period
- PLL/SSCG clock : PLLCR:POSW[3:0] bit 216× main clock period
- Sub clock : CSTBR:SOSW[2:0] bit 2^{8}× sub clock period

The main clock 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 clock 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 "5.4.8 PLL Setting Register: PLLCR (PLL Configuration Register)".

The sub clock oscillation stabilization wait time is always specified by the initial value because CSTBR:SOSW[2:0] is initialized by reset (INIT or RST). Except that case, the sub oscillation stabilization wait time can be changed by setting to CSTBR:SOSW[2:0].

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 (SRCCLK) 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.

Displays 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"
- Sub clock (SBCLK) : CMONR:SCRDY ="0" , CMONR:SCRDY ="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.

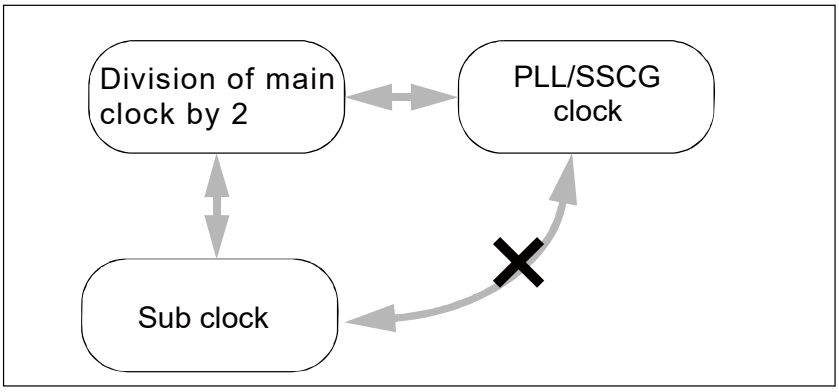
After reset (RST) the main clock (MCLK) divided by 2 is selected as the source clock (SRCCLK). After program operation the source clock can be changed by setting CSELR:CKS[1:0].

5.5.3.2 Procedure of Switching the Source Clock

The procedure of switching the source clock is shown.

The source clock (SRCCLK) cannot be directly switched from the PLL/SSCG clock (PLLSSCLK) to the sub clock (SBCLK) and from the sub clock to the PLL/SSCG clock. Switch the main clock divided by 2 once. Set the oscillation stop as necessary because the value of the oscillation enabled bit (CSELR:xCEN) is held, even though the source clock is switched.

Figure 5-12. 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 lock wait time (setting PLLCR/ CCPSSELR/ CCPSDIVR/ CCPLLFBR/ CCSSFBR0/ CCSSFBR1/ CCSSCCR0/ CCSSCCR1) --when PLL/SSCG oscillation is not enabled--

   ↓

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

   ↓

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

   ↓

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

   ↓

   PLL/SSCG oscillation begins (PCEN=0→1)

   ↓

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

   ↓

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

   ↓

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

   ↓

   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→00)

   ↓

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

3. The main clock divide by 2→sub clock

   While selecting the main clock divided by 2 as the source clock (CMONR:CKM[1:0]=01)

   ↓

   Sets the sub clock oscillation stabilization wait time (sets CSTBR:SOSW[2:0])
   –when sub oscillation is not enabled–

   ↓

   Clears the sub timer interrupt source (STIF=0)

   ↓

   (as necessary) sets sub timer interrupt enable (STIE=1)

   ↓

   The sub oscillation begins (SCEN=0→1)

   ↓

   Sub clock oscillation stabilization wait loop (loop until when SCRDY=1), or interrupt wait

   ↓

   Clears sub timer interrupt (STIF=0)

   ↓

   Switches the source clock to the sub clock (CSELR:CKS[1:0]=01→11)

   ↓

   While selecting the sub clock as the source clock (CMONR:CKM[1:0]=11)

4. The sub clock→the main clock divided by 2

   While selecting the sub clock as the source clock (CMONR:CKM[1:0]=11)

   ↓

   Sets the main clock oscillation stabilization wait time (sets CSTBR:MOSW[3:0])
   – when the main oscillation is not enabled–

   ↓

   Clears the main timer interrupt source (MTIF=0)

   ↓

   (as necessary) Sets the main timer interrupt enable (MTIE=1)

   ↓

   The main oscillation begins (MCEN=0→1)

   ↓

   The main clock oscillation stabilization wait loop (loop until when MCRDY=1), or interrupt wait

   ↓

   Clears the main timer interrupt (MTIF=0)

   ↓

   Switches the source clock to the main clock divided by 2 (CSELR:CKS[1:0]=11→01)

   ↓

   While selecting the main clock as the source clock (CMONR:CKM[1:0]=01)

Figure 5-13. Example of PLL/SSCG Mode Setting Main → PLL/SSCG

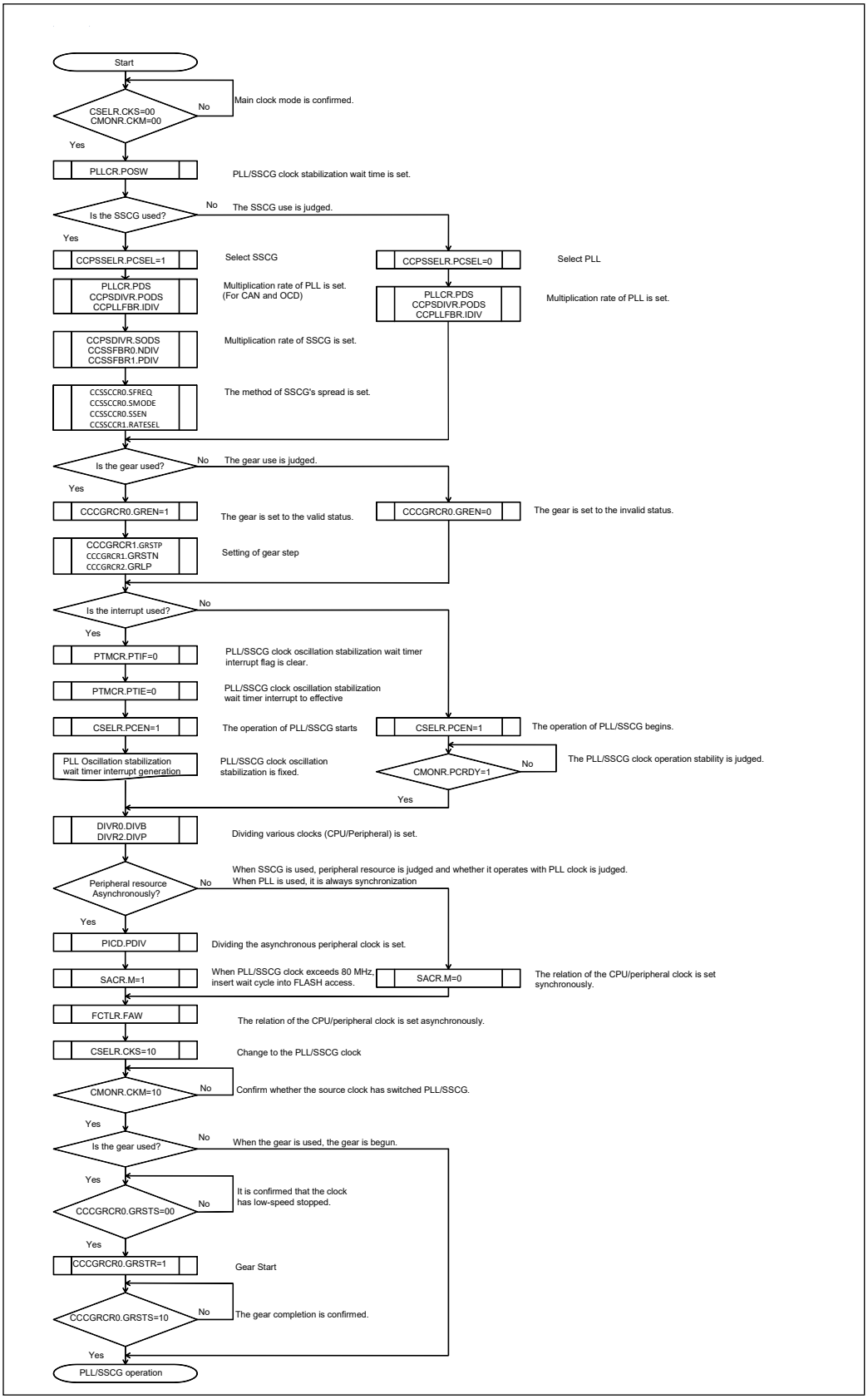
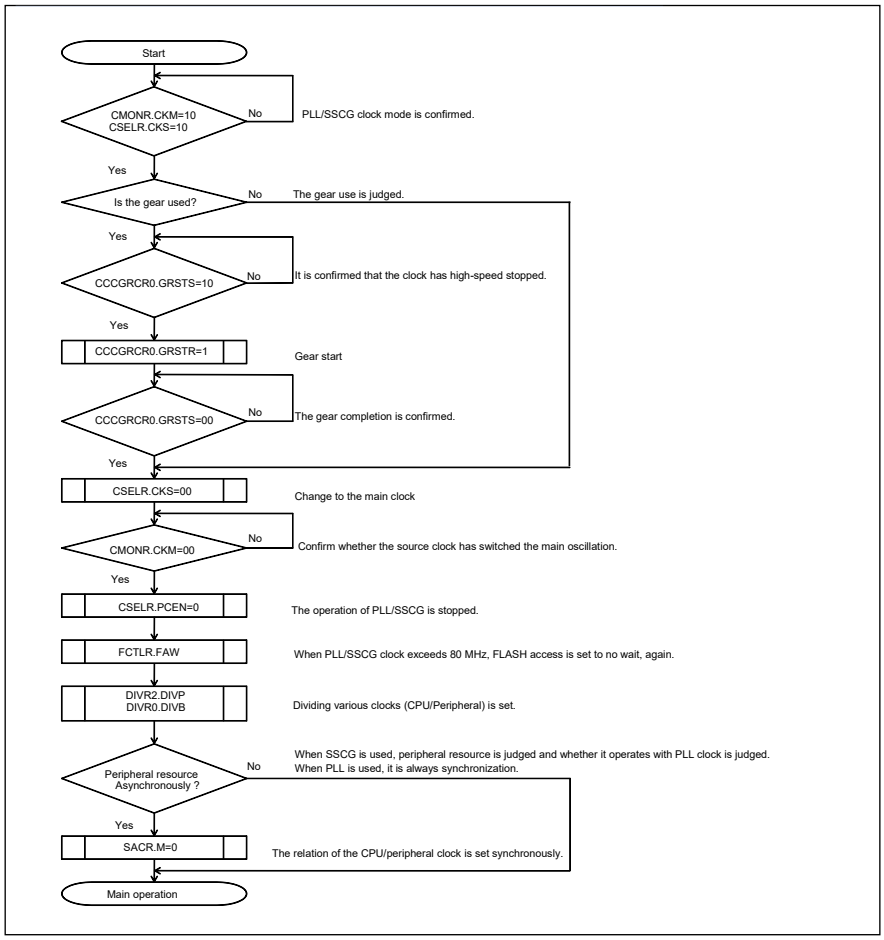


Figure 5-14. Example of PLL/SSCG Mode Setting PLL/SSCG → Main



5.5.4 Timer

The timer is shown.

5.5.4.1 Main Clock Oscillation Stabilization Wait Timer (Main Timer)
5.5.4.2 Sub Clock Oscillation Stabilization Wait Timer (Sub Timer)
5.5.4.3 PLL/SSCG Clock Oscillation Stabilization Wait Timer (PLL Timer)
5.5.4.4 Setting
5.5.4.5 Procedure for Setting the Timer Interrupt
5.5.4.6 Timer Operations
5.5.4.7 Watch Mode and Timer Interrupt

5.5.4.1 Main Clock Oscillation Stabilization Wait Timer (Main Timer)

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

The main timer is operated by the main clock (MCLK). It is used for the main clock stabilization time counter. When main clock is stabilized, the timer can be used as the timer which generates interrupt after the specified period.

5.5.4.2 Sub Clock Oscillation Stabilization Wait Timer (Sub Timer)

The sub clock oscillation stabilization wait timer (Sub Timer) is explained.

The sub timer is operated by the sub clock (SBCLK). This timer is used for the generation of the sub clock oscillation stabilization wait time, and in the sub clock stabilization status other than those can be used as the timer which generates interrupt after the specified period.

5.5.4.3 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 clock oscillation stabilization wait time. This timer cannot be used for a general-purposed timer.

5.5.4.4 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 cleared (MTMCR:MTC=1), the main timer 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".

If the sub timer operation is enabled (STMCR:STE=1), the count operation of the sub timer starts. If the sub timer operation is disabled (STMCR:STE=0), the count operation of the sub timer stops and the sub timer is cleared.

If the sub timer is cleared (STMCR:STC=1), the sub timer is cleared. STMCR:STC=1 is read until clear. The period of interrupt can be set by STMCR:STS[2:0]. When STMCR:STIE=1, if STMCR:STIF=1, the sub timer interrupt occurs. STMCR.STIF is cleared by writing "0".

Note:

For setting the period of the timer interrupt (MTS and STS), set the period more than PCLK1 × 5 clock. When the period of the timer interrupt is set to the extremely short time, the interrupt source may not be set.

5.5.4.5 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)/(STMCR:STIE=0)
and the interrupt flag clear(MTMCR:MTIF=0)/(STMCR:STIF=0)

↓

Sets the timer operation disable (MTMCR:MTE=0)/(STMCR:STE=0)

↓

Verifies MTC=0/STC=0

↓

Sets the period of the timer (MTMCR:MTS=1000 to 1111)/(STMCR:STS=000 to 111)

↓

Sets the timer interrupt enable (MTMCR:MTIE=1)/(STMCR:STIE=1)

↓

Sets the timer operation enable (MTMCR:MTE=1)/(STMCR:STE=1)

↓

The interrupt occurs after setting time

↓

To the interrupt routine

↓

Sets the interrupt flag clear (MTMCR:MTIF=0)/(STMCR:STIF=0)

↓

Verifies the interrupt flag (MTMCR:MTIF=0)/(MTMCR:STIF=0)

↓

Program operations

↓

RETI

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

5.5.4.6 Timer Operations

Timer operations are shown.

When 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".

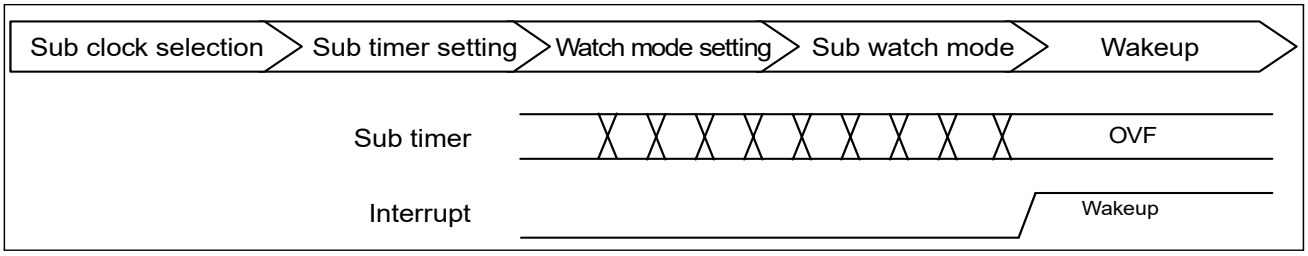
While STMCR:STE=1, the sub timer counts up by the sub clock (SBCLK). If the timer overflows by the period which is selected by STMCR:STS[2:0], STMCR:STIF is "1".

5.5.4.7 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/sub timer interrupt or RTC interrupt. The example for switching of the watch mode in the setting of wake-up from the sub timer is shown as follows.

Figure 5-15. Wake-up from the Watch Mode



5.5.5 Notes When Clocks Conflict

Notes when clocks conflict is shown.

Notes that if peripheral interrupt activated by the very low frequency 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 over again.

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 to "00"->"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 to "01"->"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 to "10"->"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 to "11"->"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, the PLL reference clock is controlled so as to be supplied even if CSELR:PCEN is cleared while communicating the MDI high speed. The value of the register related to PLL is maintained and not updated. However, when software sets CSELR:PCEN=0, the value of the register related to PLL can be freely updated (write).

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 usually. The register related to PLL is as follows.

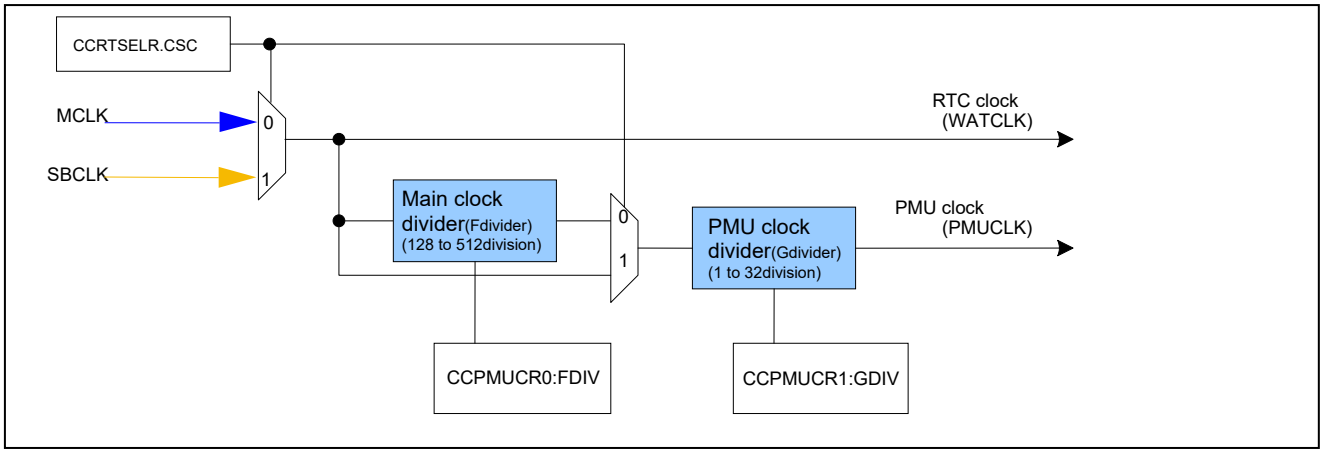
- CCPSDIVR:PODS
- CCPLLFBR: IDIV
- PLLCR:PDS

5.5.8 About PMU Clock (PMUCLK)

The PMU clock (PMUCLK) is shown.

The PMU clock is an operation clock of power management unit (PMU). Complete the setting of this clock before controlling the standby mode.

Figure 5-16. Watch/Power Management Clock Generation Unit



The frequency of the PMU clock can be calculated by the following expressions.

- When CCRTSELR:CSC=0 (main clock is selected)

  PMU clock frequency=
  (Main clock frequency) / (CCPMUCR0: FDIV [1:0] division ratio) / (CCPMUCR1:GDIV[4:0] division ratio)

- When CCRTSELR:CSC=1 (sub clock is selected)

  PMU clock frequency=(Sub clock frequency) / (CCPMUCR1:GDIV[4:0] division ratio)

  Moreover, observe the following specification limitation to the PMU clock. (There is a possibility that the shutdown processing is not normally done when this limitation is not defended.)

  1. Selects the clock under the oscillation about CCRTSELR:CSC.[1]
  2. The PMU clock must use the machine of F divider frequency to become 32 kHz or less.
  3. Please use the machine of G dividing frequency to become 1/4 of the frequencies of peripheral clock (PCLK1).

  [1]: Always CCRTSELR:CSC="0" is always read for single clock products.

It explains each specification limitation as follows.

1. Selects the clock under the oscillation about CCRTSELR:CSC.

   Please confirm the CMONR: MCRDY register and the CMONR: SCRDY register to the oscillation of the main clock and a sub-clock. Moreover, when the CCRTSELR:CSC register is rewritten, the processing of the handshaking of the main clock and a sub-clock (clock transfer) is generated. If both clocks are oscillating (CMONR:MCRDY=CMONR:SCRDY=1), the change operation is not normally completed for this period. Please confirm the status of the clock transfer by the CCRTSELR:CST register.

2. The PMU clock must use the machine of F divider frequency to become 32 kHz or less.

   The PMU clock is used to control the power switch, and the frequency of 32 kHz or less is recommended for the reasons for the stabilization at the pressure rising time when the power supply is input etc.

   As for the PMU clock, the main clock is selected for CCRTSELR:CSC=0 as a source clock. Please set the CCPMUCR0:FDIV register so that the frequency of the PMU clock may become 32 kHz or less. The machine of F divider frequency does not influence operation for CCRTSELR:CSC=1.

FDIV[1:0]	Division Rate	Target Main Oscillation Frequency
00	128 division(initial value)	4MHz
01	256 division	8MHz
10	384 division	12MHz
11	512 division	16MHz

3. Please use the machine of G dividing frequency to become 1/4 of the frequencies of peripheral clock (PCLK1). Signal transfer between peripheral clock (PCLK) and PMU clock (PMUCLK) needs 4 PMU clock cycles.

   When the source clock of peripheral clock(PCLK1) is sub oscillation clock (CMONR:CKM=10), the frequency of peripheral clock(PCLK1) should be set quadruple (or more higher) frequency of PMU clock. It can be set by CCPMUCR1:GDIV register.

   When the source clock of peripheral clock(PCLK1) is main oscillation clock (CMONR:CKM=00 or CMONR:CKM=01). If the frequency of peripheral clock(PCLK1) is slower than 128kHz (depends on DIVR0:DIVB and DIVR2:DIVP), CCPMUCR1:GDIV register should be set as same.

GDIV[4:0]	Division Ratio
00000	Do not divide (initial value)
00001	2 division
…	…
11110	31 division
11111	32 division

[Reference]
The frequency of the peripheral clock (PCLK1) can be calculated by the following expressions:

Peripheral clock (PCLK1) frequency = (Clock frequency selecting it by CMONR:CKM) / (DIVR0:DIVB[2:0] division ratio) /(DIVR2:DVP[3:0] division ratio)

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

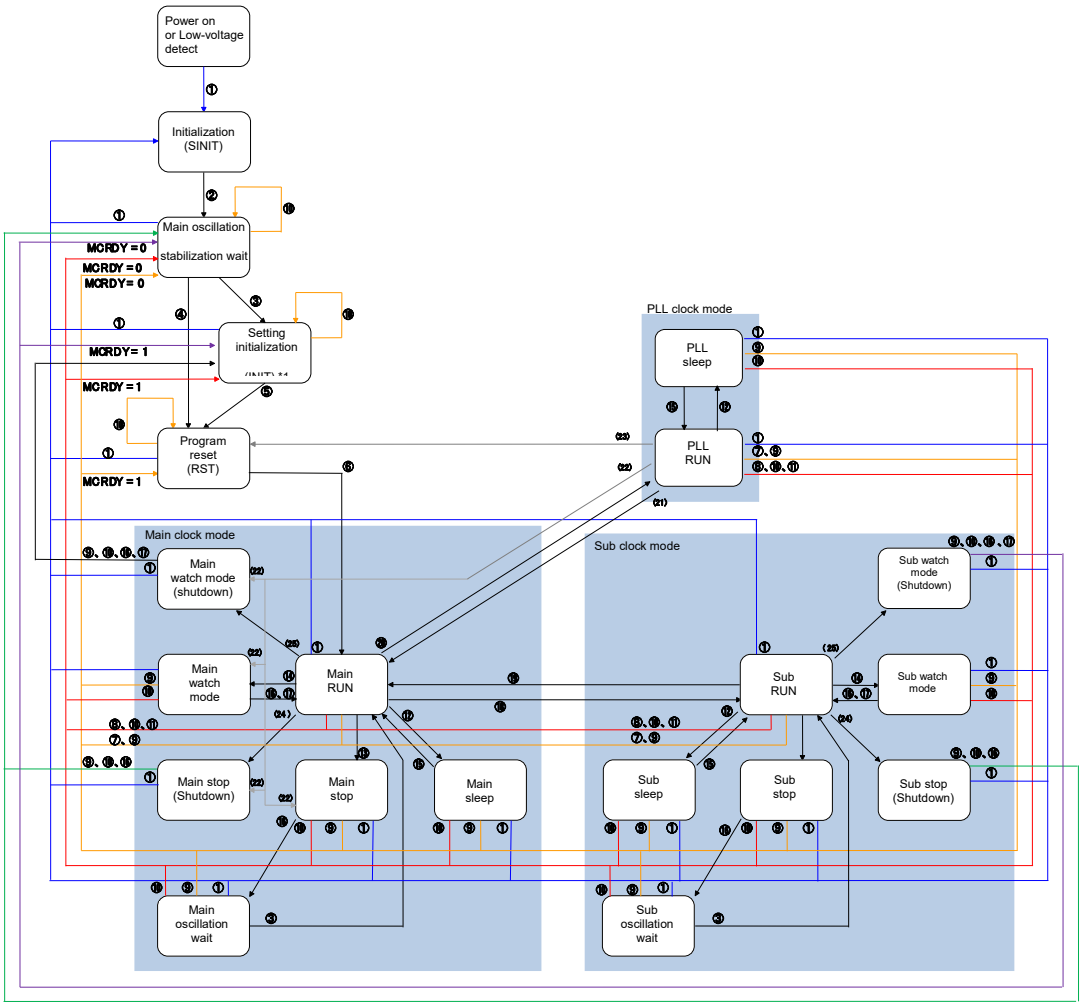
6.2.3 Priority of State Transition Requests

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



① Power-on reset or internal low-voltage detection or external reset and simultaneous assert of NMI
② Power-on reset release and internal low-voltage release and external reset and release simultaneous assert of NMI
③ End of oscillation stabilization wait
④ End of oscillation stabilization wait (if the reset factor is ⑦ or ⑨)
⑤ INIT release
⑥ RST release
⑦ Software reset
⑧ Software watchdog reset (including irregular) or software reset (irregular)
⑨ External reset input (NMI disabled) or external low-voltage detection
⑩ External reset input (NMI disabled + irregular) or external low-voltage detection (irregular)
⑪ Hardware watchdog reset (including irregular)
⑫ Sleep mode (write instruction)
⑬ Stop mode (write instruction)
⑭ Watch mode (write instruction)
⑮ Interrupt (including ⑯ and ⑰)
⑯ Interrupt (clock not required)/NMI
⑰ Main timer interrupt/Sub timer interrupt/RTC interrupt
⑱ Switch from main to sub (write instruction)
⑲ Switch from sub to main (write instruction)
⑳ Switch from main to PLL (write instruction)
(21) Switch from PLL to main (write instruction)
(22) Illegal standby mode transition
(23) Illegal standby mode transition detection reset
(24) Stop mode and shutdown (write instruction)
(25) Watch mode and shutdown (write instruction)

There is a register not reset when returning from the watch mode (Shutdown) and returning from the stop mode (Shutdown). See "Limitations of the standby control power shutdown/usually" in "Chapter: Power Consumption Control" for detail.

Notes:

- The transition may be different from above diagram when connecting to OCD tool. See "Chapter: On Chip Debugger (OCD)" for details.
- The sub clock mode is not transmitted to because single clock products do not include the sub clock input.

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 (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, sub 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. Transits to the RUN state by some specific (no clock required) effective interrupts, main timer interrupts, sub timer interrupts and watch counter interrupts. For details, see "Chapter: Power Consumption Control".

Watch Mode (Power Shutdown) State

The device is stopped while the power supply unnecessary for the watch mode is turned off. The state transits by program operation. The power supply for the internal circuit is turned off and the internal circuits other than the oscillation circuits (the main clock generation unit and the sub clock generation unit) are stopped. Stop PLL oscillation before going into the watch mode (power shutdown) state. It is also possible to use the external pins altogether (except for some pins) for high impedance by the settings. Transits to the setting initialization (INIT) state by some specific (no clock required) effective interrupts, the main timer interrupt, the sub timer interrupt and the watch counter interrupt. 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".

Stop (Power Shutdown) State

The device is stopped while the power supply unnecessary for the stop state is turned off. The state transits by program operation. The power supply for the internal circuit is turned off and all the internal circuits are stopped. Stop PLL oscillation before going into the stop (power shutdown) state. It is also possible to use the external pins altogether (except for some pins) for high impedance by the settings. Transits to the main oscillation stabilization wait (reset) state by NMI interrupt. For details, see "Chapter: Power Consumption Control".

Main Oscillation Stabilization Wait, Sub 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} × 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 sub oscillation circuit and 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 (Single Clock Product)

Device State	Main Clock	Regulator Mode
Main RUN	Oscillation	Main mode
Main sleep	Oscillation	Main mode
Main watch mode	Oscillation	Standby mode
Main watch mode (Shutdown)	Oscillation	Standby mode
Main stop	Stop	Standby mode
Main stop (Shutdown)	Stop	Standby mode
Main Oscillation wait	Oscillation	Main mode
PLL RUN	Oscillation	Main mode
PLL sleep	Oscillation	Main mode

Table 6-2. Relationship between Device State and Regulator Mode (Dual Clock Product)

Device State	Main Clock	Sub Clock	Regulator Mode
Main RUN	Oscillation	Oscillation or Stop	Main mode
Main sleep	Oscillation	Oscillation or Stop	Main mode
Main watch mode	Oscillation	Oscillation or Stop	Standby mode
Main watch mode(Shutdown)	Oscillation	Oscillation or Stop	Standby mode
Main stop	Stop	Stop	Standby mode
Main stop (Shutdown)	Stop	Stop	Standby mode
Main Oscillation wait	Oscillation	Oscillation or Stop	Main mode
Sub RUN 1	Oscillation	Oscillation	Main mode
Sub RUN 2	Stop	Oscillation	Sub mode
Sub sleep 1	Oscillation	Oscillation	Main mode
Sub sleep 2	Stop	Oscillation	Sub mode
Sub watch mode	Oscillation or Stop	Oscillation	Standby mode
Sub watch mode (Shutdown)	Oscillation or Stop	Oscillation	Standby mode
Sub stop	Stop	Stop	Standby mode
Sub stop (Shutdown)	Stop	Stop	Standby mode
Sub Oscillation wait 1	Oscillation	Oscillation	Main mode
Sub Oscillation wait 2	Stop	Oscillation	Sub mode
PLL RUN	Oscillation	Oscillation or Stop	Main mode
PLL sleep	Oscillation	Oscillation or Stop	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
- Recovery reset from stand by (power shutdown)

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 whether or not 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

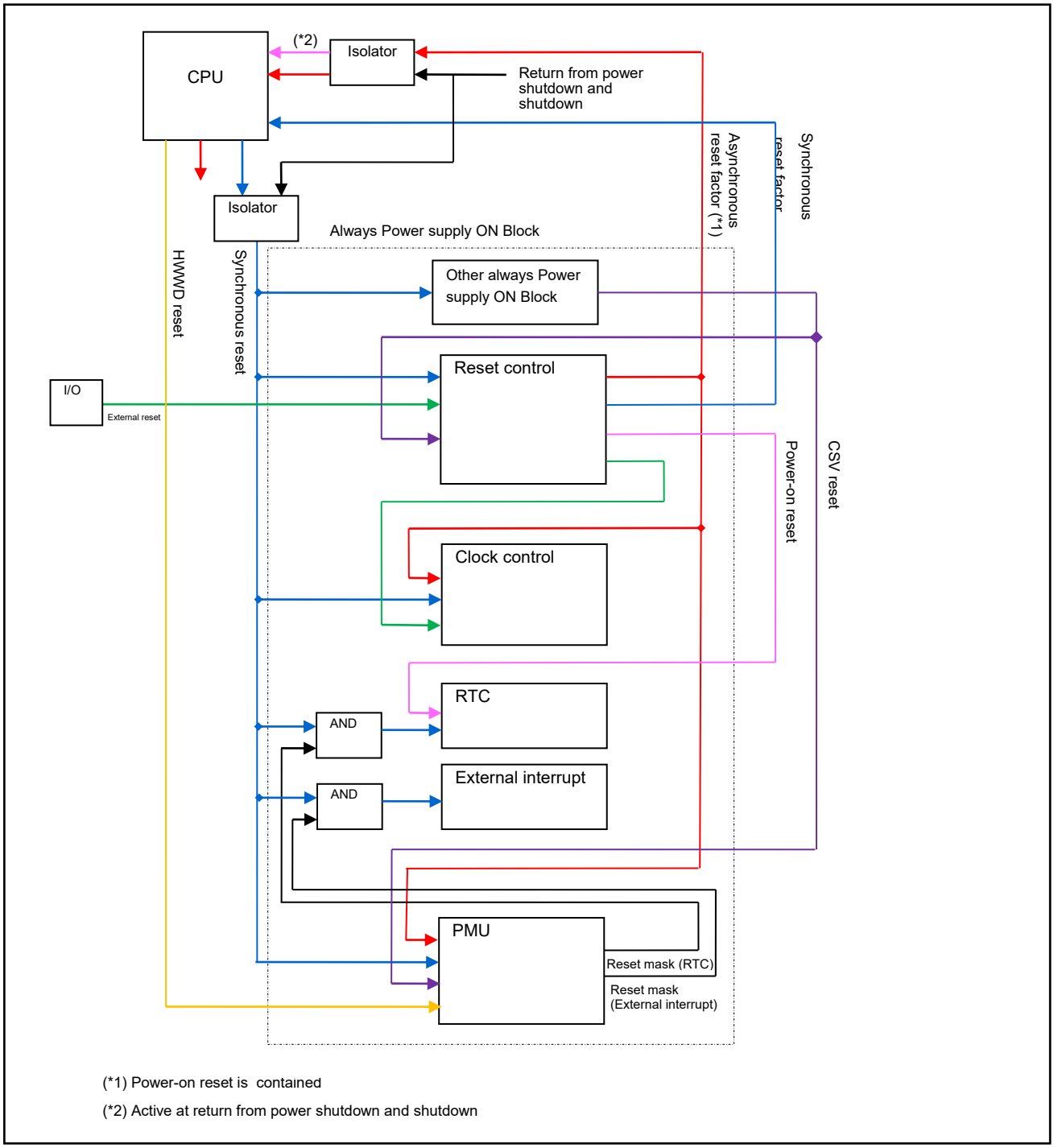


Figure 7-2. Configuration Diagram of Reset (Reset Control)

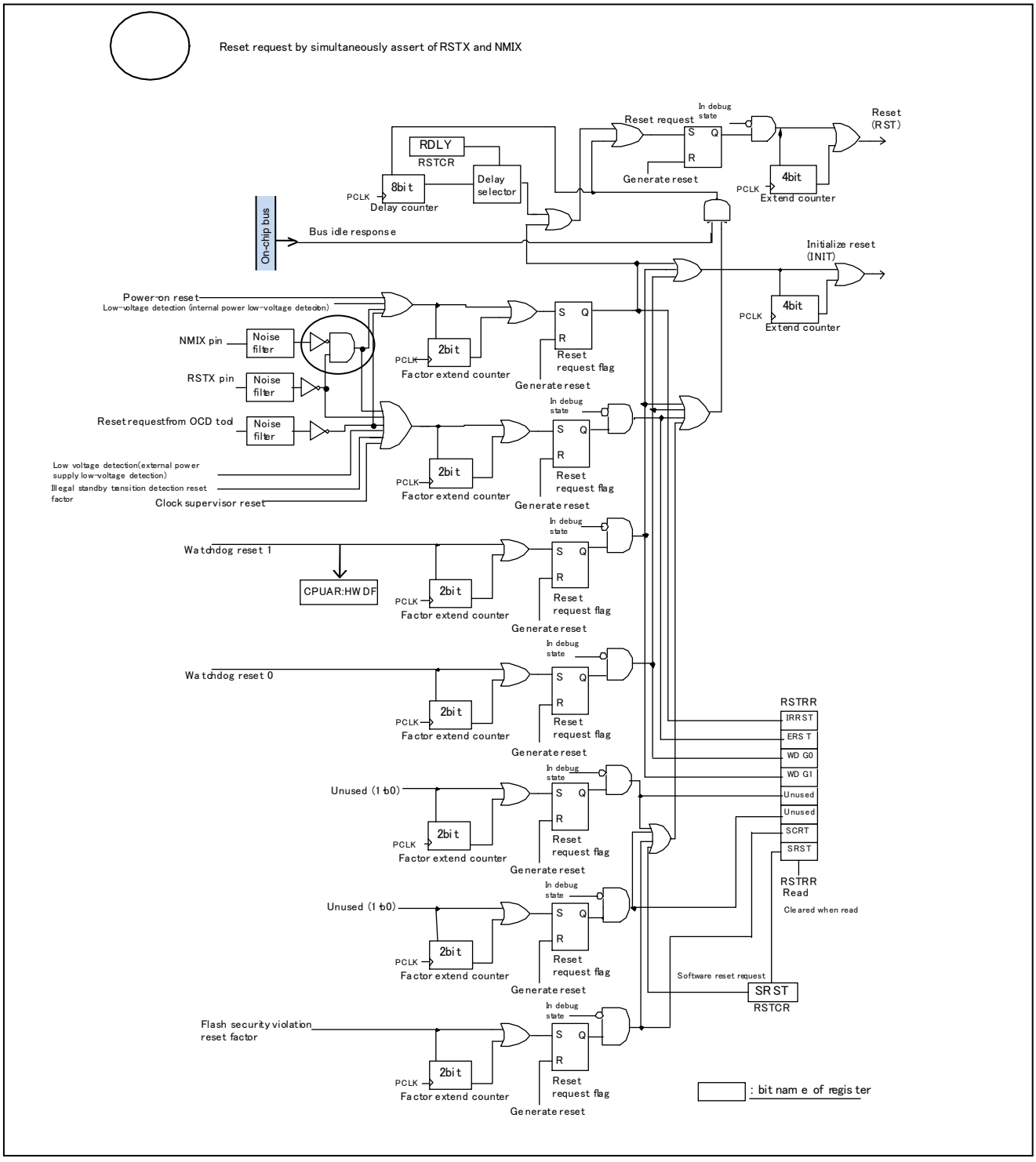
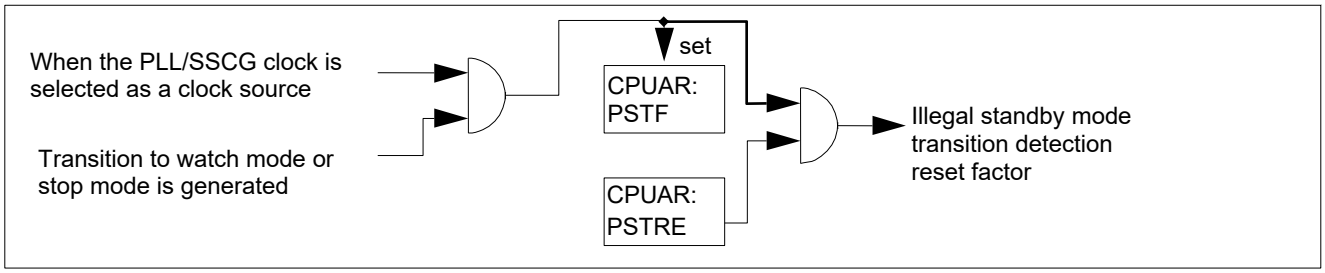


Figure 7-3. Generation Diagram of Illegal Standby Mode Transition Detection Reset Factor



7.4 Registers

This section explains the registers of the reset.

Table 7-1. Registers 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
0x0590	PMUSTR	Reserved	Reserved	Reserved	PMU Status Register

Note:

Please note that the register of "Chapter: Power Consumption Control" is allocated in address 0x0482, 0x0591, and 0x0592.

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.

Note:

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

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	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Attribute	R,WX	R,WX	R,WX	R,WX	RX,WX	RX,WX	R,WX	R,WX

[1]: Due to a reset factor.

[bit7] IRRST (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, External Low-voltage Detection, Clock Supervisor Reset 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)

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)

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)

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

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 × 2 cycles
001	PCLK × 4 cycles
010	PCLK × 8 cycles
011	PCLK × 16 cycles
100	PCLK × 32 cycles
101	PCLK × 64 cycles
110	PCLK × 128 cycles
111	PCLK × 256 cycles (Initial value)

[bit0] SRST (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.

In the RSTCR reading in the debugging state, reset is not generated.

SRST	Software Reset
0	No output (initial value)
1	The set request is output by RSTCR reading.

7.4.3 CPU Abnormal Operation Register: CPUAR

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	[1]	[1]	[1]	[1]
Attribute	R/W	R0,WX	R0,WX	R0,WX	RX,WX	R(RM1),W	R(RM1),W	R(RM1),W

[1]: It will be initialized to "0" by RSTX pin asserts (including simultaneous assert with NMIX). It will not be initialized by the other reset factors.

[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

The set factor is given to priority when a set factor and a clear factor are generated at the same time.

[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 timer1 (Hardware watchdog) reset factor has been generated.	Clear this bit
1	Watchdog timer1 (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.

7.4.4 PMU Status Register: PMUSTR (Power Management Unit STatus register)

The bit configuration of the PMU status register is shown.

This register indicates the PMU status.

PMUSTR: Address0590$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PMUST	Reserved					PONR_F	RSTX_F
Initial value	0	0	0	0	0	0	1	[1]
Attribute	R,W	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R,W	R,W

[1]: It will be initialized to "1" by RSTX pin asserts (including simultaneous assert with NMIX). It will not be initialized by the other reset factors.

[bit7] PMUST (Power Management Unit Status)

The state immediately before shows information on whether it was a shutdown mode.

PMUST	PMU Status
0	Operation return from initial state and initialization reset
1	Operation return from Shutdown mode

This bit is cleared by writing "0". "1" writing is invalid. This bit is just initialized by the power-on reset, the low-voltage detection, and the simultaneous assert of RSTX and NMIX only; is not initialized by the other reset factors. Therefore, confirm the reset factors before judging whether the status is the retuning from the power- shutdown mode or not.

[bit6 to bit2] Reserved

"0" is always read. Please be sure to write "0".

[bit1] PONR_F (Power On Reset Flag)

This bit is a power-on reset detection flag.

PONR_F	Power-on Reset
0	No detection
1	Detection

This bit is cleared by writing "0". "1" writing is invalid.
This bit is not initialized in reset factors other than power-on reset.

[bit0] RSTX_F (Resetx Input Flag)

This bit is an external reset detection flag.

RSTX_F	RSTX Input Reset
0	No detection
1	Detection

This bit is cleared by writing "0". "1" writing is invalid.
This bit is not cleared by the power-on reset. Be sure to use after clear.

7.5 Operation

This section explains each operation of the reset feature of this product.

7.5.1 Reset Level
7.5.2 Reset Factor
7.5.3 Reset Acceptance
7.5.4 Reset Issue
7.5.5 Reset Sequence
7.5.6 Notes

7.5.1 Reset Level

The reset level is explained.

The following two levels of resets are available with this product.

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)

Initialize reset (INIT) is explained.

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 a sub oscillation circuit and, PLL become 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 explained.

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

This section explains each reset factor of this product.

7.5.2.1 Power-on Reset
7.5.2.2 RSTX Pin Input
7.5.2.3 Watchdog Reset 0
7.5.2.4 Watchdog Reset 1
7.5.2.5 External Low-voltage Detection Reset
7.5.2.6 Illegal Standby Mode Transition Detection Reset
7.5.2.7 Internal Low-voltage Detection Reset
7.5.2.8 Flash Security Violation Reset
7.5.2.9 Software Reset (RSTCR:SRST)
7.5.2.10 Recovery from Standby (Power Interception)

7.5.2.1 Power-on Reset

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. Whether or not 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.

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. The reset from this reset source is detected as irregular reset. After the detection, 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 a software reset request)
MOV     R0, R0                  ; Dummy processing for pipeline adjustment
NOP                             ; Dummy processing for pipeline adjustment
```

7.5.2.10 Recovery from Standby (Power Interception)

Recovery from standby (power interception) is shown.

On majority of the block, the operation similar as super initialize reset (SINIT) is executed by the start from the standby. However, power-on reset factor is always at the power-on block, the detection is not displayed in the reset source register (RSTRR) . The factors are displayed in the PMU status register (PMUSTR), and please confirm this register, when the microcontroller reactivates.

Reset by this reset factor issues the initialization reset (INIT).

7.5.3 Reset Acceptance

This section explains the acceptance processing of each reset factor.

7.5.3.1 Generation of Reset Request

7.5.3.2 Acceptance of Reset Request

7.5.3.3 Reset Issue Delay Counter

7.5.3.4 Irregular Reset

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

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, it will generate an irregular reset.

Once an irregular reset is generated, the following processing will be executed.

- Regardless of the type of reset factor, initialize reset (INIT) will be issued.
- 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

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)

7.5.4.2 Initialize Reset (INIT)

7.5.4.3 Reset (RST)

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} × main clock cycle.

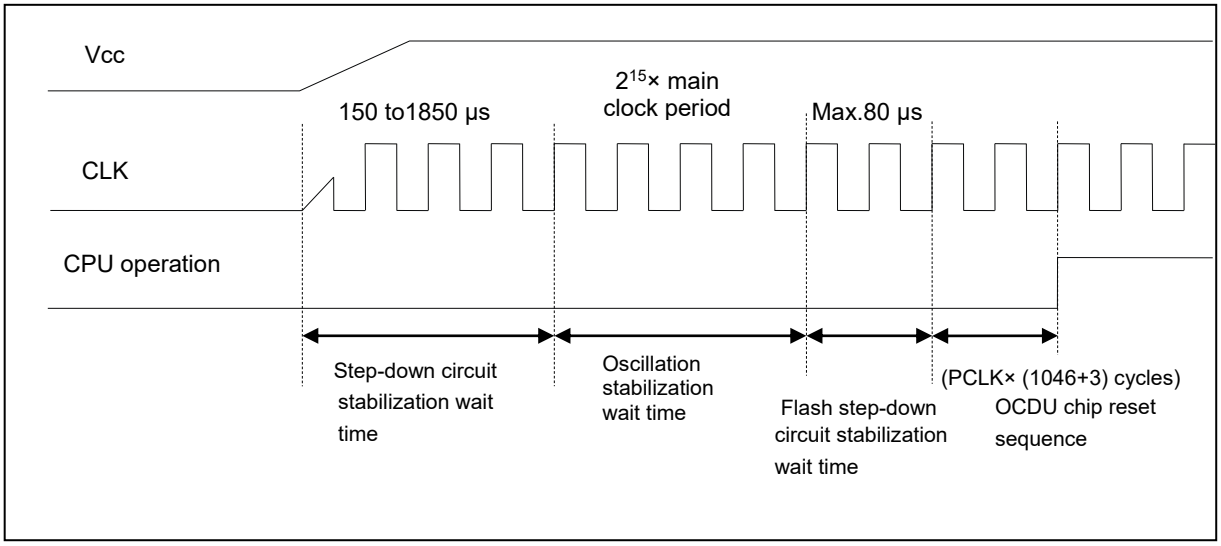
Table 7-2. Oscillation Stabilization Wait Time (SINIT)

Type	Main Clock Oscillation Stabilization Wait Time
Power-on reset	2^{15}× Main clock cycle
Internal low-voltage detection	2^{15}× Main clock cycle
Simultaneous assert of RSTX and NMIX	2^{15}× Main clock cycle

Note:

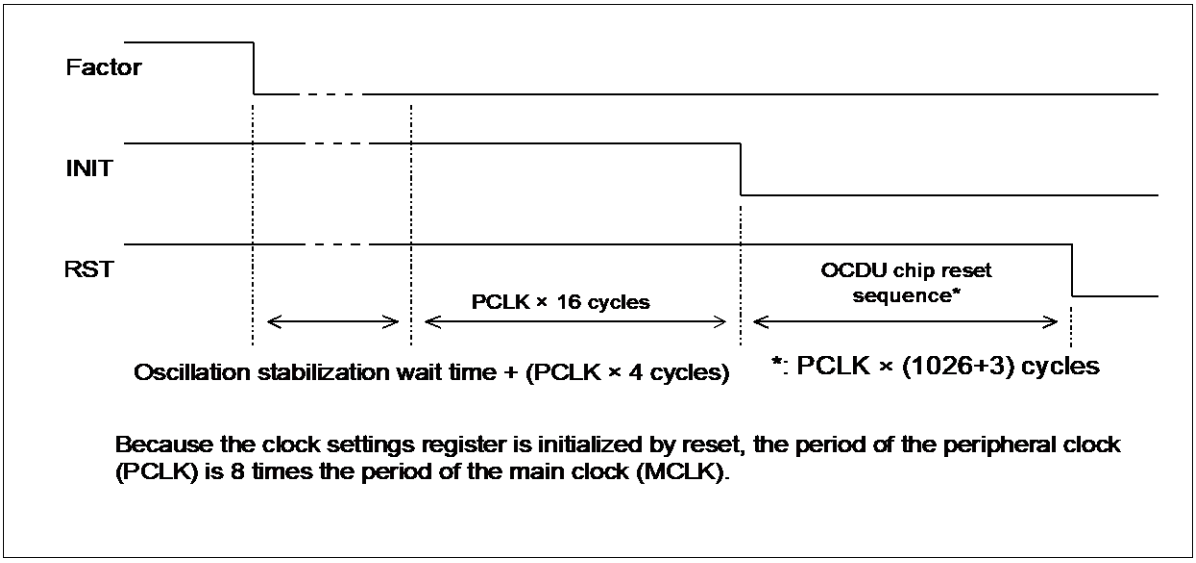
The oscillation stabilization wait time shown in the above table does not include the regulator stabilization wait time and FLASH stabilization wait time associated with the power-on and voltage restore. These stabilization wait time (150 µs to 1850 µs and maximum 80 µs) are needed at power-on reset.

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



7.5.4.2 Initialize Reset (INIT)

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.

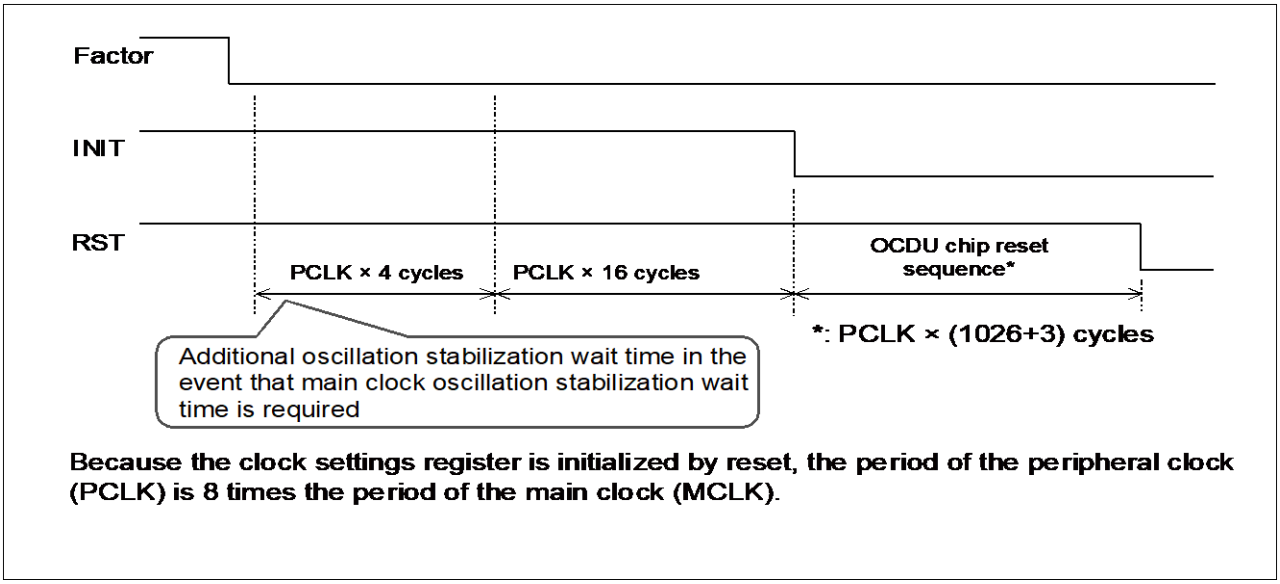
If the main clock is inactive such as in a stop mode, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15} × main clock cycle).

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}× 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 "7.5.1.1. Initialize Reset (INIT)").

While this reset is being issued, all clocks become active.

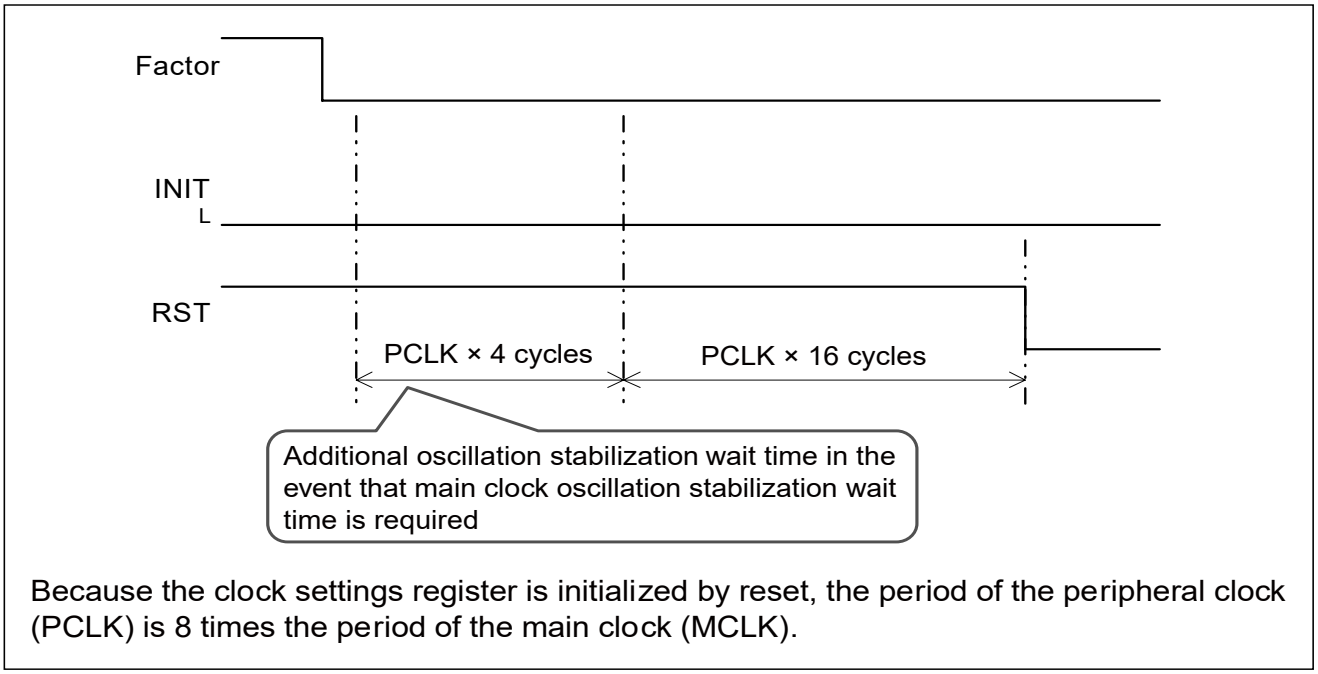
If the main clock is inactive such as in a stop mode before the reset, it takes the main clock oscillation stabilization wait time. Since the register of the clock control part will be initialized by a reset, the oscillation stabilization wait time is the default value of this product (2^{15} × main 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} × 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

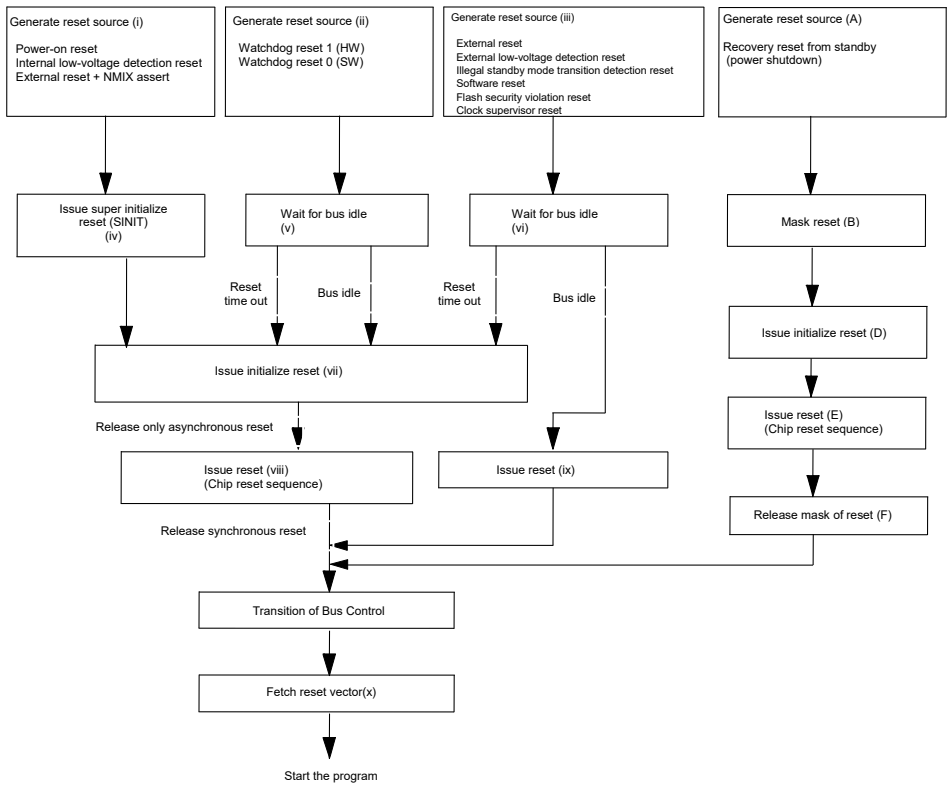


7.5.5 Reset Sequence

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 2-1 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 a bus idle status after generating reset source, so the status move to (ix).

-Super initialize reset (SINIT) is issued at the recovery from standby (power shutdown) (A). However, because of preventing a reset to following block the reset without SINIT to this block will be masked (B) during the reset period.

(1) RTC (only watch mode)

(2) External interrupt block

(3) Power management unit

(4) Clock generation block (only sub-clock selection register)

(5) LCD controller

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 × peripheral clock (PCLK) cycle.
After that, a reset cycle will be maintained by the period of peripheral clock (PCLK) × 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

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

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

Reset vector fetch is shown.

After the reset release, the CPU starts fetching the reset vector (at 0x000FFFFC).

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

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,E_BPCLR register).

7.5.6 Notes

Notes are shown.

During return form standby watch mode (power-shutdown) and standby stop mode (power-shutdown), an internal reset is issued. Therefore any reset source without power-on reset, internal low-voltage detection reset, reset by simultaneous assert of RSTX and NMIX will not be accepted.

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

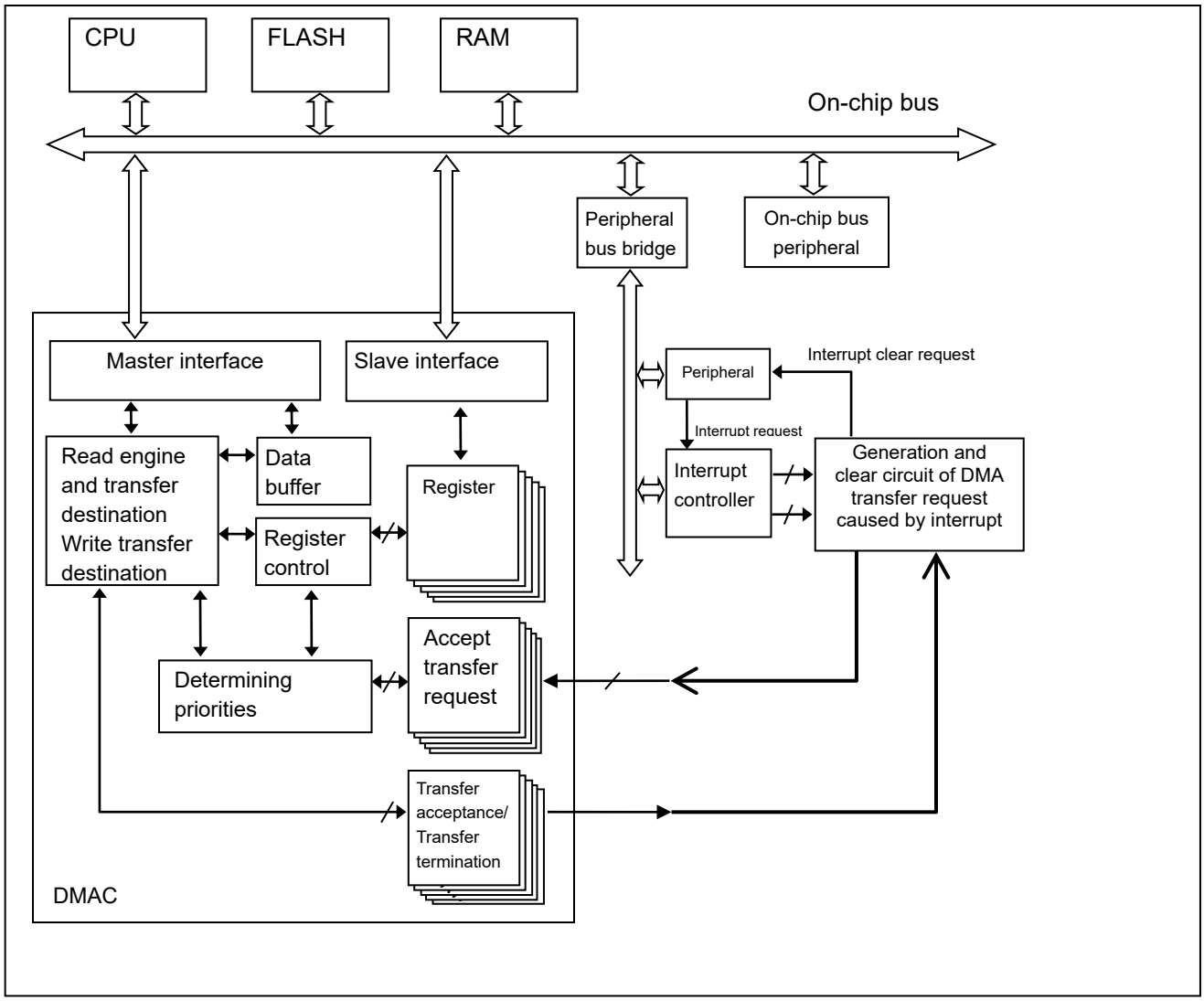
This section explains the features of the DMA controller (DMAC).

- Channels : 16 channels
- Address space : 32-bit address space (4GB)
- 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 interrupt (for the transfer request by peripheral interrupt, you should select interrupt by channels. See "Chapter: Generation and Clearing of DMA Transfer Requests".)
- 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 > ch.8 > ch.9 > ch.10 > ch.11 > ch.12 > ch.13 > ch.14 > ch.15)
  - ☐ 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 block 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

Address	Registers				Register Function
	+0	+1	+2	+3	
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
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
0x0C80	DCCR8				DMA channel control register 8
0x0C84	DCSR8		DTCR8		DMA channel status register 8 DMA transfer count register 8
0x0C88	DSAR8				DMA transfer source address register 8
0x0C8C	DDAR8				DMA transfer destination address register 8
0x0C90	DCCR9				DMA channel control register 9

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0C94	DCSR9		DTCR9		DMA channel status register 9 DMA transfer count register 9
0x0C98	DSAR9				DMA transfer source address register 9
0x0C9C	DDAR9				DMA transfer destination address register 9
0x0CA0	DCCR10				DMA channel control register 10
0x0CA4	DCSR10		DTCR10		DMA channel status register 10 DMA transfer count register 10
0x0CA8	DSAR10				DMA transfer source address register 10
0x0CAC	DDAR10				DMA transfer destination address register 10
0x0CB0	DCCR11				DMA channel control register 11
0x0CB4	DCSR11		DTCR11		DMA channel status register 11 DMA transfer count register 11
0x0CB8	DSAR11				DMA transfer source address register 11
0x0CBC	DDAR11				DMA transfer destination address register 11
0x0CC0	DCCR12				DMA channel control register 12
0x0CC4	DCSR12		DTCR12		DMA channel status register 12 DMA transfer count register 12
0x0CC8	DSAR12				DMA transfer source address register 12
0x0CCC	DDAR12				DMA transfer destination address register 12
0x0CD0	DCCR13				DMA channel control register 13
0x0CD4	DCSR13		DTCR13		DMA channel status register 13 DMA transfer count register 13
0x0CD8	DSAR13				DMA transfer source address register 13
0x0CDC	DDAR13				DMA transfer destination address register 13
0x0CE0	DCCR14				DMA channel control register 14

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0CE4	DCSR14		DTCR14		DMA channel status register 14 DMA transfer count register 14
0x0CE8	DSAR14				DMA transfer source address register 14
0x0CEC	DDAR14				DMA transfer destination address register 14
0x0CF0	DCCR15				DMA channel control register 15
0x0CF4	DCSR15		DTCR15		DMA channel status register 15 DMA transfer count register 15
0x0CF8	DSAR15				DMA transfer source address register 15
0x0CFC	DDAR15				DMA transfer destination address register 15
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

This section explains the DMA control register (DMACR).

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 enabled

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 DCCRn: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 DCCRn: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 15: DCCR0 to 15

This section explains the bit configuration for DMA channel control register 0 to 15 (DCCR0 to 15).

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 15: 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 enabled

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 (DCCRn: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 DCCRn:BLK. When writing "1" again and detecting a new transfer request, the operation restarts.

CE	Channel Operation Enabled
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 enabled

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 : DCCRn:TM = 10_B
- Transfer source address count : DCCRn:SAC = 10_B
- Transfer destination address count : DCCRn:DAC = 10_B
- Transfer size : DCCRn:TS = 11_B
- Demand transfer mode by software request : DCCRn:RS = 00_B and DCCRn:TM = 11_B

As for the interrupt factor, refer to the status register (DCSRn).

AIE	Abnormal Completion Interrupt Enabled
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, refer to the status register (DCSRn).

SIE	Transfer Suspend Interrupt Enabled
0	Disabled (initial value)
1	Enabled

[bit24] NIE (Normal Completion Interrupt Enable): Normal completion interrupt enabled

This bit controls the generation of interrupts when completing DMA transfers successfully. After completing transfers as many times as set by transfer count (DTCRn:DTC) or when writing "1" to the corresponding channel's DCCRn:CE bit at the time the transfer count is "0", the operation will complete normally. As for the interrupt factor, see the status register (DCSRn).

NIE	Normal Completion Interrupt Enabled
0	Disabled (initial value)
1	Enabled

[bit23, bit22] Reserved

Always write "0" to these bits. The read value is "0".

[bit21, bit20] RS[1:0] (Request Source): DMA transfer request source

These bits select the transfer request source for the channel.

RS[1:0]	DMA Transfer Request Source
00	Software (initial value)
01	Interrupts
10	Reserved (setting is prohibited)
11	Reserved (setting is prohibited)

[bit19, bit18] Reserved

Always write "0" to these bits. The read value is "0".

[bit17, bit16] TM[1:0] (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 (DCCR: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[1:0] (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 (DCCRn: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). As for the setting, 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[1:0] (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 (DCCRn: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", DCCRn: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, DCCRn: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[1:0] (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 :halfword
10	32-bit :word
11	Reserved (setting is prohibited)

Set values to DSARn and DDARn registers so as not to cause a misalignment for the transfer size specified in these bits.

[bit3 to bit0] BLK[3:0] (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 DCCRn:TS bit.

BLK[3:0]	Block Size
0000	1 byte
0001	2 bytes
0010	3 bytes
0011	4 bytes
0100	5 bytes
0101	6 bytes
0110	7 bytes
0111	8 bytes
1000	9 bytes
1001	10 bytes
1010	11 bytes
1011	12 bytes
1100	13 bytes
1101	14 bytes
1110	15 bytes
1111	16 bytes

8.4.3 DMA Channel Status Register 0 to 15: DCSR0 to 15

This section explains the bit configuration for DMA channel status register 0 to 15 (DCSR0 to 15).

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 15: 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 operation of the channels. Writing "1" to the corresponding DCCRn: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 DCCRn:CE makes the operation stop.

Writing this bit is ignored.

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 : DCCRn:TM = 10B
- Transfer source address count : DCCRn:SAC = 10B
- Transfer destination address count : DCCRn:DAC = 10B
- Transfer size : DCCRn:TS = 11B
- Demand transfer mode by software request : DCCRn:RS = 00B and DCCRn:TM = 11B

When having allowed the abnormal completion interrupt (DCCRn:AIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is ignored.

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 suspension 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 (DCCRn:SIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is ignored.

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 "DCCRn:CE" bit at the time the transfer count is "0", the operation will complete normally. When having allowed the normal completion interrupt (DCCRn:NIE), writing "0" to this bit clears the interrupt. Writing "1" to this bit is ignored.

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 15: DTCR0 to 15

This section explains the bit configuration for DMA transfer count register 0 to 15 (DTCR0 to 15).

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 15: 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 transfer times. 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 DCCRn:TCR is "1", the value is returned to the initial value after data transfer.

8.4.5 DMA Transfer Source Register 0 to 15: DSAR0 to 15: (DMA Source Address Register 0 to 15)

This section explains the bit configuration for DMA transfer source register 0 to 15 (DSAR0 to 15).

These registers are 32-bit registers to indicate the transfer source address of each DMAC channel, and each channel has these registers separately. This register must be accessed as a 32-bit data.

DSAR0 to 15: 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] DSA[31:0] (DMA Source Address): DMA transfer source address

These registers indicate the transfer source address. If an increment or a decrement is set by DCCRn:SAC, the address is updated according to the transfer size (DCCRn:TS). Also, the dedicated reload register is provided. If DCCRn:SAR 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 DCCRn:TS.

If the DMA transfer request source has a peripheral interrupt (DCCRn: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 15: DDAR0 to 15 (DMA Destination Address Register 0 to 15)

This section explains the bit configuration for DMA transfer destination register 0 to 15 (DDAR0 to 15).

These registers are 32-bit registers to indicate the transfer destination address of each DMAC channel, and each channel has these registers separately. These registers must be accessed as a 32-bit data.

DDAR0 to 15: 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

[bit 31 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 DCCRn:DAC, the address is updated according to the transfer size (DCCRn:TS). Also, the dedicated reload register is provided. If DCCRn: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 DCCRn:TS.

If the DMA transfer request source has a peripheral interrupt (DCCRn: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 NMI flag register (DNMIR).

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 ignored.

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	Stops the DMA transfer by the user NMI. (initial value)
1	Does 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 (DILVR).

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[4:0] (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	Suppresses the DMA transfer when any peripheral interrupt request is issued. (initial value)
11110	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 1E$_H$ is issued.
11101	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 1D$_H$ is issued.
11100	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 1C$_H$ is issued.
11011	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 1B$_H$ is issued.
11010	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 1A$_H$ is issued.
11001	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 19$_H$ is issued.
11000	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 18$_H$ is issued.
10111	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 17$_H$ is issued.
10110	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 16$_H$ is issued.

LVL[4:0]	DMA Suppression Control
10101	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 15H is issued.
10100	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 14H is issued.
10011	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 13H is issued.
10010	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 12H is issued.
10001	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 11H is issued.
10000	Does 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).

Configuration

The following explains the setting items common to all channels and the items to be set separately for each channel.

- Common Items for All Channels

  "8.5.1 DMA Operation Enable" explains the register settings for the entire DMAC control.

8.5.1 DMA Operation Enable

This section explains the DMA operation enable..

The entire DMAC operation can be controlled using the DMACR:DME.

- DMA operation disabled (DMACR:DME = 0)
- DMA operation enabled (DMACR:DME = 1)

Channel Priority

A channel priority can be set by the 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.

8.5.2 Separate Items for Each Channel

This section explains the separate items for each channel of the DMA controller (DMAC).

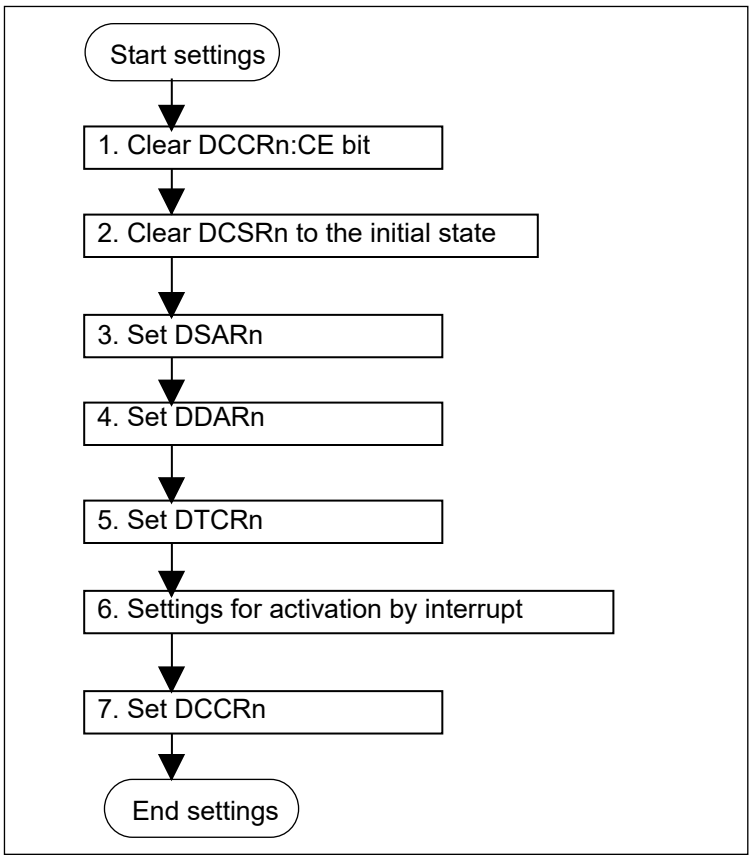
The following explains both the items to be set separately for each channel and the register setup procedure.

Register Setup Procedure

The channel registers must be set in the following procedure. When you set the DCCRn:CE bit to "1", be sure to set the DTCRn to 1 or a higher value.

1. Clear the DCCRn:CE bit to disable the channel operation.
2. Clear each bit of DCSRn register to initialize the channel status flag.
3. Set the transfer source address (to be used when the transfer starts) in the DSARn register.
4. Set the transfer destination address (to be used when the transfer starts) in the DDARn register.
5. Set the transfer count in the DTCRn register. This count must be 1 or a larger value.
6. 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 the "Chapter: Generation and Clearing of DMA Transfer Requests" about the ICSEL and IORR registers.)
7. Set the DCCRn register. During this time, the channel operation is enabled when the DCCRn: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 DSARn:DSA.

Set the transfer destination address (to be used when the transfer starts) using the DDARn:DDA.

Align the transfer source and destination addresses based on the transfer size (DDCRn: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 DTCRn.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 DCCRn:CE.

- Disable the channel operation (DCCRn:CE = 0)
- Enable the channel operation (DCCRn:CE = 1)

When the software is selected at the transfer request source and when the DCCRn:CE bit is set, the channel operation is enabled and data transfer is started.

Interrupt Permission Setting

Enable an interrupt during abnormal completion, using the DCCRn:AIE.

- Disable an abnormal completion interrupt (DCCRn:AIE = 0)
- Enable an abnormal completion interrupt (DCCRn:AIE = 1)

Using the DCCRn: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 (DCCRn:SIE = 0)
- Enable a transfer suspend interrupt during detection of transfer stop request (DCCRn:SIE = 1)

Enable an interrupt during normal completion, using the DCCRn:NIE.

- Disable a normal completion interrupt (DCCRn:NIE = 0)
- Enable a normal completion interrupt (DCCRn:NIE = 1)

Transfer Request Source Setting

Set the transfer request source to accept a transfer request using the DCCRn:RS.

- Request by software (DCCRn:RS = 00)
- Request by an interrupt (DCCRn:RS = 01)

Transfer Mode Setting

Set the DMA transfer mode using the DCCRn:TM.

- Block transfer (DCCRn:TM = 00)
- Burst transfer (DCCRn: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 combination (5).

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 (DCCRn:RS[1:0])	Transfer Source (DSAR)	Transfer Destination (DDAR)		
(1)	Request by software (DCCRn:RS[1:0] = 00)	Any combination		Supported	ST= 0, DT= 0
(2)	Peripheral interrupt (DCCRn:RS[1:0] = 01)	[1]	[2]	Supported	ST= 1, DT= 0
(3)		[2]	[1]	Supported	ST= 0, DT= 1
(4)		[1]	[1]	Supported	ST= 0, DT= 1
(5)		[2]	[2]	Not supported	-

[1]: Address range of the peripheral under control of 16-bit peripheral bus or 32-bit peripheral bus

[2]: 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.

Transfer Address Reload Setting

Using the DCCRn: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.) (DCCRn:SAR=0)
- The transfer source address is returned to the initial value at the end of transfer. (DCCRn:SAR=1)

Using the DCCRn: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.) (DCCRn:DAR=0)
- The transfer destination address is returned to the initial value at the end of transfer. (DCCRn:DAR=1)

Transfer Address Update Setting

Using the DCCRn:SAC, set the updating of transfer source address for DMA transfer.

- Address is increased. (DCCRn:SAC = 00)
- Address is decreased. (DCCRn:SAC = 01)
- Address is fixed. (DCCRn:SAC = 11)

Using the DCCRn.DAC, set the updating of transfer destination address for DMA transfer.

- Address is increased. (DCCRn:DAC = 00)
- Address is decreased. (DCCRn:DAC = 01)
- Address is fixed. (DCCRn:DAC = 11)

Transfer Count Reload Setting

Using the DCCRn: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.) (DCCRn:TCR=0)
- The transfer count is returned to the initial value at the end of transfer. (DCCRn: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 DCCRn.TS, set the size of data to be sent by a single DMA transfer (8-bit/16-bit/32-bit).

- 8-bit (DCCRn:TS = 00)
- 16-bit (DCCRn:TS = 01)
- 32-bit (DCCRn:TS = 10)

Using the DCCRn: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 (DCCRn:TS), is transferred for the number of times being set by the block size.

8.5.3 Operations

This section explains DMAC operations.

This section explains the DMAC operations as follows.

- Channel status check
- Data transfer

Channel Status Check

Each DMAC channel status can be checked using the DCSRn register.

- When the channel operation is enabled (the channel is active), the DCSRn:CA bit is "1". When the channel is stopped, its status is shown as "0".
- If data transfer terminates abnormally, the DCSRn:AC bit is set to "1".
- If data transfer is suspended by the transfer stop request, the DCSRn:SP bit is set to "1".
- When data transfer terminates normally, the DCSRn:NC bit is set to "1".

Data writing to the DCSRn:CA bit is ignored.

The DCSRn:AC, DCSRn:SP, and DCSRn: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 DCCRn: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.

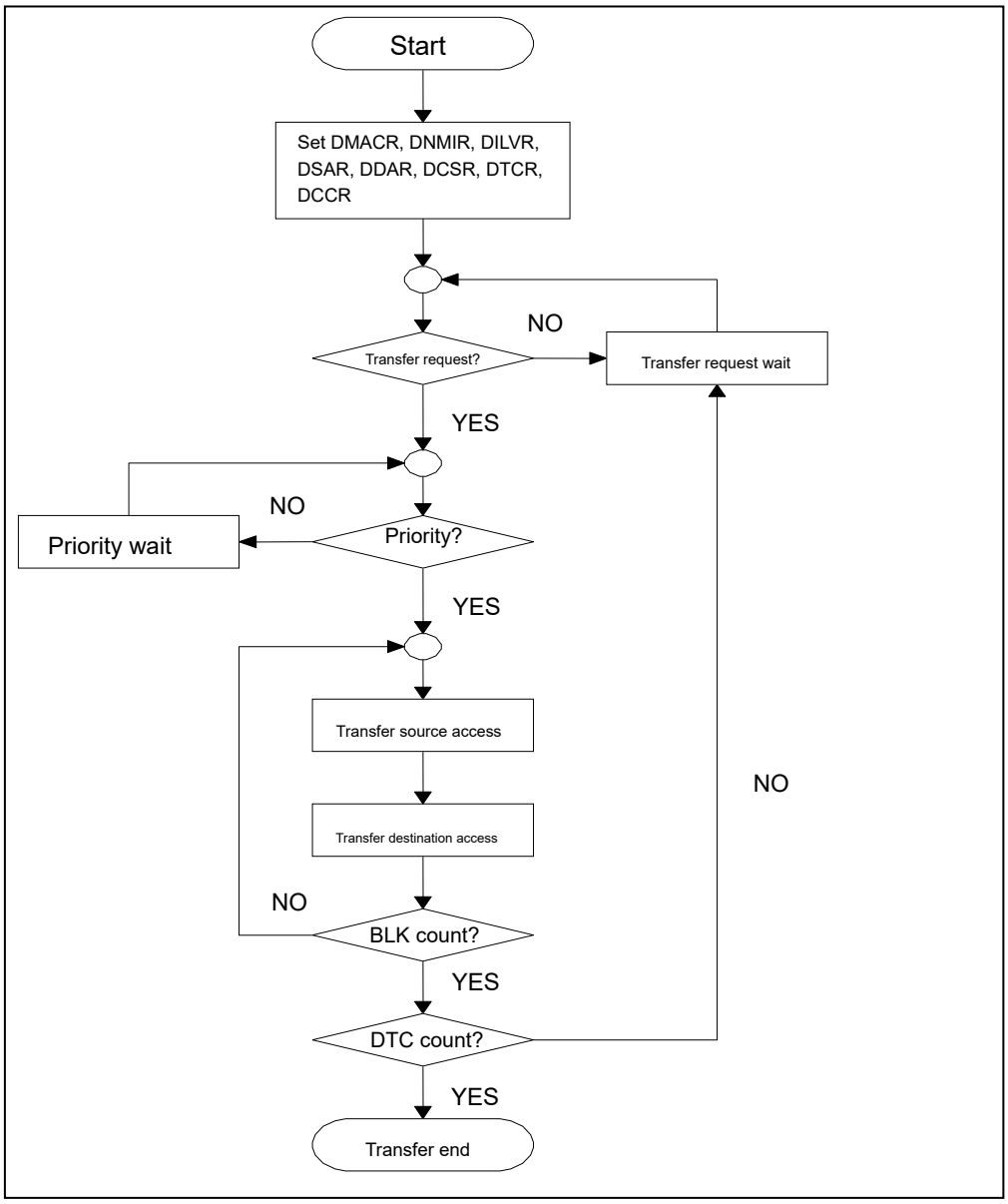
- Transfer Mode

  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 DCCRn:TS bit is transferred for the number of times being set by the block size.

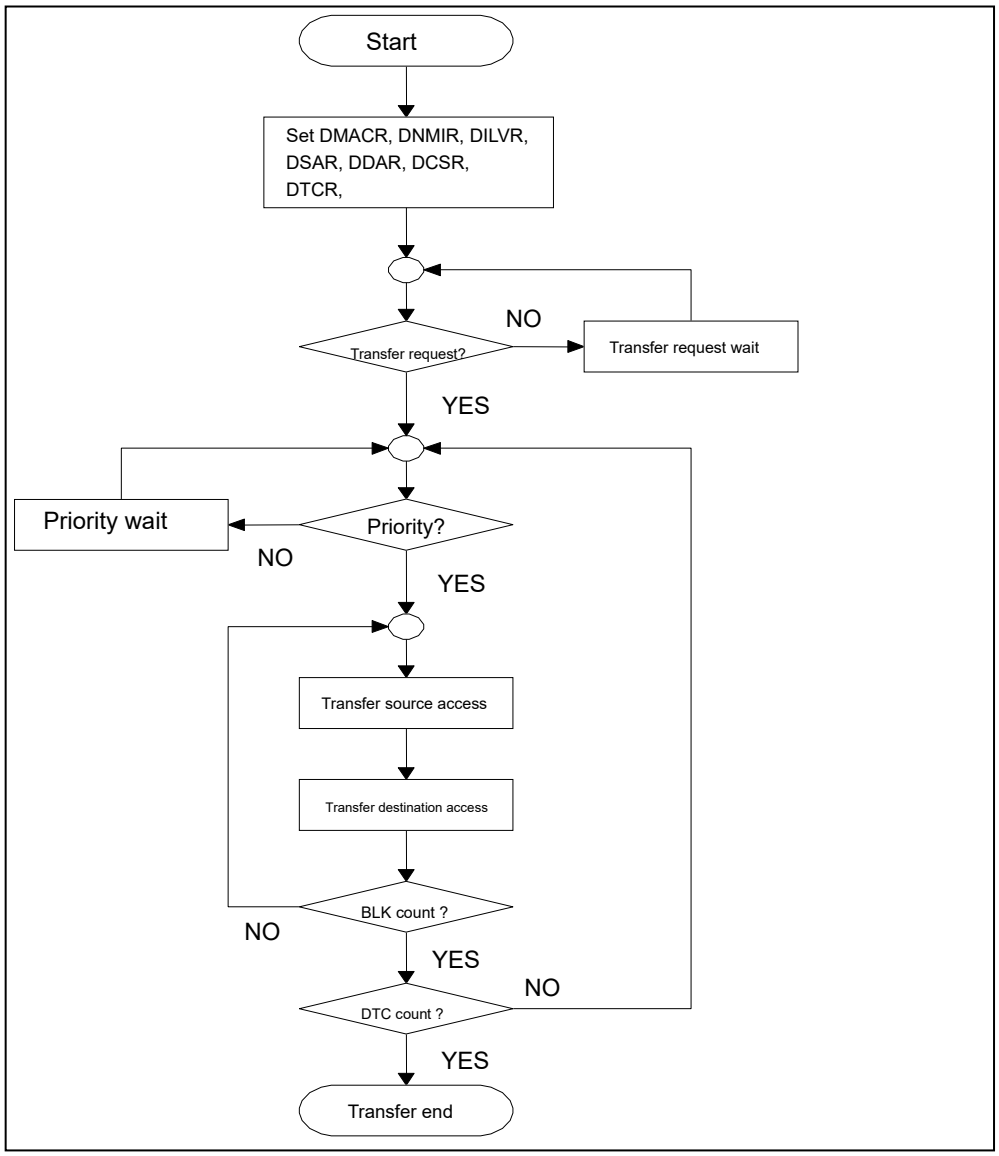
Figure 8-3. Each Transfer Mode (Block Transfer)



☐ Burst Transfer Mode

1-time transfer request causes the continuous data transfer until the end of transfer. (Data having the size set by the DCCRn:TS bit is transferred continuously for the block size × number of transfer times.)

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 DCCRn: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 DCCRn:CE bit is cleared automatically.

  - ☐ Request by Interrupt

    If the channel operation is enabled (DCCRn: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 the section "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	Remarks
Request by Software	Set the DCCRn:CE bit to "1".	Set the DCCRn:CE bit to "1".	-
Request by Interrupt	Edge detection	Edge detection	-

Also, the relationship between the detected transfer request and the DMACR:DME and DCCRn: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)

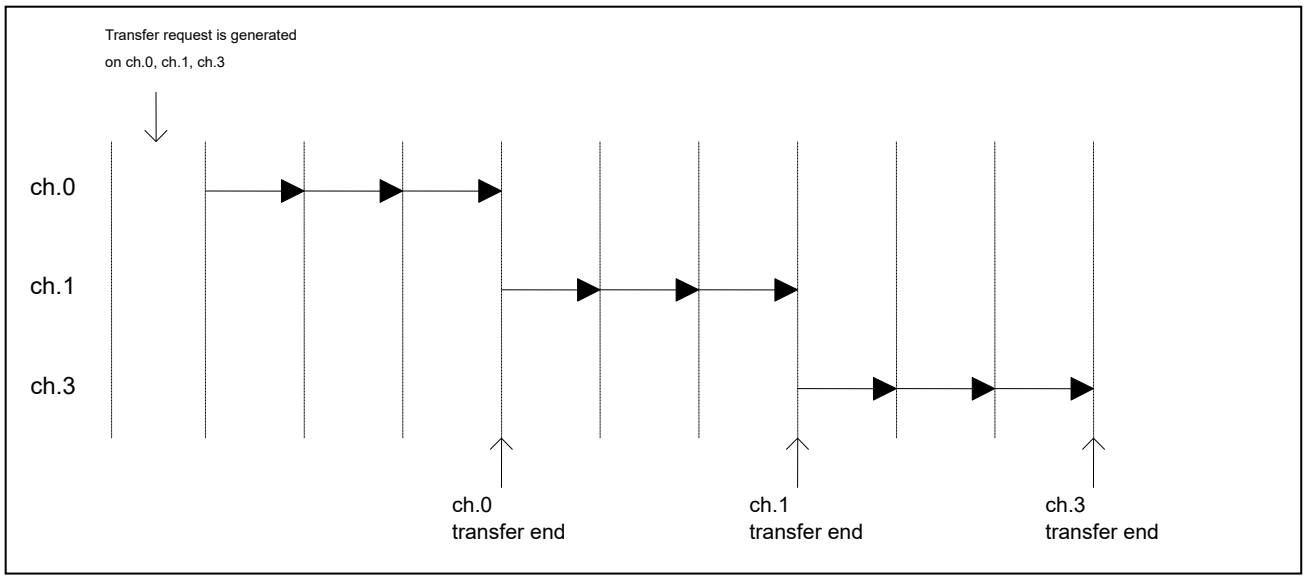
    The channel priority is fixed in the sequence of "ch.0 > ch.1 > ch.2 > ch.3". The following gives an example.

    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 data transfer occurs 3 times.

Figure 8-5. Data Transfer Example 1 If Channel Priority Is Fixed

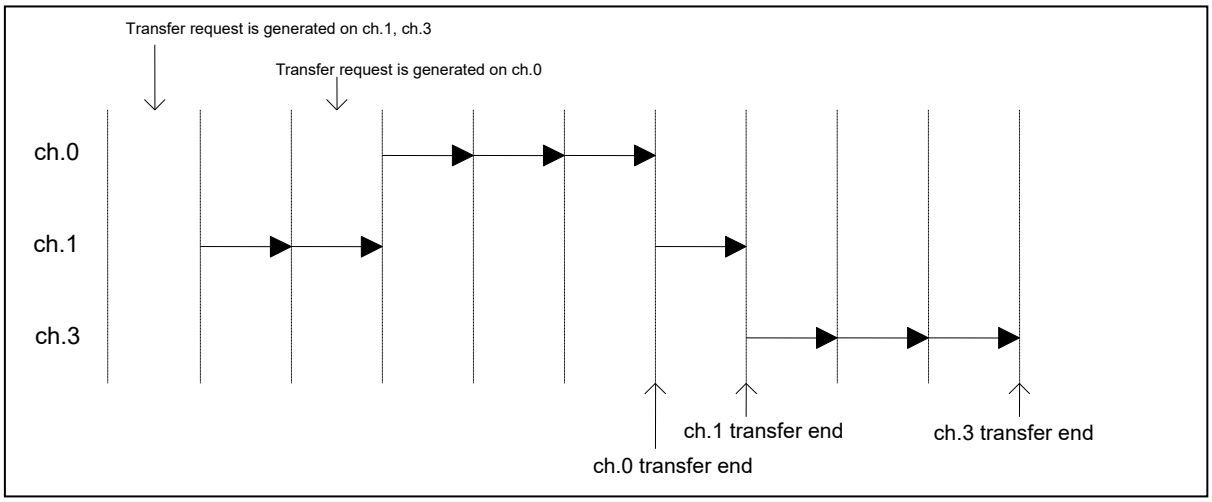


Example 2: If transfer requests are issued simultaneously for ch.1 and ch.3 and if a transfer request on ch.0 is issued during data transfer on ch.1, the data transfer on ch.1 is temporarily stopped and data transfer on ch.0 is started. During this time, the channel transition occurs in units of blocks. 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 occurs 3 times.

Figure 8-6. Data Transfer Example 2 If Channel Priority Is Fixed



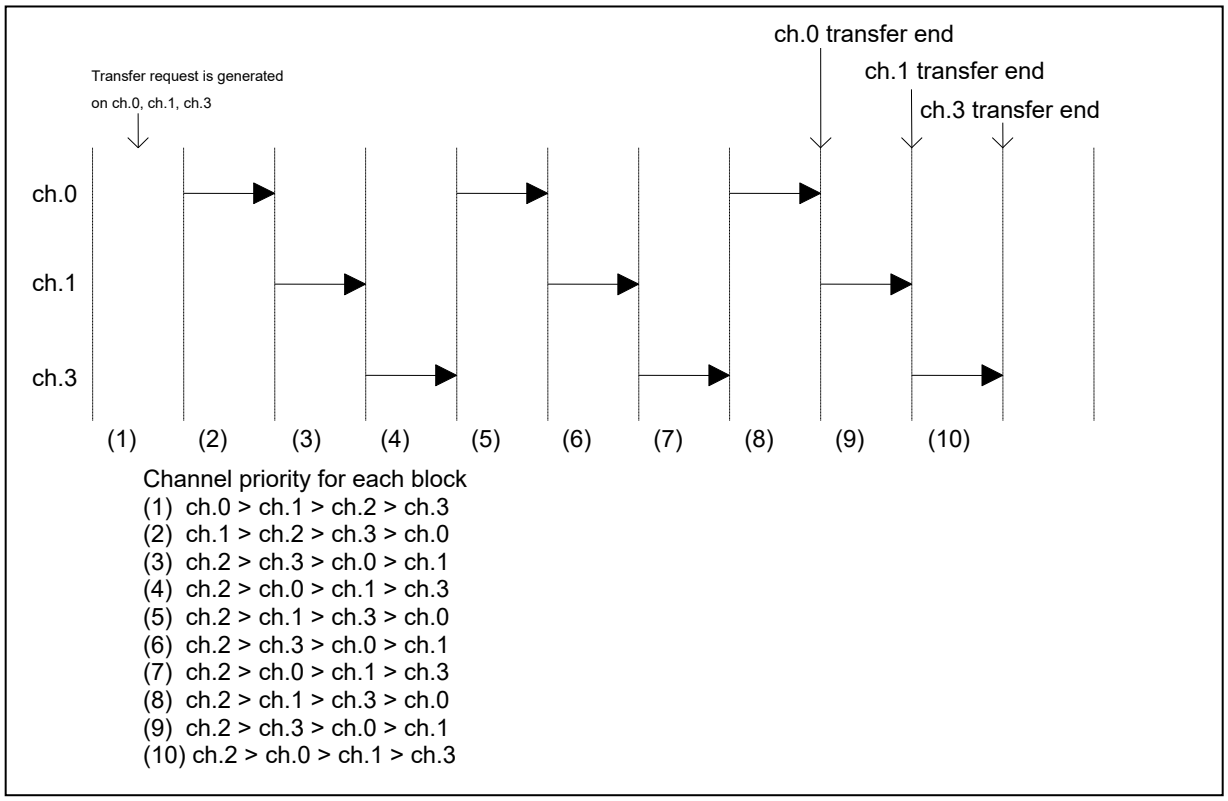
☐ Round Robin (DMACR:AT = 1)

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 occurs 3 times.

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 DCCRn: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 (DCCRn:TS). If fixed, the address value does not change. Table 5-4 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.

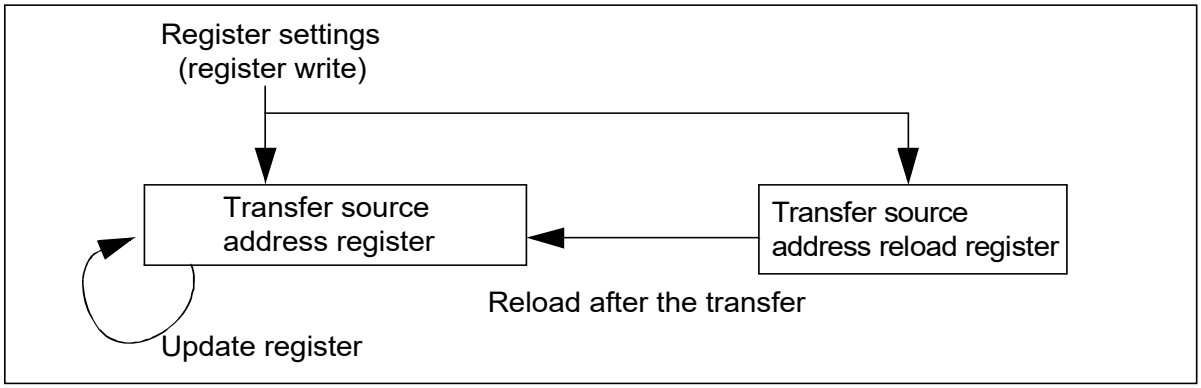
- Reloading of Transfer Source Address

  If the reloading of transfer source address has been set, the DSARn:DSA bit is returned to the initial value after the data transfer.

  If the reloading of transfer source address is disabled, the DSARn:DSA bit indicates the next access address of the last address after the current data transfer.

  If the specified number of times of transfer is suspended or abnormally terminated, the DSARn: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



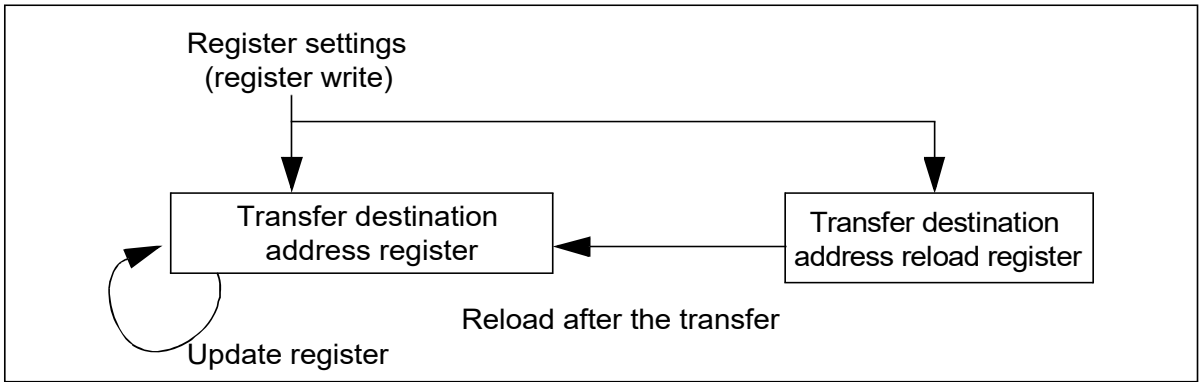
- Reloading of Transfer Destination Address Register

  If the reloading of the transfer destination address has been set, the DDARn:DDA bit is returned to the initial value after the data transfer.

  If the reloading of the transfer destination address is disabled, the DDARn:DDA bit indicates the next access address of the last address after the current data transfer.

  If the specified number of times of transfer is suspended or abnormally terminated, the DDARn: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



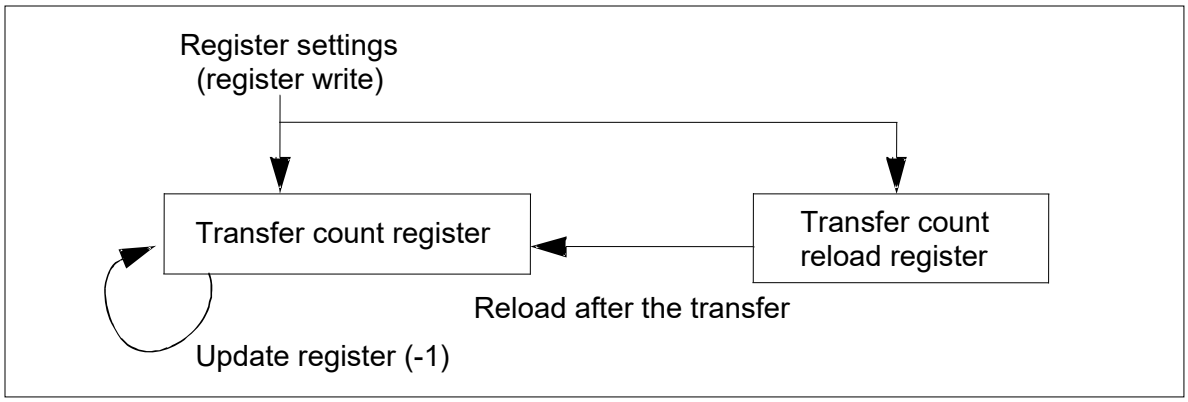
- Reloading of Transfer Count

  If the reloading of the transfer count has been set, the DTCRn:DTC bit is returned to the initial value after the data transfer.

  If reloading of the transfer count is disabled, the DTCRn:DTC bit is set to "0" after the data transfer.

  If the specified number of times of transfer is suspended or abnormally terminated, the DTCRn:DTC bit indicates the remaining transfer count regardless of the reload setting of the transfer count.

Figure 8-10. Reloading of Transfer Count Address Register



The DCCRn: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. DCCRn:CE Bit at the End of Transfer

	Software request	Non-software request
If the reloading of transfer count is set	The DCCRn:CE bit is cleared	The DCCRn:CE bit is not cleared
If the reloading of transfer count is disabled	The DCCRn:CE bit is cleared	The DCCRn: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 DCCRn: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 DMACR:DME 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 DCCRn:CE bit is cleared

  If the DCCRn: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 DCCRn:CE bit is cleared, the already detected transfer request is cleared. To restart data transfer, set the DCCRn: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:

  - ☐ Multi-function serial interface

    If a PE, FRE, or ORE flag is set

  - ☐ LIN-UART

    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 (DCSRn) is set to "1".
  - ☐ The CE bit of DMA channel control registers (DCCRn) 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 flags described in paragraphs (A) and (B) to make the transfer stop request invalid.
  2. Set the SP bit of DMA channel status registers (DCSRn) of the corresponding channel to "0".
  3. Set the CE bit of DMA channel control registers (DCCRn) 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 (DTCRn:DTC) end. When terminated normally, the DCSRn:NC bit of the corresponding channel is set. Also, the DCCRn: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 DCCRn:CE bit of the channel is not cleared.

    If the transfer count (DTCRn:DTC) is "0" and if the DCCRn:CE bit of the corresponding channel is set to "1", the DCSRn:NC bit is set in the similar way as for the normal termination. Before setting the DCCRn:CE bit to "1", be sure to set the DTCRn: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 DCSRn:AC bit of the corresponding channel is set. Also, the DCCRn:CE bit is cleared and data transfer is stopped.

    The items not allowed to set to registers are listed below:

    1. Transfer mode : DCCRn:TM = 10
    2. Transfer source address count : DCCRn:SAC = 10
    3. Transfer destination address count : DCCRn:DAC = 10
    4. Transfer size : DCCRn:TS = 11
    5. Demand transfer mode by software request : DCCRn:RS = 00 and DCCRn: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 (DCSRn) to check the interrupt request cause or to clear the interrupt request.

  - ☐ Interrupt Request at Normal Termination

    If the normal termination interrupt of a channel is enabled (DCCRn:NIE=1), the DMAC issues the interrupt request at the normal termination.

    However, the DCSRn:NC bit of the corresponding channel must be set regardless of the normal termination interrupt setting (DCCRn:NIE).

    Clear the interrupt request by clearing the DCSRn:NC bit of the corresponding channel.

  - ☐ Interrupt Request at Abnormal Termination

    If the abnormal termination interrupt of a channel is enabled (DCCRn:AIE=1), the DMAC issues the interrupt request at the abnormal termination. However, the DCSRn:AC bit of the corresponding channel is set regardless of the abnormal termination interrupt (DCCRn:AIE) setting.

    Clear the interrupt request by clearing the DCSRn: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 (DCCRn:AIE=1), the DMAC issues the interrupt request if data transfer is suspended by a transfer stop request. However, the DCSRn:SP bit of the corresponding channel is set regardless of the transfer suspension interrupt (DCCRn:SIE) settings.

    Clear the interrupt request by clearing the DCSRn: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 NMIH bit is set to "0", DMAC sets NMIH flag when user NMI occurs and suppresses DMA transfer after the block unit transfer is done.

    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

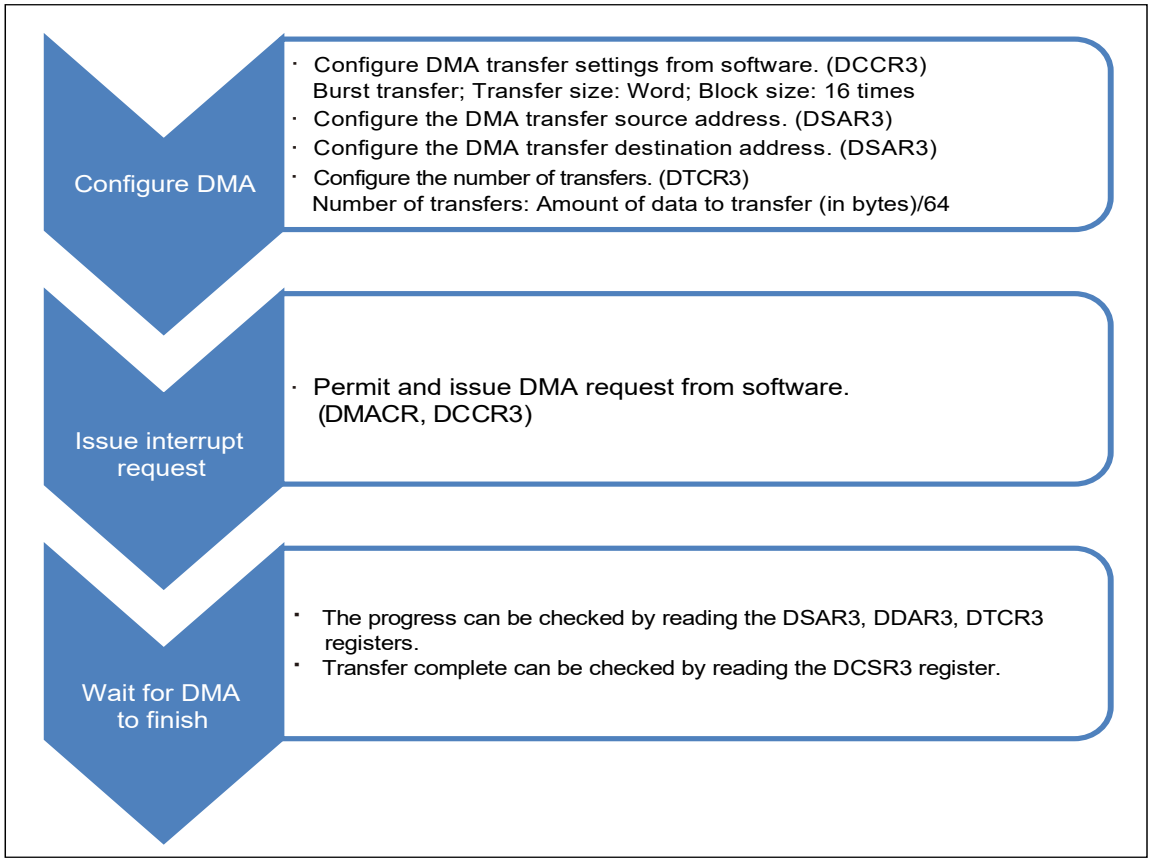
LVL[4:0]	DMA Suppress Control
11111	Suppresses the DMA transfer when any peripheral interrupt request is issued. (initial value)
11110	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than $1E_H$ is issued.
11101	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than $1D_H$ is issued.
11100	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than $1C_H$ is issued.
11011	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than $1B_H$ is issued.
11010	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than $1A_H$ is issued.
11001	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 19_H is issued.
11000	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 18_H is issued.
10111	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 17_H is issued.
10110	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 16_H is issued.
10101	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 15_H is issued.
10100	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 14_H is issued.
10011	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 13_H is issued.
10010	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 12_H is issued.
10001	Suppresses the DMA transfer when a peripheral interrupt request having a level higher than 11_H is issued.
10000	Does not suppress the DMA transfer when a peripheral interrupt request is issued.

8.6 DMA Usage Examples

DMA usage examples are shown.

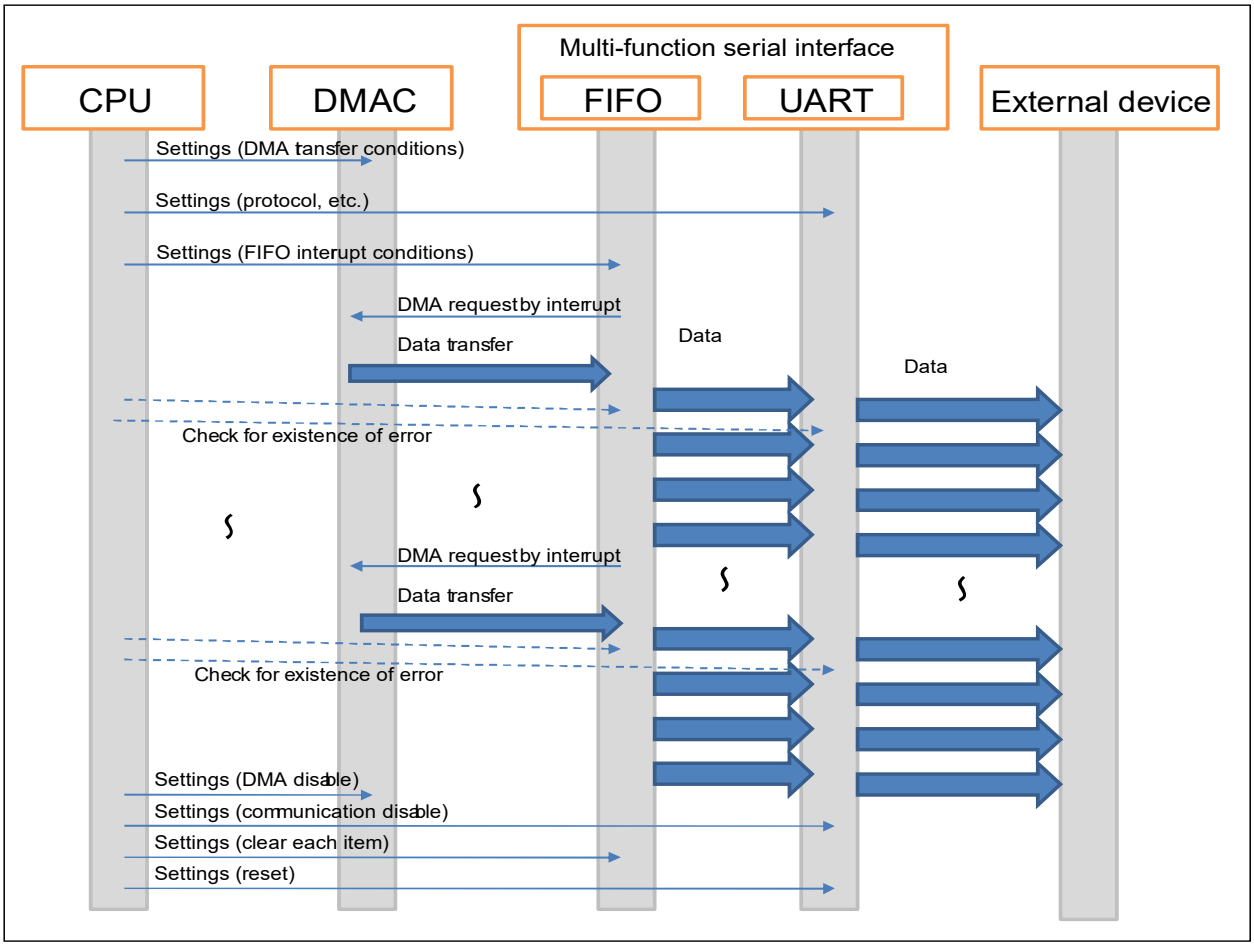
The following gives an example of memcpy instruction in every 64-byte data using the DMA. This is the simplest DMA transfer example.

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 series 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.

Transfer Request Generation Setting

For each 16-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.

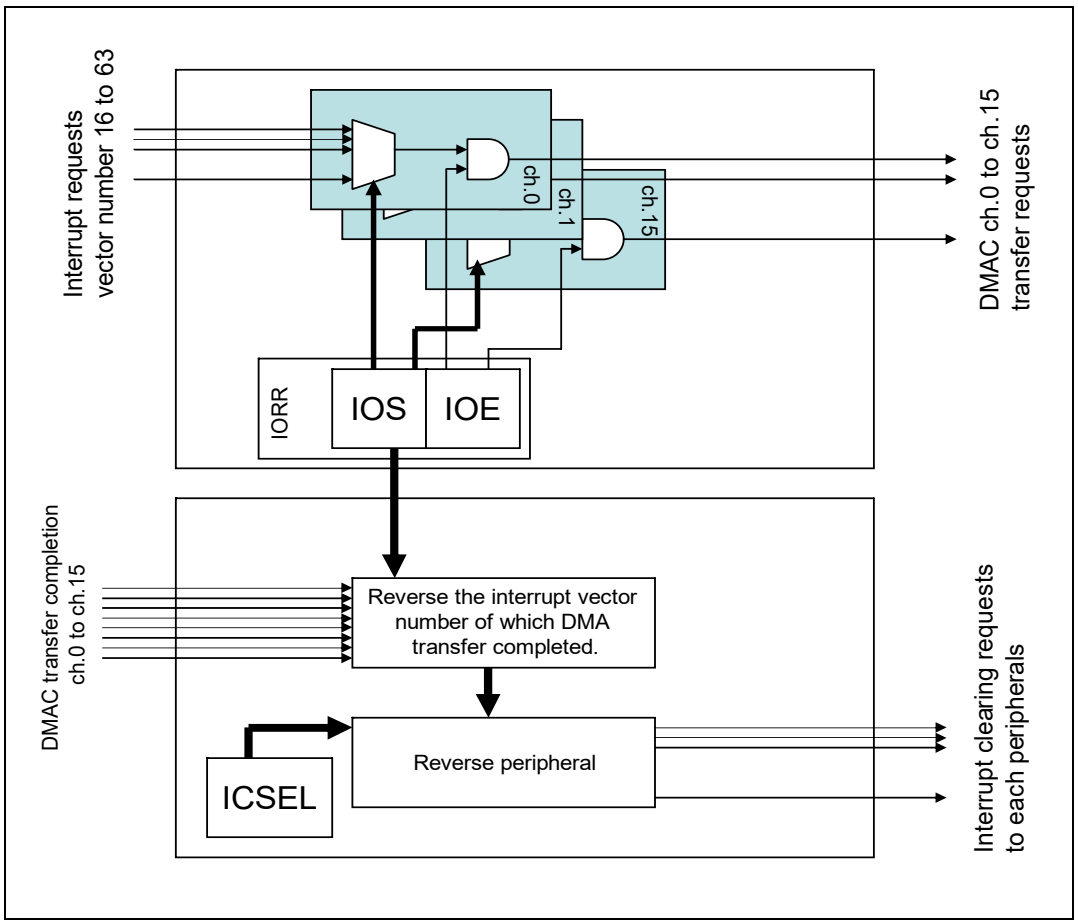
Interrupt Clearing Setting

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. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0400	ICSEL0	ICSEL1	ICSEL2	ICSEL3	DMA clear request register 0 (for vector number #16) DMA clear request register 1 (for vector number #17) DMA clear request register 2 (for vector number #18) DMA clear request register 3 (for vector number #19)
0x0404	ICSEL4	ICSEL5	ICSEL6	ICSEL7	DMA clear request register 4 (for vector number #38) DMA clear request register 5 (for vector number #39) DMA clear request register 6 (for vector number #40) DMA clear request register 7 (for vector number #41)
0x0408	ICSEL8	ICSEL9	ICSEL10	ICSEL11	DMA clear request register 8 (for vector number #42) DMA clear request register 9 (for vector number #43) DMA clear request register 10 (for vector number #44) DMA clear request register 11 (for vector number #46)
0x040C	ICSEL12	ICSEL13	ICSEL14	ICSEL15	DMA clear request register 12 (for vector number #47) 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	Reserved	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)
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

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0498	IORR8	IORR9	IORR10	IORR11	IO transfer request register 8 IO transfer request register 9 IO transfer request register 10 IO transfer request register 11
0x049C	IORR12	IORR13	IORR14	IORR15	IO transfer request register 12 IO transfer request register 13 IO transfer request register 14 IO transfer request register 15

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.

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 7

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #16).

EISEL[2:0]	Clear Target
000	External interrupt 0
001	External interrupt 1
010	External interrupt 2
011	External interrupt 3
100	External interrupt 4
101	External interrupt 5
110	External interrupt 6
111	External interrupt 7

9.4.2 DMA Request Clear Register 1: ICSEL1 (Interrupt Clear Select Register 1)

The bit configuration of DMA request clear register 1 is shown below.

ICSEL1: Address 0401_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 8 to 15

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #17).

EISEL[2:0]	Clear Target
000	External interrupt 8
001	External interrupt 9
010	External interrupt 10
011	External interrupt 11
100	External interrupt 12
101	External interrupt 13
110	External interrupt 14
111	External interrupt 15

9.4.3 DMA Request Clear Register 2: ICSEL2 (Interrupt Clear Select Register 2)

The bit configuration of DMA request clear register 2 is shown below.

ICSEL2: Address 0402_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							RTSEL0
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] RTSEL0 (Reload Timer Selection): Interrupt clear selection bit for reload timer 0/1

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #18).

Note:

An interrupt request signal is shared by reload timer ch.0 and ch.4, and another one is shared by reload timer ch.1 and ch.5.Therefore, ch.4 must not assert interrupt request when ch.0 is requesting DMA transfer. Ch.5 must not assert interrupt request when ch.1 is requesting DMA transfer as well.

RTSEL0	Clear Target
0	Reload timer 0
1	Reload timer 1

9.4.4 DMA Request Clear Register 3: ICSEL3 (Interrupt Clear Select Register 3)

The bit configuration of DMA request clear register 3 is shown below.

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

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #19).

Note:

An interrupt request signal is shared by reload timer ch.2 and ch.6.Therefore, ch.6 must not assert interrupt request when ch.2 is requesting DMA transfer.

RTSEL1	Clear Target
0	Reload timer 2
1	Reload timer 3

9.4.5 DMA Request Clear Register 4: ICSEL4 (Interrupt Clear Select Register 4)

The bit configuration of DMA request clear register 4 is shown below.

ICSEL4: Address 0404_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							SG_RX_SEL0
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] SG_RX_SEL0 (SG_RX Selection0): Interrupt clear selection bit for sound generator ch.0 / LIN-UART ch.7 reception completion

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #38).

SG_RX_SEL0	Clear Target
0	Sound generator ch.0
1	LIN-UART ch.7 reception completion

9.4.6 DMA Request Clear Register 5: ICSEL5 (Interrupt Clear Select Register 5)

The bit configuration of DMA request clear register 5 is shown below.

ICSEL5: Address 0405_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							SG_RX_SEL1
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] SG_RX_SEL1 (SG_RX Selection1): Interrupt clear selection bit for sound generator ch.1 / LIN-UART ch.7 transmission completion

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #39).

SG_RX_SEL1	Clear Target
0	Sound generator ch.1
1	LIN-UART ch.7 transmission completion

9.4.7 DMA Request Clear Register 6: ICSEL6 (Interrupt Clear Select Register 6)

The bit configuration of DMA request clear register 6 is shown below.

ICSEL6: Address 0406_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					PPGSEL0[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] PPGSEL0[2:0] (PPG Selection0): Interrupt clear selection bits for PPG0, 1, 10, 11, 20, 21

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #40).

PPGSEL0[2:0]	Clear Target
000	PPG0
001	PPG1
010	PPG10
011	PPG11
100	PPG20
101	PPG21
110	Reserved (No interrupt is cleared)
111	Reserved (No interrupt is cleared)

Note:

Setting PPGSEL0[2:0] = "3'b110","3'b111" is prohibited. During this setting, no interrupt clear will be selected.

9.4.8 DMA Request Clear Register 7: ICSEL7 (Interrupt Clear Select Register 7)

The bit configuration of DMA request clear register 7 is shown below.

ICSEL7: Address 0407_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					PPGSEL1[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] PPGSEL1[2:0] (PPG Selection1): Interrupt clear selection bits for PPG2, 3, 12, 13, 22, 23

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #41).

PPGSEL1[2:0]	Clear Target
000	PPG2
001	PPG3
010	PPG12
011	PPG13
100	PPG22
101	PPG23
110	Reserved (No interrupt is cleared)
111	Reserved (No interrupt is cleared)

Note:

Setting PPGSEL1[2:0] = "3'b110","3'b111" is prohibited. During this setting, no interrupt clear will be selected.

9.4.9 DMA Request Clear Register 8: ICSEL8 (Interrupt Clear Select Register 8)

The bit configuration of DMA request clear register 8 is shown below.

ICSEL8: Address 0408_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PPGSEL2[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] PPGSEL2[1:0] (PPG Selection2): Interrupt clear selection bits for PPG4, 5, 14, 15

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #42).

PPGSEL2[1:0]	Clear Target
00	PPG4
01	PPG5
10	PPG14
11	PPG15

9.4.10 DMA Request Clear Register 9: ICSEL9 (Interrupt Clear Select Register 9)

The bit configuration of DMA request clear register 9 is shown below.

ICSEL9: Address 0409_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PPGSEL3[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] PPGSEL3[1:0] (PPG Selection3): Interrupt clear selection bits for PPG6, 7, 16, 17

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #43).

PPGSEL3[1:0]	Clear Target
00	PPG6
01	PPG7
10	PPG16
11	PPG17

9.4.11 DMA Request Clear Register 10: ICSEL10 (Interrupt Clear Select Register 10)

The bit configuration of DMA request clear register 10 is shown below.

ICSEL10: Address 040A$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PPGSEL4[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] PPGSEL4[1:0] (PPG Selection4): Interrupt clear selection bits for PPG8, 9, 18, 19

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #44).

PPGSEL4[1:0]	Clear Target
00	PPG8
01	PPG9
10	PPG18
11	PPG19

9.4.12 DMA Request Clear Register 11: ICSEL11 (Interrupt Clear Select Register 11)

The bit configuration of DMA request clear register 11 is shown below.

ICSEL11: Address 040B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						PMSTSEL[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] PMSTSEL[1:0] (PLL, Main, Sub Timer Selection): Interrupt clear selection for main timer / sub timer / PLL timer / multi-function serial ch.8 transmission completion

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #46).

PMSTSEL[1:0]	Clear Target
00	Main timer
01	Sub timer
10	PLL timer
11	Multi-function serial ch.8 transmission completion

9.4.13 DMA Request Clear Register 12: ICSEL12 (Interrupt Clear Select Register 12)

The bit configuration of DMA request clear register 12 is shown below.

ICSEL12: Address 040C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						SG_RX_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] SG_RX_SEL[1:0] (SG_RX Selection): Interrupt clear selection for SG4 / multi-function serial ch.9 reception completion

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #47).

SG_RX_SEL[1:0]	Clear Target
00	Reserved (No interrupt is cleared)
01	Sound generator ch.4
10	Multi-function serial ch.9 reception completion
11	Reserved (No interrupt is cleared)

9.4.14 DMA Request Clear Register 13: ICSEL13 (Interrupt Clear Select Register 13)

The bit configuration of DMA request clear register 13 is shown below.

ICSEL13: Address 040D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							ICUSEL0
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] ICUSEL0: Interrupt clear selection for ICU ch.0, ch.6

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #52).

ICUSEL0	Clear Target
0	ICU ch.0
1	ICU ch.6

9.4.15 DMA Request Clear Register 14: ICSEL14 (Interrupt Clear Select Register 14)

The bit configuration of DMA request clear register 14 is shown below.

ICSEL14: Address 040E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							ICUSEL1
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] ICUSEL1: Interrupt clear selection for ICU ch.1, ch.7

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #53).

ICUSEL1	Clear Target
0	ICU ch.1
1	ICU ch.7

9.4.16 DMA Request Clear Register 15: ICSEL15 (Interrupt Clear Select Register 15)

The bit configuration of DMA request clear register 15 is shown below.

ICSEL15: Address 040F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							ICUSEL2
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] ICUSEL2: Interrupt clear selection for ICU ch.2, ch.8

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #54).

ICUSEL2	Clear Target
0	ICU ch.2
1	ICU ch.8

9.4.17 DMA Request Clear Register 16: ICSEL16 (Interrupt Clear Select Register 16)

The bit configuration of DMA request clear register 16 is shown below.

ICSEL16: Address 0410_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							ICUSEL3
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] ICUSEL3: Interrupt clear selection for ICU ch.3, ch.9

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #55).

ICUSEL3	Clear Target
0	ICU ch.3
1	ICU ch.9

9.4.18 DMA Request Clear Register 17: ICSEL17 (Interrupt Clear Select Register 17)

The bit configuration of DMA request clear register 17 is shown below.

ICSEL17: Address 0411_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							ICUSEL4
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] ICUSEL4: Interrupt clear selection for ICU ch.4, ch.10

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #56).

ICUSEL4	Clear Target
0	ICU ch.4
1	ICU ch.10

9.4.19 DMA Request Clear Register 18: ICSEL18 (Interrupt Clear Select Register 18)

The bit configuration of DMA request clear register 18 is shown below.

ICSEL18: Address 0412_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							ICUSEL5
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] ICUSEL5: Interrupt clear selection for ICU ch.5, ch.11

This bit is used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #57).

ICUSEL5	Clear Target
0	ICU ch.5
1	ICU ch.11

9.4.20 DMA Request Clear Register 19: ICSEL19 (Interrupt Clear Select Register 19)

The bit configuration of DMA request clear register 19 is shown below.

ICSEL19: Address 0413_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					OCUSEL0[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] OCUSEL0[2:0] (OCU Selection0): Interrupt clear selection bits for OCU0, 1, 6, 7, 10, 11

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #58).

OCUSEL0[2:0]	Clear Target
000	OCU0
001	OCU1
010	OCU6
011	OCU7
100	OCU10
101	OCU11
110	Reserved (No interrupt is cleared)
111	Reserved (No interrupt is cleared)

Note:

Setting OCUSEL0[2:0] = "3'b110","3'b111" is prohibited. During this setting, no interrupt clear will be selected.

9.4.21 DMA Request Clear Register 20: ICSEL20 (Interrupt Clear Select Register 20)

The bit configuration of DMA request clear register 20 is shown below.

ICSEL20: Address 0414_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					OCUSEL1[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] OCUSEL1[2:0] (OCU Selection1): Interrupt clear selection bits for OCU2, 3 , 4, 5, 8, 9

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #59).

OCUSEL1[2:0]	Clear Target
000	OCU2
001	OCU3
010	OCU4
011	OCU5
100	OCU8
101	OCU9
110	Reserved (No interrupt is cleared)
111	Reserved (No interrupt is cleared)

Note:

Setting OCUSEL1[2:0] = "3'b110","3'b111" is prohibited. During this setting, no interrupt clear will be selected.

9.4.22 DMA Request Clear Register 21: ICSEL21 (Interrupt Clear Select Register 21)

The bit configuration of DMA request clear register 21 is shown below.

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/ SG2

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #60).

BT_SG_SEL0[1:0]	Clear Target
00	Base Timer0 IRQ0
01	Base Timer0 IRQ1
10	Sound generator ch.2
11	Reserved (No interrupt is cleared)

Note:

Setting BT_SG_SEL0[1:0] ="2'b11" is prohibited. During this setting, no interrupt clear will be selected.

9.4.23 DMA Request Clear Register 22: ICSEL22 (Interrupt Clear Select Register 22)

The bit configuration of DMA request clear register 22 is shown below.

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_SEL [1:0] (BT_SG_Selection1): Interrupt clear selection bits for Base Timer1 IRQ0, IRQ1/ SG3

These bits are used to select the peripheral that has generated the interrupt to be cleared (assigned to interrupt vector number #61).

BT_SG_SEL1[1:0]	Clear Target
00	Base Timer1 IRQ0
01	Base Timer1 IRQ1
10	Sound generator ch.3
11	Reserved (No interrupt is cleared)

Note:

Interrupts for XBS RAM single-bit error occurrence and backup RAM single-bit error occurrence shall not be covered as they do not support the IIOC.
Setting BT_SG_SEL1[1:0] ="2'b11" is prohibited. During this setting, no interrupt clear will be selected.

9.4.24 IO Transfer Request Setting Register 0 to 15: IORR0 to 15 (IO triggered DMA Request Register for ch. 0 to 15)

The bit configuration of IO transfer request setting register 0 to 15 is shown below.

If the DMA transfer request generation factor is specified as a peripheral interrupt request, these registers are used to 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 15: Address 0490_H to $049F_H$ (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

[bit7] Reserved

Always write "0" to this bit. The read value is always "0".

[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	No DMA transfer request output -- The interrupt request generated by the peripheral is not used as a DMA transfer request (Initial value).
1	DMA transfer request output

[bit5 to bit0] IOS[5:0] (IO triggered DMA transfer request Select): Transfer request selection bits

These registers 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 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

This section explains the configuration of the generation and clearing of DMA transfer requests.

The operating sequence is as follows:

1. On the IORR, set the interrupt vector number of the transfer request source peripheral and the IOE bit.
2. Set ICSEL if multiple peripherals is assigned to the vector number selected in step 1.
3. Set the interrupt configuration-related registers for the peripheral.
4. Configure the DMAC.

9.5.2 Notes

This section explains notes of the generation and clearing of DMA transfer requests.

- Do not change the IORR and ICSEL registers when the DMAC enables DMA transfer requests issued by peripherals.
- 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 Operation

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.

Features

- Interrupt vector on reset returned by the FixedVector function
- 0x0007_0024

Configuration

See "Figure 45-2" in "Chapter: Flash Memory" for the configuration diagram.

Registers

None.

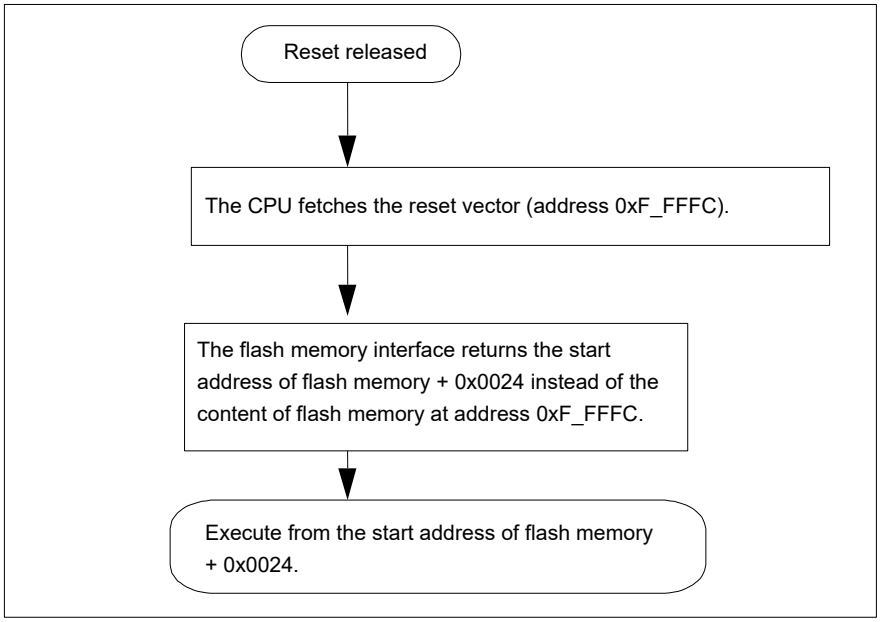
10.2 Operation

This section explains the operation of the FixedVector function.

Operation After Reset Released

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



Usage

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.

Notes

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 external bus) 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.

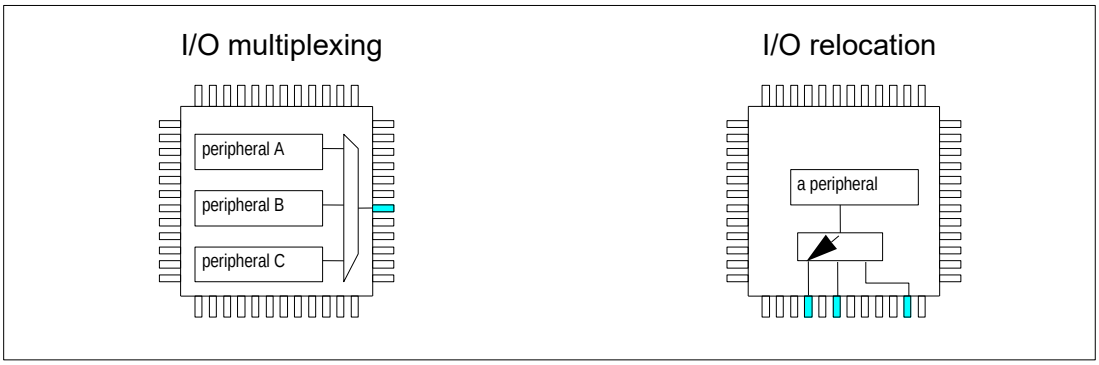
- I/O Relocation

  If one pin for one peripheral can serve multiple external pins for I/O, one of these external pins 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, I/O Relocation



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	PDR01	PDR02	PDR03	Port data register 00 to 13
0x0004	PDR04	PDR05	PDR06	PDR07	
0x0008	PDR08	PDR09	PDR10	PDR11	
0x000C	PDR12	PDR13	Reserved	Reserved	
0x0E00	DDR00	DDR01	DDR02	DDR03	Data direction register 00 to 13
0x0E04	DDR04	DDR05	DDR06	DDR07	
0x0E08	DDR08	DDR09	DDR10	DDR11	
0x0E0C	DDR12	DDR13	Reserved	Reserved	
0x0E20	PFR00	PFR01	PFR02	PFR03	Port function register 00 to 13
0x0E24	PFR04	PFR05	PFR06	PFR07	
0x0E28	PFR08	PFR09	PFR10	PFR11	
0x0E2C	PFR12	PFR13	Reserved	Reserved	
0x0E40	PDDR00	PDDR01	PDDR02	PDDR03	Input data direct read register 00 to 13
0x0E44	PDDR04	PDDR05	PDDR06	PDDR07	
0x0E48	PDDR08	PDDR09	PDDR10	PDDR11	
0x0E4C	PDDR12	PDDR13	Reserved	Reserved	
0x0E60	EPFR00	EPFR01	EPFR02	EPFR03	Extended port function register 00 to 35
0x0E64	EPFR04	EPFR05	EPFR06	EPFR07	
0x0E68	EPFR08	EPFR09	EPFR10	EPFR11	
0x0E6C	EPFR12	EPFR13	EPFR14	EPFR15	
0x0E70	EPFR16	EPFR17	EPFR18	EPFR19	
0x0E74	EPFR20	EPFR21	EPFR22	EPFR23	
0x0E78	EPFR24	EPFR25	EPFR26	EPFR27	
0x0E7C	EPFR28	EPFR29	EPFR30	EPFR31	
0x0E80	EPFR32	EPFR33	EPFR34	EPFR35	

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0E84	EPFR36	EPFR37	EPFR38	EPFR39	Extended port function register 36 to 54
0x0E88	EPFR40	EPFR41	EPFR42	EPFR43	
0x0E8C	EPFR44	EPFR45	EPFR46	EPFR47	
0x0E90	Reserved	Reserved	Reserved	Reserved	
0x0E94	EPFR52	EPFR53	EPFR54	Reserved	
0x0EA0	PPCR00	PPCR01	PPCR02	PPCR03	Port pull-up/down control register 00 to 13
0x0EA4	PPCR04	PPCR05	PPCR06	PPCR07	
0x0EA8	PPCR08	PPCR09	PPCR10	PPCR11	
0x0EAC	PPCR12	PPCR13	Reserved	Reserved	
0x0EC0	PPER00	PPER01	PPER02	PPER03	Port pull-up/down enable register 00 to 13
0x0EC4	PPER04	PPER05	PPER06	PPER07	
0x0EC8	PPER08	PPER09	PPER10	PPER11	
0x0ECC	PPER12	PPER13	Reserved	Reserved	
0x0EE0	PILR00	PILR01	PILR02	PILR03	Port input level selection register 00 to 13
0x0EE4	PILR04	PILR05	PILR06	PILR07	
0x0EE8	PILR08	PILR09	PILR10	PILR11	
0x0EEC	PILR12	PILR13	Reserved	Reserved	
0x0F00	EPILR00	EPILR01	EPILR02	EPILR03	Extended port input level selection register 00 to 13
0x0F04	EPILR04	EPILR05	EPILR06	EPILR07	
0x0F08	EPILR08	EPILR09	EPILR10	EPILR11	
0x0F0C	EPILR12	EPILR13	Reserved	Reserved	
0x0F20	PODR00	PODR01	PODR02	PODR03	Port output drive register 00 to 13
0x0F24	PODR04	PODR05	PODR06	PODR07	
0x0F28	PODR08	PODR09	PODR10	PODR11	
0x0F2C	PODR12	PODR13	Reserved	Reserved	
0x0F34	Reserved	EPODR01	EPODR02	EPODR03	Extended port output drive register 01,02,03,06,07,08
0x0F38	EPODR06	EPODR07	EPODR08	Reserved	
0x0F40	PORTEN	Reserved	Reserved	Reserved	Port input enable register

11.4.1 Port Data Register 00 to 13: PDR00-13

The bit configuration of port data register 00 to 13 is shown below.

These registers hold the output levels of the pins corresponding to individual ports that are in output mode.

PDR00-13: Address 0000_H, 0001_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[7:0] (Port): Port data setting bits

These bits set the output level of external pins P000, P001, ..., when the ports are in output mode. PDR00:P[7:0] is for external pins P007 to P000
PDR01:P[7:0] is for external pins P017 to P010
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.

PDR13:P5 is a reserved bit. Reading or writing has no effects.
PDR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.2 Data Direction Register 00 to 13: DDR00-13

The bit configuration of data direction register 00 to 13 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.

DDR00-13: Address $0E00_H$, $0E01_H$, • • • (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[7:0] (Port): Data direction selection bits

These bits set the I/O direction of external pins P000, P001, ..., when the ports are in output mode.
DDR00:P[7:0] is for external pins P007 to P000
DDR01:P[7:0] is for external pins P017 to P010
DDR02:P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	Input (Initial value)
1	Output

DDR05:P7 is a reserved bit. Reading or writing has no effects.
DDR13:P[5] is a reserved bit. Reading or writing has no effects.
DDR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.3 Port Function Register 00 to 13: PFR00-13

The bit configuration of port function register 00 to 13 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's input pin, the corresponding bit register must be set for the port function.

PFR00-13: Address $0E20_H$, $0E21_H$, • • • (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
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

[1]: Initial value of the each bits can be referred “B. I/O map” in “Appendix”.

[bit7 to bit0] P[7:0] (Port): Port function selection bits

These bits are used to set the port function.
PFR00:P[7:0] is for external pins P007 to P000
PFR01:P[7:0] is for external pins P017 to P010
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 or external bus pin(set by EPFR)

PFR09:P6 is not assigned peripheral I/O pin, peripheral output pin or external bus pin function.
PFR13:P5 is a reserved bit. Reading or writing has no effects.
PFR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.4 Input Data Direct Register 00 to 13: PDDR00-13 (Port Data Direct Register 00-13)

The bit configuration of input data direct register 00 to 13 is shown below.

These registers can always show the voltage levels of individual external pins. These registers can always be read without condition.

PDDR00-13: Address $0E40_H$, $0E41_H$, • • • (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[7:0] (Port): Read bits

These bits can be read the value of the external pin.
PDDR00:P[7:0] is for external pins P007 to P000
PDDR01:P[7:0] is for external pins P017 to P010
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	Low level
1	High level

PDDR13:P5 is a reserved bit. Reading or writing has no effects.
PDDR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.5 Port Pull-up/down Control Register 00 to 13: PPCR00-13

The bit configuration of port pull-up/down control register 00 to 13 is shown below.

These registers are used to select pull-up or pull-down for each port. These registers are functioned for input condition pins only. These registers are combined with the pull-up/down enable register (PPER) for this setting.

PPCR00-13: Address 0EA0$_H$, 0EA1$_H$, • • • (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
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

[bit7 to bit0] P[7:0] (Port): Pull-up/down control selection bits

PPCR00:P[7:0] is for external pins P007 to P000
PPCR01:P[7:0] is for external pins P017 to P010
PPCR02:P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	Pull-down
1	Pull-up (Initial value)

See "List of Pin Functions" and "I/O Circuit Types" of "Chapter: Overview" for the presence of pull-up/pull-down.

PPCR13:P5 is a reserved bit. Reading or writing has no effects.
PPCR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.6 Port Pull-up/down Enable Register 00 to 13: PPER00-13

The bit configuration of port pull-up/down enable register 00 to 13 is shown below.

These registers are used to enable pull-up or pull-down each port. These registers are functioned for input condition pins only. These registers are combined with the pull-up/down control register (PPCR) for this setting.

PPER00-13: Address $0EC0_H$, $0EC1_H$, • • • (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[7:0] (Port): Pull-up/down enable selection bits

PPER00:P[7:0] is for external pins P007 to P000
PPER01:P[7:0] is for external pins P017 to P010
PPER02:P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

P[n]	Operation
0	Pull-up/down disabled (Initial value)
1	Pull-up/down enabled

See "List of Pin Functions" and "I/O Circuit Types" of "Chapter: Overview" for the presence of pull-up/pull-down of each port.

PPER13:P5 is a reserved bit. Reading or writing has no effects.
PPER13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.7 Port Input Level Selection Register 00 to 13: PILR00-13 (Port Input Level Register 00-13)

The bit configuration of port input level selection register 00 to 13 is shown below.

These registers are used to set input levels for individual ports. Glitch input may occur at a pin. Therefore, if the pin is used to supply external input clock or trigger to a peripheral, the peripheral must be disabled.

These registers, when used, are paired with the extended port input level selection register (EPILR).

PILR00-13: Address $0EE0_H$, $0EE1_H$, • • • (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	P[7:0]							
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

[bit7 to bit0] P[7:0] (Port): Port input level selection bits

PILR00:P[7:0] is for external pins P007 to P000
PILR01:P[7:0] is for external pins P017 to P010
PILR02:P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

PILR:P[n]	EPILR:P[n]	Input level	Remarks
0	0	CMOS Schmitt VIL = 0.3Vcc VIH = 0.7Vcc	
0	1	TTL VIL = 0.8[V] VIH = 2.0[V]	
1	0	Automotive VIL = 0.5Vcc VIH = 0.8Vcc	Initial value
1	1	CMOS VIL = 0.3Vcc VIH = 0.7Vcc	

PILR13:P5 is a reserved bit. Reading or writing has no effects.
PILR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.8 Extended Port Input Level Selection Register 00 to 13: EPILR00-13 (Extended Port Input Level Register 00-13)

The bit configuration of extended port input level selection register 00 to 13 is shown below.

These registers, when used, are paired with the port input level selection register (PILR). See "11.4.7 Port Input Level Selection Register 00 to 13: PILR00-13 (Port Input Level Register 00-13)".

EPILR00-13: Address $0F00_H$, $0F01_H$, • • • (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[7:0] (Port): Extended port input level selection bits

EPILR00:P[7:0] is for external pins P007 to P000
EPILR01:P[7:0] is for external pins P017 to P010
EPILR02:P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above. For settings, see the section of PILR.

EPILR13:P5 is a reserved bit. Reading or writing has no effects.
EPILR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.9 Port Output Drive Register 00 to 13: PODR00-13

The bit configuration of port output drive register 00 to 13 is shown below.

These registers are used to set drive levels for individual ports.

PODR00-13: Address 0F20H, 0F21H, • • • 0F2CH (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[7:0] (Port): Port output drive selection bits

PODR00:P[7:0] is for external pins P007 to P000
PODR01:P[7:0] is for external pins P017 to P010
PODR02:P[7:0] is for external pins P027 to P020
(A similar process continues)
The assignment is as shown above.

11.4.9.1 Pins P010 to P036

These pins are configured by the combination with EPODR.
P010 to P036

At VCCE = 5 V

PODR:P[n]	EPODR:P[n]	Operation
0	0	1 mA
0	1	Reserved
1	0	2 mA
1-	1	2 mA

Note:

Drive level at VCCE = 3.3 V should be referred the Datasheet.

11.4.9.2 Pins P060 to P087

The drive level of a pin that doubles as the output pin for a stepping motor controller can be set to be 1 mA/ 2 mA/30 mA by the combination with an extended port output drive register (EPODR) setting.

If a pin is specified as the output pin for a stepping motor controller, the driving current at the pin is set to 30 mA regardless of the PODR register setting.

P060 to P087

PODR:P[n]	EPODR:P[n]	Operation	Remarks
0	0	1 mA	Initial value
0	1	30 mA	
1	0	2 mA	
1	1	2 mA	

11.4.9.3 Other Pins than P010 to P036 nor P060 to P087

Pins other than P010 to P036 nor P060 to P087.

PODR:P[n]	Operation
0	1 mA
1	2 mA

[Only P127, P130, P132, and P133 pins]
When the multi-function serial interface is selected and I^2C has been selected by the operational mode of the multi-function serial interface, it becomes 3mA. In other cases, the setting in the above table needs to be followed.

PODR13:P5 is a reserved bit. Reading or writing has no effects.
PODR13:P[7:6] are reserved bit in the dual clock products. Reading or writing has no effects.

11.4.10 Extended Port Output Drive Register 01-03,06-08: EPODR01-03,06-08

The bit configuration of extended port output drive register 01-03,06-08 is show below.

These bits set drive level of each pins. They are used by the combination with PODR.

EPODR01: Address 0F35H (Access: Byte, Half-word, Word)

EPODR02: Address 0F36H (Access: Byte, Half-word, Word)

EPODR03: Address 0F37H (Access: Byte, Half-word, Word)

EPODR06: Address 0F38H (Access: Byte, Half-word, Word)

EPODR07: Address 0F39H (Access: Byte, Half-word, Word)

EPODR08: Address 0F3AH (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

Refer “11.4.9 Port Output Drive Register 00 to 13: PODR00-13 ” for the configuration.

EPODR03:P7 is a reserved bit. Reading or writing has no effects.

11.4.11 Extended Port Function Register 00 to 54: EPFR00-54

The bit configuration of extended port function register 00 to 54 is show below.

These registers control switching between the peripheral and the external bus, I/O relocation and I/O multi-plexing. Unlike other port registers, these registers have an enable bit for each peripheral, rather than for each pin.

When I/O relocation is executed, glitch occurs by switching and operation may happen by recognition as a signal change. Therefore, execute I/O relocation for input neglecting inputs from peripheral resource. The external interrupt flag must be cleared before the interrupt is enabled.

Pin assignment to peripheral resources is made by the registers of PFR and EPFR. However, since all registers cannot be changed at one time, I/O relocation for outputs must be executed in the port setting state (PFRxx:P[n]=0).

11.4.11.1 Extended Port Function Register 00, 01, 39: EPFR00, 01, 39

The bit configuration of extended port function register 00, 01, 39 is shown.

These registers are used to select input pins for input capture. (I/O relocation)

EPFR00: Address 0E60H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICU3E[1:0]		ICU2E[1:0]		ICU1E[1:0]		ICU0E[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

EPFR01: Address 0E61H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICU7E[1:0]		ICU6E[1:0]		ICU5E[1:0]		ICU4E[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

EPFR39: Address 0E87H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICU11E[1:0]		ICU10E[1:0]		ICU9E[1:0]		ICU8E[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

ICU0E[1:0]: Input capture ch.0 input pin selection

ICU1E[1:0]: Input capture ch.1 input pin selection

ICU2E[1:0]: Input capture ch.2 input pin selection

ICU3E[1:0]: Input capture ch.3 input pin selection

ICU4E[1:0]: Input capture ch.4 input pin selection

ICU5E[1:0]: Input capture ch.5 input pin selection

ICU6E[1:0]: Input capture ch.6 input pin selection

ICU7E[1:0]: Input capture ch.7 input pin selection

ICU8E[1:0]: Input capture ch.8 input pin selection

ICU9E[1:0]: Input capture ch.9 input pin selection

ICU10E[1:0]: Input capture ch.10 input pin selection

ICU11E[1:0]: Input capture ch.11 input pin selection

ICUnE[1:0] (n=0 to 11)	Operation
00	Input from ICUn_0 pin
01	Input from ICUn_1 pin
10	Input from ICUn_2 pin
11	Reserved (Input from ICUn_2 pin)

11.4.11.2 Extended Port Function Register 02 to 05: EPFR02-05

The bit configuration of extended port function register 02 to 05 is shown.

These registers are used to enable output and to select output/input pins for reload timer. (I/O relocation and I/O multiplexing)

EPFR02: Address 0E62H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			TOT0E[2:0]			TIN0E[1:0]	
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

EPFR03: Address 0E63H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			TOT1E[2:0]			TIN1E[1:0]	
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

EPFR04: Address 0E64H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			TOT2E[2:0]			TIN2E[1:0]	
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

EPFR05: Address 0E65H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			TOT3E[2:0]			TIN3E[1:0]	
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

TOT0E[2:0]: Reload timer ch.0 TOT output pin selection

TIN0E[1:0]: Reload timer ch.0 TIN input pin selection

TOT1E[2:0]: Reload timer ch.1 TOT output pin selection

TIN1E[1:0]: Reload timer ch.1 TIN input pin selection

TOT2E[2:0]: Reload timer ch.2 TOT output pin selection

TIN2E[1:0]: Reload timer ch.2 TIN input pin selection

TOT3E[2:0]: Reload timer ch.3 TOT output pin selection

TIN3E[1:0]: Reload timer ch.3 TIN input pin selection

TOTnE[2:0] (n=0 to 3)	Operation
000	No output
xx1	Output from TOTn_0 pin
x1x	Output from TOTn_1 pin
1xx	Output from TOTn_2 pin

TINnE[1:0] (n=0 to 3)	Operation
00	Input from TINn_0 pin
01	Input from TINn_1pin
10	Input from TINn_2pin
11	Reserved (Input from TINn_2 pin)

11.4.11.3 Extended Port Function Register 06 to 09, 33, 34: EPFR06-09, 33, 34

The bit configuration of extended port function register 06 to 09, 33 and 34 is shown.

These registers are used to enable output and to select output/input pins for LIN-UART. (I/O relocation and I/O multiplexing)

Note:

Please set SCK/SOT/SIN of LIN-UART to the same group (SCKn/SOTn/SINn or SCKn_1/SOTn_1/ SINn_1). It is a prohibition to do relocations as disjointedly as the following examples.

Prohibition example: SCKn_0/SOTn_1/SINn_0

EPFR06: Address 0E66H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT2E[1:0]		SCK2E[1:0]		SIN2E
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

EPFR07: Address 0E67H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT3E[1:0]		SCK3E[1:0]		SIN3E
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

EPFR08: Address 0E68H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT4E[1:0]		SCK4E[1:0]		SIN4E
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

EPFR09: Address 0E69H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT5E[1:0]		SCK5E[1:0]		SIN5E
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

EPFR33: Address 0E81H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT6E[1:0]		SCK6E[1:0]		SIN6E
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

EPFR34: Address 0E82H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT7E[1:0]		SCK7E[1:0]		SIN7E
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

SOT2E[1:0]: LIN-UART ch.2 SOT output pin selection

SCK2E[1:0]: LIN-UART ch.2 SCK output/input pin selection

SIN2E : LIN-UART ch.2 SIN input pin selection

SOT3E[1:0]: LIN-UART ch.3 SOT output pin selection

SCK3E[1:0]: LIN-UART ch.3 SCK output/input pin selection

SIN3E : LIN-UART ch.3 SIN input pin selection

SOT4E[1:0]: LIN-UART ch.4 SOT output pin selection

SCK4E[1:0]: LIN-UART ch.4 SCK output/input pin selection

SIN4E: LIN-UART ch.4 SIN input pin selection

SOT5E[1:0]: LIN-UART ch.5 SOT output pin selection

SCK5E[1:0]: LIN-UART ch.5 SCK output/input pin selection

SIN5E: LIN-UART ch.5 SIN input pin selection

SOT6E[1:0]: LIN-UART ch.6 SOT output pin selection

SCK6E[1:0]: LIN-UART ch.6 SCK output/input pin selection

SIN6E: LIN-UART ch.6 SIN input pin selection

SOT7E[1:0]: LIN-UART ch.7 SOT output pin selection

SCK7E[1:0]: LIN-UART ch.7 SCK output/input pin selection

SIN7E: LIN-UART ch.7 SIN input pin selection

SOTnE[1:0] (n=2 to 7)	Operation
00	No output
x1	Output from SOTn_0 pin
1x	Output from SOTn_1 pin

SCKnE[1:0] (n=2 to 7)	Operation
00	Input from SCKn_0 / No output
01	Input from SCKn_0 / Output from SCKn_0
10	Input from SCKn_1 / Output from SCKn_1
11	Reserved (Input from SCKn_1 / Output from SCKn_1)

SINnE (n=2 to 7)	Operation
0	Input from SINn_0 pin
1	Input from SINn_1 pin

11.4.11.4 Extended Port Function Register 10 to 15, 45, 46: EPFR10-15,45,46

The bit configuration of extended port function register 10 to 15,45 and 46 is shown.

These registers are used to enable output and to select output pins for PPG. (I/O relocation and I/O multiplexing)

EPFR10: Address 0E6A$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	PPG1E[3:0]				PPG0E[2:0]		
Initial value	1	0	0	0	0	0	0	0
Attribute	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EPFR11: Address 0E6B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		PPG3E[2:0]			PPG2E[2:0]		
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

EPFR12: Address 0E6C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		PPG5E[2:0]			PPG4E[2:0]		
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

EPFR13: Address 0E6D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		PPG7E[2:0]			PPG6E[2:0]		
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

EPFR14: Address 0E6E$_H$ (Access: Byte, Half-word, Word)

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		PPG9E[2:0]			PPG8E[2:0]		
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

EPFR15: Address 0E6F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	PPG12E[1:0]		PPG11E[1:0]		PPG10E[2:0]		
Initial value	1	0	0	0	0	0	0	0
Attribute	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EPFR45: Address 0E8D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PPG17E	PPG16E	PPG15E[1:0]		PPG14E[1:0]		PPG13E[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

EPFR46: Address 0E8E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		PPG23E	PPG22E	PPG21E	PPG20E	PPG19E	PPG18E
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

PPG0E[2:0]: PPG ch.0 output pin selection

PPG2E[2:0]: PPG ch.2 output pin selection

PPG3E[2:0]: PPG ch.3 output pin selection

PPG4E[2:0]: PPG ch.4 output pin selection

PPG5E[2:0]: PPG ch.5 output pin selection

PPG6E[2:0]: PPG ch.6 output pin selection

PPG7E[2:0]: PPG ch.7 output pin selection

PPG8E[2:0]: PPG ch.8 output pin selection

PPG9E[2:0]: PPG ch.9 output pin selection

PPG10E[2:0]: PPG ch.10 output pin selection

PPGnE[2:0] (n=0,2 to 10)	Operation
000	No output
xx1	Output from PPGn_0 pin
x1x	Output from PPGn_1 pin
1xx	Output from PPGn_2 pin

PPG1E[3:0]: PPG ch.1 output pin selection

PPGnE[3:0] (n=1)	Operation
0000	No output
xxx1	Output from PPGn_0 pin
xx1x	Output from PPGn_1 pin
x1xx	Output from PPGn_2 pin
1xxx	Output from PPGn_3 pin

PPG11E[1:0]: PPG ch.11 output pin selection

PPG12E[1:0]: PPG ch.12 output pin selection

PPG13E[1:0]: PPG ch.13 output pin selection

PPG14E[1:0]: PPG ch.14 output pin selection

PPG15E[1:0]: PPG ch.15 output pin selection

PPGnE[1:0] (n=11 to 15)	Operation
00	No output
01	Output from PPGn_0 pin
1x	Output from PPGn_1 pin

PPG16E: PPG ch.16 output pin selection

PPG17E: PPG ch.17 output pin selection

PPG18E: PPG ch.18 output pin selection

PPG19E: PPG ch.19 output pin selection

PPG20E: PPG ch.20 output pin selection

PPG21E: PPG ch.21 output pin selection

PPG22E: PPG ch.22 output pin selection

PPG23E: PPG ch.23 output pin selection

PPGnE (n=16 to 23)	Operation
0	No output
1	Output from PPGn_0 pin

11.4.11.5 Extended Port Function Register 16 to 20: EPFR16-20

The bit configuration of extended port function register 16 to 20 is shown.

These registers are used to enable the segment of LCD controller, common output and reference voltage input. (I/O multiplexing)

EPFR16: Address 0E70H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SEG7E	SEG6E	SEG5E	SEG4E	SEG3E	SEG2E	SEG1E	SEG0E
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

EPFR17: Address 0E71H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SEG15E	SEG14E	SEG13E	SEG12E	SEG11E	SEG10E	SEG9E	SEG8E
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

EPFR18: Address 0E72H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SEG23E	SEG22E	SEG21E	SEG20E	SEG19E	SEG18E	SEG17E	SEG16E
Initial value	1	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EPFR19: Address 0E73H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SEG31E	SEG30E	SEG29E	SEG28E	SEG27E	SEG26E	SEG25E	SEG24E
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

EPFR20: Address 0E74H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	V3E	V2E	V1E	V0E	COM3E	COM2E	COM1E	COM0E
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

SEGnE (n=0 to 31): LCD controller duty driving driver segment *n* output enable

COMnE (n=0 to 3): LCD controller driver common *n* output enable

VnE (n=0 to 2): LCD controller reference voltage *n* input enable

V3E: LCD controller reference voltage selection

SEGnE (n=0 to 31)	Operation
0	Disable LCDC segment output
1	Enable LCDC segment output

COMnE (n=0 to 2) are similar to the above.

VnE (n=0 to 2)	Operation
0	Disable LCDC reference voltage input
1	Enable LCDC reference voltage input

To select V3 , set both V3E and PFR05.bit7.

{V3E, PFR05.bit7}	Operation
{1,1}	Use V3 pin as the LCDC V3 reference voltage input.
The others	VCC is used for the LCDC V3 reference voltage. The pin whose pin number is 40 can be used as P057/RDY/FRCK3_1 input.

It is forbidden to switch the LCDC V3 reference voltage selection while the Main clock oscillation is stopped.

11.4.11.6 Extended Port Function Register 21 to 23: EPFR21-23

The bit configuration of extended port function register 21 to 23 is shown.

These registers are used to enable stepping motor controller output. (I/O multiplexing)

EPFR21: Address 0E75$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWM2M1E	PWM2P1E	PWM1M1E	PWM1P1E	PWM2M0E	PWM2P0E	PWM1M0E	PWM1P0E
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

EPFR22: Address 0E76$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWM2M3E	PWM2P3E	PWM1M3E	PWM1P3E	PWM2M2E	PWM2P2E	PWM1M2E	PWM1P2E
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

EPFR23: Address 0E77$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PWM2M5E	PWM2P5E	PWM1M5E	PWM1P5E	PWM2M4E	PWM2P4E	PWM1M4E	PWM1P4E
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

PWM2MnE (n=0 to 5): SMC channel *n* PWM control (M2) output enable

PWM2PnE (n=0 to 5): SMC channel *n* PWM control (P2) output enable

PWM1MnE (n=0 to 5): SMC channel *n* PWM control (M1) output enable

PWM1PnE (n=0 to 5 : SMC channel *n* PWM control (P1) output enable

PWM2MnE (n=0 to 5)	Operation
0	SMC channel n PWM M2 output disabled (Initial value)
1	SMC channel n PWM M2 output enabled

PWM2PnE, PWM1MnE and PWM1PnE (n=0 to 5) are also similar to PWM2MnE.

11.4.11.7 Extended Port Function Register 24: EPFR24

The bit configuration of extended port function register 24 is shown.

This register is used to enable CAN output. (I/O multiplexing)

EPFR24: Address 0E78$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					TX2E	TX1E	TX0E
Initial value	1	1	1	1	1	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W

TXnE (n=0 to 2): CAN channel n transmission data output enabled

TXnE (n=0 to 2)	Operation
0	CAN channel n output disabled (Initial value)
1	CAN channel n output enabled

11.4.11.8 Extended Port Function Register 25: EPFR25

The bit configuration of extended port function register 25 is shown.

This is a reserved register.

EPFR25: Address 0E79H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					EPFR25D[2:0]		
Initial value	1	1	1	1	1	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W0	R/W0	R/W0

EPFR25D[2:0]: Reserved bits

"0" must be written to the bits.

11.4.11.9 Extended Port Function Register 26: EPFR26

The bit configuration of extended port function register 26 is shown.

This register is used to enable base timer output. (I/O multiplexing)

EPFR26: Address 0E7A$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				TIB1E	TIB0E	TIA1E	TIA0E
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

TIBnE (n=0, 1): Reserved bits

Setting to these bits does not affect on the operation.

TIAnE (n=0, 1): Base timer TIOAn output enable

TIAnE (n=0, 1)	Operation
0	Base timer TIOAn output disabled (Initial value)
1	Base timer TIOAn output enabled

11.4.11.10 Extended Port Function Register 27, 30: EPFR27, 30

The bit configuration of extended port function register 27 and 30 is shown.

These registers are used to enable the real-time clock and sound generator output. (I/O multiplexing and I/O relocation)

EPFR27: Address 0E7B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			WOTE	SGO1E	SGA1E	SGO0E	SGA0E
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

EPFR30: Address 0E7E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SGO4E[1:0]		SGA4E[1:0]		SGO3E	SGA3E	SGO2E	SGA2E
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

SGAnE (n=0 to 3): Sound generator channel *n* SGA output enable

SGOnE (n=0 to 3): Sound generator channel *n* SGO output enable

WOTE: Real time clock overflow output enable

SGAnE (n=0 to 3)	Operation
0	Sound generator channel n SGA output disabled (Initial value)
1	Sound generator channel n SGA output enabled

SGOnE (n=0 to 3) and WOTE are also similar to the above.

SGA4E [1:0]: Sound generator channel 4 SGA output enable

SGO4E [1:0]: Sound generator channel 4 SGO output enable

SGA4E[1:0]	Operation
00	No output
01	Output from SGA4_0 pin
10	Output from SGA4_1 pin
11	Forbidden

SGO4E[1:0] is also similar to the above.

11.4.11.11 Extended Port Function Register 28: EPFR28

The bit configuration of extended port function register 28 is shown.

This register is used to enable free-run timer clock input. (I/O multiplexing)

EPFR28: Address 0E7C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				FRCK3E	FRCK2E	FRCK1E	FRCK0E
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

[bit0] FRCK0E: Free-run timer ch.0 clock input selection

[bit1] FRCK1E: Free-run timer ch.1 clock input selection

[bit2] FRCK2E: Free-run timer ch.2 clock input selection

[bit3] FRCK3E: Free-run timer ch.3 clock input selection

FRCKnE (n=0 to 3)	Operation
0	Input from FRCKn_0 pin (Initial value)
1	Input from FRCKn_1 pin

11.4.11.12 Extended Port Function Register 29, 31, 32: EPFR29, 31, 32

The bit configuration of extended port function register 29,31 and 32 is shown.

This register is used to enable output compare output. (I/O multiplexing and I/O relocation)

EPFR29: Address 0E7D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OCU3E[1:0]		OCU2E[1:0]		OCU1E[1:0]		OCU0E[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

EPFR31: Address 0E7F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OCU7E[1:0]		OCU6E[1:0]		OCU5E[1:0]		OCU4E[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

EPFR32: Address 0E80$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	OCU11E[1:0]		OCU10E[1:0]		OCU9E[1:0]		OCU8E[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

OCUnE[1:0] (n=0 to 11): Output compare channel *n* output enable

OCUnE[1:0] (n=0 to 11)	Operation
00	No output
01	Output from OCUn_0 pin
10	Output from OCUn_1 pin
11	Forbidden

11.4.11.13 Extended Port Function Register 35 to 38: EPFR35-38

The bit configuration of extended port function register 35 to 38 is shown.

These registers are used to enable multi-function serial interface output. (I/O multiplexing and I/O relocation)

EPFR35: Address 0E83H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT0E[1:0]		SCK0E[1:0]		SIN0E
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

EPFR36: Address 0E84H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT1E[1:0]		SCK1E[1:0]		SIN1E
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

EPFR37: Address 0E85H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SOT8E[2:0]			SCK8E[2:0]			SIN8E[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

EPFR38: Address 0E86H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			SOT9E[1:0]		SCK9E[1:0]		SIN9E
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

SOT0E[1:0]: multi-function serial interface ch.0 SOT output/input pin selection

SCK0E[1:0]: multi-function serial interface ch.0 SCK output/input pin selection

SIN0E: multi-function serial interface ch.0 SIN input pin selection

SOT1E[1:0]: multi-function serial interface ch.1 SOT output/input pin selection

SCK1E[1:0]: multi-function serial interface ch.1 SCK output/input pin selection

SIN1E: multi-function serial interface ch.1 SIN input pin selection

SOT8E[2:0]: multi-function serial interface ch.8 SOT output/input pin selection

SCK8E[2:0]: multi-function serial interface ch.8 SCK output/input pin selection

SIN8E[1:0] : multi-function serial interface ch.8 SIN input pin selection

SOT9E[1:0] : multi-function serial interface ch.9 SOT output/input pin selection

SCK9E[1:0] : multi-function serial interface ch.9 SCK output/input pin selection

SIN9E: multi-function serial interface ch.9 SIN input pin selection

Relocations of I^2C pins of ch.0 and ch.1 are not supported. Select relocation "_0" when ch.0 or ch.1 are used for I^2C function. The ch.8 and ch.9 do not support I^2C input/output.

SCK0,1,9 pin selection (n=Channel number)

SCK0E1 SCK1E1 SCK9E1	SCK0E0 SCK1E0 SCK9E0	Operation
0	0	Input from SCKn_0 / No output
0	1	Input from SCKn_0 / Output from SCKn_0
1	0	Input from SCKn_1 / Output from SCKn_1
1	1	Forbidden

Relocations of I^2C pins of ch.0 and ch.1 are not supported. Set "01" for ch.0 and ch.1 when they are configured for I^2C.

SOT0,1,9 pin selection (n=Channel number)

SOT0E1 SOT1E1 SOT9E1	SOT0E0 SOT1E0 SOT9E0	Operation
0	0	Input from SOTn_0 / No output
0	1	Input from SOTn_0 / Output from SOTn_0
1	0	Input from SOTn_1 / Output from SOTn_1
1	1	Forbidden

Relocations of I^2C pins of ch.0 and ch.1 are not supported. Set "01" for ch.0 and ch.1 when they are configured for I^2C.

SIN0,1,9 pin selection (n=Channel number)

SIN0E SIN1E SIN9E	Operation
0	Input from SINn_0
1	Input from SINn_1

Relocations of I^2C pins of ch.0 and ch.1 are not supported. Set “0” for ch.0 and ch.1 when they are configured for I^2C.

SCK8 pin selection

SCK8E2	SCK8E1	SCK8E0	Operation
0	0	0	Input from SCK8_0 / No output
0	0	1	Input from SCK8_0 / Output from SCK8_0
0	1	0	Input from SCK8_1 / Output from SCK8_1
1	0	0	Input from SCK8_2 / Output from SCK8_2
The others			Forbidden

SOT8 pin selection

SOT8E2	SOT8E1	SOT8E0	Operation
0	0	0	Input from SOT8_0 / No output
0	0	1	Input from SOT8_0 / Output from SOT8_0
0	1	0	Input from SOT8_1 / Output from SOT8_1
1	0	0	Input from SOT8_2 / Output from SOT8_2
The others			Forbidden

SIN8 pin selection

SIN8E1	SIN8E0	Operation
0	0	Input from SIN8_0
0	1	Input from SIN8_1
1	0	Input from SIN8_2
1	1	Forbidden

11.4.11.14 Extended Port Function Register 40, 41: EPFR40, 41

The bit configuration of extended port function register 40 and 41 is shown.

These registers are used to select input pins for Up/Down counter. (I/O relocation)

EPFR40: Address 0E88H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		ZIN0E[1:0]		BIN0E[1:0]		AIN0E[1:0]	
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

EPFR41: Address 0E89H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					ZIN1E	BIN1E	AIN1E
Initial value	1	1	1	1	1	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W

AIN0E[1:0]	Operation
00	Input from AIN0_0 pin (Initial value)
01	Input from AIN0_1 pin
10	Input from AIN0_2 pin
11	Forbidden

BIN0E, ZIN0E are also similar to the above.

AIN1E	Operation
0	Input from AIN1_0 pin (Initial value)
1	Input from AIN1_1 pin

BIN1E, ZIN1E are also similar to the above.

11.4.11.15 Extended Port Function Register 42: EPFR42

The bit configuration of extended port function register 42 is shown.

This register is used to enable output signal of D/A converter. (I/O multiplexing)

EPFR42: Address 0E8A$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						DAS1	DAS0
Initial value	1	1	1	1	1	1	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W

[bit0] DAS0: D/A converter 0 output data selection enable

[bit1] DAS1: D/A converter 1 output data selection enable

DASn (n=0,1)	Operation
0	DAOn output disabled (Initial value)
1	DAOn output enabled

11.4.11.16 Extended Port Function Register 43, 44: EPFR43, 44

The bit configuration of extended port function register 43 and 44 is shown.

These registers are used to select input pins for external interrupt. (I/O relocation)

EPFR43: Address 0E8B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	INT7E	INT6E	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E
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

EPFR44: Address 0E8C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	INT15E	INT14E	INT13E	INT12E	INT11E	INT10E	INT9E	INT8E
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

INTnE (n=0 to 15)	Operation
0	Input from INTn_0 pin (Initial value)
1	Input from INTn_1 pin

11.4.11.17 Extended Port Function Register 47: EPFR47

The bit configuration of extended port function register 47 is shown.

The upper byte and lower byte of the data pins of the external bus interface can be swapped using this register.

EPFR47: Address 0E8F$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							EBDSWPE
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] EBDSWPE: Enable swap

EBDSWPE	Operation
0	Normal connection Pin num. 131 D16 Pin num. 132 D17 ~ Pin num. 137 D22 Pin num. 138 D23 Pin num. 139 D24 Pin num. 140 D25 ~ Pin num. 3 D30 Pin num. 4 D31
1	Swapped connection Pin num. 131 D24 Pin num. 132 D25 ~ Pin num. 137 D30 Pin num. 138 D31 Pin num. 139 D16 Pin num. 140 D17 ~ Pin num. 3 D22 Pin num. 4 D23

11.4.11.18 Extended Port Function Register 48 to 51: EPFR48-51

The bit configuration of extended port function register 48 to 51 is shown.

They are reserved registers. Do not use them.

EPFR48: Address 0E90$_H$ (Access: Byte, Half-word, Word)

EPFR49: Address 0E91$_H$ (Access: Byte, Half-word, Word)

EPFR50: Address 0E92$_H$ (Access: Byte, Half-word, Word)

EPFR51: Address 0E93$_H$ (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	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX

11.4.11.19 Extended Port Function Register 52: EPFR52

The bit configuration of extended port function register 52 is shown.

This register configures SDATA[1:0] and SPI_SIO0,1 pins connection from/to HS_SPI.

EPFR52: Address 0E94H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							HSSWAP
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] HSSWAP: HS_SPI D0,D1 swap enable

HSSWAP	Operation
0	Normal connection SPI_SIO0 input → SDATA[0] input SPI_SIO1 input → SDATA[1] input SDATA[0] output → SPI_SIO0 output SDATA[1] output → SPI_SIO1 output
1	Swapped connection SPI_SIO0 input → SDATA[1] input SPI_SIO1 input → SDATA[0] input SDATA[0] output → SPI_SIO1 output SDATA[1] output → SPI_SIO0 output

11.4.11.20 Extended Port Function Register 53, 54: EPFR53, 54

The bit configuration of extended port function register 53 and 54 is shown.

These registers are used to enable HS_SPI output. (I/O multiplexing)

EPFR53: Address 0E95H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved			HSCKE	HSDE[3:0]			
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

EPFR54: Address 0E96H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				HSSELE[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

[bit4] HSCKE: HS_SPI SCK output enable

HSCKE	Operation
0	Output disabled
1	Output enabled

[bit3 to bit0] HSDE[3:0]: HS_SPI SDATAn (n=0 to 3) output enable

HSDE[n] (n=0 to 3)	Operation
0	Output disabled
1	Output enabled

[bit3 to bit0] HSSELE[3:0]: HS_SPI SSELn (n=0 to 3) output enable

HSSELE[n] (n=0 to 3)	Operation
0	Output disabled
1	Output enabled

Set both HSCKE and HSSELE to 0(output disabled) and configure corresponding pins as peripheral input (See 11.5.1.2 Peripheral Input Assignment) when you use the HS_SPI in slave mode.

11.4.12 Port Input Enable Register: PORTEN (Port Enable Register)

The bit configuration of the port input enable register is shown below.

This register contains control-bit to enable port input. 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. For information on pins whose inputs are blocked, see "D. Pin status in CPU Status" in "Appendix". After each port pin is configured for its function by software, Global PORT Enable (PORTEN.GPORTEN) bit must be set to “1” to enable input.

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	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W

[bit0] GPORTEN (Global Port Enable): Global input enable

GPORTEN	Operation
0	Most of pins are set input-blocked to cut off pass-through current at unstable condition. See "Pin status in CPU Status" in "Appendix" for the pin that is input-blocked at initial state by reset.
1	The input is enabled by this bit.

11.5 Operation

This section explains operations of I/O ports.

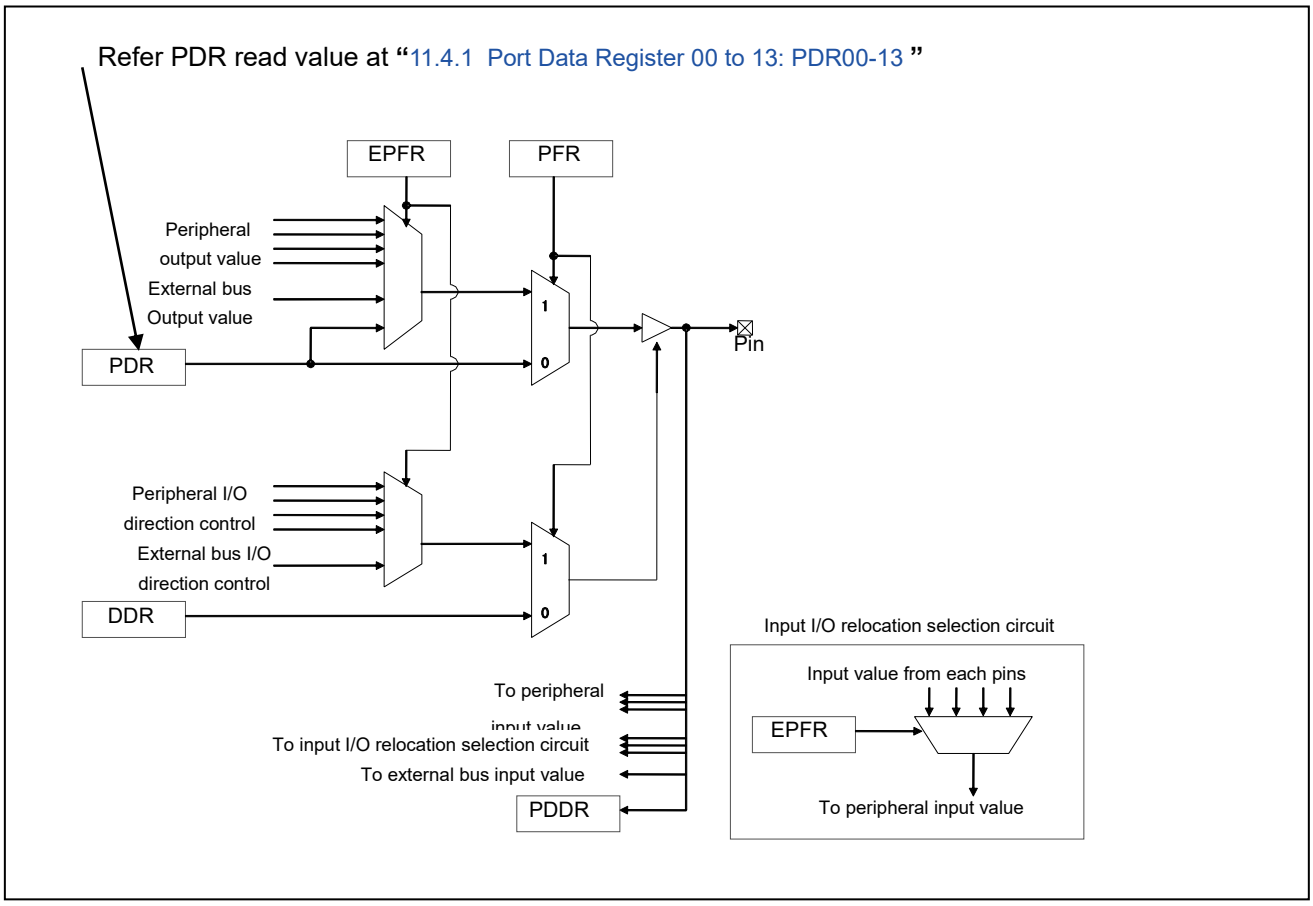
11.5.1 Pin I/O Assignment
11.5.2 EPFR Setting Priority
11.5.3 Notes on Input I/O Relocation Setting
11.5.4 Input Blocked by GPORTEN
11.5.5 Notes on Pins with the A/D Converter Function
11.5.6 Setting when Using the Base Timer TIOA1 Pin
11.5.7 Operation at Wake Up from Power Shutdown
11.5.8 Notes on Switching the I/O Port Function
11.5.9 Port Enable Function at Using Specification Resources

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 then functions as a port, also set the DDR and PDR values in advance if necessary. When doing this, 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 information on the setting method, see "Chapter: A/D Converter".

11.5.1.1 Peripheral I/O (Bidirectional) Pin Assignment

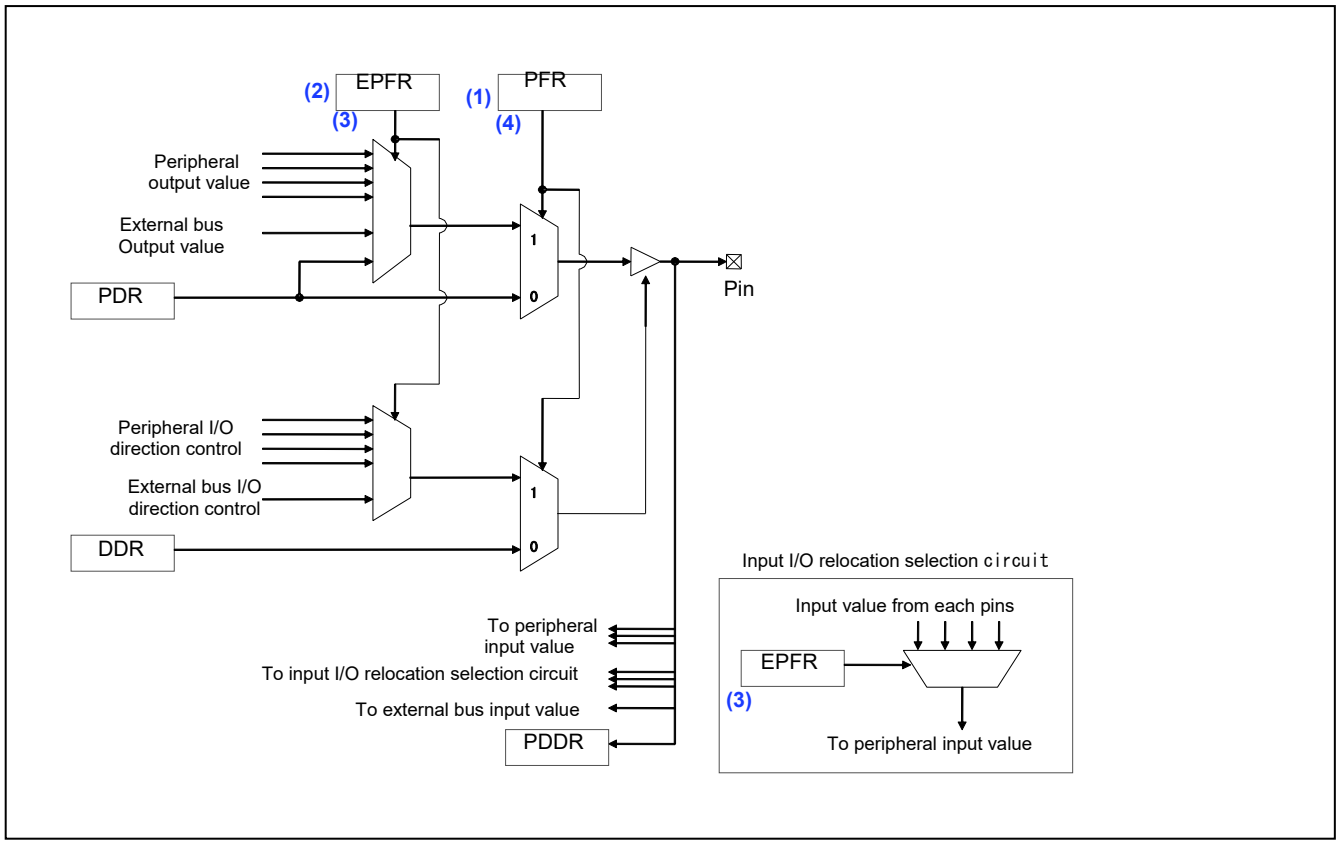
The peripheral I/O (bidirectional) pin assignment is shown below.

Preparation

- Since the pin once functions as a port as the result of step (1), also 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 information on the setting method, see "Chapter: A/D Converter".

1. Set the PFR for the applicable pin to enable the port function.
2. Disable the EPFRs for all other peripherals to be used by the relevant pin.
3. If the relevant pin is also used for the external bus or the relevant peripheral is one of the targets of I/O multiplexing, set the EPFR of the relevant peripheral. In addition, if the relevant peripheral has the I/O relocation function, 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

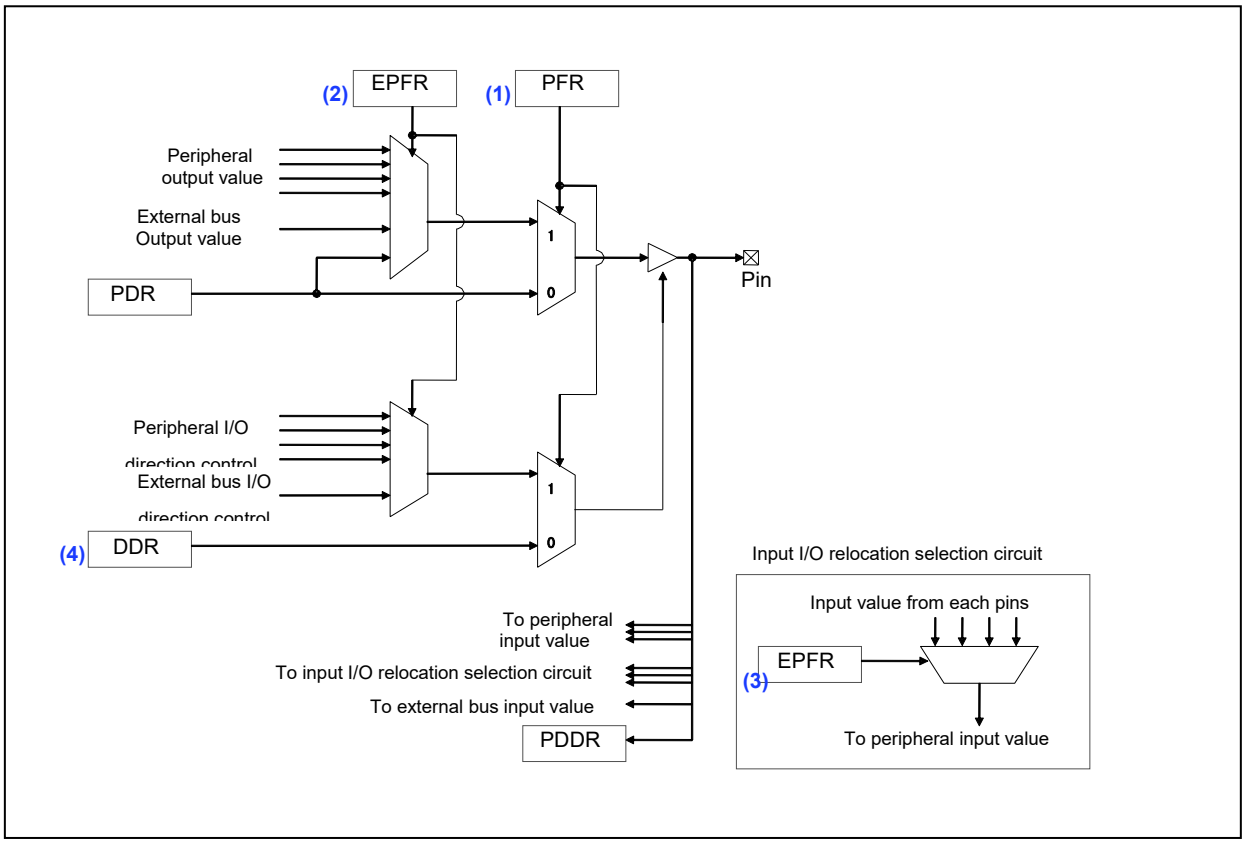
The peripheral input assignment is shown below.

Preparation

- Since the pin will once function 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 information on the setting method, see "Chapter: A/D Converter".

1. Set the PFR of the applicable pin to enable the port function.
2. Disable the EPFRs for all other peripherals to be used by the relevant pin.
3. An addition, if the relevant peripheral has the I/O relocation function, set the EPFR of the relevant peripheral.
4. 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.
Example: Since INT10_0 and PPG2_2 are assigned to the same pin (pin number 104, P111), external interrupt 10 can be generated at the PPG2 output by setting peripheral output of PPG2_2 and relocation of external interrupt 10 to "0".

11.5.1.3 Peripheral Output Assignment

The peripheral output assignment is shown below.

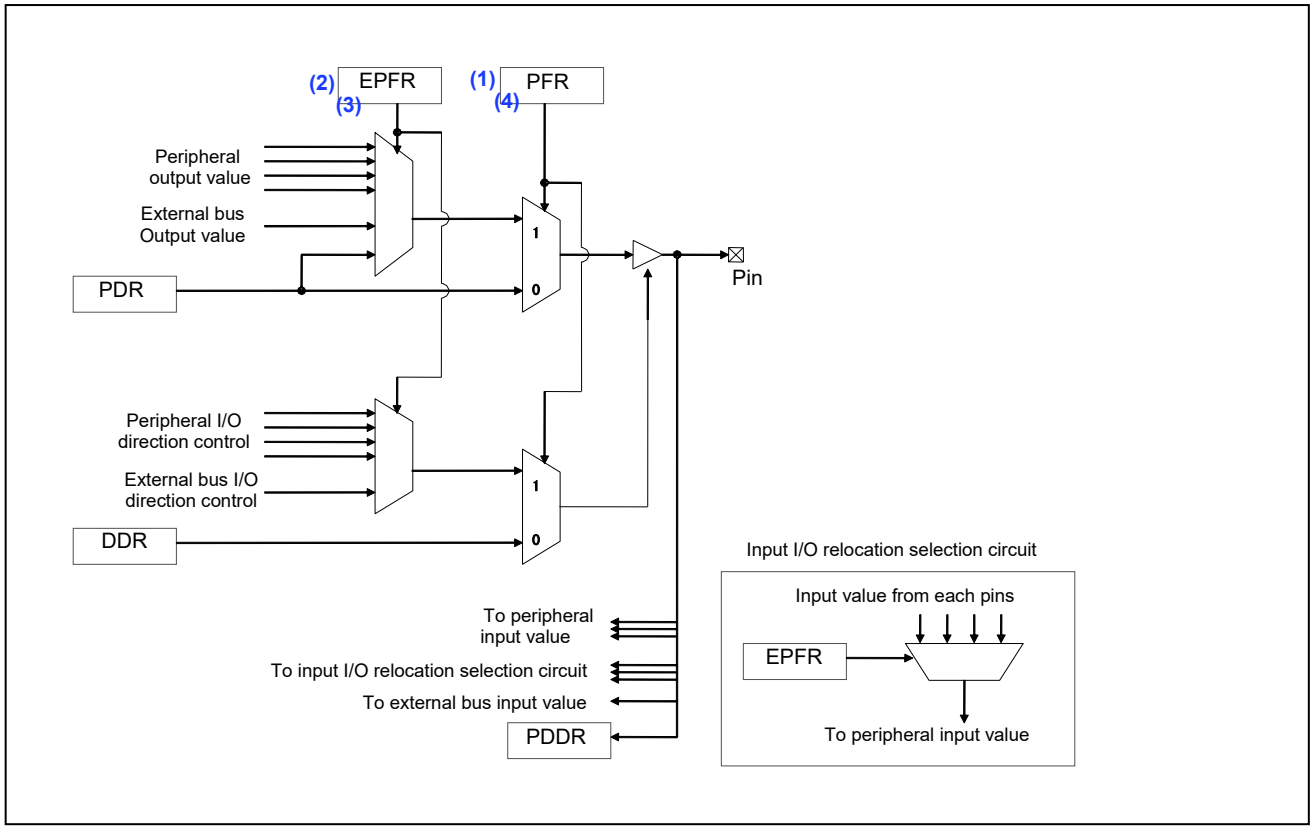
The setting method is the same as that described in "11.5.1.1 Peripheral I/O (Bidirectional) Pin Assignment".

Preparation

- Since the pin will once function 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 information on the setting method, see "Chapter: A/D Converter".

1. Set the PFR of the applicable pin to enable the port function.
2. Disable the EPFRs for all other peripherals to be used by the relevant pin.
3. If the relevant pin is also used for the external bus or the relevant peripheral is one of the targets of I/O multiplexing, set the EPFR of the relevant peripheral. In addition, if the relevant peripheral has the I/O relocation function, 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 External Bus Assignment

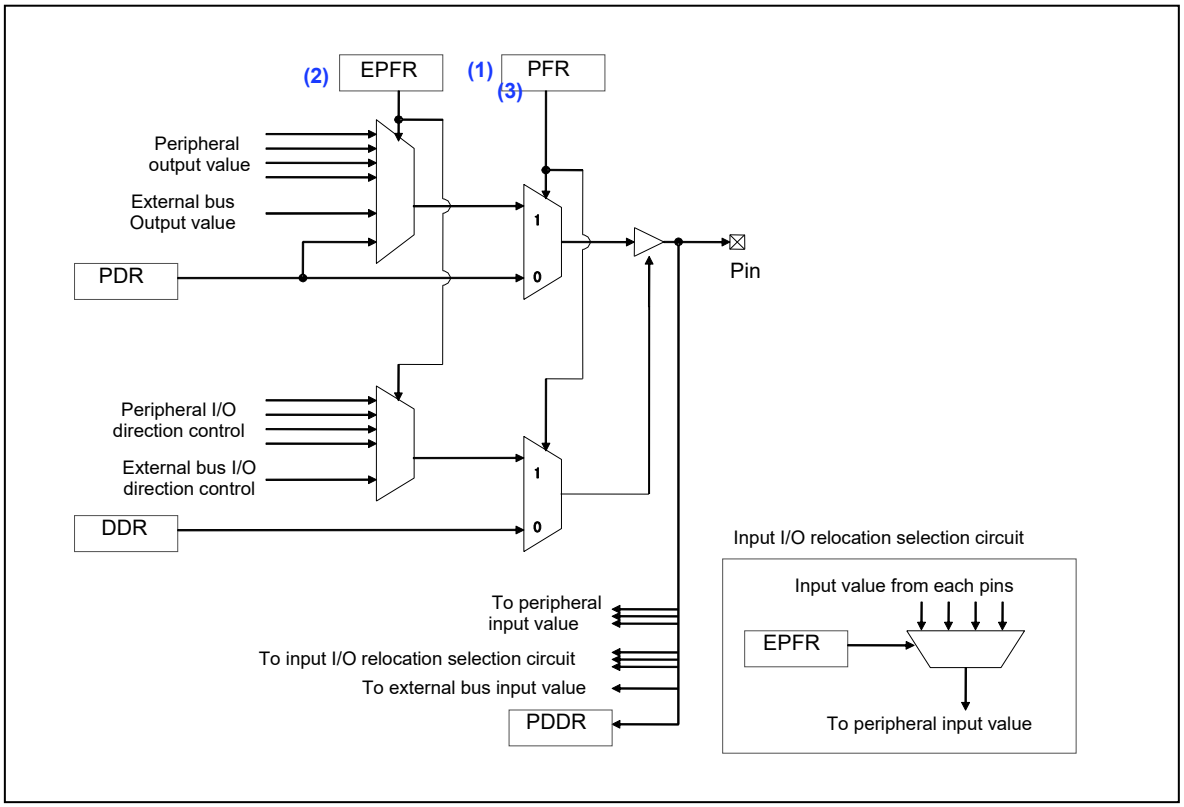
The external bus assignment is shown below.

Preparation

- Since the pin will once function 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 information on the setting method, see "Chapter: A/D Converter".

1. Set the PFR for the applicable pin to enable the port function.
2. Disable the EPFRs for all other peripherals that use the same pin as the external bus.
3. Set the PFR for the peripheral.

Figure 11-6. External Bus Assignment Procedure



11.5.1.5 Port Function (Input) Assignment

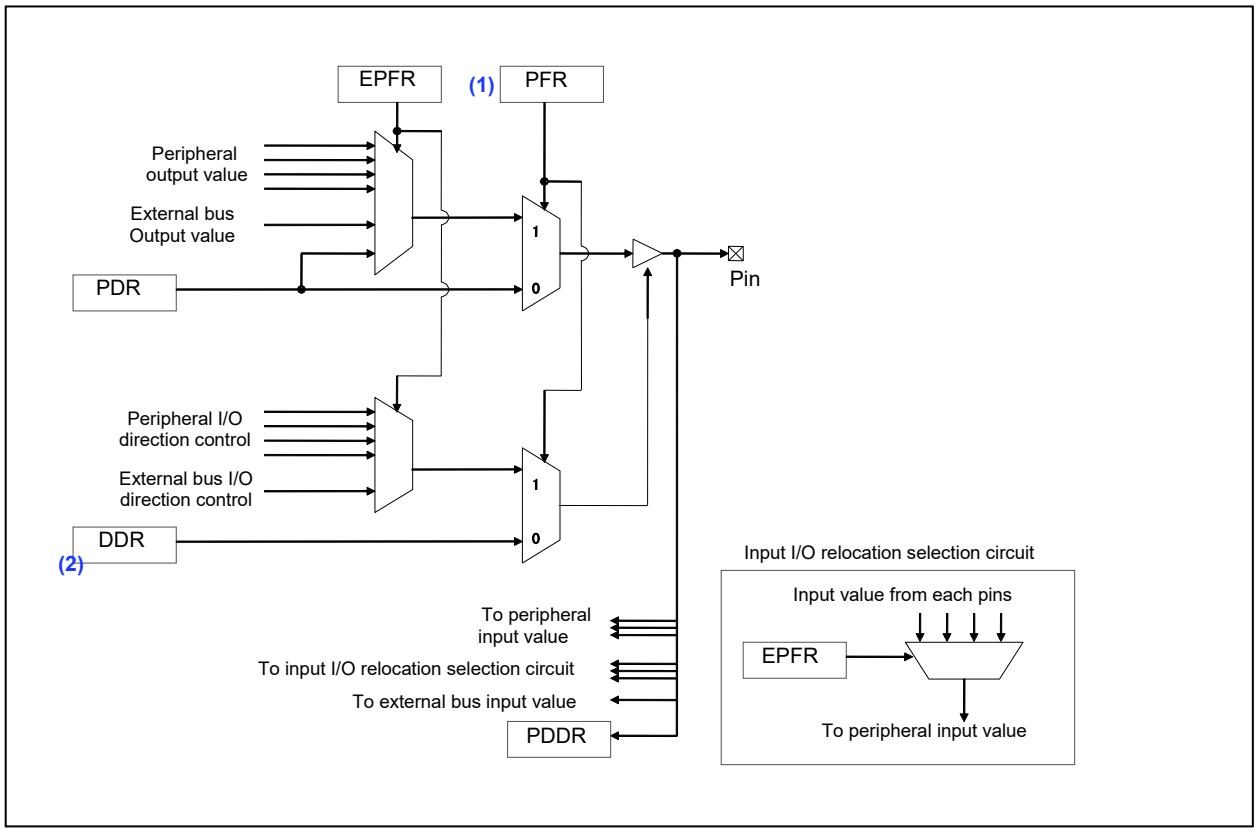
The port function (input) assignment is shown below.

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 information on the setting method, see "Chapter: A/D Converter".

1. Set the PFR to enable the port function.
2. Set the DDR for input.

Figure 11-7. Port Function (Input) Assignment Procedure



11.5.1.6 Port Function (Output) Assignment

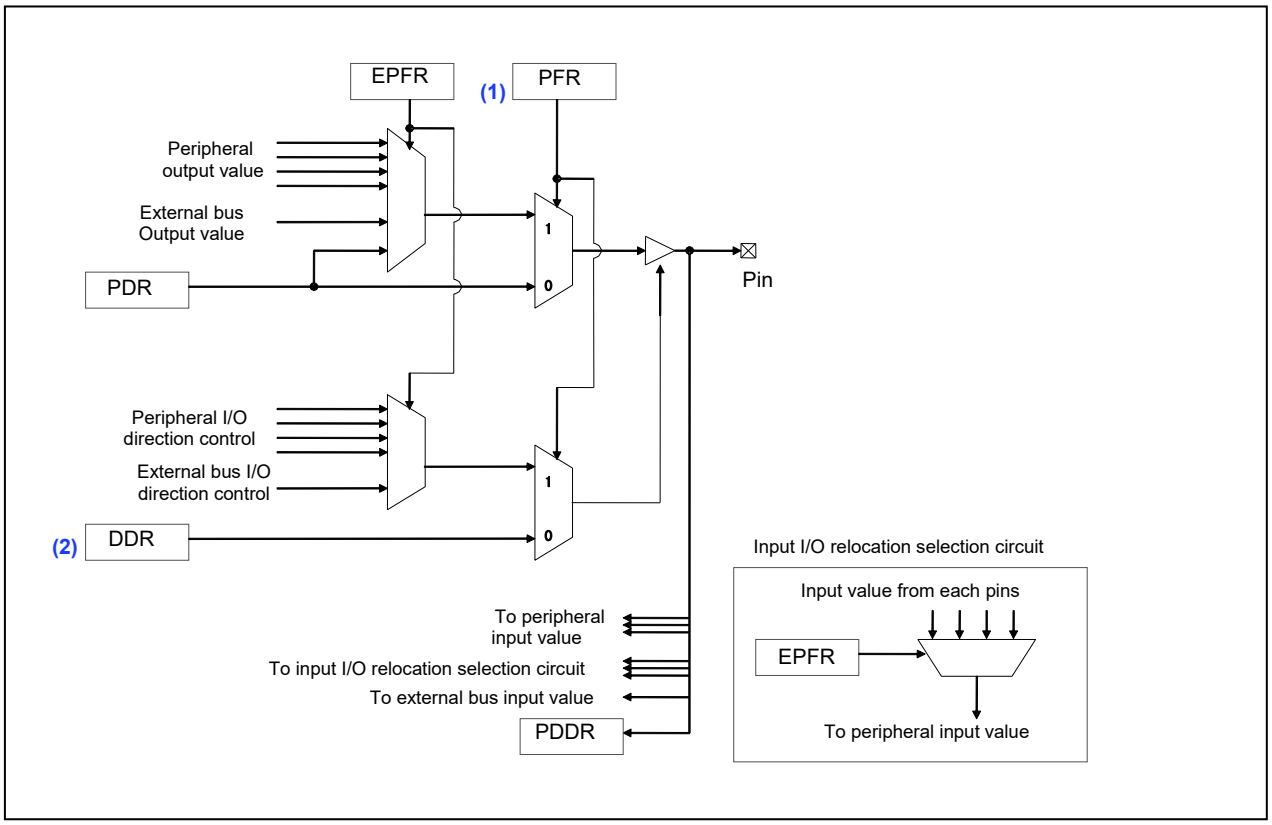
The port function (Output) assignment is shown below.

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 information on the setting method, see "Chapter: A/D Converter".

1. Set the PFR to enable the port function.
2. Set the DDR for output.

Figure 11-8. Port Function (Output) Assignment Procedure



11.5.1.7 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: A/D Converter".

Since the A/D converter assignment is given the highest priority, no other configuration is required.

11.5.1.8 D/A Converter Output Assignment

The D/A converter output assignment is shown below.

Same to “11.5.1.3 Peripheral Output Assignment”.

11.5.1.9 LCD Controller V0, V1, V2, V3 (LCDC Reference Voltage Input) Assignment

The LCD controller V0, V1, V2, V3 (LCDC reference voltage input) assignment is shown below.

Preparation

- Since the pin will once function as a port as the result of step (1), set the DDR and PDR values in advance if necessary.

1. Set corresponding PFR (PFR05.bit4/5/6/7) to port function.
2. Disable EPFR for the peripheral which shares same pin.
3. Set corresponding EPFR (EPFR20.bit4/5/6/7) to LCDC reference voltage input.
4. Set corresponding PFR (PFR05.bit4/5/6/7) to peripheral function.

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. LCD output, D/A converter output
2. CAN
3. Multi-function serial interface
4. LIN-UART
5. HS_SPI
6. PPG
7. Sound Generator
8. Real time clock
9. Base Timer
10. Reload timer
11. Output compare

11.5.3 Notes on Input I/O Relocation Setting

Notes on input I/O relocation setting are shown below.

When switching an input pin to another pin, if there is a difference between pin levels before and after the switch, the I/O relocation change may become a trigger input to the peripheral that uses the relevant pin as a trigger.

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 "D. Pin Status in CPU Status" in " Appendix " for the pin that becomes input blocked. See "11.4.12 Port Input Enable Register: PORTEN (Port Enable Register)" for the method of releasing the input blocked state.

During input signal is disconnected by GPORTEN setting, value of "0" is read as state of the disconnected pins.

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 "Port I/O Mode" in advance. For information on the setting method, see "Chapter: A/D Converter". 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 PFR13) and extended port function register (EPFR00 to EPFR54) settings.

11.5.6 Setting when Using the Base Timer TIOA1 Pin

Setting when using the base timer TIOA1 pin is shown below.

If the base timer TIOA1 pin is to be used, 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. If the base timer TIOA1 pin is to be used, it must be set for peripheral input for base timer I/O mode 1 (see "11.5.1.2 Peripheral Input Assignment 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").

11.5.7 Operation at Wake Up from Power Shutdown

The operation at wake up from the power shutdown is shown below.

When PMUCTLR:IOCTMD bit is set, the I/O state is kept during the wake up sequence from the power shutdown. The maintenance of the I/O state continues until PMUCTLR:IOCT is set.

When PMUCTLR:IOCTMD bit is cleared , the maintenance is effective during the wake up sequence. The register setting of the I/O port shall be effective after the wake-up completion.

On waking up from power shutdown, there is a case that the maintenance of the I/O latch is not released.

After waking up from power shutdown, PMUCTLR.IOCT bit must be written "1" for releasing the maintenance of I/O

11.5.8 Notes on Switching the I/O Port Function

Notes on switching the I/O port function is shown below.

When the port function is changed (general purpose port to peripheral function or peripheral function to general purpose port), it has possibilities that port outputs short spike.

Short spike is the same logic level as PDR value.

It is happened in the case of switching with direction change.

If this output is critical for the system, please set the certain value on PDR in prior to change port function.

11.5.9 Port Enable Function at Using Specification Resources

Port enable function at using specification resource is described below.

Value of "0" is read as state of pin if the pin is used as ADC or DAC or LCD or SMC function.

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 the 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

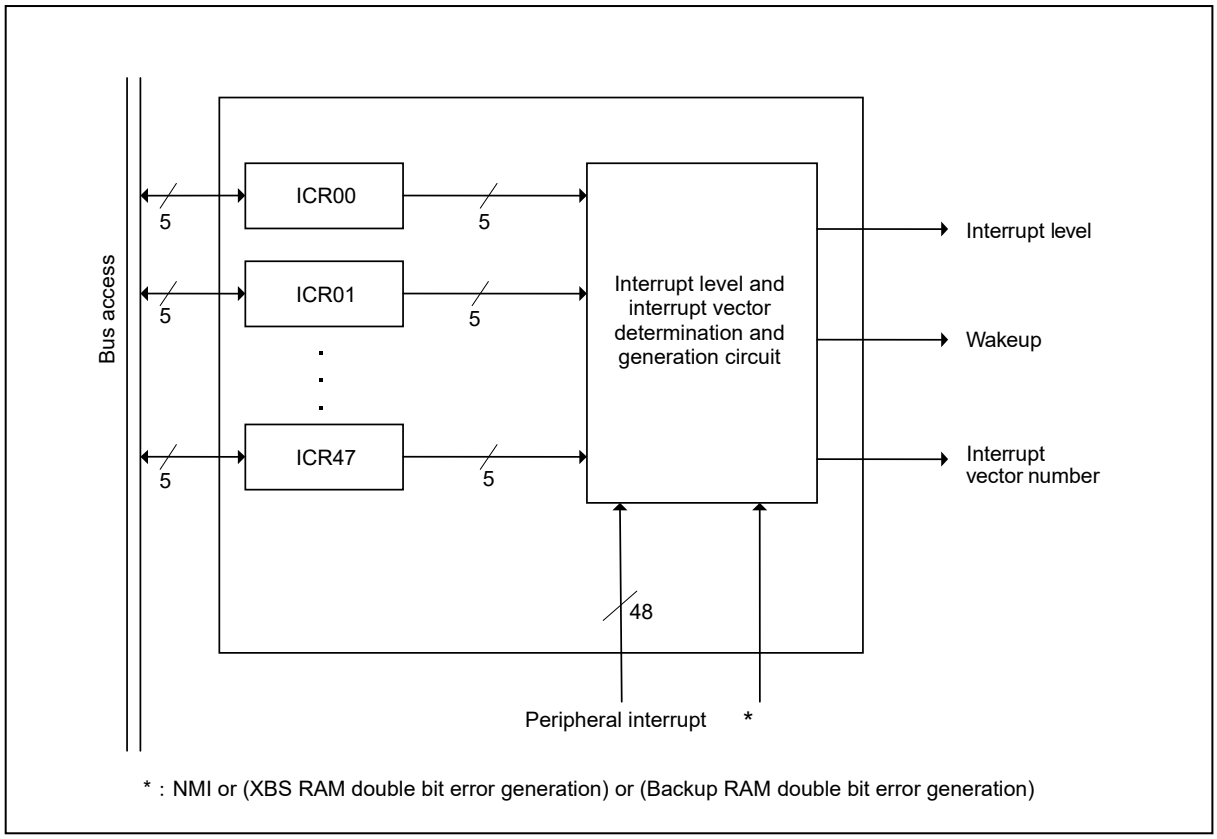
This module has the following functions.

- Detecting NMI requests and peripheral interrupt requests
- Priority determination (by level and interrupt vector)
- Transmitting the interrupt level of the source with the highest priority to the CPU
- Transmitting the interrupt vector number of the source 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



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

This section explains the bit configuration of the interrupt control registers 00 to 47 (ICR00 to ICR47).

One register is provided for each interrupt input to set the level for the corresponding interrupt request.

ICR00 to ICR47: Address 0440_H to $046F_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	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)

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).

Setup

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.)

Starting

Start the configured peripheral.

Determining Priorities

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.

Recovering From Stop Mode

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).

Recovering From Standby Mode (Power-shutdown)

When the interrupt level is higher than ICR=0x1F (interrupt disable) and the standby return factor is more effective in the state that the interrupt factor has been generated, the thing that changes to the state of the power supply interception cannot be done. The instruction execution is continued as it is.

It returns immediately through the power supply interception return sequence though it changes to the state of the power supply interception because the interrupt level does not become a standby return factor in the state that ICR=0x1F (interrupt disable) and the interrupt factor have been generated once because it is a state with the factor of the power supply interception return. (It is executed from the reset vector.)

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 INT15).

13.2 Features

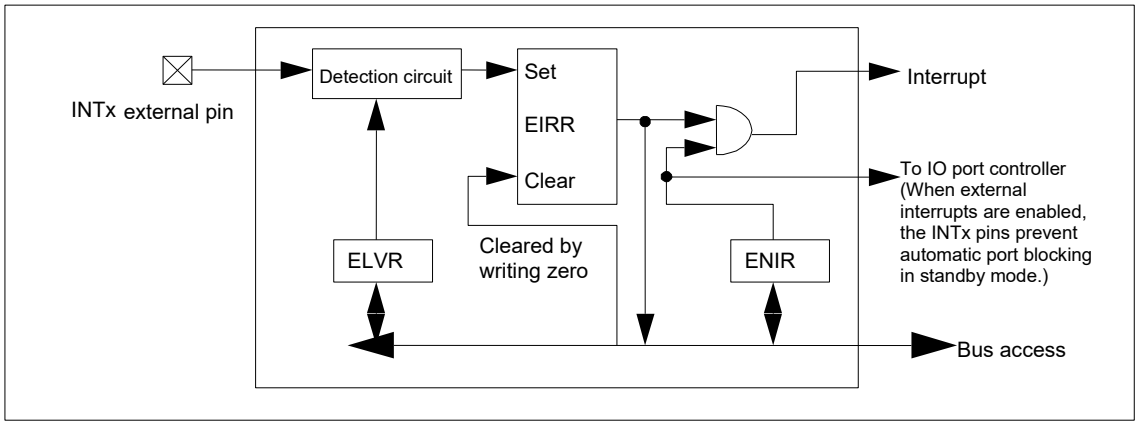
This section explains features of the external interrupt input.

- 16 systems external interrupt input pins (INT0 to INT15)
- Interrupt detection factors:4 types: ("L" level, "H" level, rising edge, 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_0 / INT0_1
1	0x0550	INT1_0 / INT1_1
2	0x0550	INT2_0 / INT2_1
3	0x0550	INT3_0 / INT3_1
4	0x0550	INT4_0 / INT4_1
5	0x0550	INT5_0 / INT5_1
6	0x0550	INT6_0 / INT6_1
7	0x0550	INT7_0 / INT7_1
8	0x0554	INT8_0 / INT8_1
9	0x0554	INT9_0 / INT9_1
10	0x0554	INT10_0 / INT10_1
11	0x0554	INT11_0 / INT11_1
12	0x0554	INT12_0 / INT12_1
13	0x0554	INT13_0 / INT13_1
14	0x0554	INT14_0 / INT14_1
15	0x0554	INT15_0 / INT15_1

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	EIRR1	ENIR1	ELVR1		External interrupt factor register 1 External interrupt enable register 1 External interrupt request level register 1

13.4.1 External Interrupt Factor Register 0/1: EIRR0/EIRR1 (External Interrupt Request Register 0/1)

The bit configuration of external interrupt factor register 0/1 (EIRR0/EIRR1) is shown below.

This register holds information that an external interrupt factor has been generated.

EIRR0: Address 0550_H (Access: Byte, Half-word, Word)

EIRR1: Address 0554_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
Initial value	X	X	X	X	X	X	X	X
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 to bit0] ER7 to ER0 (External interrupt Request7 to 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	Does not influence operation

- EIRR0:ER0 corresponds to INT0 pin, EIRR0:ER1 to INT1 pin, ..., EIRR0:ER7 to INT7 pin, EIRR1:ER0 to INT8 pin, ..., EIRR1:ER7 to INT15 pin.
- Writing "1" to these bits doesn't influence operation.
- 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 except recovery from standby (power shutdown) when PMUCTLR:IOCTMD=1.

13.4.2 External Interrupt Enable Register 0/1: ENIR0/ENIR1 (Enable Interrupt Request Register 0/1)

The bit configuration of external interrupt enable register 0/1 (ENIR0/ENIR1) is shown below.

This register enables external interrupt inputs.

ENIR0: Address 0551$_H$ (Access: Byte, Half-word, Word)

ENIR1: Address 0555$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
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] EN7 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. Holds interrupt requests but does not output them. (initial value)
1	Interrupt request enabled. Enables interrupt requests.

- ENIR0:EN0 corresponds to INT0 pin, ENIR0:EN1 to INT1 pin, ..., ENIR0:EN7 to INT7 pin, ENIR1:EN0 to INT8 pin, ..., ENIR1:EN7 to INT15 pin.
- This register will be initialized by all reset factors except recovery from standby (power shutdown) when PMUCTLR:IOCTMD=1.

13.4.3 External Interrupt Request Level Register 0/1: ELVR0/ELVR1

The bit configuration of external interrupt request level register 0/1 (ELVR0/ELVR1) is shown below.

This register selects detection conditions for external interrupt requests.

ELVR0: Address 0552H (Access: Byte, Half-word, Word)

ELVR1: Address 0556H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4
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
	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 bit1] LB7 to LB0 (Level select B): Level select B

[bit14 to bit0] LA7 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

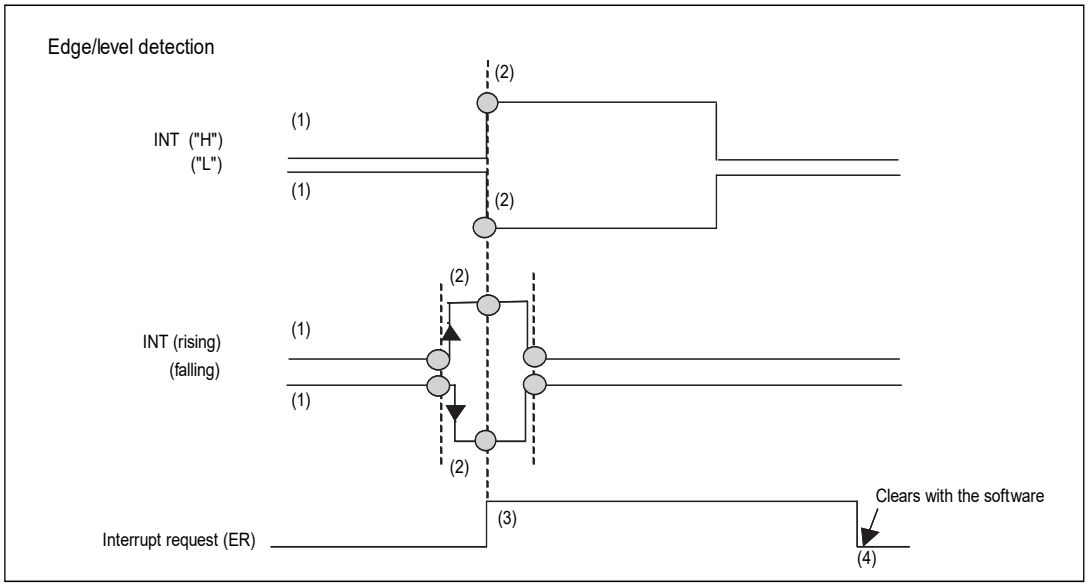
When the request input is a level (LAn, LBn =00 or 01), the corresponding bit (ERn) will turn back to "1" if INTn pin input is still in the effective levels after setting the external interrupt request bit (ERn) to "0".

- ELVR0:LA/LB0 corresponds to INT0 pin, ELVR0:LA/LB1 to INT1 pin, ..., ELVR0:LA/LB7 to INT7 pin, ELVR1:LA/LB0 to INT8 pin, ..., ELVR1:LA/LB7 to INT15 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 except recovery from standby (power shutdown) when PMUCTLR:IOCTMD=1.

13.5 Operation

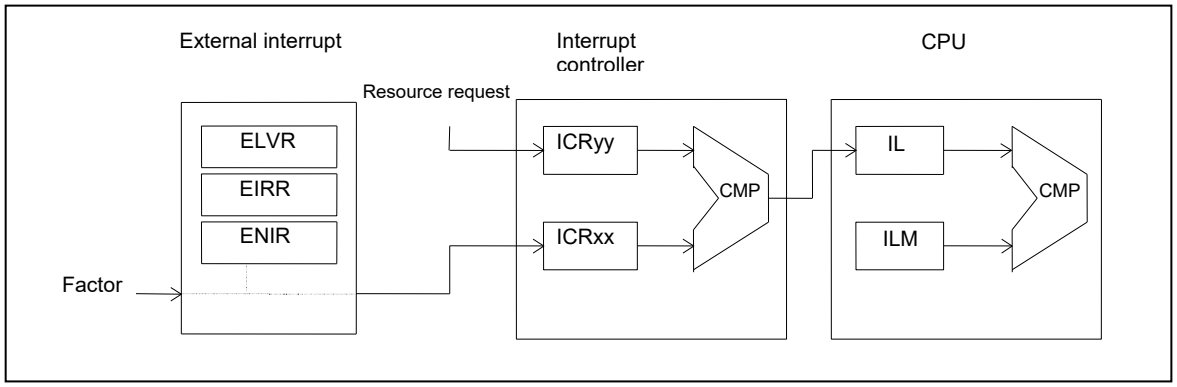
This section explains the operation of the external interrupt input.

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



- 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 resource was found to have the highest priority in the result for examining the priority in interrupts concurrently occurred in the interrupt controller.

- Transition to standby mode

  Channels not to be used should be moved to disable state before letting them go into the standby mode. For the enabled channel, the standby mode automatic input/output blocked feature to the external pin will also be suppressed. See "Chapter: Power Consumption Control" for the automatic input/output blocked feature.

- Setting procedure of external interrupts

  When setting registers which reside in the external interrupt unit, follow the steps 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.

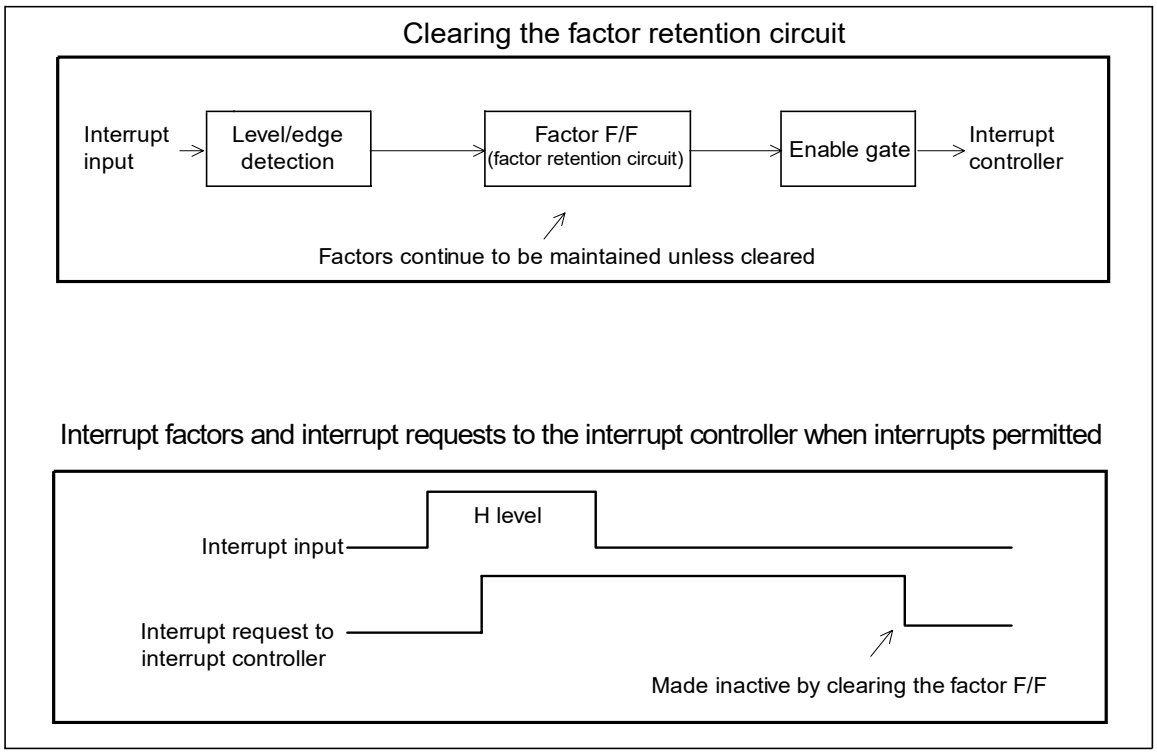
  This has to be done to avoid generating erroneous interrupt factors at the time of setting register or in interrupt enable state.

- 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



13.6 Setting

This section explains setting of the external interrupt input.

Figure 13-5. Necessary Settings for Using External Interrupts

Settings	Setting Register	Setting Method
Detection level settings	External interrupt request level setting register (ELVR0, ELVR1)	See "About Detection Levels and Their Setting Procedures" in "13.7 Q&A".
Make external pins to use 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 INT15	-

13.7 Q&A

This section explains Q&A of the external interrupt input.

About Detection Levels and Their Setting Procedures

Four levels: ("L" level, "H" level, rising edge, falling edge)

Set the detection level bits as follows: (ELVRy:LBn, LAn) (n=0 to 7, y=0, 1).

Operation Modes	Detection Level Bits (LBn, LAn) n=0 to 7
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/ENIR1:EN0 to EN7).

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/EIRR1:ER0 to ER7).

Operation	Interrupt Request Bit (ERn)
To clear interrupt requests	Write "0".

13.8 Notes

This section explains the notes of the external interrupt input.

The external interrupt input register is not initialized when returned from the standby clock mode (power shutdown) and the standby stop mode (power shutdown) when PMUCTLR:IOCTMD=1. To maintain the status before it returns and the status under return, set the device in the status of the I/O maintenance by setting PMUCTLR:IOCTMD before setting standby. And, release the I/O maintenance by setting PMUCTLR:IOCT after the I/O port is set. See "Chapter: Power Consumption Control" for the details of the PMUCTLR register.

Moreover, the internal reset is issued at the return from the standby watch mode (power shutdown) and the standby stop mode (power shutdown) when PMUCTLR:IOCTMD=1. Therefore, only the reset factors (power-on reset, internal low-voltage detection, and simultaneous assert of RSTX and NMIX) are accepted. At this time, the register of the external interrupt input is not initialized. If the reset input from RSTX pin input or the external low-voltage detection flag is set after the start-up, initialize the external interrupt input register before using.

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

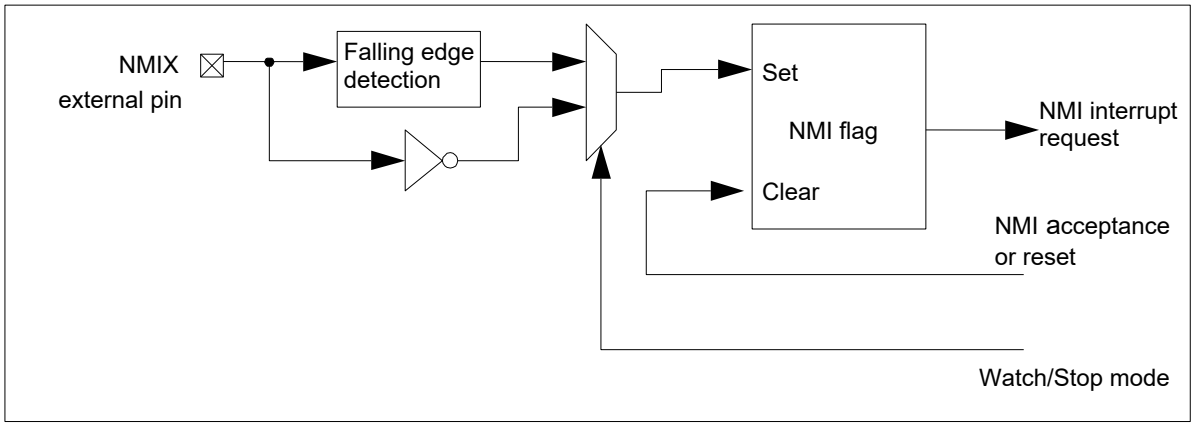
This section explains features of the NMI input

Can be used in both stop mode (Power-shutdown is included) and watch mode (Power-shutdown is included).

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

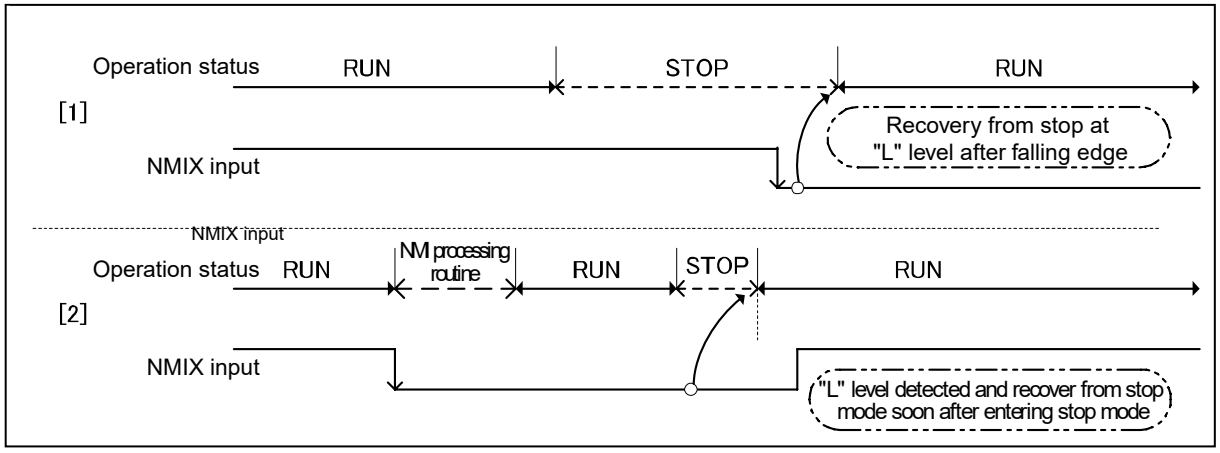
The NMI request detector has an NMI flag that is set for an NMI request and is cleared only if an interrupt for the NMI itself is accepted or reset occurs. The NMI flag cannot be read or written.

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). Similarly, the power-shutdown will not be controlled when the status changes to the stop mode (power-shutdown) without setting the NMIX pin to the "H" level. 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. NMI request is not accepted during return from the standby mode (shutdown) because the internal reset is generated.

Figure 14-2. Recovering from Stop Mode

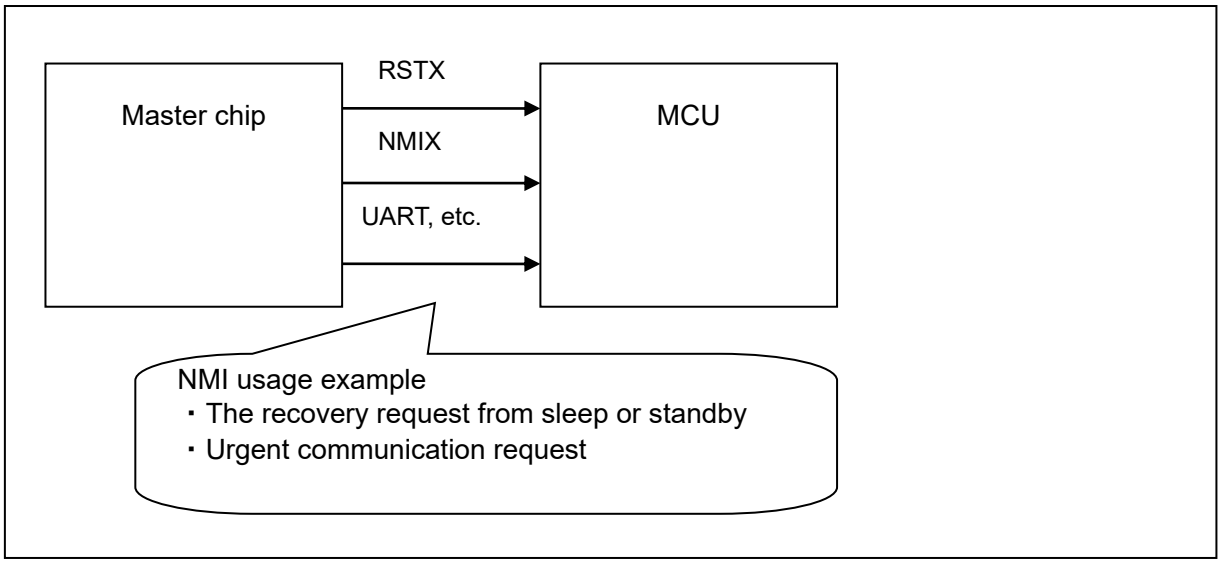


Note: The watch mode and the watch mode (power-shutdown) are similarly controlled.

14.6 Usage Example

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 cancelled by software.

15.2 Features

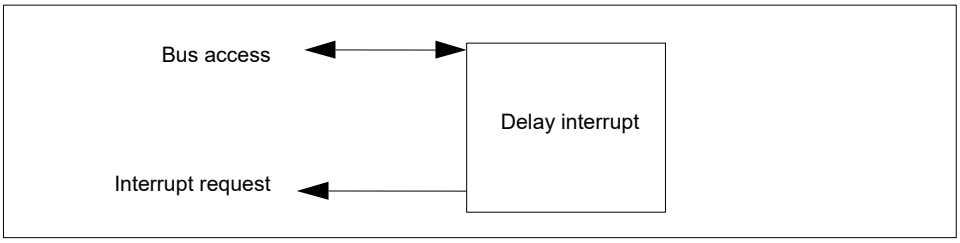
This section explains features of the delay interrupt.

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 source.

DLYI	Description
Write 0	Clears the delay interrupt source
Write 1	Generates the delay interrupt source

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 cancelled 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 source. Writing "0" to this bit cancels the delay interrupt source.

This bit functions like a standard interrupt source 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 in 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

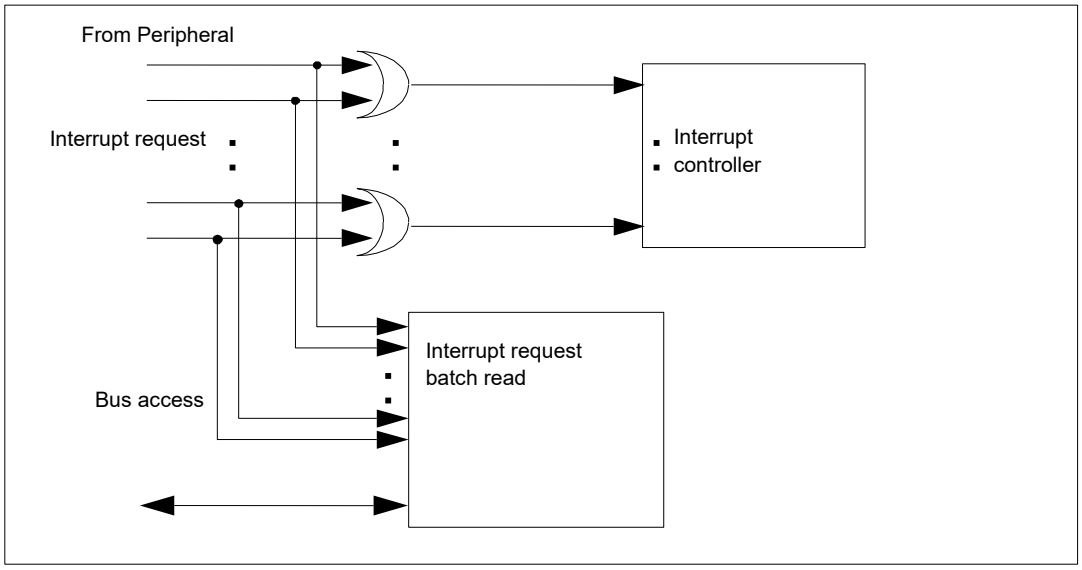
This section shows features of the interrupt request batch read.

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 Function
	+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 (#36)
0x0424	IRPR6H	IRPR6L	IRPR7H	IRPR7L	Interrupt request batch read register 6 upper-order (#45) 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 (#49)
0x0428	IRPR8H	IRPR8L	IRPR9H	IRPR9L	Interrupt request batch read register 8 upper-order (#50) Interrupt request batch read register 8 lower-order (#51) Interrupt request batch read register 9 upper-order (#52) Interrupt request batch read register 9 lower-order (#53)
0x042C	IRPR10H	IRPR10L	IRPR11H	IRPR11L	Interrupt request batch read register 10 upper-order (#54) Interrupt request batch read register 10 lower-order (#55) Interrupt request batch read register 11 upper-order (#56) 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	IRPR14L	IRPR15H	Reserved	Interrupt request batch read register 14 upper-order (#62) Interrupt request batch read register 14 lower-order (#62) Interrupt request batch read register 15 upper-order (#15)

#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 explained.

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	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] RTIR0 (Reload Timer Interrupt Request 0): Reload Timer 0, 4 Interrupt Request

[bit6] RTIR1 (Reload Timer Interrupt Request 1): Reload Timer 1, 5 Interrupt Request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

Note:

The RTIR0 shows logical OR value of the IRQ from Reload Timer ch.0 and the IRQ from Reload Timer ch.4. The RTIR1 shows logical OR value of the IRQ from Reload Timer ch.1 and the IRQ from Reload Timer ch.5.

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 explained.

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, 6 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.

Note:

The RTIR2 shows logical OR value of the IRQ from Reload Timer ch.2 and the IRQ from Reload Timer ch.6.

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 explained.

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): Multi-Function-Serial-Interface ch.0 reception completion Interrupt Request

[bit6] ISIR0 (Multi-Function-Serial-Interface Status Interrupt Request 0): Multi-Function-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 explained.

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 explained.

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
	SGIR0	RXIR7	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] SGIR0 (SG Interrupt Request 0): Sound Generator 0 Interrupt Request

[bit6] RXIR7 (RX Interrupt Request 7): LIN-UART7 reception completion Interrupt Request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.6 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 explained.

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
	SGIR1	TXIR7	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] SGIR1 (SG Interrupt Request 1): Sound Generator 1 Interrupt Request

[bit6] TXIR7 (TX Interrupt Request 7): LIN-UART7 transmission completion Interrupt Request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.7 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 explained.

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
	PPGIR0	PPGIR1	PPGIR10	PPGIR11	PPGIR20	PPGIR21	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX

[bit7] PPGIR0 (PPG Interrupt Request 0): PPG0 Interrupt Request

[bit6] PPGIR1 (PPG Interrupt Request 1): PPG1 Interrupt Request

[bit5] PPGIR10 (PPG Interrupt Request 10): PPG10 Interrupt Request

[bit4] PPGIR11 (PPG Interrupt Request 11): PPG11 Interrupt Request

[bit3] PPGIR20 (PPG Interrupt Request 20): PPG20 Interrupt Request

[bit2] PPGIR21 (PPG Interrupt Request 21): PPG21 Interrupt Request

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 explained.

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
	PPGIR2	PPGIR3	PPGIR12	PPGIR13	PPGIR22	PPGIR23	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX

[bit7] PPGIR2 (PPG Interrupt Request 2): PPG2 Interrupt Request

[bit6] PPGIR3 (PPG Interrupt Request 3): PPG3 Interrupt Request

[bit5] PPGIR12 (PPG Interrupt Request 12): PPG12 Interrupt Request

[bit4] PPGIR13 (PPG Interrupt Request 13): PPG13 Interrupt Request

[bit3] PPGIR22 (PPG Interrupt Request 22): PPG22 Interrupt Request

[bit2] PPGIR23 (PPG Interrupt Request 23): PPG23 Interrupt Request

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 explained.

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
	PPGIR4	PPGIR5	PPGIR14	PPGIR15	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] PPGIR4 (PPG Interrupt Request 4): PPG4 Interrupt Request

[bit6] PPGIR5 (PPG Interrupt Request 5): PPG5 Interrupt Request

[bit5] PPGIR14 (PPG Interrupt Request 14): PPG14 Interrupt Request

[bit4] PPGIR15 (PPG Interrupt Request 15): PPG15 Interrupt Request

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 explained.

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
	PPGIR6	PPGIR7	PPGIR16	PPGIR17	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] PPGIR6 (PPG Interrupt Request 6): PPG6 Interrupt Request

[bit6] PPGIR7 (PPG Interrupt Request 7): PPG7 Interrupt Request

[bit5] PPGIR16 (PPG Interrupt Request 16): PPG16 Interrupt Request

[bit4] PPGIR17 (PPG Interrupt Request 17): PPG17 Interrupt Request

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 explained.

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
	PPGIR8	PPGIR9	PPGIR18	PPGIR19	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] PPGIR8 (PPG Interrupt Request 8): PPG8 Interrupt Request

[bit6] PPGIR9 (PPG Interrupt Request 9): PPG9 Interrupt Request

[bit5] PPGIR18 (PPG Interrupt Request 18): PPG18 Interrupt Request

[bit4] PPGIR19 (PPG Interrupt Request 19): PPG19 Interrupt Request

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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #36)

IRPR5L: Address 0423$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CANIR2	UDCIR0	UDCIR1	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] CANIR2 (CAN Interrupt Request 2): CAN ch.2 Interrupt Request

[bit6] UDCIR0 (UpDown Counter Interrupt Request 0): Up/Down counter ch.0 Interrupt Request

[bit5] UDCIR1 (UpDown Counter Interrupt Request 1): Up/Down counter ch.1 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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #45)

IRPR6H: Address 0424$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		RXIR8	ISIR8	HS_RIR	Reserved		
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX

[bit5] RXIR8 (multifunction serial RX Interrupt Request 8): Multi-function Serial interface ch.8 Reception Completion Interrupt Request

[bit4] ISIR8 (multifunction serial Inform Status Interrupt Request 8): Multi-function Serial interface ch.8 Status Interrupt Request

[bit3] HS_RIR (HS_SPI RX Interrupt Request): HS_SPI Reception 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 explained.

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	STIR	PTIR	TXIR8	HS_TIR	Reserved		
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX

[bit7] MTIR (Main Timer Interrupt Request): Main Timer Interrupt Request

[bit6] STIR (Sub Timer Interrupt Request): Sub Timer Interrupt Request

[bit5] PTIR (PLL Timer Interrupt Request): PLL Timer Interrupt Request

[bit4] TXIR8 (multifunction serial TX Interrupt Request 8): Multifunction Serial interface ch.8 Transmission Completion Interrupt Request

[bit3] HS_TIR (HS_SPI TX Interrupt Request): HS_SPI Transmission 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 explained.

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
	Reserved	SUBIR	SGIR4	RXIR9	ISIR9	Reserved		
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX	R0,WX

[bit6] SUBIR (SUB Interrupt Request): Clock Calibration (Sub) Interrupt Request

[bit5] SGIR4 (SG Interrupt Request 4): Sound Generator 4 Interrupt Request

[bit4] RXIR9 (multifunction serial RX Interrupt Request 9) Multi-function Serial Interface ch.9 Reception Completion Interrupt Request

[bit3] ISIR9 (multifunction serial Inform Status Interrupt Request 9) Multi-function Serial Interface ch.9 Status Interrupt Request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.16 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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #49)

IRPR7L: Address 0427$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved						CRIR	TXIR9
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R,WX	R,WX

[bit1] CRIR (CR clock calibration Interrupt Request): Clock Calibration (CR) Interrupt Request

[bit0] TXIR9 (multifunction serial TX Interrupt Request 9): Multi-function Serial Interface ch.9 Transmission 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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #50)

IRPR8H: Address 0428$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FRTIR0	FRTIR2	FRTIR4	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] FRTIR0 (FRT Interrupt Request 0): Free-run Timer ch.0 Interrupt Request

[bit6] FRTIR2 (FRT Interrupt Request 2): Free-run Timer ch.2 Interrupt Request

[bit5] FRTIR4 (FRT Interrupt Request 4): Free-run Timer ch.4 Interrupt Request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.18 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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #51)

IRPR8L: Address 0429_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FRTIR1	FRTIR3	FRTIR5	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] FRTIR1 (FRT Interrupt Request 1): Free-run Timer ch.1 Interrupt Request

[bit6] FRTIR3 (FRT Interrupt Request 3): Free-run Timer ch.3 Interrupt Request

[bit5] FRTIR5 (FRT Interrupt Request 5): Free-run Timer ch.5 Interrupt Request

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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #52)

IRPR9H: Address 042A$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUIR0	ICUIR6	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] ICUIR0 (ICU Interrupt Request 0): Input Capture ch.0 Interrupt Request

[bit6] ICUIR6 (ICU Interrupt Request 6): Input Capture ch.6 Interrupt Request

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)

The bit configuration of the interrupt request batch read register 9 lower-order is explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #53)

IRPR9L: Address 042B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUIR1	ICUIR7	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] ICUIR1 (ICU Interrupt Request 1): Input Capture ch.1 Interrupt Request

[bit6] ICUIR7 (ICU Interrupt Request 7): Input Capture ch.7 Interrupt Request

Read Value of Each Bit	Meaning
0	No interrupt request has been issued.
1	An interrupt request has been issued.

16.4.21 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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #54)

IRPR10H: Address 042C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUIR2	ICUIR8	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] ICUIR2 (ICU Interrupt Request 2): Input Capture ch.2 Interrupt Request

[bit6] ICUIR8 (ICU Interrupt Request 8): Input Capture ch.8 Interrupt Request

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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #55)

IRPR10L: Address 042D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUIR3	ICUIR9	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] ICUIR3 (ICU Interrupt Request 3): Input Capture ch.3 Interrupt Request

[bit6] ICUIR9 (ICU Interrupt Request 9): Input Capture ch.9 Interrupt Request

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 explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #56)

IRPR11H: Address 042E$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICUIR4	ICUIR10	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] ICUIR4 (ICU Interrupt Request 4): Input Capture ch.4 Interrupt Request

[bit6] ICUIR10 (ICU Interrupt Request 10): Input Capture ch.10 Interrupt Request

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 explained.

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
	ICUIR5	ICUIR11	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] ICUIR5 (ICU Interrupt Request 5): Input Capture ch.5 Interrupt Request

[bit6] ICUIR11 (ICU Interrupt Request 11): Input Capture ch.11 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 explained.

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
	OCUIR0	OCUIR1	OCUIR6	OCUIR7	OCUIR10	OCUIR11	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX

[bit7] OCUIR0 (OCU Interrupt Request 0): Output Compare ch.0 Interrupt Request

[bit6] OCUIR1 (OCU Interrupt Request 1): Output Compare ch.1 Interrupt Request

[bit5] OCUIR6 (OCU Interrupt Request 6): Output Compare ch.6 Interrupt Request

[bit4] OCUIR7 (OCU Interrupt Request 7): Output Compare ch.7 Interrupt Request

[bit3] OCUIR10 (OCU Interrupt Request 10): Output Compare ch.10 Interrupt Request

[bit2] OCUIR11 (OCU Interrupt Request 11): Output Compare ch.11 Interrupt Request

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 explained.

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
	OCUIR2	OCUIR3	OCUIR4	OCUIR5	OCUIR8	OCUIR9	Reserved	
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R0,WX	R0,WX

[bit7] OCUIR2 (OCU Interrupt Request 2): Output Compare ch.2 Interrupt Request

[bit6] OCUIR3 (OCU Interrupt Request 3): Output Compare ch.3 Interrupt Request

[bit5] OCUIR4 (OCU Interrupt Request 4): Output Compare ch.4 Interrupt Request

[bit4] OCUIR5 (OCU Interrupt Request 5): Output Compare ch.5 Interrupt Request

[bit3] OCUIR8 (OCU Interrupt Request 8): Output Compare ch.8 Interrupt Request

[bit2] OCUIR9 (OCU Interrupt Request 9): Output Compare ch.9 Interrupt Request

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 explained.

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	SGIR2	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] 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

[bit5] SGIR2 (SG Interrupt Request 2): Sound Generator 2 Interrupt Request

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 explained.

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	SGIR3	XB_ECC_SE	BR_ECC_SE	Reserved		
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,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

[bit5] SGIR3 (SG Interrupt Request 3): Sound Generator 3 Interrupt Request

[bit4] XB_ECC_SE: XBS RAM single bit error generation Interrupt Request

[bit3] BR_ECC_SE: Backup RAM single bit error generation Interrupt Request

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 explained.

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 14 Lower-order: IRPR14L (Interrupt Request Peripheral Read register 14L)

The bit configuration of the interrupt request batch read register 14 lower-order is explained.

This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #62)

IRPR14L: Address 0435$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DMAC8IR	DMAC9IR	DMAC10IR	DMAC11IR	DMAC12IR	DMAC13IR	DMAC14IR	DMAC15IR
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] DMAC8IR (DMAC 8 Interrupt Request): DMAC ch.8 Interrupt Request

[bit6] DMAC9IR (DMAC 9 Interrupt Request): DMAC ch.9 Interrupt Request

[bit5] DMAC10IR (DMAC 10 Interrupt Request): DMAC ch.10 Interrupt Request

[bit4] DMAC11IR (DMAC 11 Interrupt Request): DMAC ch.11 Interrupt Request

[bit3] DMAC12IR (DMAC 12 Interrupt Request): DMAC ch.12 Interrupt Request

[bit2] DMAC13IR (DMAC 13 Interrupt Request): DMAC ch.13 Interrupt Request

[bit1] DMAC14IR (DMAC 14 Interrupt Request): DMAC ch.14 Interrupt Request

[bit0] DMAC15IR (DMAC 15 Interrupt Request): DMAC ch.15 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 Upper-order: IRPR15H (Interrupt Request Peripheral Read register 15H)

The bit configuration of the interrupt request batch read register 15 upper-order is explained.

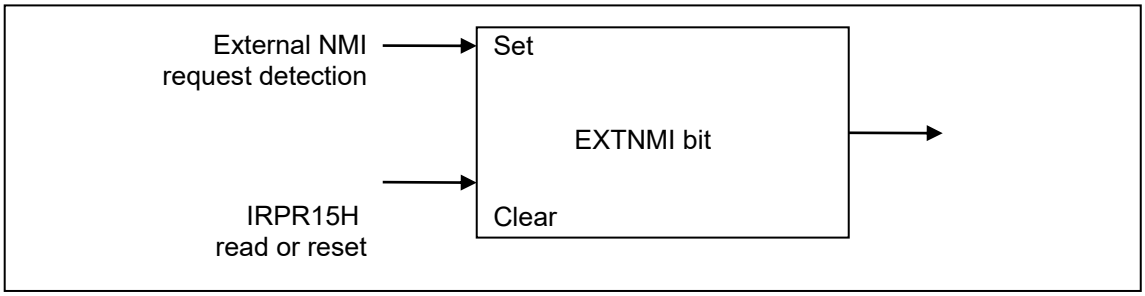
This register indicates the peripheral that has issued the interrupt request. (Interrupt vector number #15)

IRPR15H: Address 0436$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EXTNMI	XB_ECC_DE	BR_ECC_DE	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] EXTNMI: External NMI Request

The EXTNMI bit is set by detecting external NMI request, and cleared by reading this register. NMI request is not retained during return from the standby mode (shut-down) because the internal reset is generated.



[bit6] XB_ECC_DE: XBS RAM double bit error generation Interrupt Request

[bit5] BR_ECC_DE: Backup RAM double bit error generation 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.

Note:

This register does not provide a function that can be used to input external interrupts.
Read registers EIRR0 and EIRR1, which are used to input external interrupts.

17. PPG

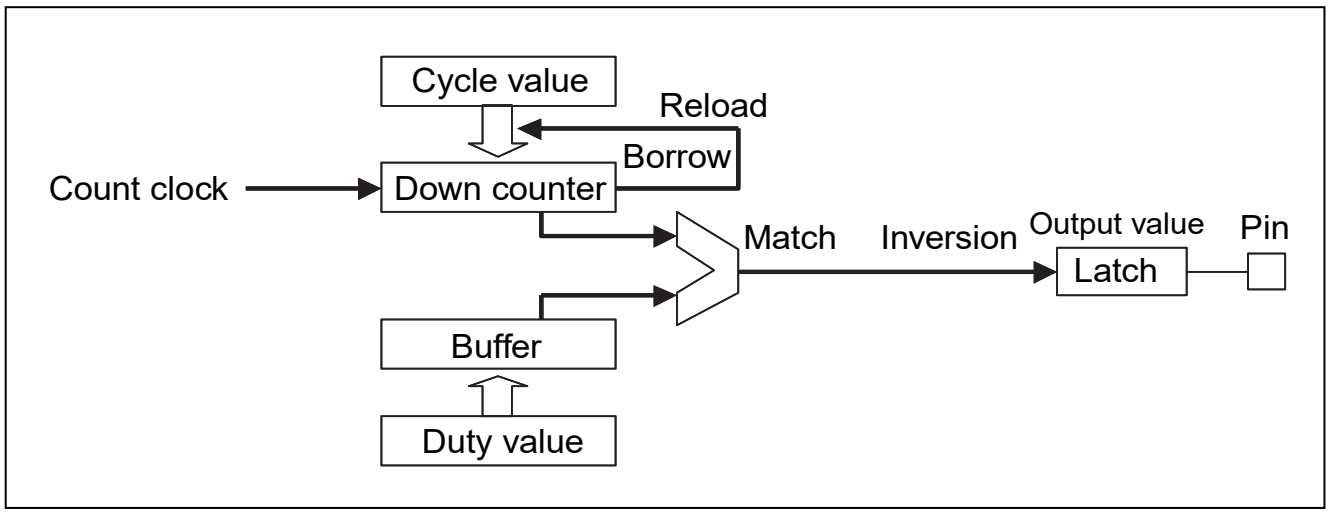
This chapter explains the PPG.

17.1 Overview
17.2 Features
17.3 Configuration
17.4 Registers
17.5 Operation
17.6 Setting
17.7 Q&A
17.8 Sample Programs
17.9 Notes

17.1 Overview

This section explains an overview of the PPG.

The programmable pulse generator (PPG) is used to generate one-shot (rectangular wave) or pulse width modulation (PWM) outputs. The PPG can be used in a wide range of applications because the cycle and duty of its output can be freely changed by software.

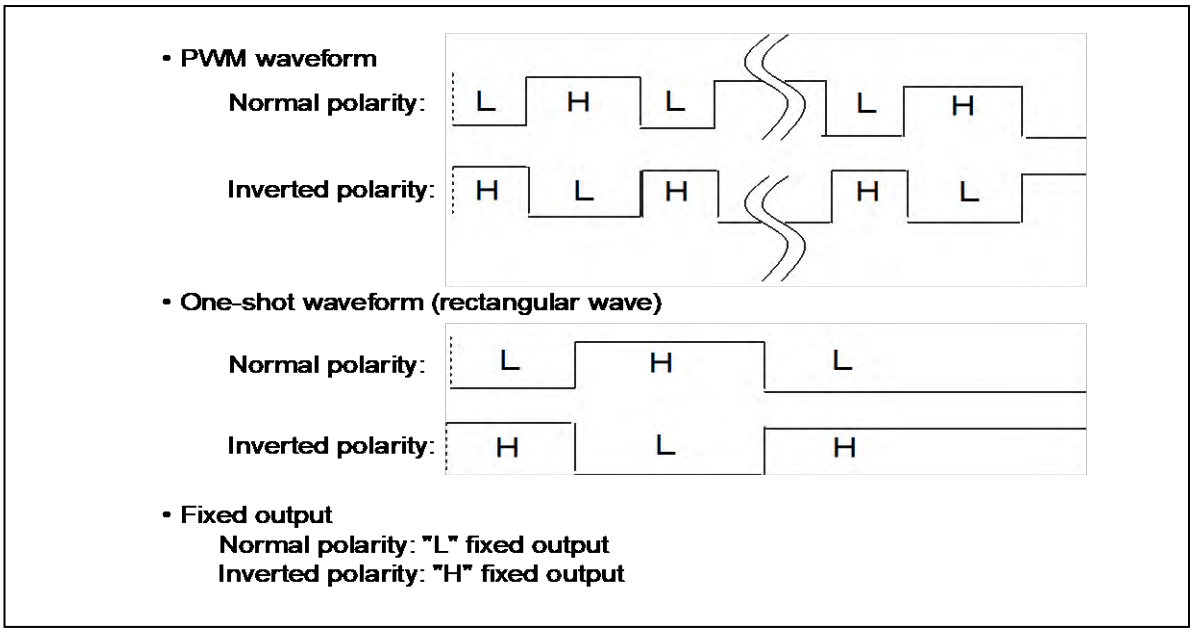


17.2 Features

This section explains features of the PPG.

- Number of PPG: 24 (Output: 24 channels, PPG0, PPG1, PPG2, PPG3, PPG4, PPG5, PPG6, PPG7, PPG8, PPG9, PPG10, PPG11, PPG12, PPG13, PPG14, PPG15, PPG16, PPG17, PPG18, PPG19, PPG20, PPG21, PPG22, PPG23)
- Count clock: Selects from 4 types (Peripheral clock (PCLK)/1, /4, /16, /64)
- Cycle:
  Cycle = Count clock × (PCSR register value + 1)
  (Example) Count clock = 16 MHz (62.5 ns), PCSR value = 63999
  Cycle = 62.5 ns × (63999+1) = 4ms
- Duty:
  Duty = Count clock × (PDUT register value + 1)
- Output waveform: 6 types shown in the figure below:

Figure 17-1. Output Waveforms



Interrupt factors: One of the following four interrupts is selected

- Software trigger and trigger input
- Borrow occurrence on the counter (match with the specified cycle)
- Duty match
- Borrow occurrence on the counter (match with the specified cycle) or duty match

Activation triggers

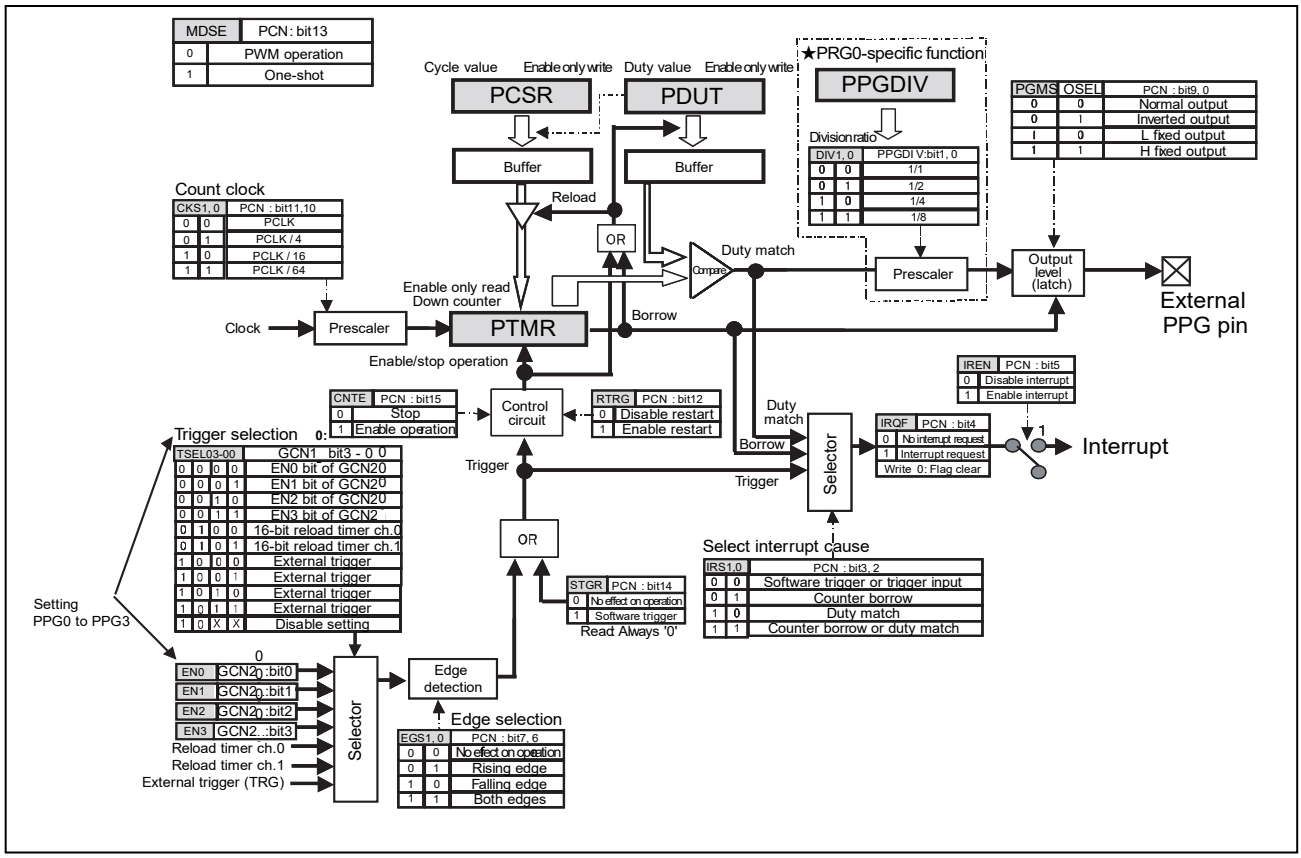
- Software trigger (set with software trigger bit)
- Internal trigger:

  Trigger with register written
  Trigger with reload timer
- External trigger

17.3 Configuration

This section explains a configuration of the PPG.

Figure 17-2. Configuration Diagram (for Each Channel)



17.4 Registers

This section explains registers of the PPG.

Table of Base Addresses (Base_addr) and External Pins

Table 17-1. Table of Base Addresses and External Pins

Channel	Base_addr	External Pin	
		PPG Output	Trigger Input
0	0x026C	PPG0_0/PPG0_1/PPG0_2	TRG0
1	0x0274	PPG1_0/PPG1_1/PPG1_2/PPG1_3	
2	0x027C	PPG2_0/PPG2_1/PPG2_2	
3	0x0284	PPG3_0/PPG3_1/PPG3_2	
4	0x028C	PPG4_0/PPG4_1/PPG4_2	TRG1
5	0x0294	PPG5_0/PPG5_1/PPG5_2	
6	0x029C	PPG6_0/PPG6_1/PPG6_2	
7	0x02A4	PPG7_0/PPG7_1/PPG7_2	
8	0x02AC	PPG8_0/PPG8_1/PPG8_2	TRG2
9	0x02B4	PPG9_0/PPG9_1/PPG9_2	
10	0x02BC	PPG10_0/PPG10_1/PPG10_2	
11	0x0150	PPG11_0/PPG11_1	
12	0x0158	PPG12_0/PPG12_1	TRG3
13	0x0160	PPG13_0/PPG13_1	
14	0x0168	PPG14_0/PPG14_1	
15	0x0170	PPG15_0/PPG15_1	

Channel	Base_addr	External Pin	
		PPG Output	Trigger Input
16	0x0178	PPG16_0	TRG4
17	0x0180	PPG17_0	
18	0x0188	PPG18_0	
19	0x0190	PPG19_0	
20	0x0198	PPG20_0	TRG5
21	0x01A0	PPG21_0	
22	0x01A8	PPG22_0	
23	0x01B0	PPG23_0	

Registers map

Table 17-2. Registers map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0144	GCN13		Reserved	GCN23	General control register 13 General control register 23
0x0148	GCN14		Reserved	GCN24	General control register 14 General control register 24
0x014C	GCN15		Reserved	GCN25	General control register 15 General control register 25
0x0150	PTMR11		PCSR11		PPG timer register 11 PPG cycle setting register 11
0x0154	PDUT11		PCN11		PPG duty setting register 11 PPG control status register 11
0x0158	PTMR12		PCSR12		PPG timer register 12 PPG cycle setting register 12
0x015C	PDUT12		PCN12		PPG duty setting register 12 PPG control status register 12
0x0160	PTMR13		PCSR13		PPG timer register 13 PPG cycle setting register 13
0x0164	PDUT13		PCN13		PPG duty setting register 13 PPG control status register 13
0x0168	PTMR14		PCSR14		PPG timer register 14 PPG cycle setting register 14
0x016C	PDUT14		PCN14		PPG duty setting register 14 PPG control status register 14
0x0170	PTMR15		PCSR15		PPG timer register 15 PPG cycle setting register 15
0x0174	PDUT15		PCN15		PPG duty setting register 15 PPG control status register 15
0x0178	PTMR16		PCSR16		PPG timer register 16 PPG cycle setting register 16
0x017C	PDUT16		PCN16		PPG duty setting register 16 PPG control status register 16

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0180	PTMR17		PCSR17		PPG timer register 17 PPG cycle setting register 17
0x0184	PDUT17		PCN17		PPG duty setting register 17 PPG control status register 17
0x0188	PTMR18		PCSR18		PPG timer register 18 PPG cycle setting register 18
0x018C	PDUT18		PCN18		PPG duty setting register 18 PPG control status register 18
0x0190	PTMR19		PCSR19		PPG timer register 19 PPG cycle setting register 19
0x0194	PDUT19		PCN19		PPG duty setting register 19 PPG control status register 19
0x0198	PTMR20		PCSR20		PPG timer register 20 PPG cycle setting register 20
0x019C	PDUT20		PCN20		PPG duty setting register 20 PPG control status register 20
0x01A0	PTMR21		PCSR21		PPG timer register 21 PPG cycle setting register 21
0x01A4	PDUT21		PCN21		PPG duty setting register 21 PPG control status register 21
0x01A8	PTMR22		PCSR22		PPG timer register 22 PPG cycle setting register 22
0x01AC	PDUT22		PCN22		PPG duty setting register 22 PPG control status register 22
0x01B0	PTMR23		PCSR23		PPG timer register 23 PPG cycle setting register 23
0x01B4	PDUT23		PCN23		PPG duty setting register 23 PPG control status register 23
0x025C	GCN10		Reserved	GCN20	General control register 10 General control register 20
0x0260	GCN11		Reserved	GCN21	General control register 11 General control register 21

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0264	GCN12		Reserved	GCN22	General control register 12 General control register 22
0x0268	Reserved			PPGDIV	PPG0 output division setting register
0x026C	PTMR0		PCSR0		PPG timer register 0 PPG cycle setting register 0
0x0270	PDUT0		PCN0		PPG duty setting register 0 PPG control status register 0
0x0274	PTMR1		PCSR1		PPG timer register 1 PPG cycle setting register 1
0x0278	PDUT1		PCN1		PPG duty setting register 1 PPG control status register 1
0x027C	PTMR2		PCSR2		PPG timer register 2 PPG cycle setting register 2
0x0280	PDUT2		PCN2		PPG duty setting register 2 PPG control status register 2
0x0284	PTMR3		PCSR3		PPG timer register 3 PPG cycle setting register 3
0x0288	PDUT3		PCN3		PPG duty setting register 3 PPG control status register 3
0x028C	PTMR4		PCSR4		PPG timer register 4 PPG cycle setting register 4
0x0290	PDUT4		PCN4		PPG duty setting register 4 PPG control status register 4
0x0294	PTMR5		PCSR5		PPG timer register 5 PPG cycle setting register 5
0x0298	PDUT5		PCN5		PPG duty setting register 5 PPG control status register 5
0x029C	PTMR6		PCSR6		PPG timer register 6 PPG cycle setting register 6
0x02A0	PDUT6		PCN6		PPG duty setting register 6 PPG control status register 6

Address	Registers				Register Function
	+0	+1	+2	+3	
0x02A4	PTMR7		PCSR7		PPG timer register 7 PPG cycle setting register 7
0x02A8	PDUT7		PCN7		PPG duty setting register 7 PPG control status register 7
0x02AC	PTMR8		PCSR8		PPG timer register 8 PPG cycle setting register 8
0x02B0	PDUT8		PCN8		PPG duty setting register 8 PPG control status register 8
0x02B4	PTMR9		PCSR9		PPG timer register 9 PPG cycle setting register 9
0x02B8	PDUT9		PCN9		PPG duty setting register 9 PPG control status register 9
0x02BC	PTMR10		PCSR10		PPG timer register 10 PPG cycle setting register 10
0x02C0	PDUT10		PCN10		PPG duty setting register 10 PPG control status register 10

17.4.1 PPG Cycle Setting Register: PCSR

The bit configuration of the PPG cycle setting register (PCSR) is shown below.

The PPG cycle setting register (PCSR) specifies a cycle of the PPG.

PCSR: Address Base_addr + 02_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	S15	S14	S13	S12	S11	S10	S9	S8
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	S7	S6	S5	S4	S3	S2	S1	S0
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

- The PPG cycle setting register has a buffer. Data transfer from the buffer to the counter occurs automatically when a borrow occurs on the counter.
- Be sure to set the PPG duty setting register (PDUT) after the PPG cycle setting register is rewritten.
- PPG cycle setting registers must be accessed in half-word (16-bit) or word (32-bit).
  (See "17.9 Notes".)

17.4.2 PPG Duty Setting Register: PDUT

The bit configuration of the PPG duty setting register (PDUT) is shown below.

The PPG duty setting register (PDUT) specifies the duty of the PPG output waveform.

PDUT: Address Base_addr + 04_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D9	D8
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
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

- The PPG duty setting register has a buffer. Data transfer from the buffer to the counter occurs automatically when a borrow occurs on the counter.
- For the PPG duty setting register, set a value that is smaller than the value set for the PPG cycle setting register (PCSR).
  (See "17.9 Notes".)
- If an equal value is set on the PPG duty setting register and the PPG cycle setting register (PCSR), the result is as follows:
  - ☐ If the polarity is normal (OSEL = "0"), the output is always "H".
  - ☐ If the polarity is inverted (OSEL = "1"), the output is always "L".
    (The OSEL bit is the output polarity selection bit on the PPG control status register (PCN).)
- PPG duty setting registers must be accessed in half-word (16-bit) or word (32-bit).
  (See "17.9 Notes".)

17.4.3 PPG Control Status Register: PCN

The bit configuration of the PPG control status register (PCN) is shown below.

The PPG control status register (PCN) controls operation of the PPG and shows status of the PPG as well.

PCN: Address Base_addr + 06_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CNTE	STRG	MDSE	RTRG	CKS1	CKS0	PGMS	-
Initial value	0	0	0	0	0	0	0	-
Attribute	R/W	R0,W	R/W	R/W	R/W	R/W	R/W	R1,WX
Rewrite while in operation	○	○	×	×	×	×	○	×

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	Reserved	OSEL
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W0	R/W
Rewrite while in operation	×	×	○	○	×	×	×	×

○: Rewrite enabled

×: Rewrite disabled (See "17.9 Notes")

[bit15] CNTE: Timer operation enable

CNTE	Operation
0	Inactive
1	Active

This bit enables the operation of the PPG.

[bit14] STRG: Software trigger

STRG	Operation
0	PPG operation is not influenced by the value written to this bit (which is always "0" when read).
1	A software trigger is generated.

If this bit is set to "1", the PPG is activated by a software trigger. The software trigger activates the PPG independent of the trigger generated by the EN bit.

[bit13] MDSE: Mode selection

MDSE	Mode
0	PWM operation
1	One-shot operation

- If this bit is set to "0", the PPG is enabled to perform PWM operation, thus generating a sequence of pulses.
- If this bit is set to "1", the PPG generates only one pulse.

[bit12] RTRG: Restart enable

RTRG	Operation
0	Restart disabled
1	Restart enabled

When the restart enable bit is set to "1", the PPG is enabled to restart with a trigger (such as software, an internal factor, or an external factor).

[bit11, bit10] CKS1, CKS0: Count clock selection

CKS1	CKS0	Down Counter Count Clock Selection
0	0	Peripheral clock (PCLK)
0	1	Division of the peripheral clock frequency by 4
1	0	Division of the peripheral clock frequency by 16
1	1	Division of the peripheral clock frequency by 64

[bit9] PGMS: PPG output mask selection

PGMS	Operation
0	No output mask
1	Output mask (Output is fixed to "L": OSEL = "0")

- When this bit is set to "1", the PPG output can be clamped to "L" or "H" regardless of the mode selection, cycle, and duty settings.
- The output level can be specified by the output polarity selection bit (PCNn:OSEL). (n = 0 to 23)

Note:

Cancel output mask by setting "1" to "0" to this bit before the duty match within the period.

[bit8] - : Undefined bit

The read value is always "1". This does not affect the writing operation.

[bit7, bit6] EGS1, EGS0: Trigger input edge selection

EGS1	EGS0	Selected Edge
0	0	Writing does not affect on the operation
0	1	Rising edge
1	0	Falling edge
1	1	Both edges (rising or falling)

Select a source edge for activation with the trigger input edge selection bits (ESG[1:0]) to the trigger input selected by the trigger specification bits (GCN10/11/12/13/14/15:TSEL3/2/1/0) of the PPG registers.

[bit5] IREN: Interrupt request enable

IREN	Operation
0	Interrupt request disabled
1	Interrupt request enabled

[bit4] IRQF: Interrupt request flag

IRQF	Read	Write
0	No interrupt request	Clears the interrupt request flag.
1	Interrupt request present	Writing does not affect on the operation

If this bit is set to "0" when the interrupt request flag (IRQF) = "1", the interrupt request flag (IRQF ="1") that is set takes precedence.

[bit3, bit2] IRS1, IRS0: Interrupt factor selection

IRS1	IRS0	Selection
0	0	Software trigger or trigger input
0	1	Borrow occurrence on the counter (match with the specified cycle)
1	0	Counter matched with the specified duty value
1	1	Borrow occurrence on the counter (matched with the specified cycle) or counter matched with duty value

These bits select the operation that generates an interrupt request.

[bit1] Reserved

"0" should be written to this bit.

[bit0] OSEL: PPG output polarity selection

OSEL	Operation
0	Normal polarity
1	Inverted polarity

If the PPG output mask selection bit (PCNn:PGMS) is set to "1", setting the output polarity selection bit (OSEL) to "0" or "1" causes the output to be clamped to "L" or "H", respectively. (n = 0 to 23)

17.4.4 General Control Register 10-13: GCN10 to GCN13

The bit configuration of the general control register 10-13 (GCN10 to GCN13) is shown below.

The general control register selects the trigger input for PPG0 to PPG15.

GCN10: PPG0 to PPG3
GCN11: PPG4 to PPG7
GCN12: PPG8 to PPG11
GCN13: PPG12 to PPG15

GCN10: Address 025C$_H$ (Access: Half-word)

GCN11: Address 0260$_H$ (Access: Half-word)

GCN12: Address 0264$_H$ (Access: Half-word)

GCN13: Address 0144$_H$ (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TSEL3[3:0]				TSEL2[3:0]			
Initial value	0	0	1	1	0	0	1	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
	TSEL1[3:0]				TSEL0[3:0]			
Initial value	0	0	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 bit12] TSEL3[3:0]: Trigger specification for PPG3, PPG7, PPG11 and PPG15

[bit11 to bit8] TSEL2[3:0]: Trigger specification for PPG2, PPG6, PPG10 and PPG14

[bit7 to bit4] TSEL1[3:0]: Trigger specification for PPG1, PPG5, PPG9 and PPG13

[bit3 to bit0] TSEL0[3:0]: Trigger specification for PPG0, PPG4, PPG8 and PPG12

TSEL0[3:0] (PPG0/4/8/12) TSEL1[3:0] (PPG1/5/9/13) TSEL2[3:0] (PPG2/6/10/14) TSEL3[3:0] (PPG3/7/11/15)				Activation trigger specification
0	0	0	0	EN0 bit (GCN20/21/22/23 register)
0	0	0	1	EN1 bit (GCN20/21/22/23 register)
0	0	1	0	EN2 bit (GCN20/21/22/23 register)
0	0	1	1	EN3 bit (GCN20/21/22/23 register)
0	1	0	0	16-bit reload timer 0
0	1	0	1	16-bit reload timer 1
1	0	0	0	External trigger
1	0	0	1	External trigger
1	0	1	0	External trigger
1	0	1	1	External trigger
1	1	X	X	Setting is prohibited
Other than above				Setting is prohibited (See "17.9 Notes")

When an edge that is specified with the trigger input edge selection bits (PCNn:EGS[1:0]) (n = 0 to 15) is detected for the specified activation trigger, selected PPG0 to PPG15 will be activated.

17.4.5 General Control Register 14, 15: GCN14, GCN15

The bit configuration of the general control register 14,15 (GCN14, GCN15) is shown below.

The general control register selects the trigger input for PPG16 to PPG23.
GCN14: PPG16 to PPG19
GCN15: PPG20 to PPG23

GCN14: Address 0148$_H$ (Access: Half-word)

GCN15: Address 014C$_H$ (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TSEL3[3:0]				TSEL2[3:0]			
Initial value	0	0	1	1	0	0	1	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
	TSEL1[3:0]				TSEL0[3:0]			
Initial value	0	0	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 bit12] TSEL3[3:0]: Trigger specification for PPG19 and PPG23

[bit11 to bit8] TSEL2[3:0]: Trigger specification for PPG18 and PPG22

[bit7 to bit4] TSEL1[3:0]: Trigger specification for PPG17 and PPG21

[bit3 to bit0] TSEL0[3:0]: Trigger specification for PPG16 and PPG20

TSEL0[3:0] (PPG16/20) TSEL1[3:0] (PPG17/21) TSEL2[3:0] (PPG18/22) TSEL3[3:0] (PPG19/23)				Activation Trigger Specification
0	0	0	0	EN0 bit (GCN24/25 register)
0	0	0	1	EN1 bit (GCN24/25 register)
0	0	1	0	EN2 bit (GCN24/25 register)
0	0	1	1	EN3 bit (GCN24/25 register)
0	1	0	0	16-bit reload timer 2
0	1	0	1	16-bit reload timer 3
1	0	0	0	External trigger
1	0	0	1	External trigger
1	0	1	0	External trigger
1	0	1	1	External trigger
1	1	X	X	Setting is prohibited
Other than above				Setting is prohibited (See "17.9 Notes")

When an edge that is specified with the trigger input edge selection bits (PCNn:EGS[1:0]) (n = 16 to 23) is detected for the specified activation trigger, selected PPG16 to PPG23 will be activated.

17.4.6 General Control Register 20-25: GCN20 to GCN25

The bit configuration of the general control register 20-25 (GCN20 to GCN25) is shown below.

The general control register generates the internal trigger level with software for PPG0 to PPG23.

GCN20: PPG0 to PPG3
GCN21: PPG4 to PPG7
GCN22: PPG8 to PPG11
GCN23: PPG12 to PPG15
GCN24: PPG16 to PPG19
GCN25: PPG20 to PPG23

GCN20: Address 025F$_H$ (Access: Byte)

GCN21: Address 0263$_H$ (Access: Byte)

GCN22: Address 0267$_H$ (Access: Byte)

GCN23: Address 0147$_H$ (Access: Byte)

GCN24: Address 014B$_H$ (Access: Byte)

GCN25: Address 014F$_H$ (Access: Byte)

	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	R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	R/W

[bit7 to bit4] Reserved

These bits must always be written to "0". (See "9. Notes ".)

[bit3] EN3: trigger input

[bit2] EN2: trigger input

[bit1] EN1: trigger input

[bit0] EN0: trigger input

ENn	Internal Triggers ENn
0	Sets the level to "L"
1	Sets the level to "H"

- Sets the internal trigger level.
- When one of the EN trigger inputs (EN0, EN1, EN2, EN3) is selected for the trigger specification bits (TSEL3, TSEL2, TSEL1, TSEL0) in the general control registers (GCN10 to GCN15), selected EN becomes the trigger input bit for the PPG.
- When the state selected with the trigger input edge selection bits (EGS[1:0]) of the PPG control status register is activated by the trigger input bits (selected EN0, EN1, EN2 and EN3) with software, this trigger will activate the PPG.

17.4.7 PPG Timer Register: PTMR

The bit configuration of the PPG timer register (PTMR) is shown below.

The PPG timer register (PTMR) allows reading the PPG timer count down values of PPG0 to PPG23.

PTMR: Address Base_addr + 00_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	T15	T14	T13	T12	T11	T10	T9	T8
Initial value	1	1	1	1	1	1	1	1
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
	T7	T6	T5	T4	T3	T2	T1	T0
Initial value	1	1	1	1	1	1	1	1
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

- The count value of the 16-bit down counter can be read from these bits.
- The PPG timer register (PTMRn) cannot be read correctly by the byte access. (n = 0 to 23)

17.4.8 PPG0 Output Division Setting Register: PPGDIV

The bit configuration of the PPG0 output division setting register (PPGDIV) is shown below.

The PPG0 output division setting register (PPGDIV) sets the output division ratio for PPG0.

PPGDIV: Address 026B$_H$ (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	-	-	DIV1	DIV0
Initial value	-	-	-	-	-	-	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W

[bit7 to bit2] - : Undefined

The read value is always "1". Writing does not affect the operation.

[bit1, bit0] DIV1, DIV: division ratio setting

DIV1	DIV0	Division Ratio
0	0	1/1
0	1	1/2
1	0	1/4
1	1	1/8

Sets the division ratio for PPG0 output.

Note:

Following restrictions will apply for setting 1/2, 1/4 and 1/8 divisions.

- The duty of the output waveform is fixed to 50%.
- Setting the one-shot operation (PCN:MDSE = 1) is prohibited.
- Setting the PPG reversed output function (PCN:OSEL = 1) is prohibited.
- Setting the PPG fixed output state (PCN:PGMS, OSEL = 01, 10, 11) is prohibited.
- Setting is prohibited when PCSR = PDUT.

17.5 Operation

This section explains the operation of the PPG.

There are 24 of PPG (Programmable Pulse Generator) to output programmable pulses independently/ systematically.

Followings are explanations for each operation mode.

17.5.1 PWM Operation

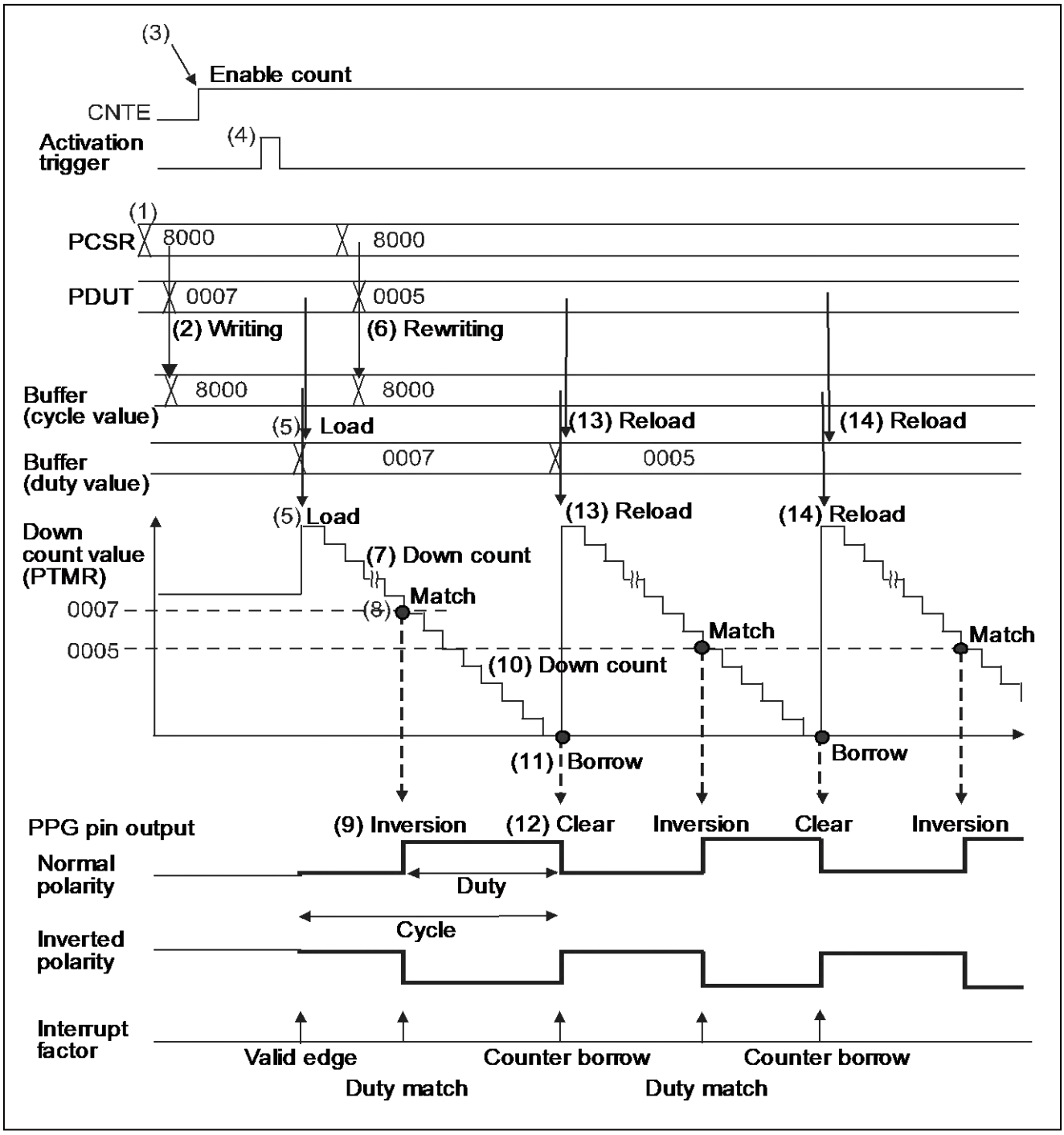
17.5.2 One-shot Operation

17.5.3 Restart Operation

17.5.1 PWM Operation

This section explains the PWM operation of the PPG.

During the PWM operation, programmable variable-duty pulses are output at the PPG pin.



(1) Writing cycle values
(2) Writing duty value and transferring cycle value to the buffer
(3) Enabling of PPG operation
(4) Activation trigger generation
(5) Loading cycle value and duty value
(6) Rewriting duty value and transferring cycle value to the buffer
(7) Counter decrement
(8) The down counter matches the duty value
(9) Output level inversion at the PPG pin
(10) Counter decrement
(11) Counter borrow occurrence
(12) Clearing PPG pin output level (restoration to normal state)
(13) Reloading cycle value
(14) Reloading duty value
(15) Repeat step (6) to (14) (See "17.9 Notes").

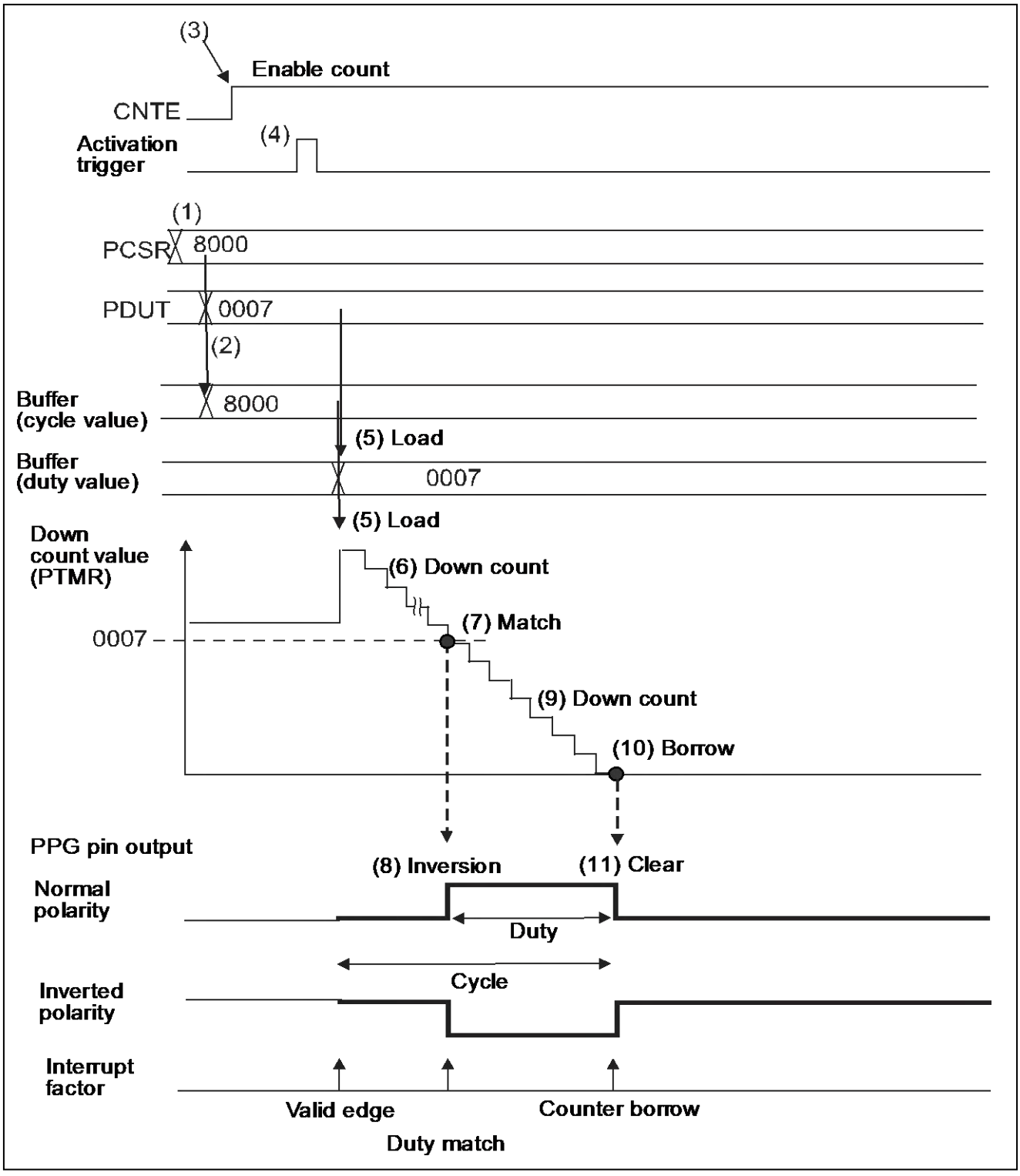
Calculation formulas:

Cycle = {Cycle value (PCSR) + 1} × Count clock
Duty = {Duty value (PDUT) + 1} × Count clock
Time to pulse output = [Cycle value (PCSR) - duty value (PDUT)] × Count clock

17.5.2 One-shot Operation

This section explains the one-shot operation of the PPG.

During the one-shot operation, one-shot pulses are output at the PPG pin.



(1) Writing cycle values
(2) Writing duty value and transferring cycle value to the buffer
(3) Enabling of PPG operation
(4) Activation trigger generation
(5) Loading cycle value and duty value
(6) Counter decrement
(7) The down counter matches the duty value
(8) Output level inversion at the PPG pin
(9) Counter decrement
(10) Counter borrow occurrence
(11) Clearing PPG pin output level (restoration to normal state)
(12) End of operation sequence
(See "17.9 Notes")

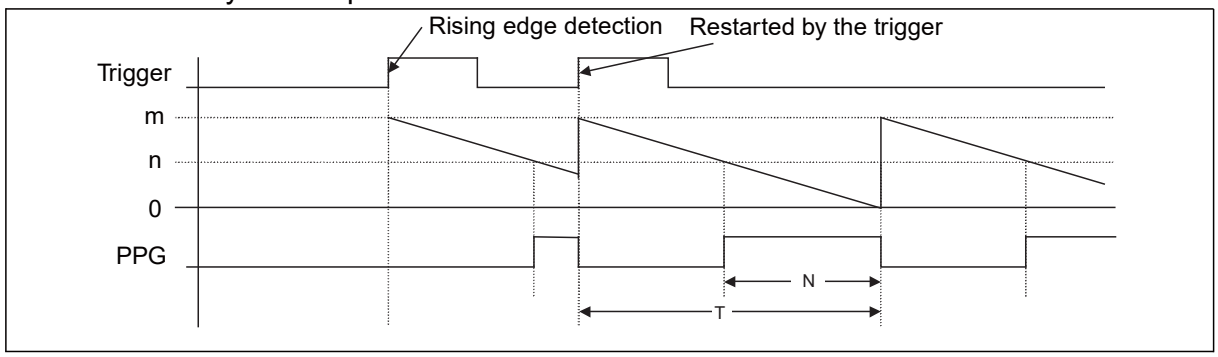
17.5.3 Restart Operation

This section explains the restart operation of the PPG.

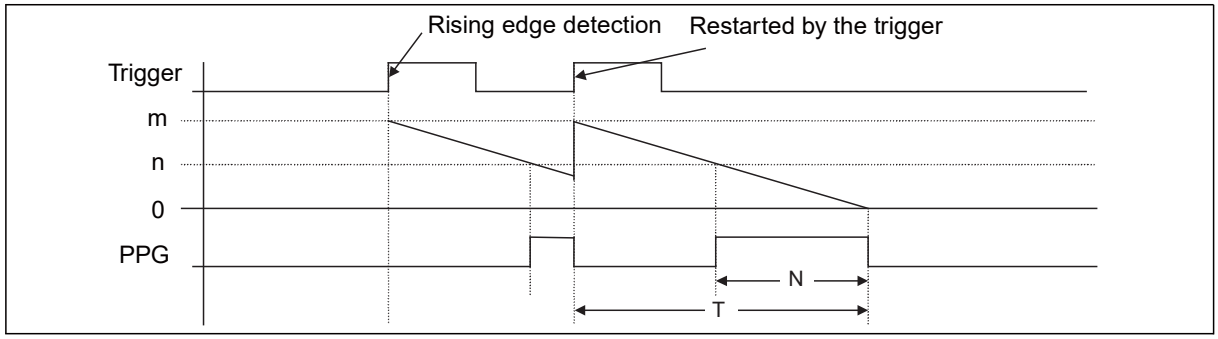
The restart operation is as follows:

* N = duty, T = cycle, m = cycle setting register (PCSR) value, n = duty setting register (PDUT) value

- Restarted by PWM operation



- Restarted by one-shot operation



When restart operation is disabled, second and latter triggers will be invalid for both the PWM operation and the one-shot operation.
(Triggers after the down counter is stopped will still be valid even if second and latter triggers occur.)

17.6 Setting

This section explains setting of the PPG.

Table 17-3. Settings Required for PPG Operation

Setting	Setting Register	Setting Method
Cycle and duty value setting	PPG cycle setting (PCSR0 to PCSR23) PPG duty setting (PDUT0 to PDUT23)	See 17.7.1
Enabling PPG operation	PPG control status (PCN0 to PCN23)	See 17.7.2
Operation mode selection (PWM/one-shot)		See 17.7.3
Restart enable		See 17.7.4
Count clock selection		See 17.7.5
PPG output mask selection		See 17.7.6
Trigger selection Software trigger External trigger Internal trigger (reload timer, GCN20/21/22/23/24/25:EN bit)	PPG control status (PCN0 to PCN23)	See 17.7.7
	Trigger input from TRG pin	
	General control 10/11/12/13/14/15 (GCN10/11/12/13/14/15)	
Output polarity selection	PPG control status (PCN0 to PCN23)	See 17.7.8
PPG pin output setting	Set the pins as peripheral output. For setting, see the "Chapter: I/O Ports".	
Trigger generation External trigger Software trigger	Trigger input from TRG pin	See 17.4.3
	PPG control status (PCN0 to PCN23)	
Reload timer	See "Chapter: Reload Timer".	
GCN20/21/22/23/24/25:EN bit	General control 20/21/22/23/24/25 (GCN20/21/22/23/24/25)	See 17.4.6

Table 17-4. Settings Required for Stopping PPG Operation

Setting	Setting Register	Setting Method
PPG stop bit setting	PPG control status (PCN0 to PCN23)	See 17.7.11

Table 17-5. Settings Required for Fixing Output level

Setting	Setting Register	Setting Method
Output polarity selection	PPG control status (PCN0 to PCN23)	See 17.7.8
PPG output mask selection		See 17.7.6
Setting cycle value = duty value	PPG duty setting (PDUT0 to PDUT23)	See 17.7.6

Table 17-6. Settings Required for PPG Interrupt

Setting	Setting Register	Setting Method
Setting for PPG interrupt vector and PPG interrupt level	See "Chapter: Interrupt Control (Interrupt Controller)".	See 17.7.12
PPG interrupt factor selection (Activation trigger generation, borrow generation, duty match)	PPG control status (PCN0 to PCN23)	See 17.7.13
PPG interrupt setting Interrupt request clear Interrupt request enable		See 17.7.14

17.7 Q&A

This section explains Q&A of the PPG.

17.7.1 How to Set (Rewrite) Cycle and Duty Values
17.7.2 How to Enable/Stop PPG Operation
17.7.3 How to Set PPG Operation Mode (PWM/One-shot)
17.7.4 How to Restart
17.7.5 Type and Selection of Count Clock
17.7.6 How to Fix the PPG Pin Output Level
17.7.7 Type and Selection of Activation Trigger
17.7.8 How to Reverse the Output Polarity
17.7.9 How to Change a Pin to a PPG Output Pin
17.7.10 How to Generate Activation Trigger
17.7.11 How to Stop PPG Operation
17.7.12 Interrupt-related Registers
17.7.13 Type and Selection of Interrupts
17.7.14 How to Enable/Disable/Clear Interrupt

17.7.1 How to Set (Rewrite) Cycle and Duty Values

This section explains how to set (rewrite) the cycle and duty values.

Cycle value setting and duty value setting

- Set the cycle value in the PPG cycle setting register PCSRn. (n = 0 to 23)
- Set the duty value in the PPG duty setting register PDUTn. (n = 0 to 23)
- As the PPG cycle setting register and PPG duty setting register have their own buffers, no timing consideration for writing is required.

Calculation formulas:

PCSR register value = {Cycle/Count clock} - 1
PDUT register value = {"H" width (duty)*/Count clock} - 1

*: Normal polarity (OSEL=0)

Available setting range

PCSR register value = PDUT register value to $FFFF_H$ (65535)
PDUT register value = 0 to PCSR register value

Note:

Be sure to set the duty value after the cycle is set. (See "17.9 Notes")

17.7.2 How to Enable/Stop PPG Operation

This section explains how to enable/stop the PPG operation.

Enabling PPG operation

Use the PPG operation enable bit (PCNn:CNTE). (n = 0 to 23)

Control	PPG Operation Enable Bit (CNTE)
How to stop PPG operation	Set to "0"
How to enable PPG operation	Set to "1"

Activate the PPG after the PPG operation is enabled.
(See "17.9 Notes")

17.7.3 How to Set PPG Operation Mode (PWM/One-shot)

This section explains how to set the PPG operation mode (PWM/one-shot).

Use the mode selection bit (PCNn:MDSE) for selecting an operation mode. (n = 0 to 23)

Operating Mode	Mode Selection Bit (MDSE)
How to set to PWM operation	Set to "0"
How to set to one-shot operation	Set to "1"

(See "17.9 Notes")

17.7.4 How to Restart

This section explains how to restart the PPG.

Restart enable

PPG restart can be enabled while the PPG is running.

Use the restart enable bit (PCNn:RTRG) for setting.(n = 0 to 23)
(See "17.9 Notes")

17.7.5 Type and Selection of Count Clock

This section explains the type and selection of the count clock.

Count clock selection

The count clock can be selected from the following four types in the table below:

Use the count clock selection bit (PCNn:CKS[1:0]). (n = 0 to 23)

Count Clock	Count clock Selection Bit		Example) Peripheral Clock (PCLK) = 16 MHz	
	CKS1	CKS0	Count clock	Cycle (1 to $FFFF_H$)
PCLK	0	0	16MHz	125.0 ns to 4.096 ms
PCLK/4	0	1	4MHz	500 ns to 16.384 ms
PCLK/16	1	0	1MHz	2.0 µs to 65.536 ms
PCLK/64	1	1	250kHz	8.0 µs to 262.144 ms

(See "17.9 Notes")

17.7.6 How to Fix the PPG Pin Output Level

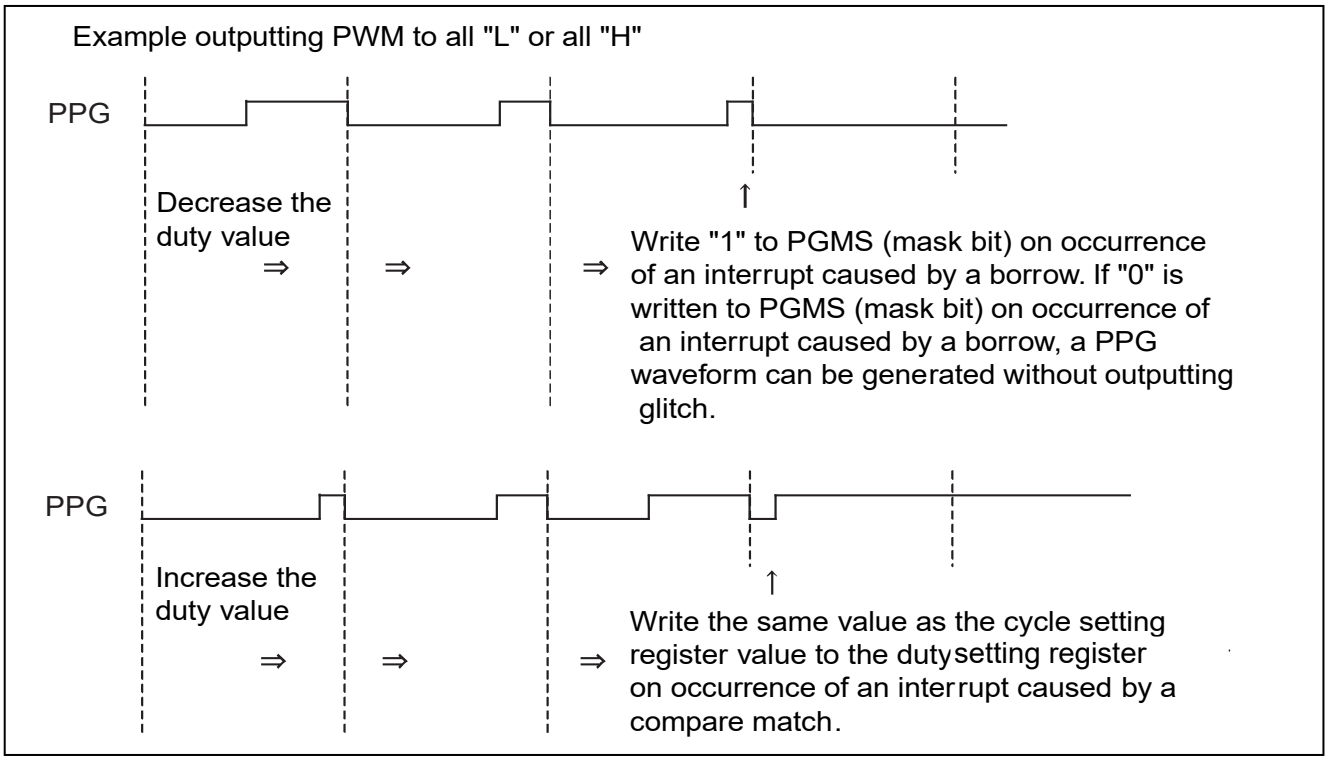
This section explains how to fix the PPG pin output level.

PPG output mask selection

The PPG pin output level can be fixed.

Use the PPG output mask selection bit (PCNn:PGMS) and duty value (PDUT) for setting.(n = 0 to 23)

PPG Pin Output	PPG Output Polarity Selection Bit (OSEL)	Setting Method
How to fix the level to "L" in normal polarity	OSEL is "0"	Set the PPG output mask selection bit (PGMS) to "1"
How to fix the level to "H" in normal polarity	OSEL is "0"	Set cycle value (PCSR) = duty value (PDUT)
How to fix the level to "H" in reversed polarity	OSEL is "1"	Set the PPG output mask selection bit (PGMS) to "1"
How to fix the level to "L" in reversed polarity	OSEL is "1"	Set cycle value (PCSR) = duty value (PDUT)



17.7.7 Type and Selection of Activation Trigger

This section explains the type and selection of the activation trigger.

Selecting Internal Trigger

- The software trigger is always valid.
- The internal trigger of PPG0 to PPG3 is the GCN20 register.
  The internal trigger of PPG4 to PPG7 is the GCN21 register.
  The internal trigger of PPG8 to PPG11 is the GCN22 register.
  The internal trigger of PPG12 to PPG15 is the GCN23 register.
  The internal trigger of PPG16 to PPG19 is the GCN24 register.
  The internal trigger of PPG20 to PPG23 is the GCN25 register.
- Use TSEL0/TSEL1/TSEL2/TSEL3 of the following general control registers for the settings for the internal triggers:

  GCN10 register (PPG0 to PPG3)
  GCN11 register (PPG4 to PPG7)
  GCN12 register (PPG8 to PPG11)
  GCN13 register (PPG12 to PPG15)
  GCN14 register (PPG16 to PPG19)
  GCN15 register (PPG20 to PPG23)

Settings for PPG0 to PPG3 are as follows:

Internal Trigger	Example for PPG0 (GCN10:TSEL0 [3:0] Setting Value)	Example for PPG1 (GCN10:TSEL1 [3:0] Setting Value)	Example for PPG2 (GCN10:TSEL2 [3:0] Setting Value)	Example for PPG3 (GCN10:TSEL3 [3:0] Setting Value)
How to select EN0 bit of the GCN20 register	Set to "0000"			
How to select EN1 bit of the GCN20 register	Set to "0001"			
How to select EN2 bit of the GCN20 register	Set to "0010"			
How to select EN3 bit of the GCN20 register	Set to "0011"			
How to select reload timer 0	Set to "0100"			
How to select reload timer 1	Set to "0101"			

Selecting External trigger

Use TSEL0/TSEL1/TSEL2/TSEL3 of the following general control registers for the settings for the external triggers:
GCN10 register (PPG0 to PPG3)
GCN11 register (PPG4 to PPG7)
GCN12 register (PPG8 to PPG11)
GCN13 register (PPG12 to PPG15)
GCN14 register (PPG16 to PPG19)
GCN15 register (PPG20 to PPG23)

Settings for PPG0 to PPG3 are as follows:

External Trigger	Example for PPG0 (GCN10:TSEL0 [3:0] Setting Value)	Example for PPG1 (GCN10:TSEL1 [3:0] Setting Value)	Example for PPG2 (GCN10:TSEL2 [3:0] Setting Value)	Example for PPG3 (GCN10:TSEL3 [3:0] Setting Value)
How to select external trigger (TRGn)	Set to any of following values. "1000", "1001", "1010", "1011"			

Specifying a same trigger to multiple PPGs will activate multiple PPGs simultaneously.
(See "17.9 Notes")

Selecting Internal/External Trigger Edge

Use trigger input edge selection bits (PCN0:EGS[1:0]) to (PCN23:EGS[1:0]) for internal/external trigger edge settings.

Selecting Internal Trigger Edge	Trigger Input Edge Selection Bits (EGS[1:0])
No trigger is detected (software trigger only)	Set to "00"
Trigger is generated at "L" → "H" (rising)	Set to "01"
Trigger is generated at "H" → "L" (falling)	Set to "10"
Trigger is generated at both edges	Set to "11"

(See "17.9 Notes")

17.7.8 How to Reverse the Output Polarity

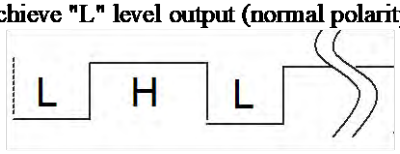
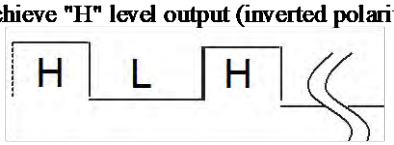
The section explains how to reverse the output polarity.

Output polarity selection

The polarity in the normal state can be specified as following table:

Use the PPG output polarity selection bit (PCNn:OSEL) for setting. (n = 0 to 23)

("Normal state" is a state which does not output pulses.)

Output Level In the Normal State	PPG Output Polarity Selection Bit (OSEL)
To achieve "L" level output (normal polarity) 	Set to "0"
To achieve "H" level output (inverted polarity) 	Set to "1"

17.7.9 How to Change a Pin to a PPG Output Pin

The section explains how to change a pin to a PPG output pin.

Set the pins as peripheral output. For setting, see "Chapter: I/O Ports".

17.7.10 How to Generate Activation Trigger

The section explains how to generate an activation trigger.

Trigger generation

The following is how to generate activation triggers.

How to activate software trigger

Use the software trigger bit (PCNn:STRG) for setting. (n = 0 to 23)

If "1" is written to the software trigger bit (STRG), the activation trigger will be generated.

This bit is always valid independent of the state of GCN10 register to GCN15 register.

How to activate with external trigger

See "17.7.7 Type and Selection of Activation Trigger". Set the pins TRG0, TRG1 and TRG2 as peripheral input. For setting, see "Chapter: I/O Ports". Then you will be able to generate the activation trigger by changing the input level for the pins TRG0, TRG1 and TRG2.

How to activate with reload timer 0/1

You need to set up and activate the reload timer. See "Chapter: Reload Timer" for details.

The activation trigger will be generated when underflow of the reload timer generated the specified edge in the reload timer output signal.

How to activate with EN trigger input bits (GCN20/21/22/23/24/25:EN[0:3])

The activation trigger will be generated by rewriting the level of the EN trigger input bits (GCN20/21/22/23/24/25:EN[0:3]) with software.

Edge	Software Setting (EN0, EN1, EN2, EN3)
Rising edge	First set the EN bit to "0", then set the EN bit to "1".
Falling edge	First set the EN bit to "1", then set the EN bit to "0".

How to activate multiple PPGs simultaneously

Multiple PPGs will be activated on trigger by specifying the same trigger (trigger input bit) from the PPG trigger specification bits.

Note:

The PPG will not be activated on the activation trigger before the PPG operation is enabled. Be sure to enable the PPG operation before generating the activation trigger. (See "17.7.2 How to Enable/Stop PPG Operation")

17.7.11 How to Stop PPG Operation

This section explains how to stop the PPG operation.

Set the PPG stop bit. (See "17.7.2 How to Enable/Stop PPG Operation".)

17.7.12 Interrupt-related Registers

This section explains the interrupt-related registers.

Setting for PPG interrupt vector and PPG interrupt level

The PPG number, interrupt level and interrupt vector are as follows:

For information on the interrupt level and interrupt vector, see "Chapter: Interrupt Control (Interrupt Controller)".

	Interrupt Vector (Default)	Interrupt Level Setting Register (ICR[4:0])
PPG0	#40 Address: $0FFF5C_H$	Interrupt level register (ICR24) Address: 00458_H
PPG1		
PPG10		
PPG11		
PPG20		
PPG21		
PPG2	#41 Address: $0FFF58_H$	Interrupt level register (ICR25) Address: 00459_H
PPG3		
PPG12		
PPG13		
PPG22		
PPG23		
PPG4	#42 Address: $0FFF54_H$	Interrupt level register (ICR26) Address: $0045A_H$
PPG5		
PPG14		
PPG15		
PPG6	# 43 Address: $0FFF50_H$	Interrupt level register (ICR27) Address: $0045B_H$
PPG7		
PPG16		
PPG17		

	Interrupt Vector (Default)	Interrupt Level Setting Register (ICR[4:0])
PPG8	# 44 Address: $0FFF4C_H$	Interrupt level register (ICR28) Address: $0045C_H$
PPG9		
PPG18		
PPG19		

Clear the interrupt request flags (PCNn:IRQF) by software before the recovery from the interrupt process as the flags will not be cleared automatically. (Write "0" to the IRQF bit) (n = 0 to 23)

17.7.13 Type and Selection of Interrupts

This section explains the type and selection of interrupts.

Selecting interrupt factor

The interrupt factor can be selected from following four factors:

Use interrupt factor setting bits (PCNn:IRS[1:0]) for setting. (n = 0 to 23)

Interrupt Factor	Interrupt Factor Setting Bits (IRS[1:0])
Software trigger or internal trigger	Set to "00"
Down counter borrow (match with the specified cycle)	Set to "01"
Duty match	Set to "10"
Down counter borrow (match with the specified cycle) or duty match	Set to "11"

17.7.14 How to Enable/Disable/Clear Interrupt

This section explains how to enable/disable/clear interrupt.

Interrupt request enable flag and interrupt request flag

Use the interrupt request enable bit (PCNn:IREN) for enabling interrupts. (n = 0 to 23)

Operation	Interrupt Request Enable Bit (IREN)
How to disable interrupt request	Set to "0"
How to enable interrupt request	Set to "1"

Use the interrupt request bit (PCNn:IRQF) for clearing interrupt requests. (n = 0 to 23)

Operation	Interrupt Request Bit (IRQF)
How to clear interrupt request	Write "0"

(See "17.9 Notes")

17.8 Sample Programs

This section explains sample programs of the PPG.

Setting procedure example 1

PWM output from PPG4, Software trigger (duty1/4), Normal polarity

Initial setting (PPG4)

↓

Activation (PPG4)

< Initial setting>

- Port — Register name.Bit name

PPG output setting for ports	See "Chapter: I/O Port".

- PPG4 control — Register name.Bit name

Control register setting	PCN4
Timer operation enable≫	.CNTE
Software trigger (unprocessed)≫	.STRG
Operation mode selection≫	.MDSE
Restart disable≫	.RTRG
Clock source selection≫	.CKS1-0
Output mask selection≫	.PGMS
Edge selection≫	.EGS1-0
Interrupt disable≫	.IREN
Interrupt flag clear≫	.IRQF
	.IRS1-0
Output polarity selection≫	.OSEL

- Cycle setting — Register name.Bit name

Cycle setting for PPG4	PCSR4

- Duty setting — Register name.Bit name

Duty setting for PPG4	PDUT4

<Activation>

- PPG4 activation — Register name.Bit name

PPG4 activation	PCN4.STRG

<Others>

(Note)

You need settings for clock and "_set_il"(numerical value) in advance. See "Chapter: Clock" and "Chapter: Interrupt Control (Interrupt Controller)" for details.

Program example 1

```
void PPG_sample_1(void)
{
        PPG4_initial();
        PPG4_start();
}


void PPG4_initial(void)
{
        PORT_SETTING_PPG4_OUT();   /* Set the PPG4 pins as peripheral input. */



        IO_PCN4.hword = 0x8000;     /* Setting value = 1000_0000_0000_0000 */
                                    /* bit15 = 1     CNTE timer enable */
                                    /* bit14 = 0     STRG  Software trigger */
                                    /* bit13 = 0     MDSE PWM  operation */
                                    /* bit12 = 0     RTRG  restart disable */
                                    /* bit11 to 10 = 00 CKS1,0 */
                                    /* bit9 = 0      PGMS PPG  output mask */
                                    /* bit8 = 0      Undefined bit */
                                    /* bit7 to 6 = 00   EGS1,0 Edge selection: disabled */
                                    /* bit5 = 0      IREN interrupt request enable */
                                    /* bit4 = 0      IRQF interrupt request flag */
                                    * bit3 to 2 = 00   IRS1,0 Interrupt factor: software trigger */
                                    /* bit1 = 0      Undefined bit */
                                    /* bit0 = 0      OSEL normal polarity */

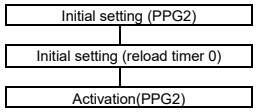

        IO_PCSR4 = 0x0909;        /* PPG cycle setting */

        IO_PDUT4 = 0x0242;        /* PPG duty ratio (1/4) setting */
}

void PPG4_start(void)
{
        IO_PCN4.bit.STRG = 1;     /* bit14 = 1 STRG Software trigger */
}
```

Setting procedure example 2

PPG one-shot output from PPG2, Reload timer ch.0 (duty1/2), Normal polarity



<Initial setting>

- Port — Register name.Bit name

PPG output setting for ports	See “Chapter: I/O Port”.

- PPG2 control — Register name.Bit name

Control register setting	PCN2
Timer operation enable≫	.CNTE
Software trigger (unprocessed)≫	.STRG
Operation mode selection≫	.MDSE
Restart disable≫	.RTRG
Clock source selection≫	.CKS1-0
Output mask selection≫	.PGMS
Edge selection≫	.EGS1-0
Interrupt disable>>	.IREN
Interrupt flag clear>>	.IRQF
	.IRS1-0
Output polarity selection>>	OSEL

- Cycle setting — Register name.Bit name

Cycle setting for PPG2	PCSR2

- Duty setting — Register name.Bit name

Duty setting for PPG2	PDUT2

- Trigger selection — Register name.Bit name

PPG2 trigger selection	GCN10.TSEL2

<Initial setting (reload timer 0)>

- Control for reload timer 0 — Register name.Bit name

Control register setting	TMCSR0
Mode selection≫	.MOD
Internal clock selection≫	.TRGM,CSL
Trigger selection≫	.TRGM
Output level selection>>	.OUTL
Reload enable≫	.RELD
Interrupt disable>>	.INTE
Interrupt flag clear>>	.UF
Count enable≫	.CNTE
Software trigger (unprocessed)≫	.TRG

- Count value

Count value setting	TMRLRA0

<Activation>

- Trigger will be input to the PPG2 by activation of the reload timer 0 — Register name.Bit name

Software trigger generation	TMCSR0.TRG

<Others>

(Note)

You need settings for clock and “_set_il”(numerical value) in advance. See “Chapter: Clock” and “Chapter: Interrupt Control (Interrupt Controller)” for details.

Program example 2

```
void PPG_sample_2(void)
{
        PPG2_initial();
        RTIM0_initial();
        RTIM0_start ();
}



void PPG2_initial(void)
{
        PORT_SETTING_PPG2_OUT();   /* Set the PPG2 pins as peripheral input */



        IO_PCN2.hword = 0x8040;     /* Setting value = 1000_0000_0100_0000 */
                                    /* bit15 = 1     CNTE timer enable */
                                    /* bit14 = 0     STRG Software trigger */
                                    /* bit13 = 0     MDSE PWM operation */
                                    /* bit12 = 0     RTRG restart disable */
                                    /* bit11 to 10 = 00 CKS1,0 */
                                    /* bit9 = 0      PGMS PPG output mask */
                                    /* bit8 = 0      Undefined bit */
                                    /* bit7 to 6 = 01   EGS1,0 edge selection: rising edge */
                                    /* bit5 = 0      IREN interrupt request enable */
                                    /* bit4 = 0      IRQF interrupt request flag */
                                    /* bit3 to 2 = 00   IRS1,0 Interrupt factor: software trigger */
                                    /* bit1 = 0      Undefined bit */
                                    /* bit0 = 0      OSEL normal polarity */



            IO_PCSR2 = 0x0909;      /* PPG cycle setting */

            IO_PDUT2 = 0x0484;      /* PPG duty ratio (1/2) setting */

        IO_GCN10.bit.TSEL2 = 4;    /* bit11 to 8 = 0100 TSEL23 to 20 Reload timer ch.0 */
}

void RTIM0_initial(void)
{
        IO_TMCSR0.hword = 0x0012; /* Setting value = 0000_0000_0001_0010 */
                                  /* bit15 to 14 = 00   MOD=00 Single mode */
                                  /* bit13 to 12 = 00   TRGM=00 No external trigger detection/
                                                        Software trigger */
                                  /* bit11 to 9 = 000   CSL=000 Count source selection
                                                        (peripheral clock/2) */
                                  /* bit8 to 6 = 000    GATE=0, EF=0 */
                                  /* bit5 = 0 OUTL=0 External output level */
                                  /* bit4 = 1 RELD=1 Reload enable */
                                  /* bit3 = 0 INTE=0 Interrupt request disabled */
                                  /* bit2 = 0 UF=0   Flag clear */
                                  /* bit1 = 1 CNTE=1 Timer operation enable/activation trigger wait */
                                  /* bit0 = 0 TRG=0  Trigger is still disabled */

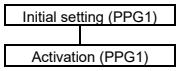


        IO_TMRLRA0 = 0xffff;     /* Initial value for counting */
}

void rtim0_start(void)
{
        IO_TMCSR0 = IO_TMCSR0 | 0x0001;  /* bit0 = 1 TRG software trigger */
}
```

Setting procedure example 3

PPG one-shot output from PPG1, High output, activation trigger (GCN20:EN1)



<Initial setting>

-Port Register name.Bit name

PPG output setting for ports	See “Chapter: I/O Port”.

- PPG1 control Register name.Bit name

Control register setting	PCN1
Timer operation enable≫	.CNTE
Software trigger (unprocessed)≫	.STRG
Operation mode selection≫	.MDSE
Restart disable≫	.RTRG
Clock source selection≫	.CKS1-0
Output mask selection≫	.PGMS
Edge selection≫	.EGS1-0
Interrupt disable>>	.IREN
Interrupt flag clear>>	.IRQF
	.IRS1-0
Output polarity selection>>	. SEL

-Cycle setting Register name.Bit name

Cycle setting for PPG1	PCSR1

-Duty setting Register name.Bit name

Duty setting for PPG1	PDUT1

-Trigger selection Register name.Bit name

PPG1 trigger selection	GCN10.TSEL1

-Trigger signal level Register name.Bit name

Trigger level = "L"	GCN20.EN1

<Activation>

-PPG1 activation Register name.Bit name

PPG1 activation	PCN4.STRG

Trigger signal level Register name.Bit name

Trigger level = "H"	GCN20.EN1

<Others>

(Note)

You need settings for clock and “_set_il”(numerical value) in advance. See “Chapter: Clock” and “Chapter: Interrupt Control (Interrupt Controller)” for details.

Program example 3

```
void PPG_sample_3(void)
{
        PPG1_initial();
        PPG1_start ();
}


void PPG1_initial(void)
{
        PORT_SETTING_PPG1_OUT();   /* Set the PPG1 pins as peripheral input */

        IO_PCN1.hword = 0xA040;     /* Setting value = 1010_0000_0100_0000 */
                                    /* bit15 = 1     CNTE timer enable */
                                    /* bit14 = 0     STRG Software trigger */
                                    /* bit13 = 1     MDSE One-shot operation */
                                    /* bit12 = 0     RTRG restart disable */
                                    /* bit11-10 = 00 CKS1,0 */
                                    * bit9 = 0      PGMS PPG output mask */
                                    /* bit8 = 0      Undefined bit */
                                    /* bit7-6 = 01   EGS1,0 edge selection: rising edge */
                                    /* bit5 = 0      IREN interrupt request enable */
                                    /* bit4 = 0      IRQF interrupt request flag */
                                    /* bit3-2 = 00   IRS1,0 Interrupt factor: software trigger */
                                    /* bit1 = 0      Undefined bit */
                                    /* bit0 = 0      OSEL normal polarity */


        IO_PCSR1 = 0x0909;        /* PPG cycle setting */

        IO_PDUT1 = 0x0484;        /* PPG duty ratio (1/2) setting */


        IO_GCN10.bit.TSEL1 = 1;    /* bit3-0 = 0001 TSEL03 to 00 EN1 bit of GCN20 */

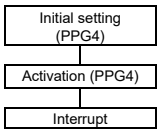
        IO_GCN20 = 0x00;          /* bit1 = 0 EN1 bit of GNC20 */
}

void PPG1_start(void)
{
        IO_PCN4.bit.STRG = 1;      /* bit14 = 1 STRG Software trigger */

        IO_GCN20 = 0x02;          /* bit1 = 1 EN1 bit of GNC20 */
}
```

Setting procedure example 4

Interval interrupt
PPG output from PPG4, Software trigger (duty1/4), Normal polarity



<Initial setting>

-Port — Register name.Bit name

PPG output setting for ports	See “Chapter: I/O Port”.

-PPG1 control — Register name.Bit name

Control register setting	PCN4
Timer operation enable≫	.CNTE
Software trigger (unprocessed)≫	.STRG
Operation mode selection≫	.MDSE
Restart disable≫	.RTRG
Clock source selection≫	.CKS1-0
Output mask selection≫	.PGMS
Edge selection≫	.EGS1-0
Interrupt disable>>	.IREN
Interrupt flag clear>>	.IRQF
	.IRS1-0
Output polarity selection>>	.OSEL

-Cycle setting — Register name.Bit name

Cycle setting for PPG4	PCSR4

-Duty setting — Register name.Bit name

Duty setting for PPG4	PDUT4

-Interrupt setting — Register name.Bit name

PPG4 interrupt level setting	ICR26
Setting for I flag	(CCR)

<Activation>

-PPG4 activation — Register name.Bit name

Interrupt enable	PCN4.IREN
PPG4 activation	PCN4.STRG

<Interrupt>

-Interrupt process — Register name.Bit name

(Given process)	
Interrupt request flag clear	PCN4.IRQF

<Interrupt vector>
Vector table setting

<Others>
(Note)
You need settings for clock and “_set_il”(numerical value) in advance. See “Chapter: Clock” and “Chapter: Interrupt Control (Interrupt Controller)” for details.

Program example 4

```
void PPG_sample_4(void)
{
        PPG4_initial();
        PPG4_start ();
}




void PPG4_initial(void)
{
        PORT_SETTING_PPG4_OUT();   /* Set the PPG4 pins as peripheral input. */

        IO_PCN4.hword = 0x8004;      /* Setting value = 1000_0000_0000_0100 */
                                     /* bit15 = 1     CNTE timer enable */
                                     /* bit14 = 0     STRG Software trigger */
                                     /* bit13 = 1     MDSE one-shot operation */
                                     /* bit12 = 0     RTRG restart disable */
                                     /* bit11 to 10 = 00 CKS1,0 */
                                     /* bit9 = 0      PGMS PPG output mask */
                                     /* bit8 = 0      Undefined bit */
                                     /* bit7 to 6 = 01   EGS1,0 edge selection: rising edge */
                                     /* bit5 = 0      IREN interrupt request enable */
                                     /* bit4 = 0      IRQF interrupt request flag */
                                     /* bit3 to 2 = 01   IRS1,0  Interrupt factor: cycle match */
                                     /* bit1 = 0      Undefined bit */
                                     /* bit0 = 0      OSEL normal polarity */


        IO_PCSR4 = 0x0909;        /* PPG cycle setting */

        IO_PDUT4 = 0x0242;        /* PPG duty ratio (1/4) setting */


        IO_ICR[26].byte = 0x10;    /* Interrupt level (given value) */
        __EI();                    /* Interrupt enable */
}

void PPG4_start(void)
{
        IO_PCN4.bit.IREN = 1; /* bit5 = 1 IREN interrupt request enable */
        IO_PCN4.bit.STRG = 1; /* bit14 = 1 STRG Software trigger */
}

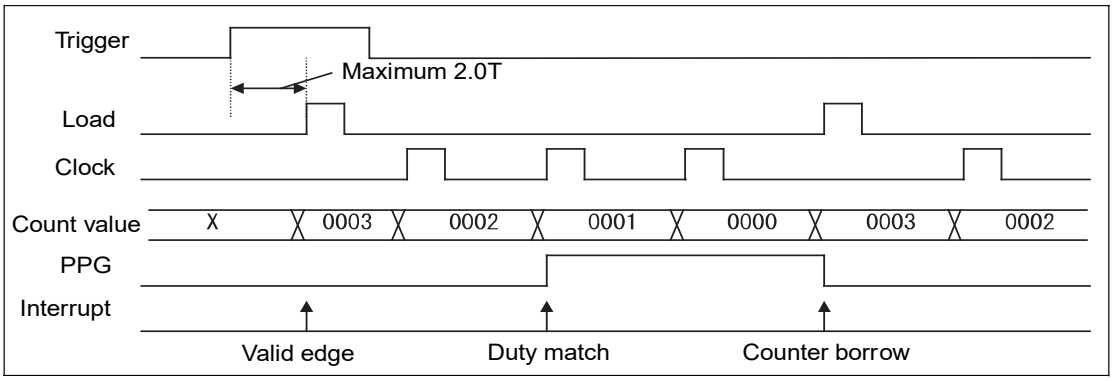
__interrupt void PPG4_int(void)
{
                                   /* Given process */
        IO_PCN4.bit.IRQF = 0; /* bit14 = 0 IRQF interrupt request flag */
}

Interrupt routine must be specified with the vector table.
#pragma intvect PPG4_int 42
```

17.9 Notes

Notes on the use of the PPG are shown in this section.

- If the timing when the interrupt request flag (PCNn:IRQF) becomes "1" and the timing to become "0" are duplicated, the operation for setting the interrupt request flag to "1" will be prioritized and the request for clearing the flag will be invalid. (n = 0 to 23)
- If the load timing and counting timing of the down counter are duplicated, the load operation will be prioritized.



- The time from the activation trigger to finish loading the counter value requires up to 2.0 T (T: peripheral clock).
- Be sure to set the duty values (PDUTn) after the cycle value is set if you make initial setting and rewriting of the cycle value PCSRn.
  (Be sure to write the values in the order of (1) PCSRn, (2) PDUTn.)
  In addition, only PDUT can be rewritten for rewriting the duty value only. (n = 0 to 23)
- When you set the duty values (PDUTn), use values smaller than the cycle values (PCSRn). When larger values are set, rewrite the duty values to the smaller ones after the PPG operation is disabled. (n = 0 to 23)
- The PPG cycle setting register PCSRn and PPG duty setting register PDUTn must be accessed in half-word (16-bit). Both upper value and lower value will not be written if the access is made in byte. (n = 0 to 23)
- To activate the PPG, the timer operation enable bit (PCNn:CNTE) must be set to "1" to enable the PPG operation before the activation or simultaneously. (n = 0 to 23)
- Do not change the configuration of the mode (MDSE), restart enable (RTRG), count clock (CKS[1:0]), trigger input edge (EGS[1:0]), interrupt factor (IRS[1:0]), internal trigger (TSEL), and output polarity selection (OSEL), while the PPG is in operation.
  If you changed the value while the PPG is in operation, first disable the PPG operation, and then retry register setting.
- When you write values to the GCN20/21/22/23/24/25, the undefined part of upper 4 bits must always be written to "0". If you have written "1" instead of "0", first stop the PPG operation, and then rewrite them.
- When values other than specified values (1100 to 1111) are set to the activation trigger selection bits (TSEL3, TSEL2, TSEL1, TSEL0) of the GCN10/11/12/13/14/15, the operation will be returned to the normal operation if you first disable the PPG operation, then write the specified values.
- When the timer operation enable bit (PCNn:CNTE) is set to "0" to disable the PPG operation while the PPGn is in operation, the PPG will be stopped with retaining count value, and clearing output level to "L" when PCNn:OSEL=0. Moreover, when the timer operation enable bit (PCNn:CNTE) is set to "1" to enable the PPG operation and a trigger is generated, cycle setting register (PCSRn) and duty setting register (PDUTn) are reloaded, and then PPG will be restarted (n = 0 to 23).

- As writing to the bits 11 and 10 (count clock selection bits CKS1 and CKS0) of the PPG control register will immediately be reflected just after the writing, setting change must be performed with the counting stopped.

- If Cycle is set equal to Duty during One-shot waveform mode, width of output H pulse is "(Cycle register value + 2) x PCLK".
- When the timer operation enable bit (PCNn: CNTE) is cleared to "0" to disable the PPG operation, it takes 3 cycles of internal clock until PPG stops the output.
- When PGMS is cleared from "1" to "0" after a matching, the pulse with specified duty is outputted in the next period (PPG cycle).
- Be sure to write the values (1) PCSRn and (2) PDUTn in that order when writing a cycle value (PCSR) and a duty value (PDUT). Notes on writing cycle values (PCSR) and duty values (PDUT) are shown below:
  1. Cycle value (PCSR) and duty value (PDUT) will be sent to buffer when writing the duty value (PDUT), and will be transferred from the buffer to the counter when an activation trigger is generated or a borrow occurs.
  2. When cycle value (PCSR) or duty value (PDUT) is rewritten in PPG operation, the writing will get reflected in the output waveform in the next cycle after the rewriting of the duty value (PDUT).
  3. No matter if only cycle value (PCSR) needs to be written, it is necessary to reset duty value (PDUT) with the same value; write the values (1) PCSR and (2) PDUT in that order.
  4. You can arbitrarily rewrite duty value (PDUT).
- While PPG output is masked (PCN.PGMS="1"), the interrupt request flag is controlled under the following conditions:
  - ☐ The interrupt request flag will never be set to "1" regardless of interrupt factors caused by duty much.
  - ☐ The interrupt request flag will be set to "1" because of borrow occurrence on the counter.
- The interrupt request flag will be set to "1" because of such interrupt factors as software trigger, external trigger, or GATE signal trigger.

18. Watchdog Timer

This chapter explains the watchdog timer.

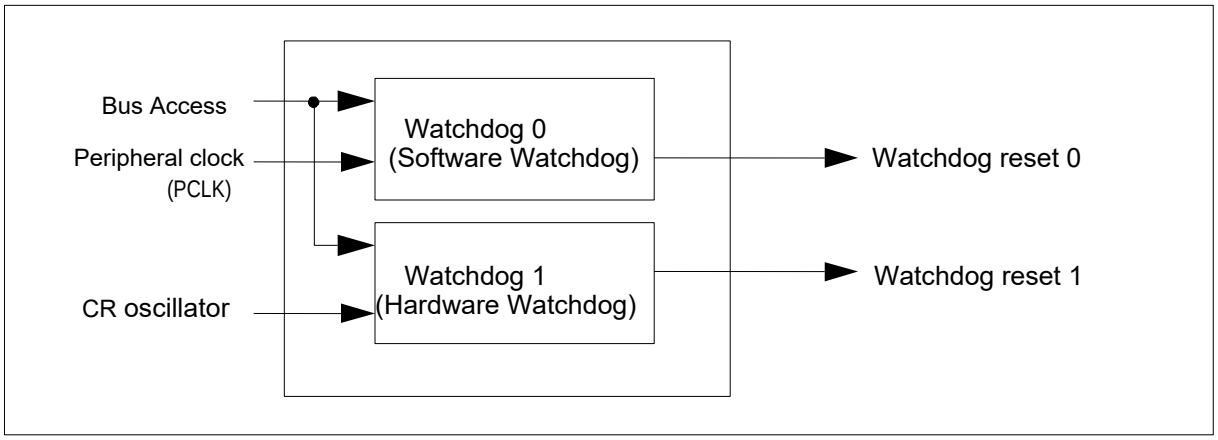
18.1 Overview
18.2 Features
18.3 Configuration
18.4 Registers
18.5 Operation
18.6 Usage Example

18.1 Overview

This section explains the overview of the watchdog timer.

This series has two watchdog timers that can detect software and hardware running out of control, and generate reset requests.

Figure 18-1 . Block Diagram (Overview)



18.2 Features

This section explains features of the watchdog timer.

Watchdog Timer 0 (Software Watchdog)

- 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 sixteen choices of the peripheral clock (PCLK) × (2^9 to 2^{24}) cycles.

- Count Stop Conditions

  The count stops while the CPU is stopped.

Watchdog Timer 1 (Hardware Watchdog)

This timer is driven by the clock generated by the built-in CR oscillator circuit immediately after the reset is released. For information on CR oscillator settings (calibration), see "Chapter: RTC/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 × 2^{15} cycles.

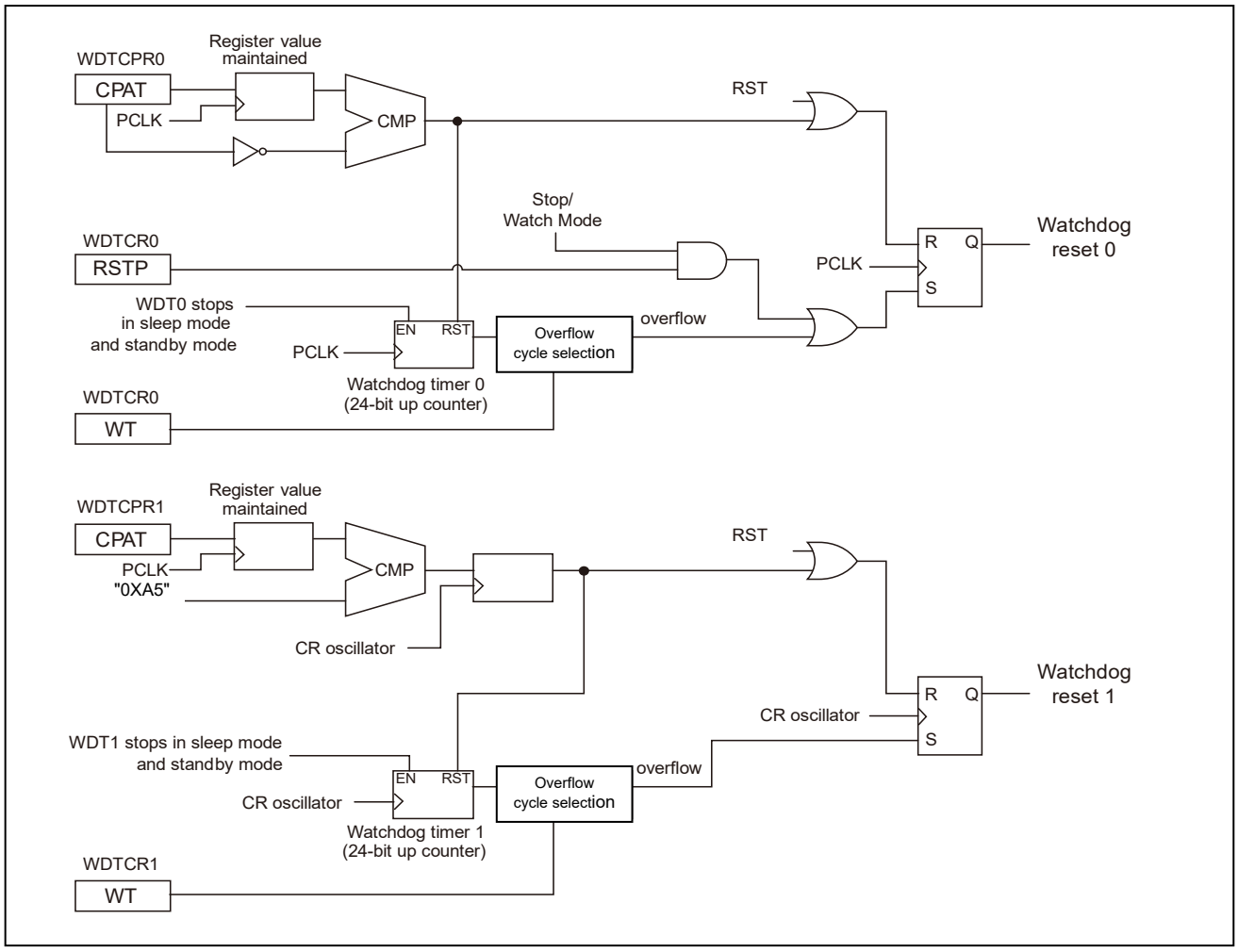
- Count Stop Conditions

  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.

18.3 Configuration

This section shows the configuration of the watchdog timer.

Figure 18-2. Block Diagram (Detailed)



18.4 Registers

This section explains the registers of the watchdog timer.

Table 18-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
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

18.4.1 Watchdog Control Register 0: WDTCR0 (Watchdog Timer Configuration Register 0)

The bit configuration of the watchdog 0 control register (WDTCR0) is explained.

This register configures each of the settings of watchdog timer 0.

Writing to this register is ignored after watchdog timer 0 activates.

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

"0" is always written to this bit. The reading value is "0".

[bit6] RSTP (Reset by Stop): Stop mode detection reset enable

This bit configures whether a reset is generated when a transition to watch mode or stop mode is detected while 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, 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 when detected

Writing to this bit is ignored after watchdog timer 0 activates.

[bit5, bit4] Reserved

"0" is always written to these bits. The reading value is "0".

[bit3 to bit0] WT[3:0] (Watchdog Timer interval): Watchdog timer cycle selection

These bits configure the number of cycles from when watchdog timer 0 was last cleared until watchdog reset 0 is issued as follows.

WT[3:0]	Watchdog Timer 0 Cycle
0000	PCLK (Peripheral Clock) × 2^9 cycles
0001	PCLK × 2^{10} cycles
0010	PCLK × 2^{11} cycles
0011	PCLK × 2^{12} cycles
0100	PCLK × 2^{13} cycles
0101	PCLK × 2^{14} cycles
0110	PCLK × 2^{15} cycles
0111	PCLK × 2^{16} cycles
1000	PCLK × 2^{17} cycles
1001	PCLK × 2^{18} cycles
1010	PCLK × 2^{19} cycles
1011	PCLK × 2^{20} cycles
1100	PCLK × 2^{21} cycles
1101	PCLK × 2^{22} cycles
1110	PCLK × 2^{23} cycles
1111	PCLK × 2^{24} cycles

Writing to these bits are ignored after watchdog timer 0 activates.

Watchdog timer 0 is not counted during periods where the CPU is not operating.

Counting is performed while the CPU is operating even if DMA transfers are being performed.

18.4.2 Watchdog Timer 0 Clear Register: WDTCPR0 (Watchdog Timer Clear Pattern Register 0)

The bit configuration of the watchdog timer 0 clear register (WDTCPR0) is explained.

This register activates or clears (delays the issue of a reset) 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

Watchdog timer 0 activates 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 previous value written. 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.

18.4.3 Watchdog Timer 1 Cycle information Register: WDTCR1 (Watchdog Timer Cycle Information Register 1)

The bit configuration of the watchdog timer 1 cycle information register (WDTCR1) is explained.

This register configures each of the settings of 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 rewritten.

[bit7 to bit4] Reserved

"0" is always read. Writing to it has no effect on operation.

[bit3 to bit0] WT[3:0] (Watchdog Timer interval) : Watchdog timer cycle selection

These bits configure the number of cycles from when watchdog timer 1 was last cleared until watchdog reset 1 is issued. The cycle is fixed to 2^{15} cycles. Writing to these bits are ignored.

WT[3:0]	Watchdog Timer 1 Cycle
0110	CR oscillator × 2^{15} cycles (initial value, fixed)

18.4.4 Watchdog Timer 1 Clear Register: WDTCPR1 (Watchdog Timer Clear Pattern Register 1)

The bit configuration of the watchdog timer 1 clear register (WDTCPR1) is explained.

This register clears watchdog timer 1 (delays the issue of a reset).

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.

18.5 Operation

This section explains the operation of the watchdog timer.

This section explains the watchdog timer function.

Software Watchdog Function

- Setup

  Before activating watchdog timer 0, set bits 3 to 0: WT[3:0] of the WDTCR0 register to select the period from clearing the watchdog timer until the reset is issued.

  Because watchdog timer 0 is only counted when the CPU is operating, set the period based on the number of program steps and the clock division setting.

  Before activating watchdog timer 0, set bit6: RSTP of the WDTCR0 register to select whether or not to generate a reset 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 is generated as soon as the CPU enters watch mode or stop mode.

  If you are using watch mode or stop mode, set RSTP=0. Writing to the RSTP bits is ignored after watchdog timer 0 activates.

- Starting

  Watchdog timer 0 starts by the first write of any data to the WDTCPR0 register after reset.

  It does not matter what the write data is.

  The WDTCPR0 register always reads out "0x00" regardless of the data written.

- Operation

  This section explains the operation of watchdog timer 0 after it has activated.

  Counting Conditions

  Watchdog timer 0 counts the rising edges of the peripheral clock (PCLK) while the CPU is operating.

  DMA transfers do not affect the operation of the count.

  The count only stops while the CPU is stopped, such as in sleep mode. Sampling of the CPU operating state is performed on the peripheral clock (PCLK), with changes in the operating state within the peripheral clock cycle ignored.

  The count is stopped in emulator mode when the ICE is connected. The count is also stopped if the watchdog reset suppression function is enabled in the debug interface functions while the ICE is connected.

  In all of the above circumstances, because the counter is not cleared but is only paused when the count is stopped, when the count resumes the count continues from the counter value prior to the stop.

  Because the peripheral clock is stopped during the oscillation stabilization wait time of the source clock, the watchdog timer count also stops.

Clearing the Timer

Once the watchdog timer has activated, the timer must be cleared before the timer period has elapsed.

Clearing the watchdog timer is performed by writing data to WDTCPR0. The write data must be the value with all bits inverted of the data previously written to WDTCPR0.

When watchdog timer 0 is activated, for example, if it is activated by writing "0x55" to WDTCPR0, the timer is cleared subsequently by alternatively writing "0xAA" then"0x55" then "0xAA" then "0x55".

Because the read value of WDTCPR0 is always "0x00", the value written previously cannot be determined by reading WDTCPR0. Storing the previously written value in a different location can be avoided by performing two consecutive writes when performing a single clear.

Reset Request Generation

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 inverse value of the previous written value is written to the clear register.

Hardware Watchdog Function

- Setup

  Bits 3 to bit0: WT[3:0] of the WDTCR1 register of watchdog timer 1 is fixed in hardware.

- Activating

  Watchdog timer 1 activates immediately after the reset is released.

- Operation

  This section explains the operation of watchdog timer 1 after it has activated.

  Counting Conditions

  Watchdog timer 1 counts the rising edges of the CR oscillator.

  The count is stopped in emulator mode when the ICE is connected. The count is also stopped if the watchdog reset suppression function is enabled in the debug interface functions while the ICE is connected.

  The count stops during sleep mode, watch mode, stop mode, and when waiting for the oscillator to stabilize when recovering from standby mode.

  Clearing the Timer

  Once the watchdog timer has activated, the timer must be cleared before the timer period has elapsed.

  Watchdog timer 1 is cleared by writing "0xA5" to WDTCPR1.

  Reset Request Generation

  Watchdog timer 1 generates a watchdog reset request under the following conditions.

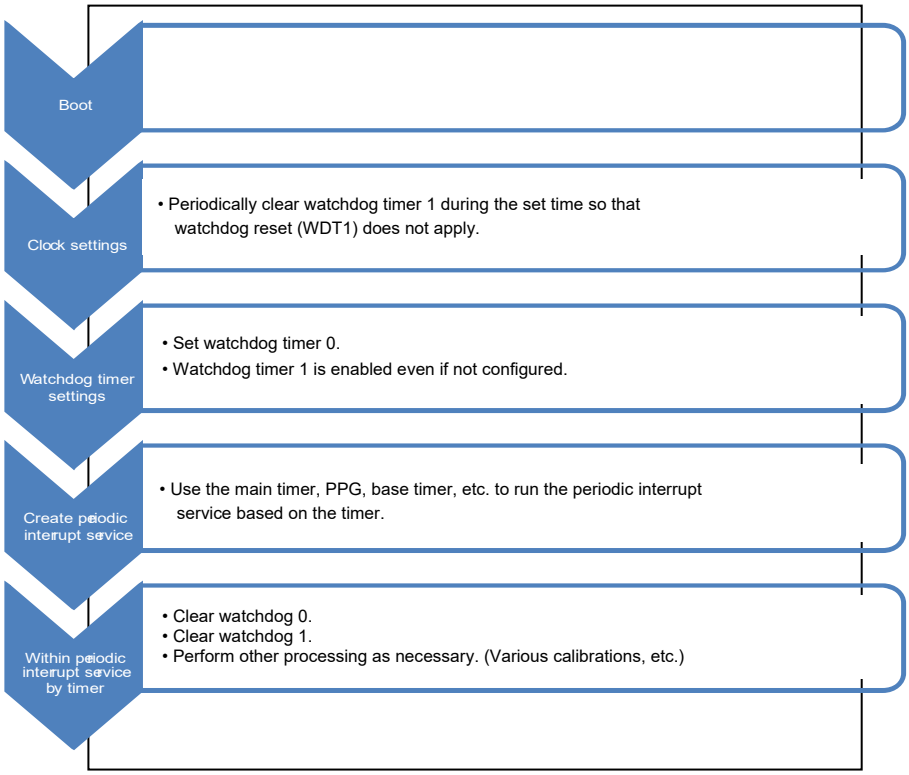
  - ☐ An overflow of the watchdog timer cycle occurs
  - ☐ A value other than "0xA5" is written to WDTCPR1

18.6 Usage Example

This section shows a usage example of the watchdog timer.

This example is provided for clearing the watchdog timer.

Figure 18-3. Example of Clearing the Watchdog Timers



19. Base Timer

This chapter explains the base timer.

19.1 Overview
19.2 Features
19.3 Configuration
19.4 Registers
19.5 Operation

19.1 Overview

This section explains the overview of the base timer.

This series includes the base timer for 2 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

19.2 Features

This section explains features of the base timer.

This series includes the base timer for 2 channels. Each channel selects and uses appropriate ones of the following functions:

19.2.1 16-/32-bit Reload Timer

19.2.2 16-bit PWM Timer

19.2.3 16-/32-bit PWC Timer

19.2.4 16-bit PPG Timer

19.2.1 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 two channels into one 32-bit reload timer.

- Operation Mode

  You can select one of the following two:

  - 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, or 256.
  - 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

19.2.2 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 Timer, 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, or 256.
  - 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

19.2.3 16-/32-bit PWC Timer

This section explains the overview of 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 two channels into one 32-bit PWC timer.

- Operation Mode

  You can select one of the following two:

  - ☐ 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, and 256.

- 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:
    1. Period from the detection of a rising edge to the detection of the falling edge
    2. 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

19.2.4 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 Timer, 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:

  - ☐ 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, or 256.
  - ☐ 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

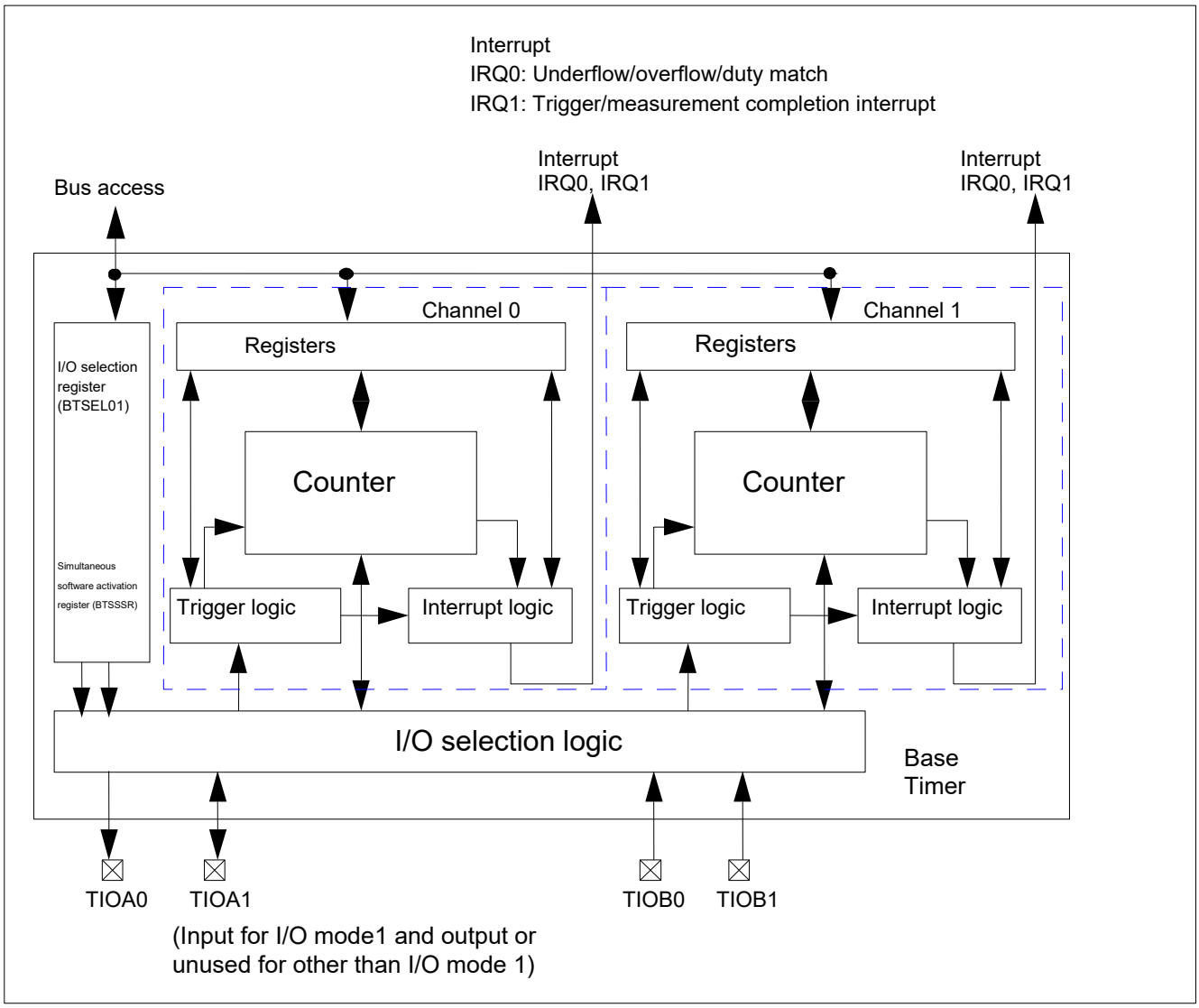
  An interrupt request can be generated in one of the following events:

  - ☐ 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.

19.3 Configuration

This section explains the configuration of the base timer.

Figure 19-1. Block Diagram (Overview)



19.4 Registers

This section explains registers of the base timer.

List of Base Addresses (Base_addr) and External Pins

Table 19-1. Table of Base Addresses (Base_addr) and External Pins

Channel Number	Base_addr	External Pin
0	0x0080	TIOA0, TIOA1, TIOB0, and TIOB1 are assigned based on the BTSEL01 register setting.
1	0x0090	

Registers Map

Table 19-2. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0080	[Common] BT0TMR		[Common] BT0TMCR		[Common] Timer register 0 [Common] Control register 0
0x0084	Reserved	[Reload timer] BT0STC [PWM] BT0STC [PPG] BT0STC [PWC] BT0STC	Reserved		[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				

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0090	[Common] BT1TMR		[Common] BT1TMCR		[Common] Timer register 1 [Common] Control register 1
0x0094	Reserved	[Reload timer] BT1STC [PWM] BT1STC [PPG] BT1STC [PWC] BT1STC	Reserved		[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

19.4.1 Common Registers

This section explains the common registers of the base timer.

The registers described here are common to various operations.

19.4.1.1 Timer Registers 0, 1: BTxTMR (Base Timer 0/1 Timer Register)

The bit configuration of timer registers 0, 1 (BTxTMR) is shown below.

These registers are used to 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 Description.

Note:

These registers must be accessed in 16-bit mode.

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

19.4.1.2 Timer Control Registers 0, 1: BTxTMCR (Base Timer 0/1 Timer Control Register)

The bit configuration of timer control registers 0, 1 (BTxTMCR) is shown below.

These registers are used to variously configure and stop the base timer and to issue software triggers.

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).

BTxTMCR: Address Base_addr + 02_H (Access: Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	CKS[2:0]			[PWM - PPG] RTGEN [Others] Reserved	[PWM - PPG] PMSK [PWC] EGS[2] [Others] Reserved	EGS[1:0]	
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W0 R0, WX[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-numbered channel
[3]:Attribute assumed for a 32-bit timer serving an odd-numbered channel or for a 16-/32-bit PWC timer

[bit15] Reserved

Always write "0" to this bit.

[bit14 to bit12] CKS[2:0] (Clock Select): Count clock selection bits

Select a count clock.

CKS[2:0]	Description	
	Clock Source	Description
000	Internal clock (Peripheral clock (PCLK))	1 division
001		4 division
010		16 division
011		128 division
100		256 division
101	[Reload timer/PWM/PPG] external clock (ECK signal) [PWC] Setting is prohibited	Rising edge
110		Falling edge
111		Both edges

In the PWC mode, settings of 101, 110, and 111 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	Does not reactivate
1	Reactivates

[PWM/PPG] [bit10] PMSK (Pulse Mask): Pulse output mask bit

Select 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

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

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)
101 110 111	Setting is prohibited

[Reload timer/PWC] [bit7] T32 (Timer 32bit): 32-bit timer selection bit

Select whether to run the 16-/32-bit timer individually by each channel or use the two channels as 32-bit timer through a cascade connection. Set this bit for both channel 0 and channel 1.

T32 (channel 0)	T32 (channel 1)	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 are used to 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
101 110 111	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

Enables/disables the counter operation.

CTEN	Description	
	Read	Write
0	Stops the operation	To be 0
1	Enables the operation	To be 1

[bit0] STRG (Software Trigger): Software trigger bit

Functions as a trigger for timer activation, etc.

In the PWC mode the read value is "0"; this bit should be cleared to "0".

Notes:

- When writing to this bit, be careful not to clear other bits.
- When writing to CTEN and FMD[2:0] simultaneously, issue a trigger as soon as the operation is enabled.

STRG	Description
0	Ignores.
1	Issues a trigger.

19.4.1.3 I/O Selection Register: BTSEL01 (Base Timer Select register ch.0 and ch.1)

The bit configuration of the I/O selection register (BTSEL01) is shown below.

These bits are used to set the I/O mode of ch.0 and ch.1 for the base timer.

Notes:

- These registers must be accessed in 8-bit mode.
- Rewrite these registers after selecting base timer reset mode (FMD2 to FMD0 = 000) by setting the base timer x timer control registers (BTxTMCR).

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 are used to set the I/O mode of ch.0 and ch.1 for the base timer.

SEL01[3:0]	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

19.4.1.4 Simultaneous Software Activation Register: BTSSSR (Base Timer Software Synchronous Start Register)

The bit configuration of the simultaneous software activation register (BTSSSR) 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 009E$_H$ (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						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 19-3.

SSSR0/1	Description
0	No effect on the operation.
1	"1" pulse to the timer input, then the corresponding channel is activated.

19.4.2 Registers for 16-/32-bit Reload Timer

This section explains registers for 16-/32-bit reload timer.

19.4.2.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

19.4.2.2 Cycle Setting Registers 0, 1: BTxPCSR (Base Timer 0/1 Pulse Counter Start Register)

19.4.2.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

The bit configuration of status control registers 0, 1 (BTxSTC) is shown below.

These registers control interrupt requests.

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).

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

[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	Disables
1	Enables

[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

19.4.2.2 Cycle Setting Registers 0, 1: BTxPCSR (Base Timer 0/1 Pulse Counter Start Register)

The bit configuration of cycle setting registers 0, 1 (BTxPCSR) 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.

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).

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

[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 two channels of a 16-bit reload timer are cascaded and it is used as the 32-bit reload timer.

- Value of even-number channel cycle setting register (BTxPCSR): Value of lower 16-bit
- Value of the 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:

- Odd-number channel base timer x cycle setting register (BTxPCSR)
- Even-number channel base timer x cycle setting register (BTxPCSR)

19.4.3 Registers for 16-bit PWM Timer

This section explains registers for 16-bit PWM timer.

19.4.3.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

19.4.3.2 Cycle Setting Registers 0, 1: BTxPCSR (Base Timer 0/1 Pulse Counter Start Register)

19.4.3.3 Duty Setting Registers 0, 1: BTxPDUT (Base Timer 0/1 Pulse Duty register)

19.4.3.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

The bit configuration of status control registers 0, 1 (BTxSTC) is shown below.

These registers control interrupt requests.

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).

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

[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	Disables.
1	Enables.

[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.

19.4.3.2 Cycle Setting Registers 0, 1: BTxPCSR (Base Timer 0/1 Pulse Counter Start Register)

The bit configuration of cycle setting registers 0, 1 (BTxPCSR) 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.

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).

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

[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

19.4.3.3 Duty Setting Registers 0, 1: BTxPDUT (Base Timer 0/1 Pulse Duty register)

The bit configuration of duty setting registers 0, 1 (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.

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).

BTxPDUT: Address Base_addr + $0A_H$ (Access: Half-word)

	bit15	bit14	- - -	bit2	bt1	bit0
	D[15:0]					
Initial value	0	0	- - -	0	0	0
Attribute	R/W	R/W	- - -	R/W	R/W	R/W

[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

19.4.4 Registers for 16-bit PPG Timer

This section explains registers for 16-bit PPG timer.

19.4.4.1Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

19.4.4.2L Width Setting Registers 0, 1: BTxPRLL (Base Timer 0/1 Pulse Length of "L" register)

19.4.4.3H Width Setting Registers 0, 1: BTxPRLH (Base Timer 0/1 Pulse Length of "H" register)

19.4.4.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

The bit configuration of status control registers 0, 1 (BTxSTC) is shown below.

These registers control interrupt requests.

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).

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

[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.

19.4.4.2 L Width Setting Registers 0, 1: BTxPRLL (Base Timer 0/1 Pulse Length of "L" register)

The bit configuration of L width setting registers 0, 1 (BTxPRLL) is shown below.

These registers set the default level for the signal output from the 16-bit PPG timer.

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).

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

[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 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.

19.4.4.3 H Width Setting Registers 0, 1: BTxPRLH (Base Timer 0/1 Pulse Length of "H" register)

The bit configuration of H width setting registers 0, 1 (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.

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).

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

[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" in "19.5.6.3 Operation in Reload Mode".

19.4.5 16-/32-bit PWC Timer Register

This section explains registers for 16-/32-bit PWC timer.

19.4.5.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

19.4.5.2 Data Buffer Registers 0, 1: BTxDTBF (Base Timer 0/1 Data Buffer Register)

19.4.5.1 Status Control Registers 0, 1: BTxSTC (Base Timer 0/1 Status Control)

The bit configuration of status control registers 0, 1 (BTxSTC) is shown below.

These registers control interrupt requests.

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).

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

[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	Measurement completion/overflow has not been occurred.	(EDIR) No effect on the operation. (OVIR) This bit is cleared.
1	Measurement completion/overflow has been occurred.	No effect on the operation.

19.4.5.2 Data Buffer Registers 0, 1: BTxDTBF (Base Timer 0/1 Data Buffer Register)

The bit configuration of data buffer registers 0, 1 (BTxDTBF) 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 two channels of a 16-bit PWC timer are cascaded and it is 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 these registers value in the following order.

- Even-channel data buffer register (BTxDTBF)
- Odd-channel data buffer register (BTxDTBF)

19.5 Operation

This section explains the operation of the base timer.

19.5.1 Selection of Timer Function
19.5.2 I/O Allocation
19.5.3 32-bit Mode Operation
19.5.4 16-/32-bit Reload Timer Operation
19.5.5 16-bit PWM Timer Operation
19.5.6 16-bit PPG Timer Operation
19.5.7 16-/32-bit PWC Timer Operation

19.5.1 Selection of Timer Function

This section explains selection of the timer function.

Select the timer function for BTxTMCR:FMD[2:0].

19.5.2 I/O Allocation

This section explains I/O allocation.

Set I/O of the base timer for the BTSEL01 register before using the timer. You can select one of the following seven:

- 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 two channels of base timer at the same time in this mode. Using this mode allows simultaneous activation of two 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[1] of the output signal from the even-number channel and stops with the falling edge[1].

- I/O Mode 5

  Simultaneous software activation mode

  More than one channel is 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[1] of the output signal from the even-number channel and stops with the falling edge[1].

- 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[1] of the output signal from the even-number channel.

  [1]: Make a setting using the trigger input selection bit (BTxTMCR:EGS).

Figure 19-2. Wiring Diagram of Each I/O Mode (1)

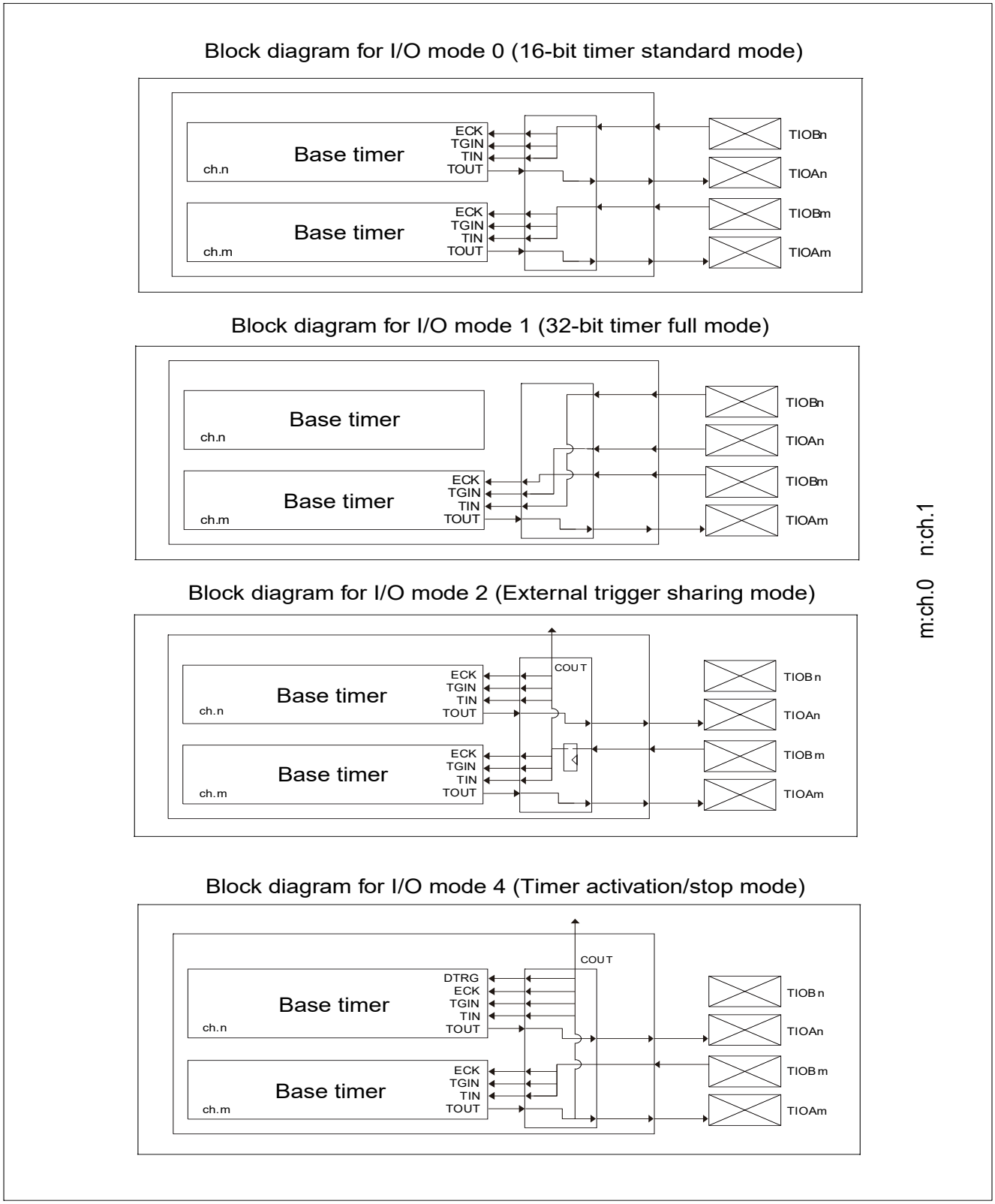
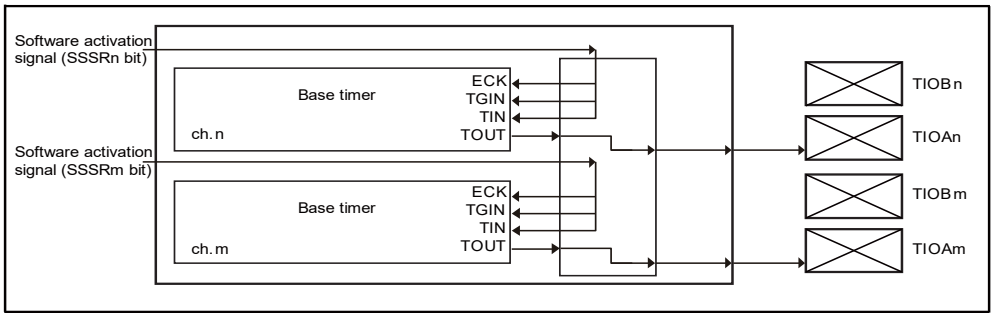
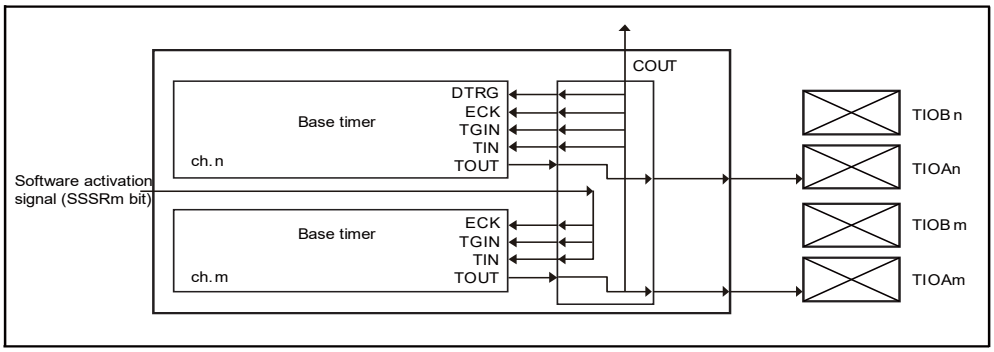


Figure 19-3. Wiring Diagram of Each I/O Mode (2)

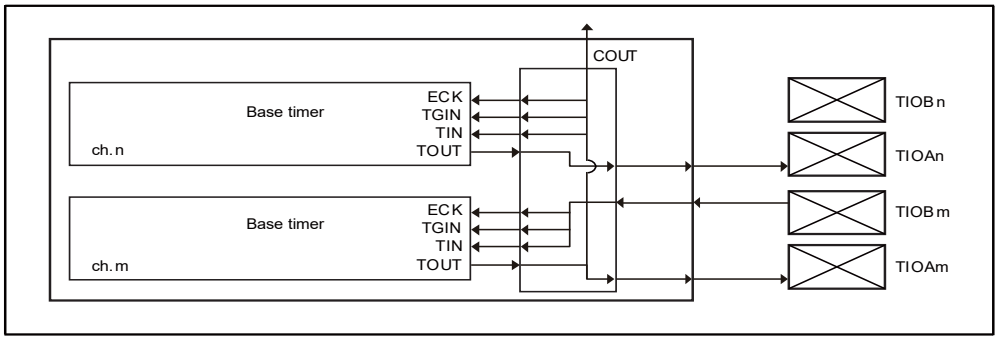
Block diagram for I/O mode 5 (simultaneous software activation mode)



Block diagram for I/O mode 6 (software activation timer activation/stop mode)



Block diagram for I/O mode 7 (timer activation mode)



m:ch.0 n:ch.1

19.5.3 32-bit Mode Operation

This section explains the 32-bit mode operation.

The reload timer and PWC timer can be operated in the 32-bit mode using two channels. The basic function/operation in the 32-bit mode is shown below.

19.5.3.1 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 two 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.

19.5.3.2 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 to the reset mode for both the even-number and odd-number channels, and make a setting in the 16-bit mode for each channel.

19.5.3.3 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 19-12 and Figure 19-28.

19.5.4 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 19-4. Block Diagram (16-bit Reload Timer Operation)

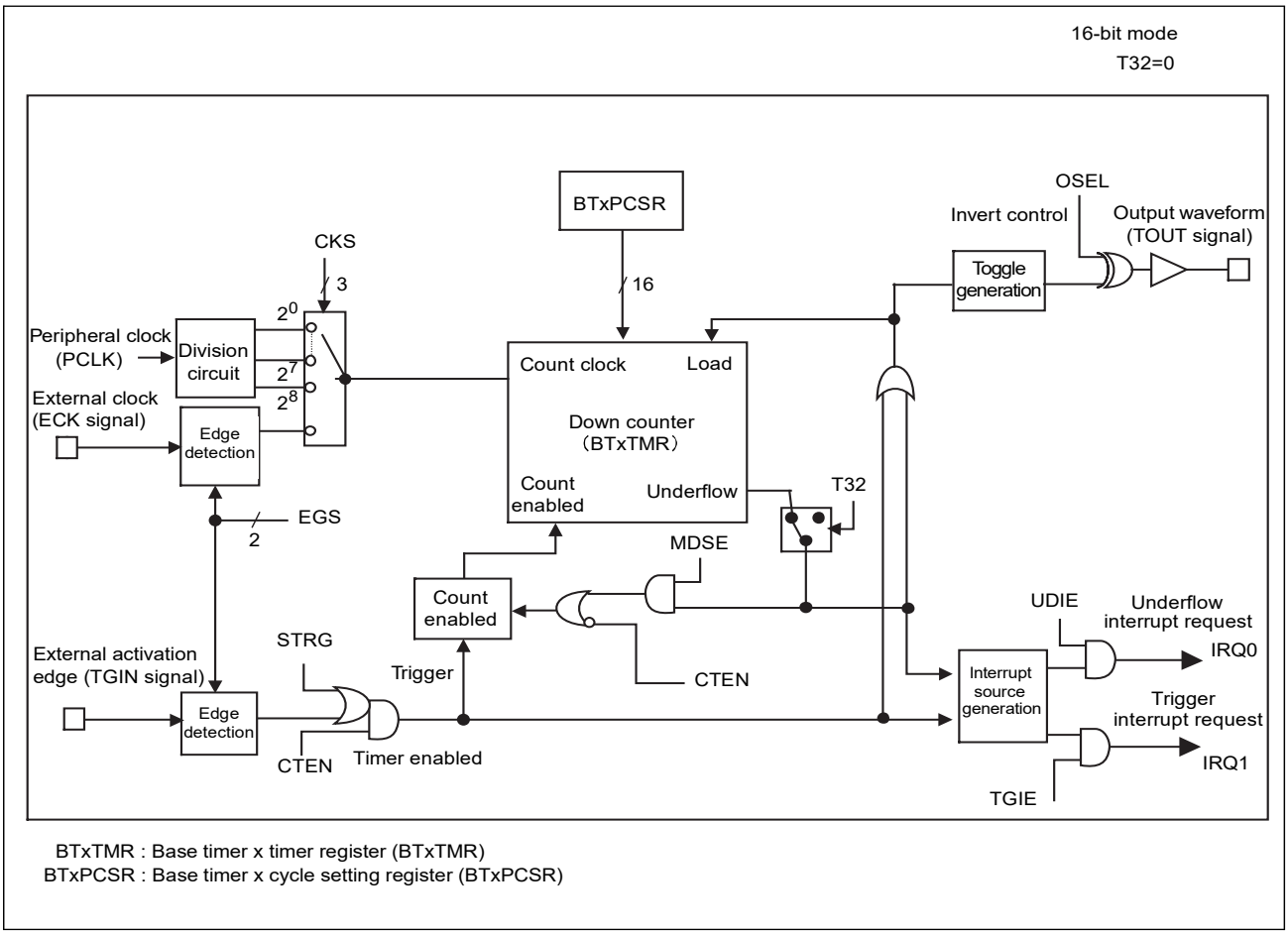
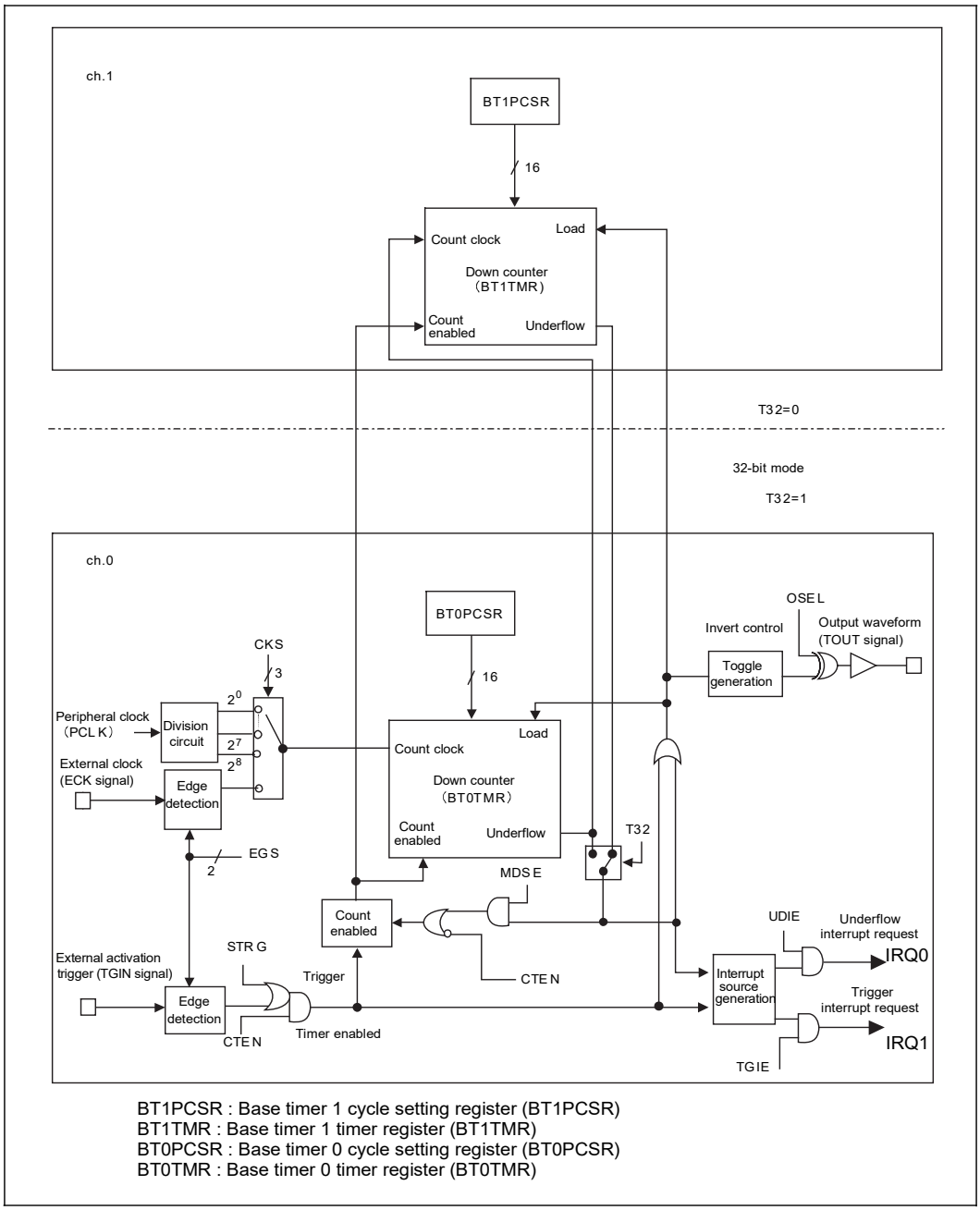


Figure 19-5. Block Diagram (32-bit Reload Timer Operation)



BT1PCSR : Base timer 1 cycle setting register (BT1PCSR)
BT1TMR : Base timer 1 timer register (BT1TMR)
BT0PCSR : Base timer 0 cycle setting register (BT0PCSR)
BT0TMR : Base timer 0 timer register (BT0TMR)

19.5.4.1 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.

19.5.4.2 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 activation trigger.
  2. Enter an activation trigger by one of the following methods:
     a. Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
     b. 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). See "19.5.2 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 19-6 and Figure 19-8 show the count start timing.

Figure 19-6. Count Start Timing (Software Trigger)

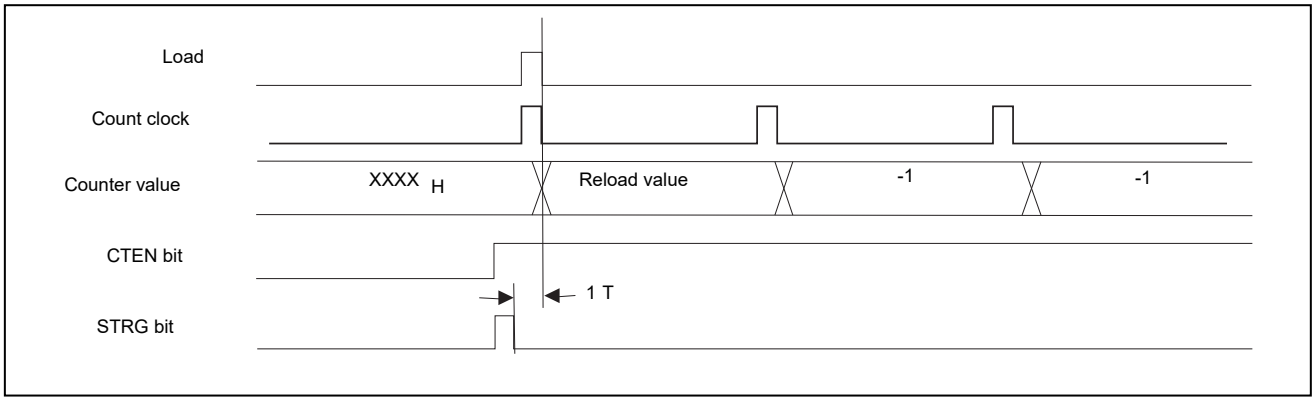
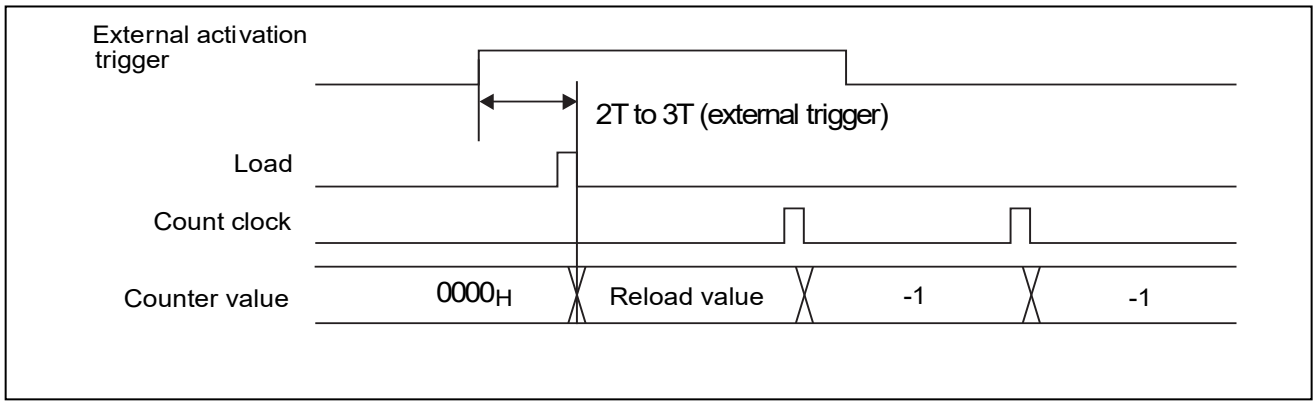


Figure 19-7. Count Start Timing (External Activation Trigger (TGIN Signal), Effective Edge = Rising Edge)

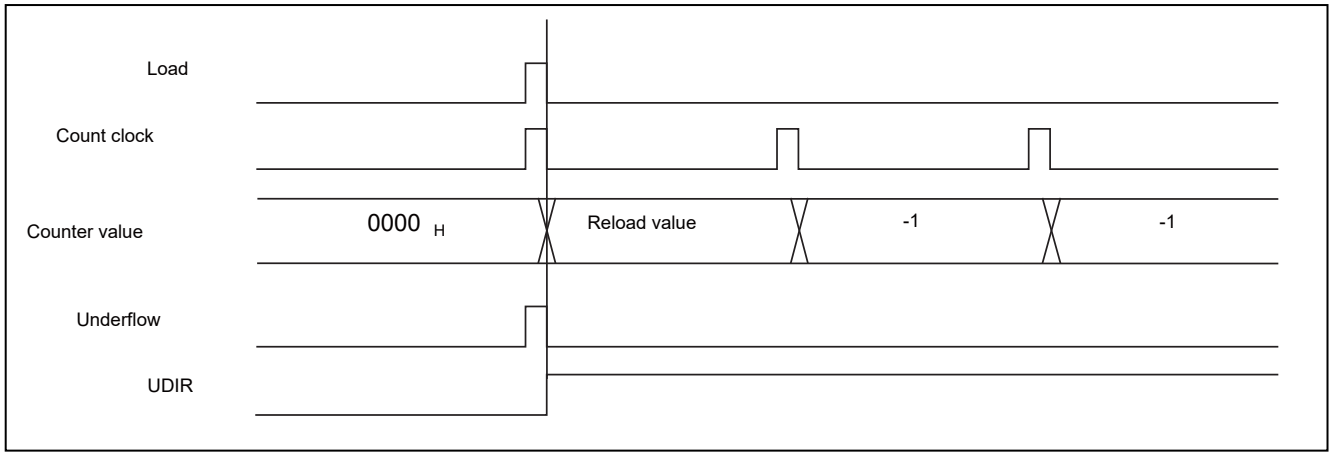


Note:

The external activation trigger (TGIN signal) entry method varies depending on the I/O mode specified by the I/O selection register (BTSEL01). See "19.5.2 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, theUDIR 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". Figure 5-7 shows the operation in case of an underflow.

Figure 19-8. Operation in Case of an Underflow



Output Waveform

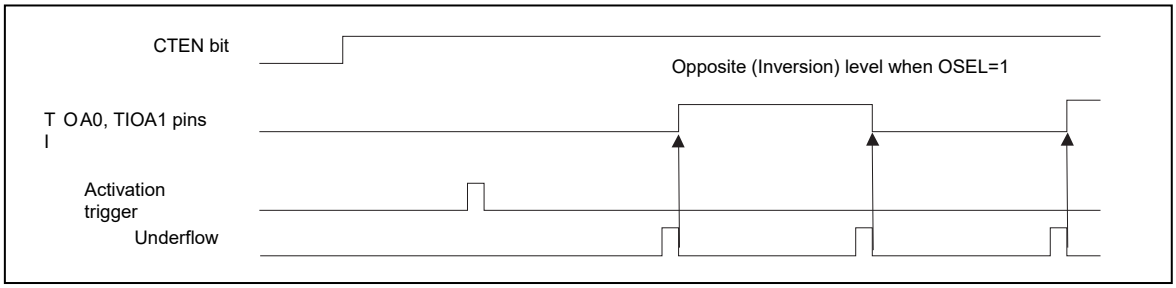
The waveform (TOUT signal) of the 16-/32-bit reload timer can be output. The waveform (TOUT signal) to be output varies according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR).

Table 19-3. 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.

Figure 19-9 shows the output waveform in reload mode.

Figure 19-9. Output Waveform in Reload Mode (Normal Polarity)



19.5.4.3 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

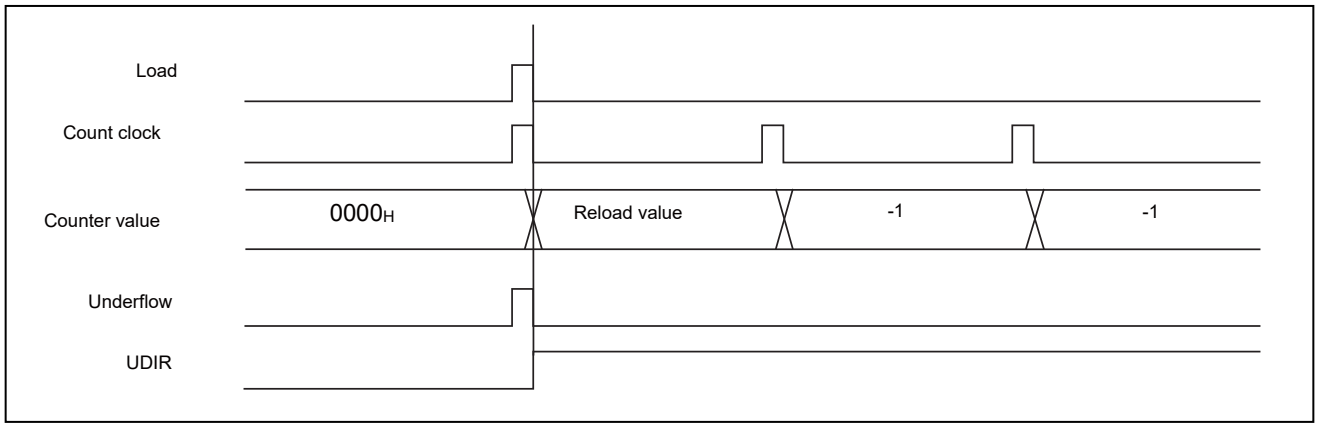
  The same operation as in reload mode. See "Overview" in "19.5.4.2 Operation in Reload Mode".

- Counting Operation

  The operation is the same as in reload mode until an underflow occurs. See "Overview". 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".

  Figure 19-10 shows the operation in case of an underflow.

Figure 19-10. 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).

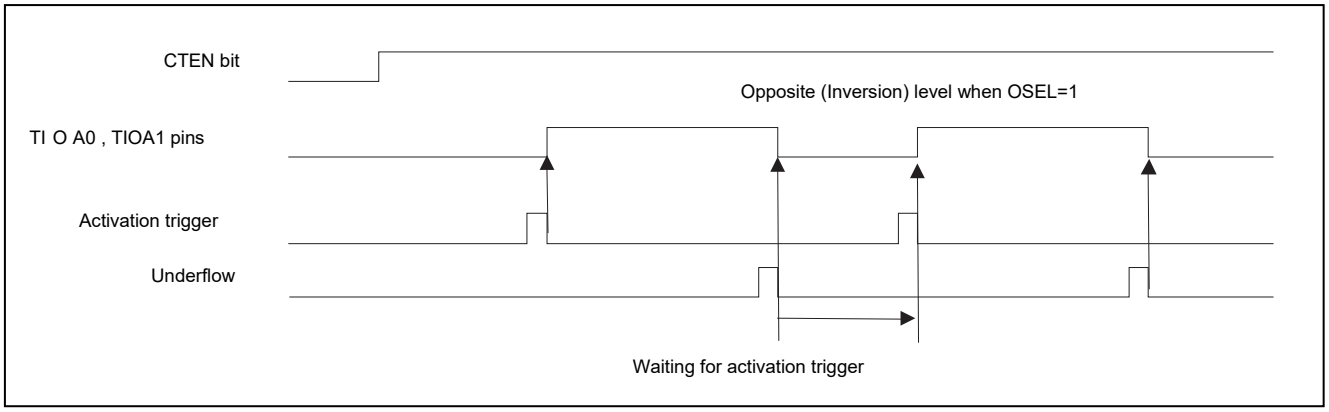
Table 19-4 shows the correspondence between output polarities and output waveforms.

Table 19-4. 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.

Figure 19-11 shows the output waveform in one-shot mode.

Figure 19-11. Output Waveform in One-shot Mode (Normal Polarity)



19.5.4.4 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-numbered channel corresponds to the lower 16-bit operation, and the odd-numbered 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. Different register settings should be used 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).
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).

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.

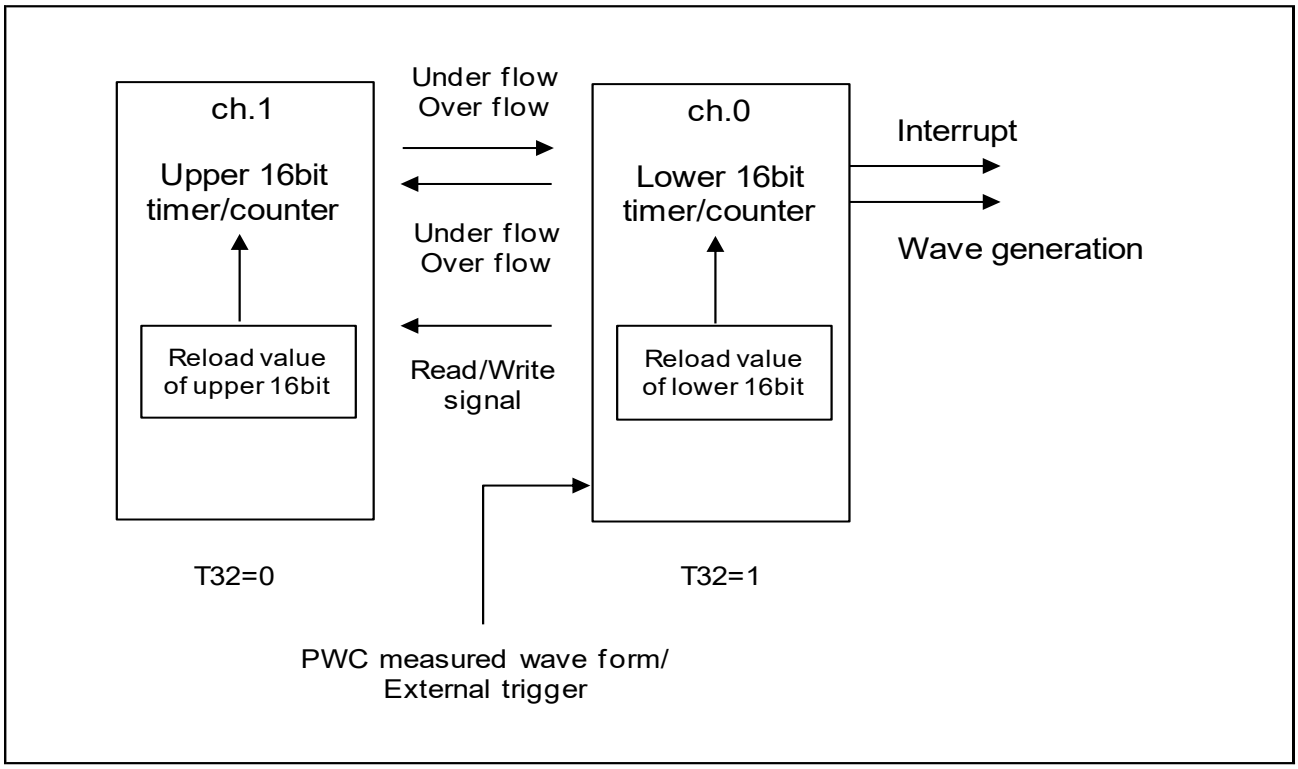
However, the counting operation conforms to the settings of the even-number channels, ignoring the settings of the registers next to the odd-number channels.

- 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 channel configuration in 32-bit timer mode is shown below.

Figure 19-12. 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-numbered 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.

19.5.4.5 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 19-5. 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 when issuing an interrupt request, see "C. List of Interrupts Vector" in entitled "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 the chapter entitled "Chapter: Interrupt Control (Interrupt Controller)".

19.5.4.6 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.

Notes on Interrupts

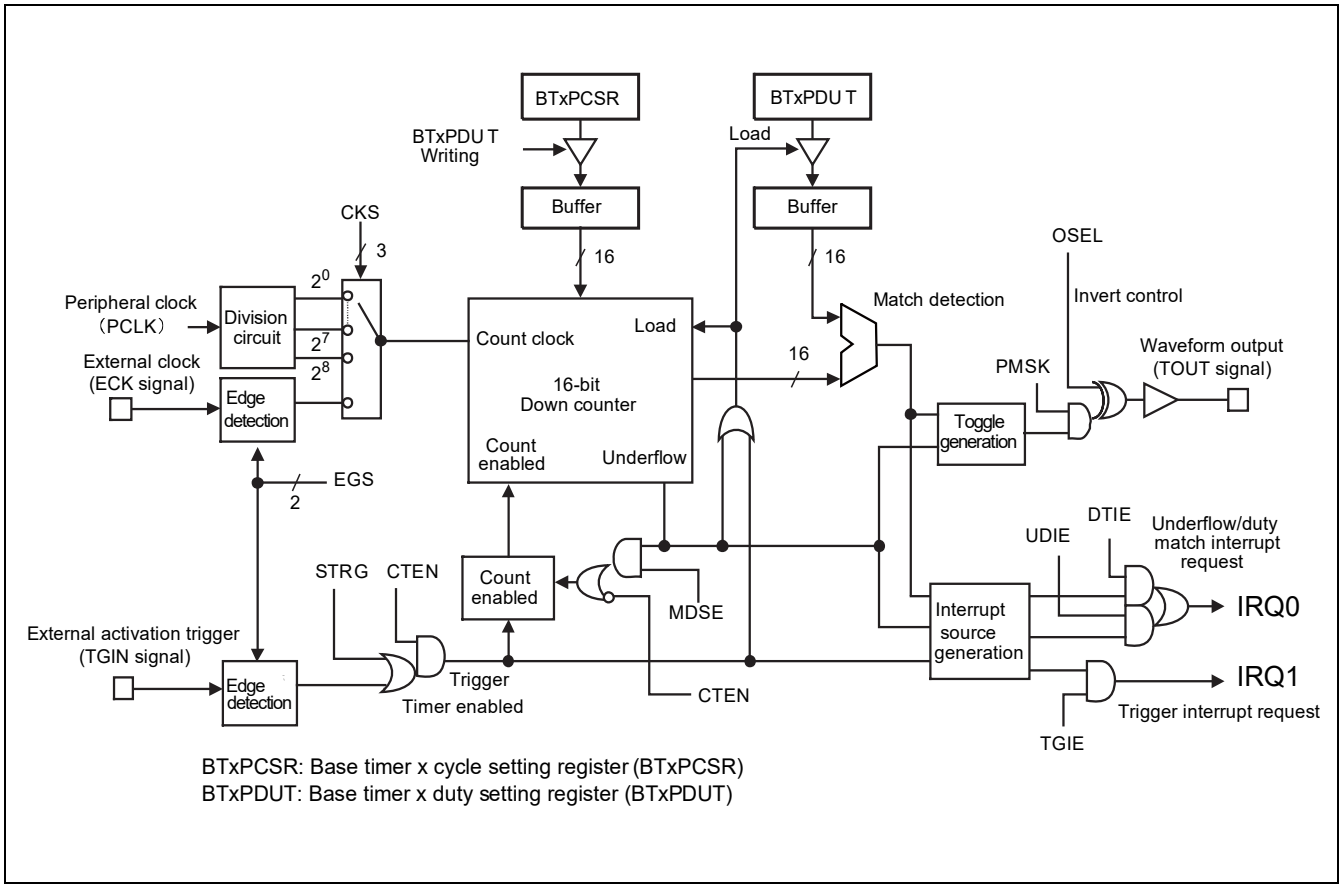
- If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

19.5.5 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 19-13. Block Diagram (16-bit PWM Timer Operation)



BTxPCSR: Base timer x cycle setting register (BTxPCSR)
BTxPDUT: Base timer x duty setting register (BTxPDUT)

19.5.5.1 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.

19.5.5.2 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:

     a. Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
     a. 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).
  - 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:

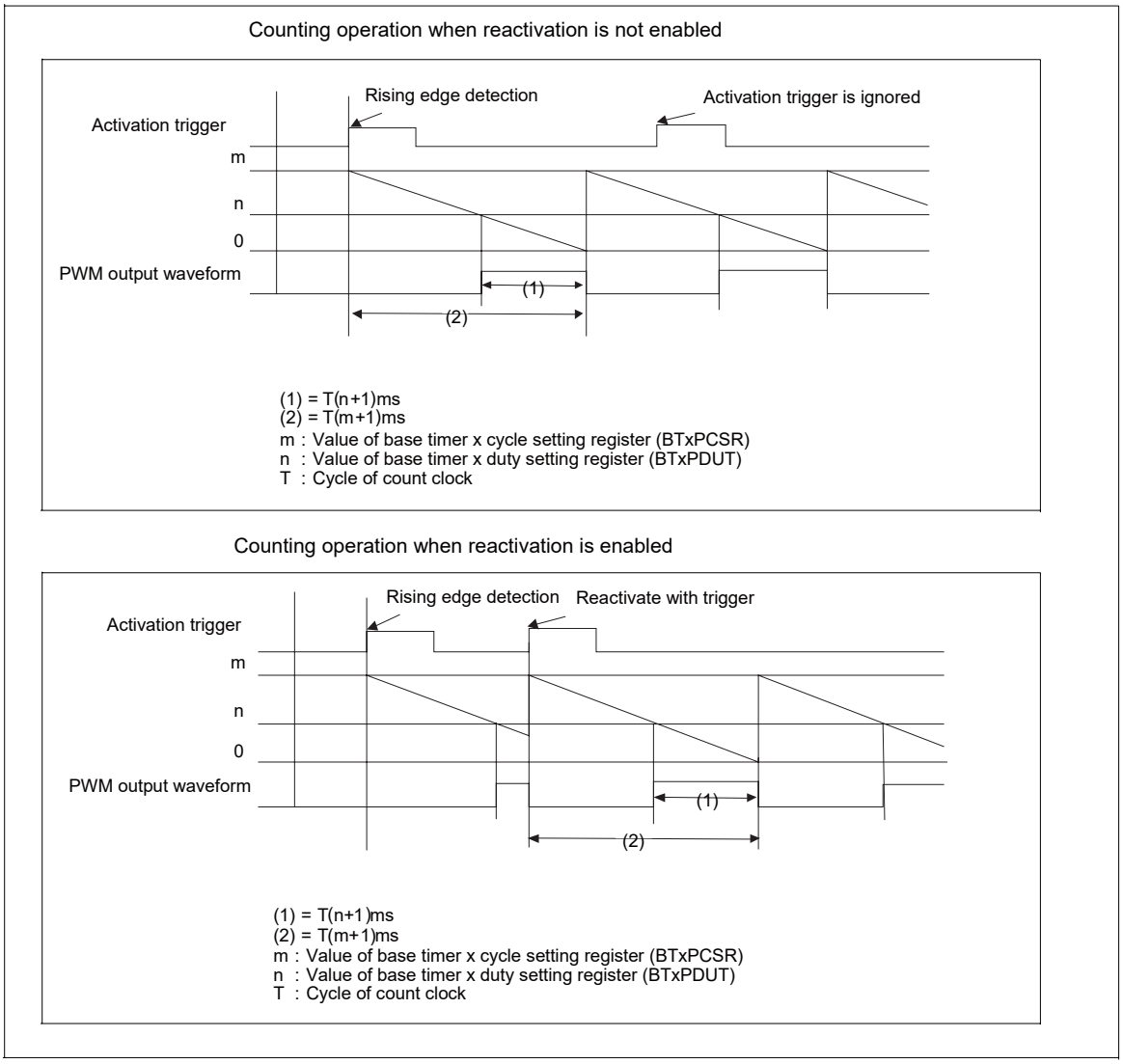
  - ☐ The DTIR bit of the status control register (BTxSTC) changes to "1".
  - ☐ The level of the output signal (TOUT) is inverted.
  - ☐ Countdown is continued. Later, when the 16-bit down counter underflows, the operation is performed as follows:
  - ☐ The UDIR bit of the status control register (BTxSTC) 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 19-14. 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

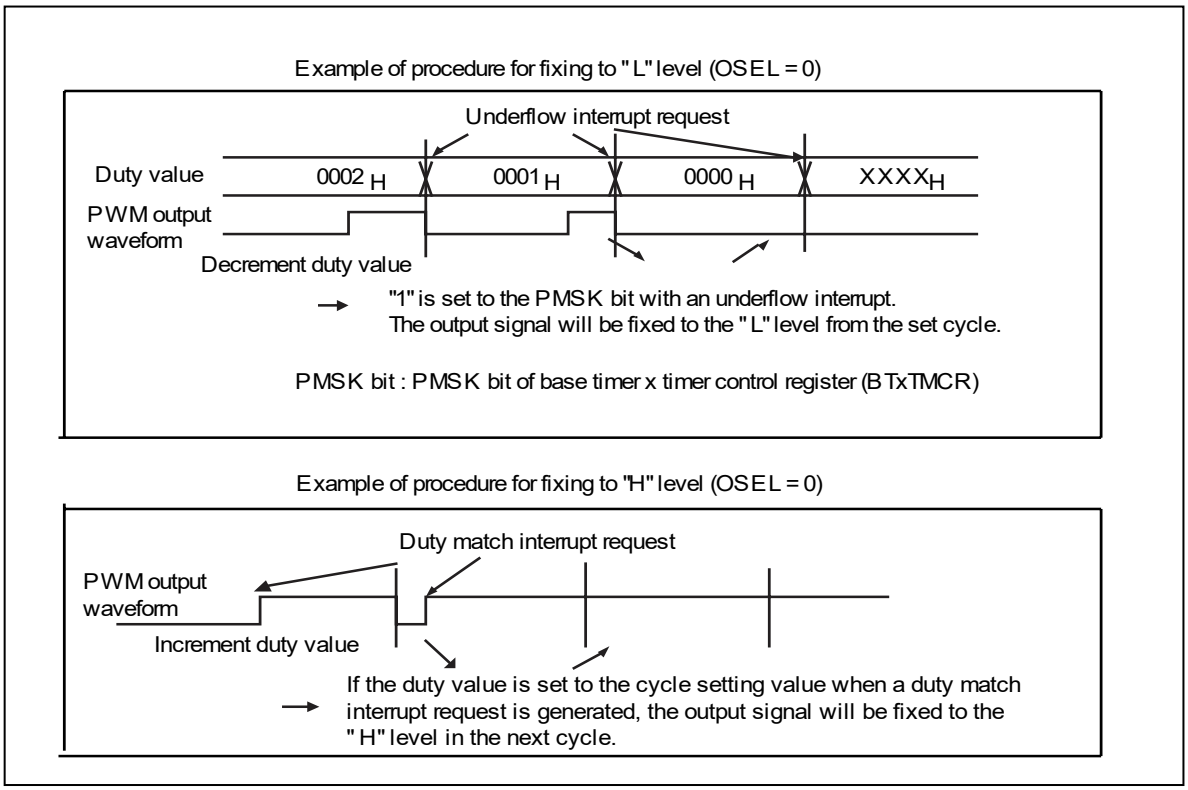
The waveform (TOUT signal) of the 16-bit PWM timer can be output. The waveform (TOUT signal) to be output varies according to the setting of the OSEL bit of the base timer x timer control register (BTxTMCR).

- 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 19-15. 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, TIOA1 pin functions

Interrupt Generation Timing

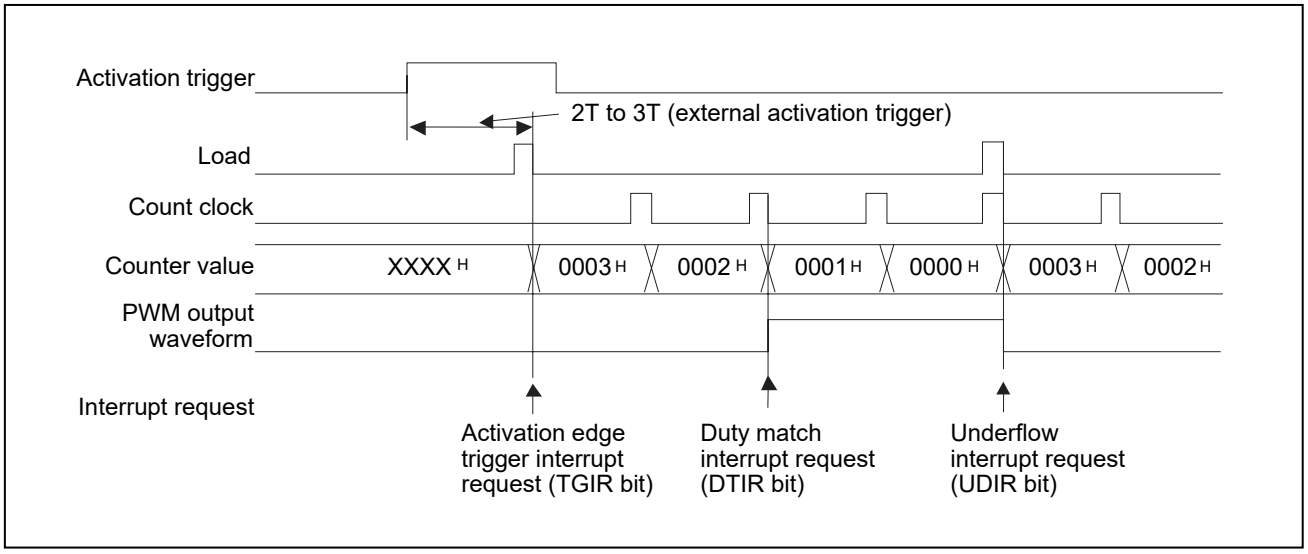
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- 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)
- When 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 19-16. Interrupt Request Generation Timing Chart



19.5.5.3 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 the section entitled "19.5.5.2 Operation in Reload Mode".

- Counting Operation

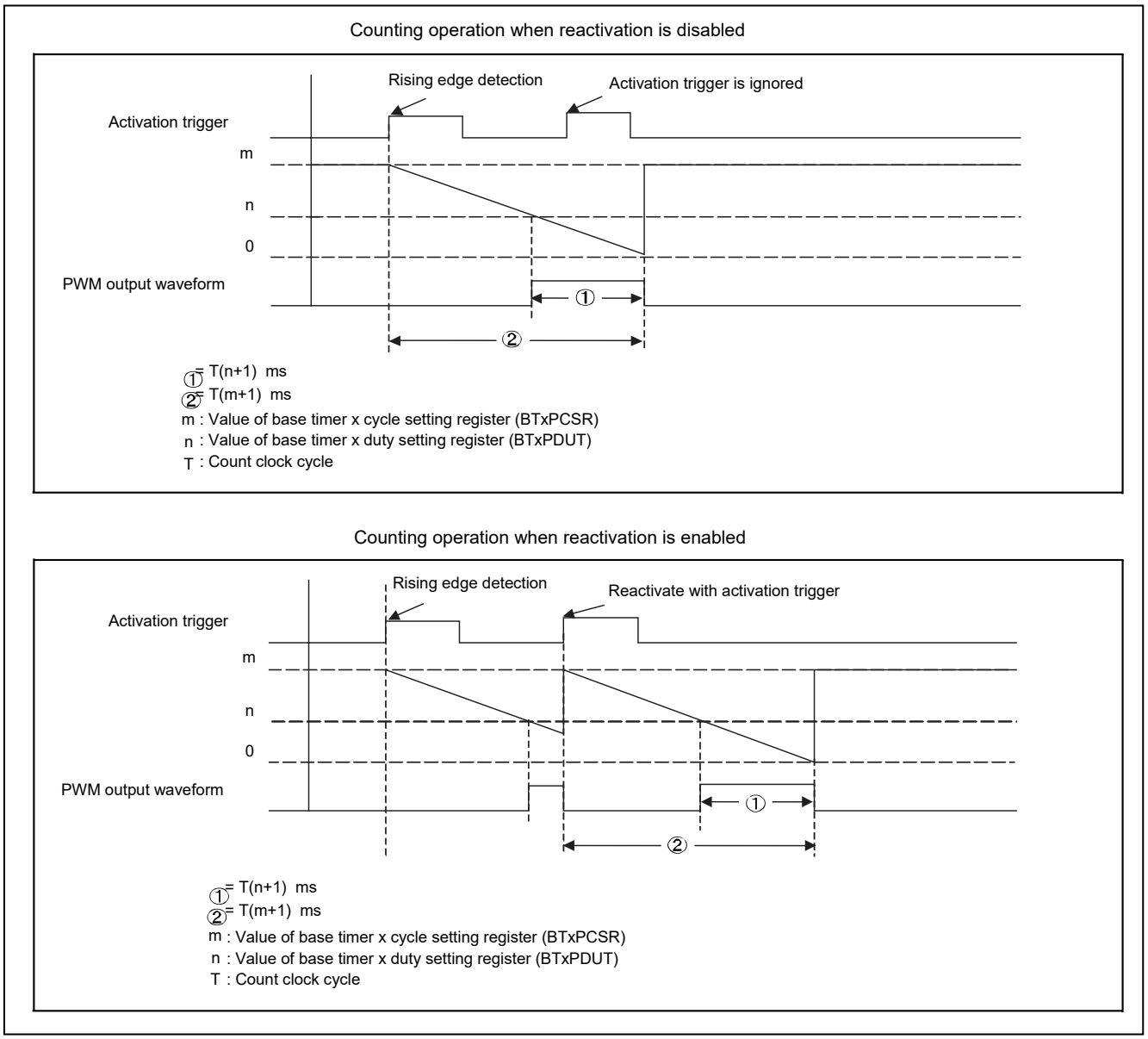
  When an activation trigger is input, the 16-bit down counter, in synchronization with the count clock, starts decreasing from the value set in the cycle setting register (BTxPCSR). When the value of the 16-bit down counter matches the value of the duty setting register (BTxPDUT), the operation is performed 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 19-17. 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 "19.5.5.2 Operation in Reload Mode".

Interrupt Generation Timing

It is the same operation as in reload mode. See "Interrupt Generation Timing" in "19.5.5.2 Operation in Reload Mode".

19.5.5.4 Interrupt

This section explains interrupts.

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 19-6. Conditions for Interrupt Generation

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 when issuing an interrupt request, see "C. List of Interrupts Vector" in entitled "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 the chapter entitled "Chapter: Interrupt Control (Interrupt Controller)".

19.5.5.5 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 Operation

- 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.

Notes on Interrupts

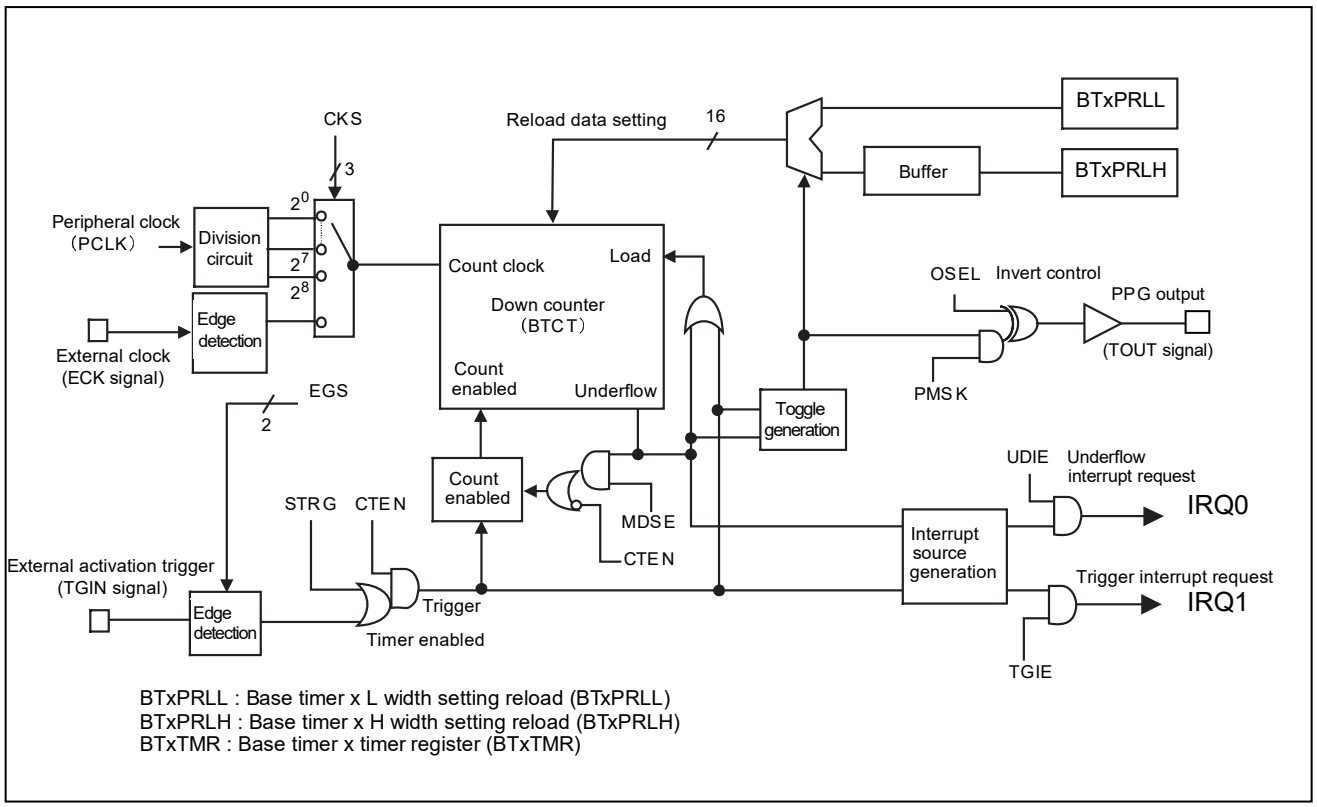
If an instruction to clear the interrupt request flag and an instruction to change the interrupt request flag to "1" occur at the same time, the flag clear instruction is ignored. The interrupt request flag is held to "1".

19.5.6 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.

Table 19-7. Block Diagram (16-bit PPG Timer Operation)



19.5.6.1 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.

19.5.6.2 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 × (L + 1)
"H" level pulse width = T × (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.

19.5.6.3 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:
     a. Set the STRG bit of the base timer x timer control register (BTxTMCR) to "1" (software trigger).
     b. 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).
  - 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).

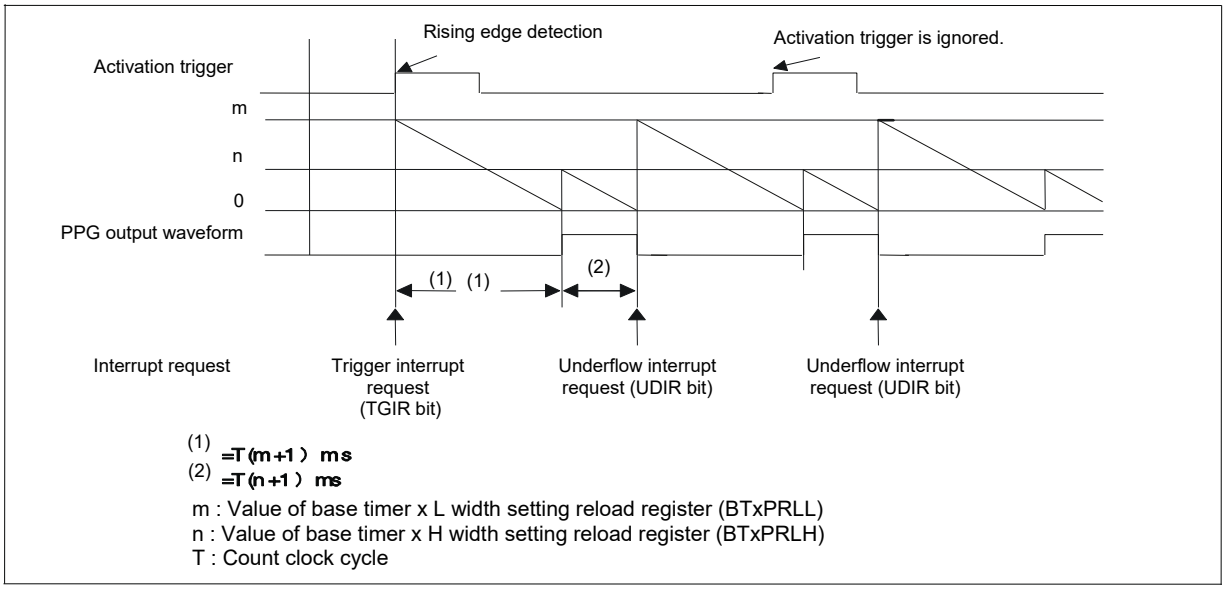
  1. The value set in the L width setting reload register (BTxPRLL) is transferred to the 16-bit down counter and the value set in the base timer x H width setting reload register (BTxPRLH) is transferred to the buffer. The 16-bit down counter begins to count down from the value of the L width setting 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 19-18. Example of Counting Operation in Reload Mode

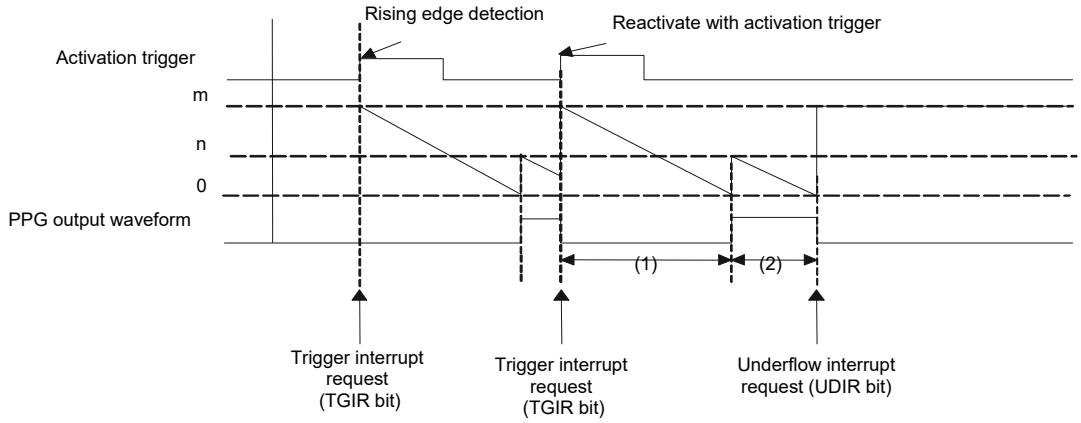
Counting operation when reactivation is disabled



(1) =T (m+1) ms
(2) =T (n+1) ms

m : Value of base timer x L width setting reload register (BTxPRLL)
n : Value of base timer x H width setting reload register (BTxPRLH)
T : Count clock cycle

Counting operation when reactivation is enabled



(1) =T (m+1) ms
(2) =T (n+1) ms

m : Value of base timer x L width setting reload register (BTxPRLL)
n : Value of base timer x H width setting reload register (BTxPRLH)
T : Count clock cycle

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, TIOA1 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:

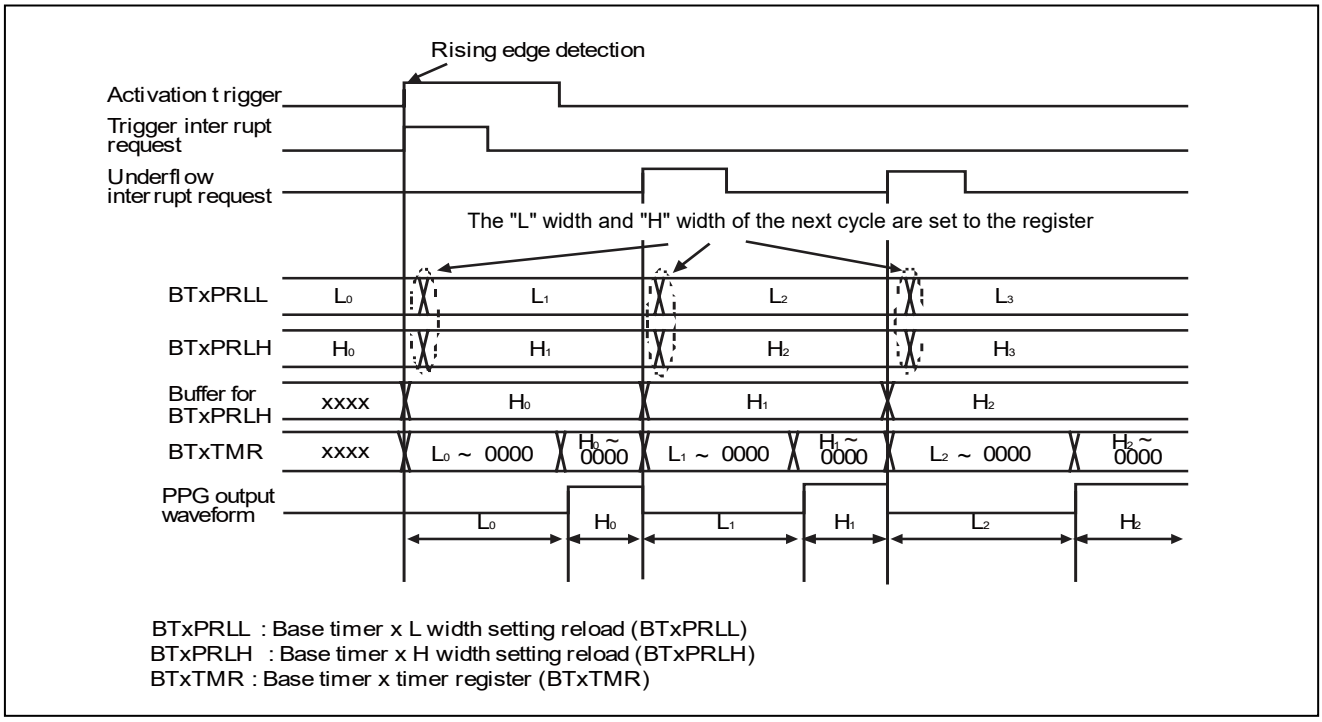
- An activation trigger is detected.
- An underflow occurs after counting down from the value of the base timer x H width setting reload register (BTxPRLH) is completed.

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 19-19. Write Timing



Interrupt Generation Timing

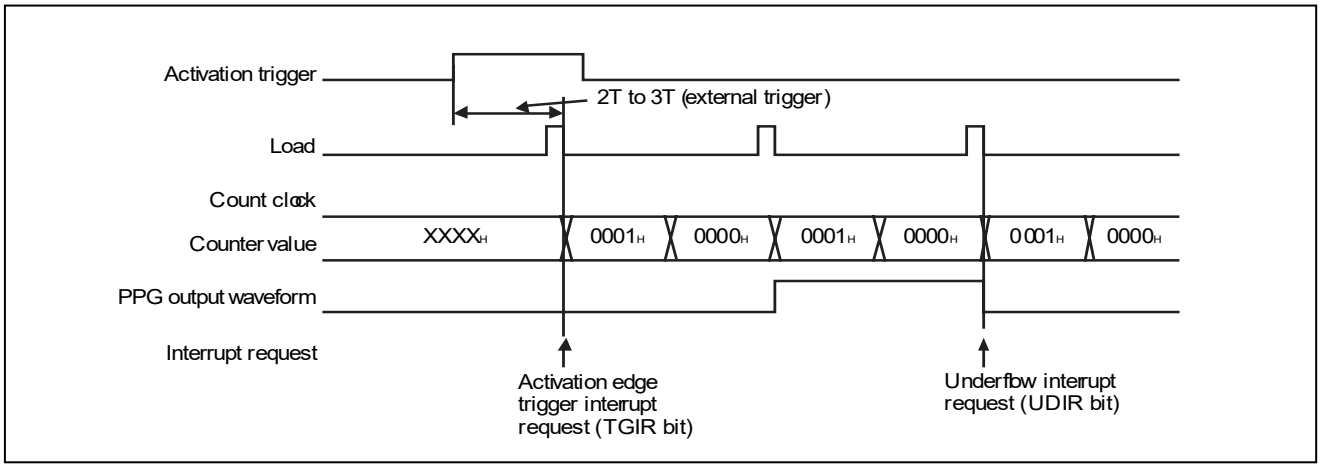
The 16-bit PPG timer can generate an interrupt request in one of the following events:

- An activation trigger is detected.
- An underflow occurs based on the value of H width setting reload register (BTxPRLH).

An example of interrupt request generation timing using the following settings is shown below.

- Value of L width setting reload register (BTxPRLL) = 0001_H
- Value of H width setting reload register (BTxPRLH) = 0001_H

Figure 19-20. Interrupt Request Generation Timing Chart



19.5.6.4 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 "19.5.6.3 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 19-21. Example of Counting Operation If Reactivation Is Not Enabled

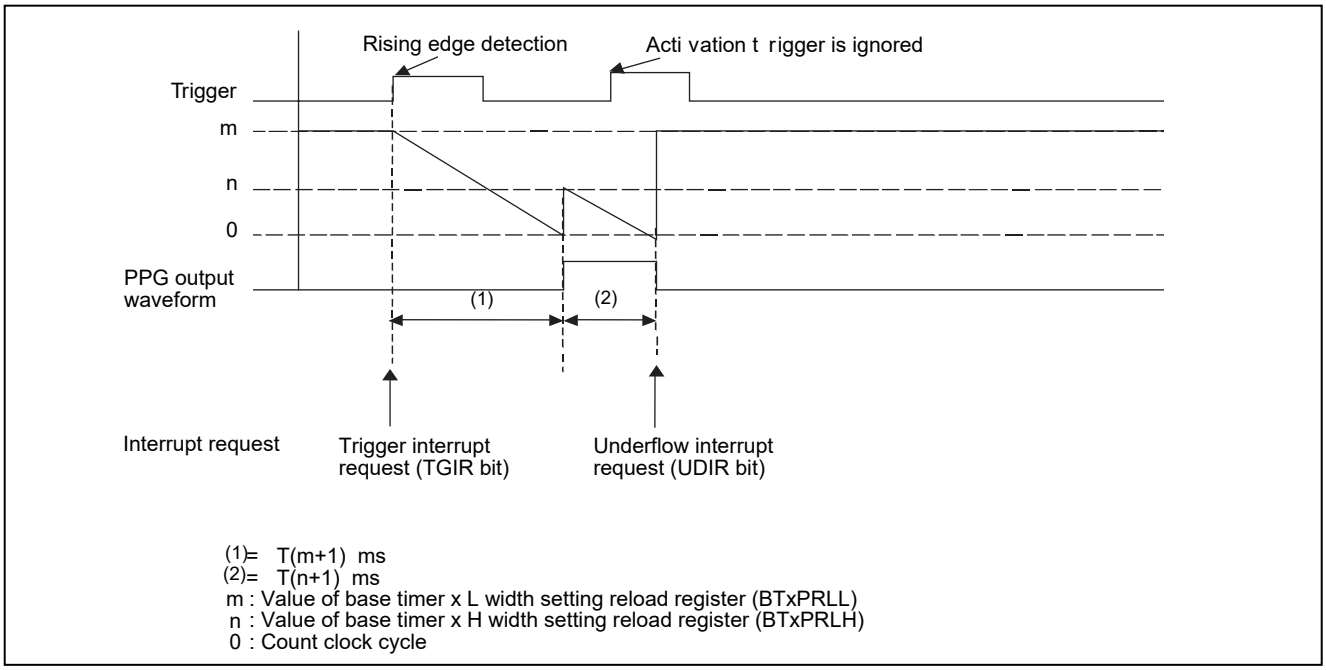
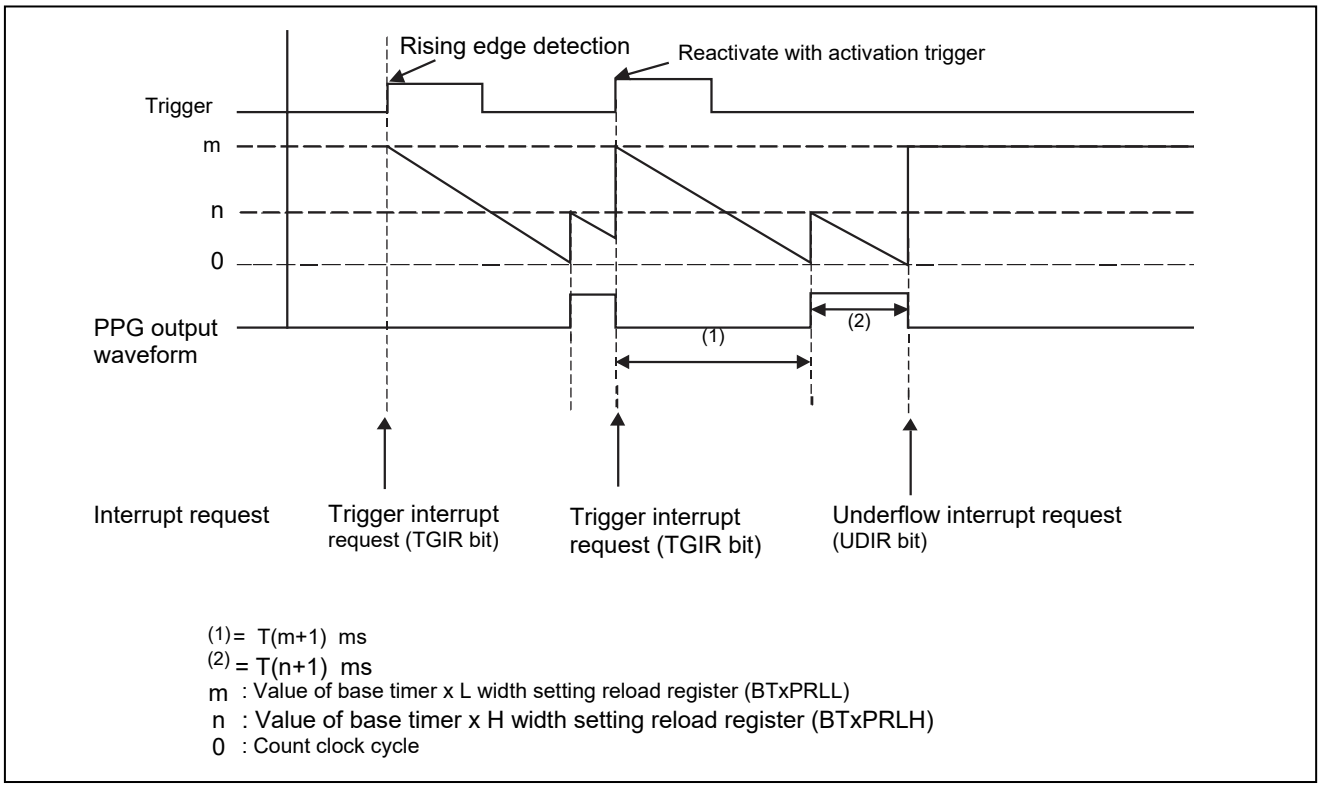


Figure 19-22. Example of Counting Operation If Reactivation Is Enabled



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, TIOA1 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 "19.5.6.3 Operation in Reload Mode".

19.5.6.5 Interrupts

This section explains interrupts of the 16-bit PPG timer operation.

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 19-8. 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 the chapter entitled "Chapter: Interrupt Control (Interrupt Controller)".

19.5.6.6 Application Notes

This section explains notes when using the 16-bit PPG timer.

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 if the FMD2 to FMD0 bits of 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

- The value loading precedes if the count timing of the 16-bit down counter and the load timing occur at the same time.
- 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.

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".

19.5.7 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 19-23. Block Diagram (16-bit PWC Timer Operation)

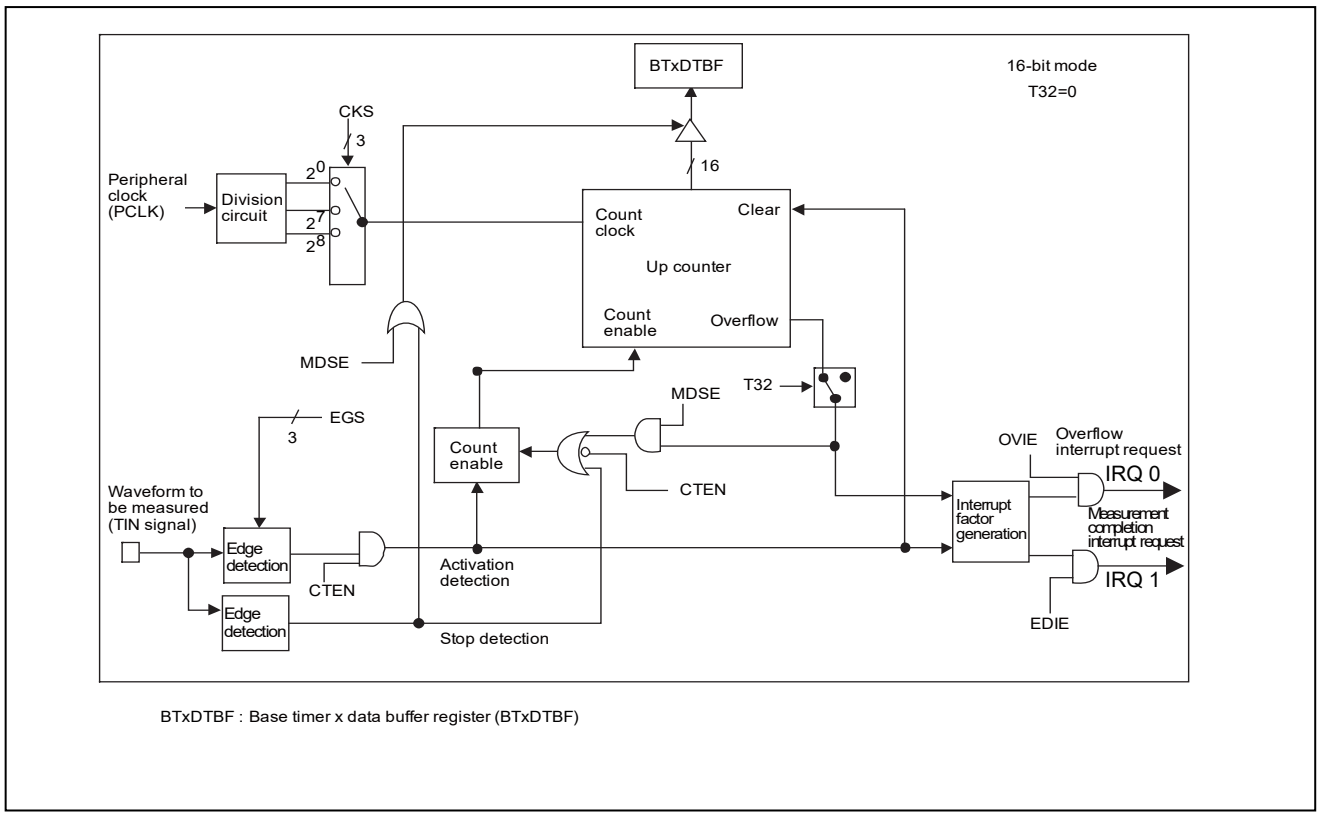
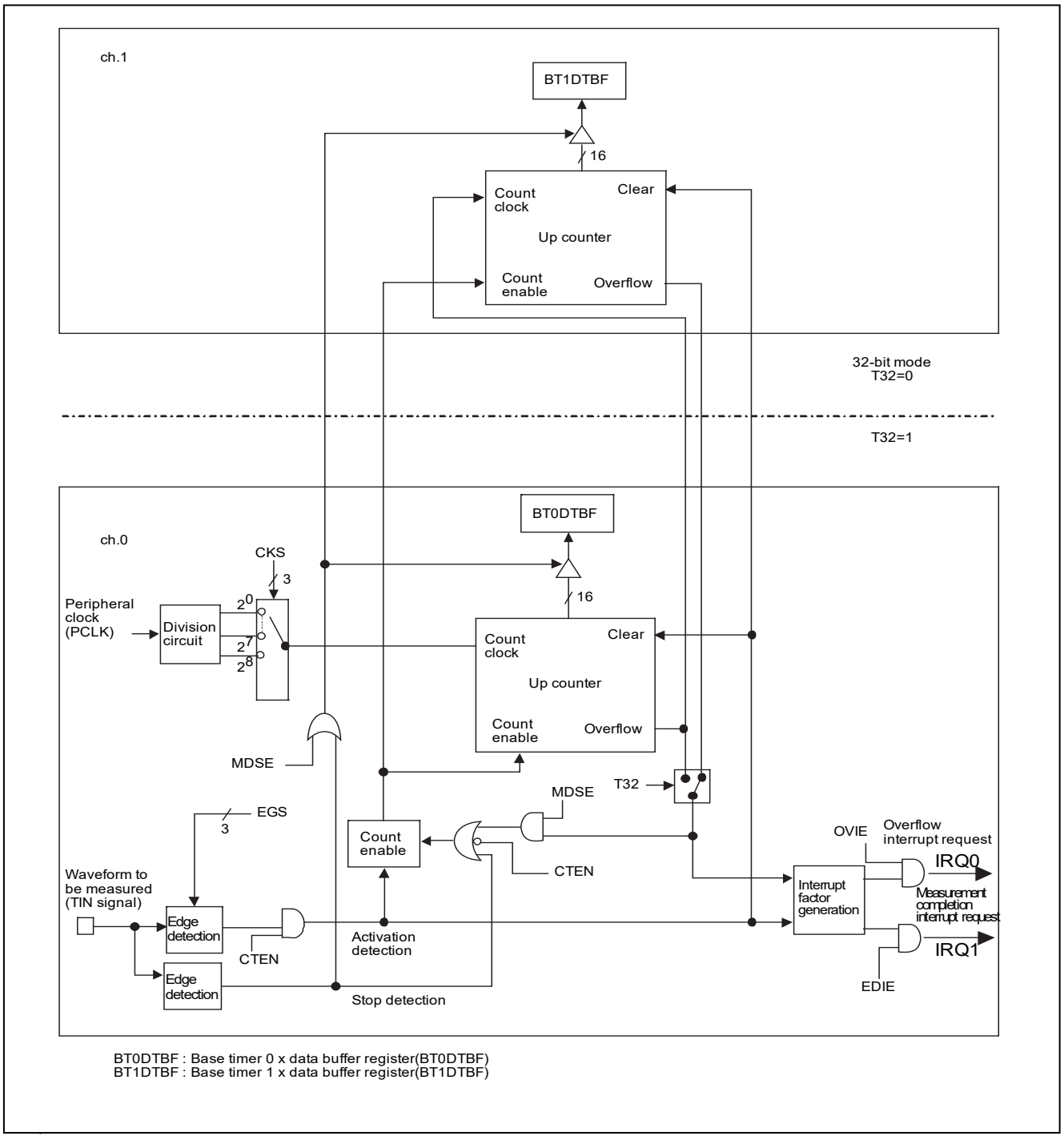


Figure 19-24. Block Diagram (32-bit PWC Timer Operation)



19.5.7.1 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). See "19.5.2 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): Two channels can be cascaded and used as a 32-bit PWC timer.

See "19.5.7.3 32-bit Timer Mode Operation" for details on the operation in 32-bit timer mode.

Note:

The T32 bit setting differs between odd-numbered and even-numbered channels when the 32-bit timer mode is selected. For details, see "19.5.7.3 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).

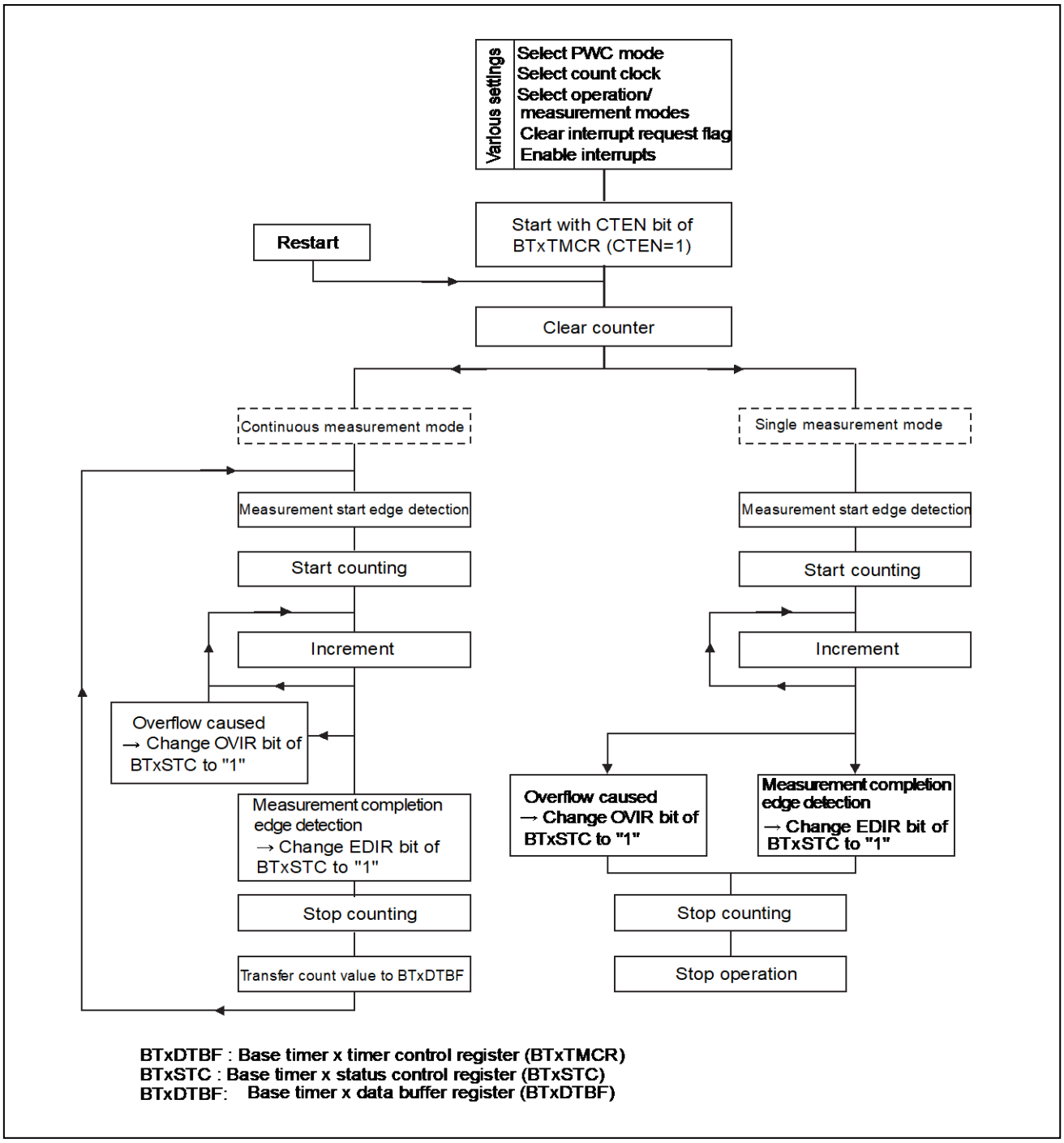
- Continuous measurement mode (MDSE = 0): In this mode, after one sequence of measurement is conducted, the input of the next measurement start edge is awaited and the detection of the next measurement start edge triggers another sequence of measurement.
- Single measurement mode (MDSE = 1): In this mode, measurement is conducted only once. Differences between the single and continuous measurement modes are listed on the table below.

Table 19-9. 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

Figure 19-25 shows the standard operation flow.

Figure 19-25. Operation Flow



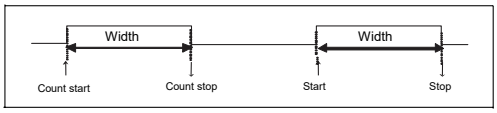
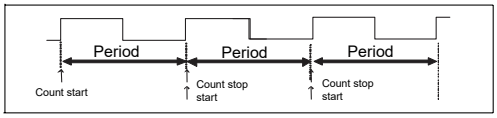
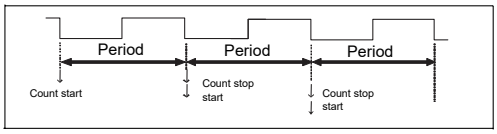
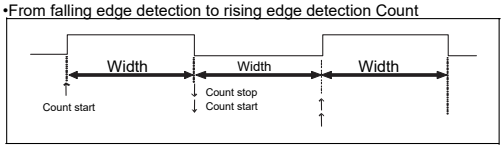
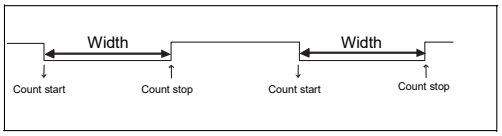
Note:

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

Either of the following five modes can be selected using EGS2 to EGS0 bits of the timer control register (BTxTMCR).

Figure 19-26. Measurement Modes and their Explanation

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
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  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

19.5.7.2 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 19-26.

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 "0000_H" 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).

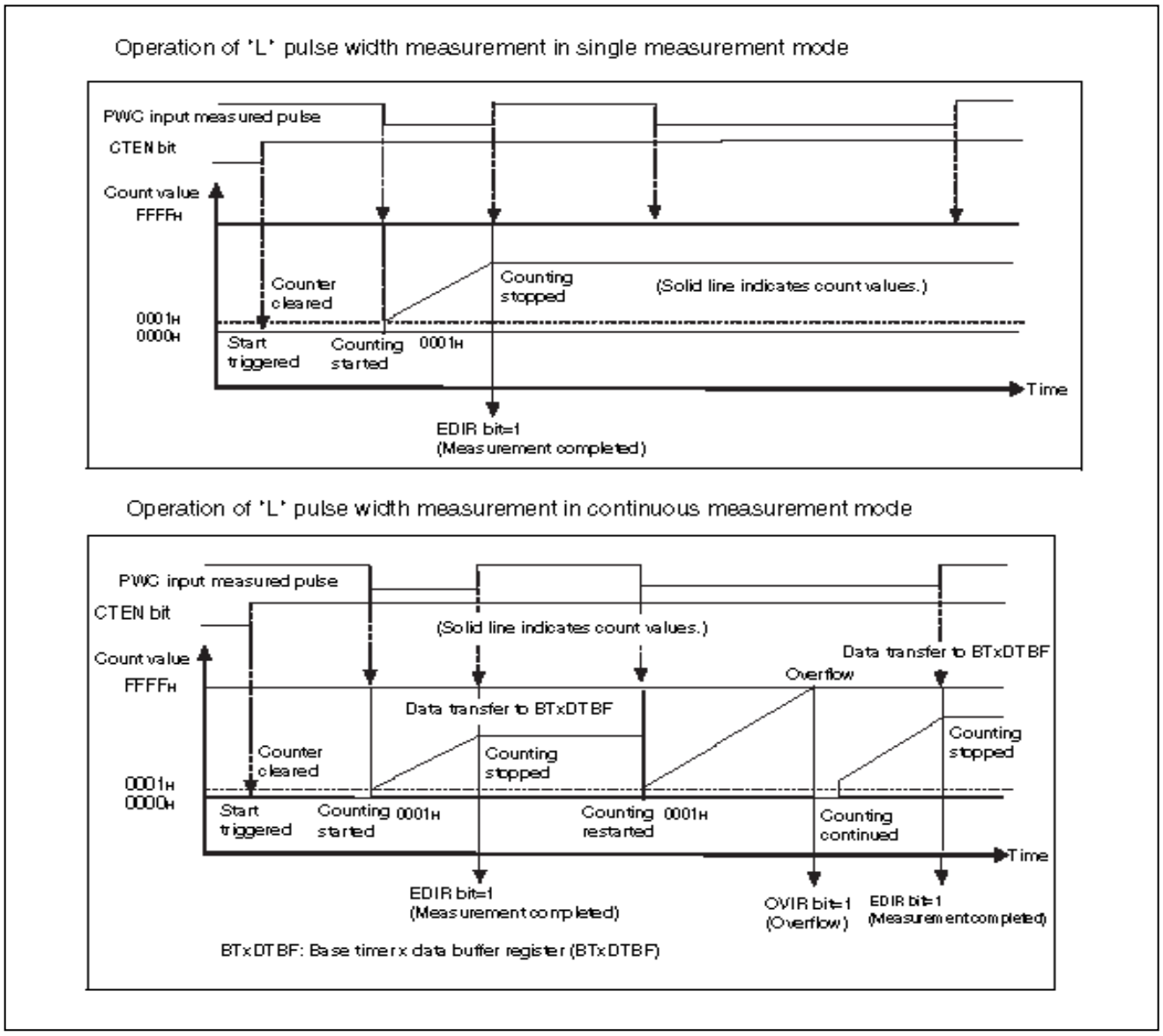
- Operation in Continuous Measurement Mode

  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).

Figure 19-27. 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 × T

n: Data buffer register (BTxDTBF) value

T: Count clock cycle

19.5.7.3 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-numbered channel corresponds to the lower 16-bit operation, and the odd-numbered channel corresponds to the upper 16-bit operation. Therefore, the up counter must be read in the order of the lower 16 bits (even-numbered channel) → the upper 16 bits (odd-numbered 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-numbered channel to "1". Also, set the T32 bit of the odd-numbered channel to "0". When setting 32-bit timer mode, set the registers using the procedure shown below.

The register setting differs between even-numbered and odd-numbered channels. In this example, channel 0 and channel 1 are connected by cascading.

1. Specify ch.0 to reset mode by setting 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)

Note:

Rewrite the T32 bit while the operation of both of the even-numbered and odd-numbered channels are 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 the 32-bit timer mode, the counting operation is basically the same as in the 16-bit timer mode. However, the counting operation conforms to the settings of the even-number channels, ignoring the settings of the registers next to the odd-number channels.

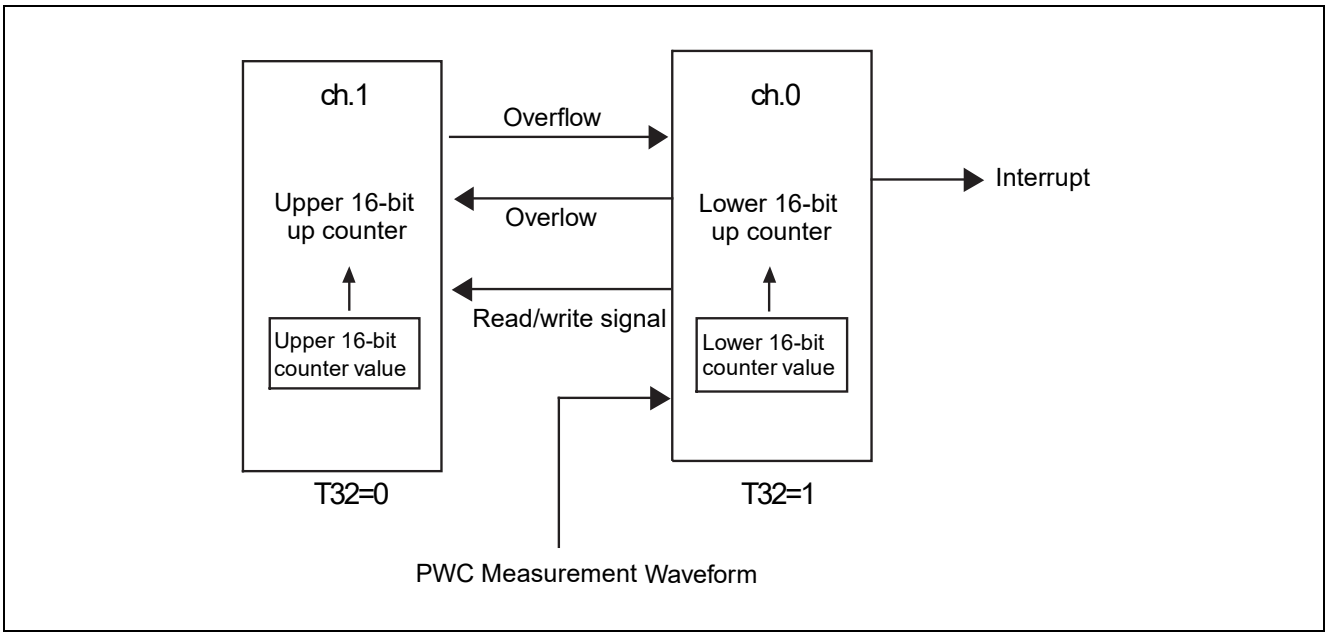
- Base timer x timer control register (BTxTMCR)
- Base timer x status control register (BTxSTC)

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-numbered 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), the lower 16-bit data of the up counter is stored in the data buffer register (BTxDTBF) of the even-numbered channel. Also, the upper 16-bit data is stored in the data buffer register (BTxDTBF) of the odd-numbered channel.

The channel configuration in 32-bit timer mode is shown below.

Figure 19-28. 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 bits (even-numbered channel) → upper 16 bits (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-numbered channel is effective.

19.5.7.4 Interrupt

This section explains interrupt 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 19-10. 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 "C. List of Interrupts Vector" in entitled "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 the chapter entitled "Chapter: Interrupt Control (Interrupt Controller)".

19.5.7.5 Application Notes

This section explains application notes of the base timer.

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 reset 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

- The value loading precedes if the count timing of the up counter and the load timing occur at the same time.
- 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 two channels of PWC timers are used as a single 32-bit PWC timer, the 16-bit PWC timer setting of the even-numbered channel is made valid. The timer setting of odd-numbered 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).

20. Reload Timer

This chapter explains the reload timer.

20.1 Overview

20.2 Features

20.3 Configuration

20.4 Registers

20.5 Operation

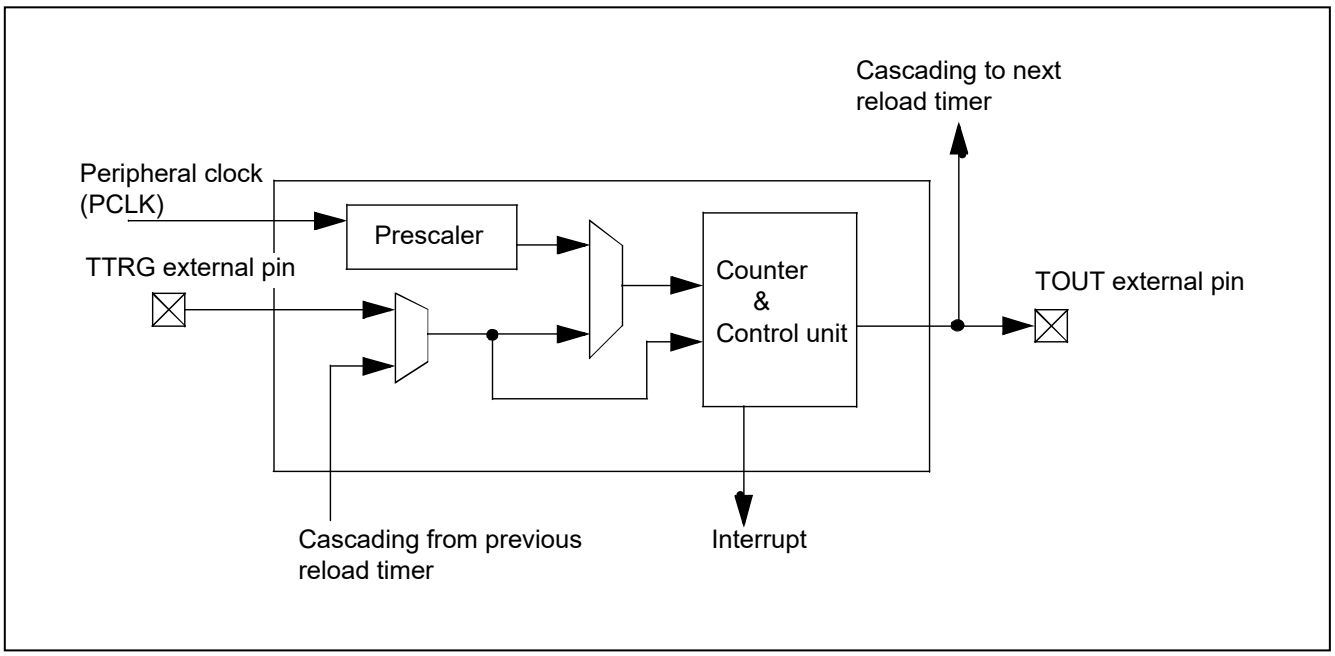
20.6 Application Note

20.1 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 20-1. Block Diagram of Reload Timer (1 Channel, Overview)



Note:

Operation clock is supplied to each group of channels individually as shown below.

Ch.0-Ch3:PCLK2

Ch.4-Ch6:PCLK1

20.2 Features

This section explains features of the reload timer.

A 7-channel reload timer is installed in this series. Each channel is configured as follows:

- 16-bit down counter ×1
- 16-bit reload register ×1
- 16-bit reload / compare/ capture register ×1
- Buffers described above ×1
- 6-bit prescaler for internal count clock creation ×1
- External trigger/event input (TTRG) ×1
- External toggle output (TOUT) ×1
- Control register ×1
- Count comparator ×1

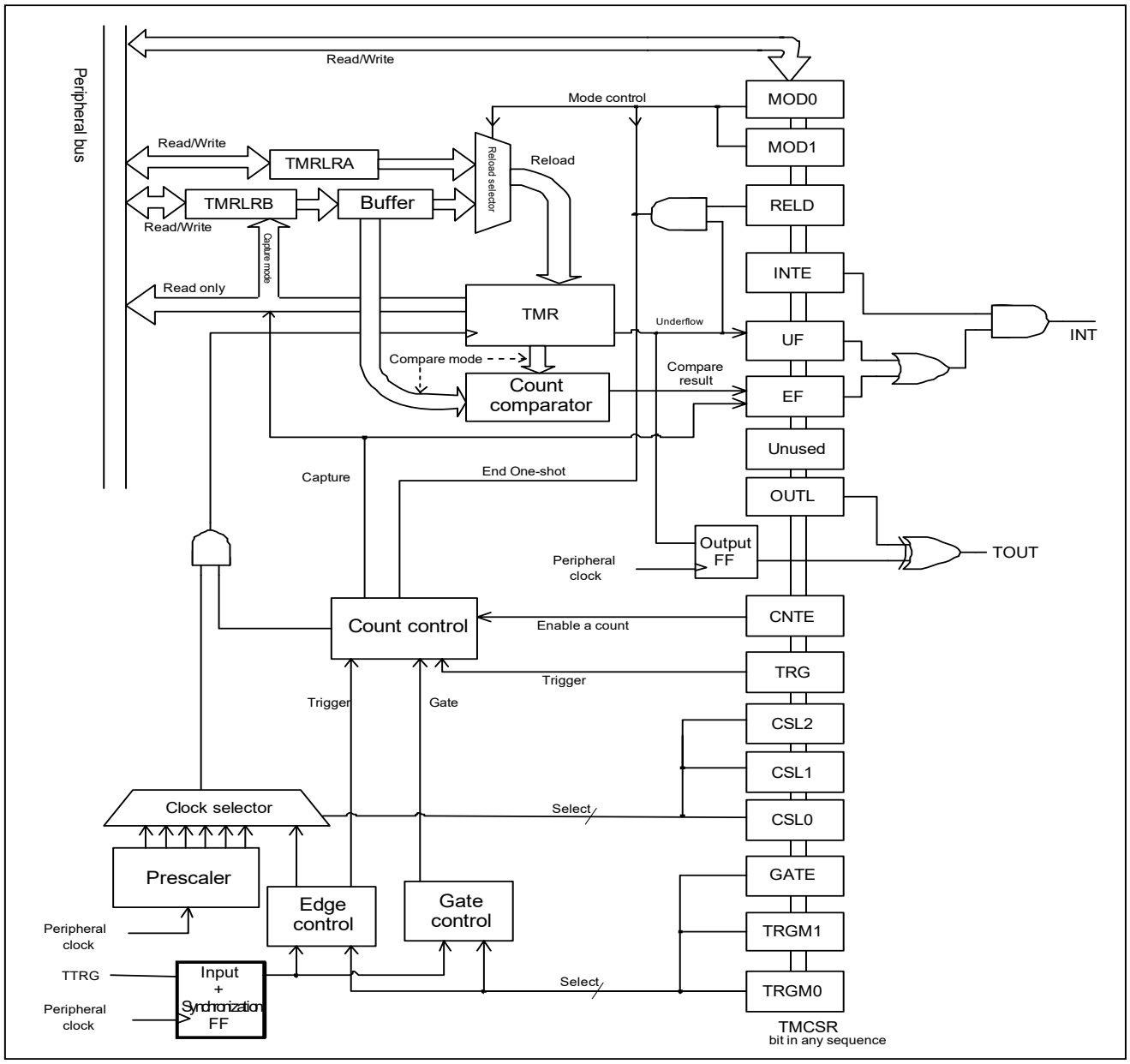
This timer, equipped with the interval timer mode/event counter mode described below, can be used for the following purposes and functions by setting the registers:

- 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.
  - ☐ Use ch.4 output for ch.5 input. Use ch.5 output for ch.6 input.

20.3 Configuration

This section explains the configuration of the reload timer.

Figure 20-2. Block Diagram of Reload Timer (1 Channel, Details)



20.4 Registers

This section explains registers of the reload timer.

Table of Base Address (Base_addr),External Pins

Table 20-1. Table of Base Address (Base_addr),External Pins

Channel	Base_addr	External Pin	
		TOUT	TTRG
0	0x0060	TOT0_0/TOT0_1/TOT0_2	TIN0_0/TIN0_1/TIN0_2
1	0x0100	TOT1_0/TOT1_1/TOT1_2	TIN1_0/TIN1_1/TIN1_2
2	0x0108	TOT2_0/TOT2_1/TOT2_2	TIN2_0/TIN2_1/TIN2_2
3	0x0110	TOT3_0/TOT3_1/TOT3_2	TIN3_0/TIN3_1/TIN3_2
4	0x0048	None	None
5	0x0050	None	None
6	0x0058	None	None

Registers Map

Table 20-2. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0048	TMRLRA4		TMR4		16-bit timer reload register A4 16-bit timer register 4
0x004C	TMRLRB4		TMCSR4		16-bit timer reload register B4 Control status register 4
0x0050	TMRLRA5		TMR5		16-bit timer reload register A5 16-bit timer register 5
0x0054	TMRLRB5		TMCSR5		16-bit timer reload register B5 Control status register 5
0x0058	TMRLRA6		TMR6		16-bit timer reload register A6 16-bit timer register 6
0x005C	TMRLRB6		TMCSR6		16-bit timer reload register B6 Control status register 6
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

20.4.1 Control Status Register: TMCSR (Timer Control and Status Register)

The bit configuration of the control status register is shown below.

These registers control 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-bit8 and bit6-bit4 and write counter operation enabling by writing CNTE=1 simultaneously. It is also possible to rewrite bit15-bit8, bit6-bit4 and write operation disabling by writing CNTE=0 simultaneously.

TMCSR: Address Base_addr + 06H (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 [1:0] (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): TTRG 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=0)]

Select an effective external edge which can be a reload trigger through TTRG input in the following manner:

TRGM[1:0]	TTRG 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 =1)]

Select the pin level which enables the counter during TTRG input in the following manner:

TRGM[1:0]	TTRG Effective Level
x0	TTRG pin "L" counted only during the input period (initial value)
x1	TTRG pin "H" counted only during the input period

[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 are count source selection bits. Select a count source from the internal clock (peripheral clock[1]) and the external event (TTRG 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:TTRG0, ch.1:TOUT0, ch.2:TOUT1, ch.3:TOUT2, ch.4:TTRG4,ch.5:TOUT4,ch.6:TOUT5)	Event counter mode
111	External event (TTRG input)	

[1]: For ch.0 to ch.3, it refers PCLK2. For ch.4 to ch.6, it refers PCLK1.

[bit8] GATE (GATE input enable): Gate input enabling bit

This bit controls the functions of the input pin (TTRG) of (bit11 to bit9:CSL[2:0]=000 to 101) at the interval timer mode specified following.

GATE	TTRG Input Pin Functions
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.[1] (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.

[1]: 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.

20.4.2 16-bit Timer Register: TMR

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[15:0] (Timer): 16-bit timer

This register can be read the counter value of the 16-bit timer. The initial value is undefined.

20.4.3 16-bit Timer Reload Register A, 16-bit Timer Reload Register B: TMRLRA, TMRLRB

The bit configuration of 16-bit timer reload register A and 16-bit timer reload register B is shown below.

TMRLRA register sets the count initial value. Function of TMRLRB depends on 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[15:0] (Timer Reload Register A) : 16-bit reload setting register A

[bit15 to bit0] TMRLRB[15:0] (Timer Reload Register B): 16-bit reload setting register B

The TMRLRA register is where the count initial value is hold. The TMRLRA can be used in all mode with regardless of the bit15, bit14:MOD[1:0] setting in the TMCSR register.

The TMRLRB is to be used by the bit15, bit14:MOD[1:0] setting in the TMCSR register specified following:

Mode	MOD[1:0]	TMRLRB Functions
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 outputted 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		
		1	0		
			1		
11	Capture	0	0	TMRLRA+1	-
			1	-	TMRLRA+1
		1	0	TMRLRA+1	
			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 time mode:

TOUT = (Setting value of this register + 1) × count source cycle

Note:

The formula described above is effective only in the interval timer mode.

20.5 Operation

This section explains the operation of the reload timer.

20.5.1 Setting
20.5.2 Operation Procedure
20.5.3 Operations of Each Counter
20.5.4 Cascade Input
20.5.5 Priority of Concurrent Operations

20.5.1 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}).

20.5.1.1 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 20-3. 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:TTRG0, ch.1:TOUT0, ch.2:TOUT1, ch.3:TOUT2, ch.4:TTRG4,ch.5:TOUT4,ch.6:TOUT5)	Event counter mode
111	External event (TTRG input)	

20.5.1.2 Timer Underflow period

The timer underflow period is shown below.

Underflow is defined as counter down-counting from 0x0000. Set the time (period) 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 period, TUF, in the interval timer mode can be represented as follows:

TUF = Peripheral clock (PCLK) period × prescaler division value (2-64) ×(Reload register value (TMRLRA/B) + 1)

20.5.1.3 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 ... Inputted from the TTRG pin.

The TTRG 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.

20.5.1.4 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 TTRG external pin.

Table 20-4. TTRG Effective Level

TRGM[0]	TTRG Effective Level
0	TTRG pin "L" counted only during the input period (initial value)
1	TTRG pin "H" counted only during the input period

20.5.1.5 Counter Operation Selection

The counter operation selection is shown below.

Select the operation in case of counter underflow in 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 20-5. 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 “20.5.3.3 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

20.5.1.6 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 section of the UF (underflow) 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 20-3. TOUT Output Change in Each Event (1 / 3)

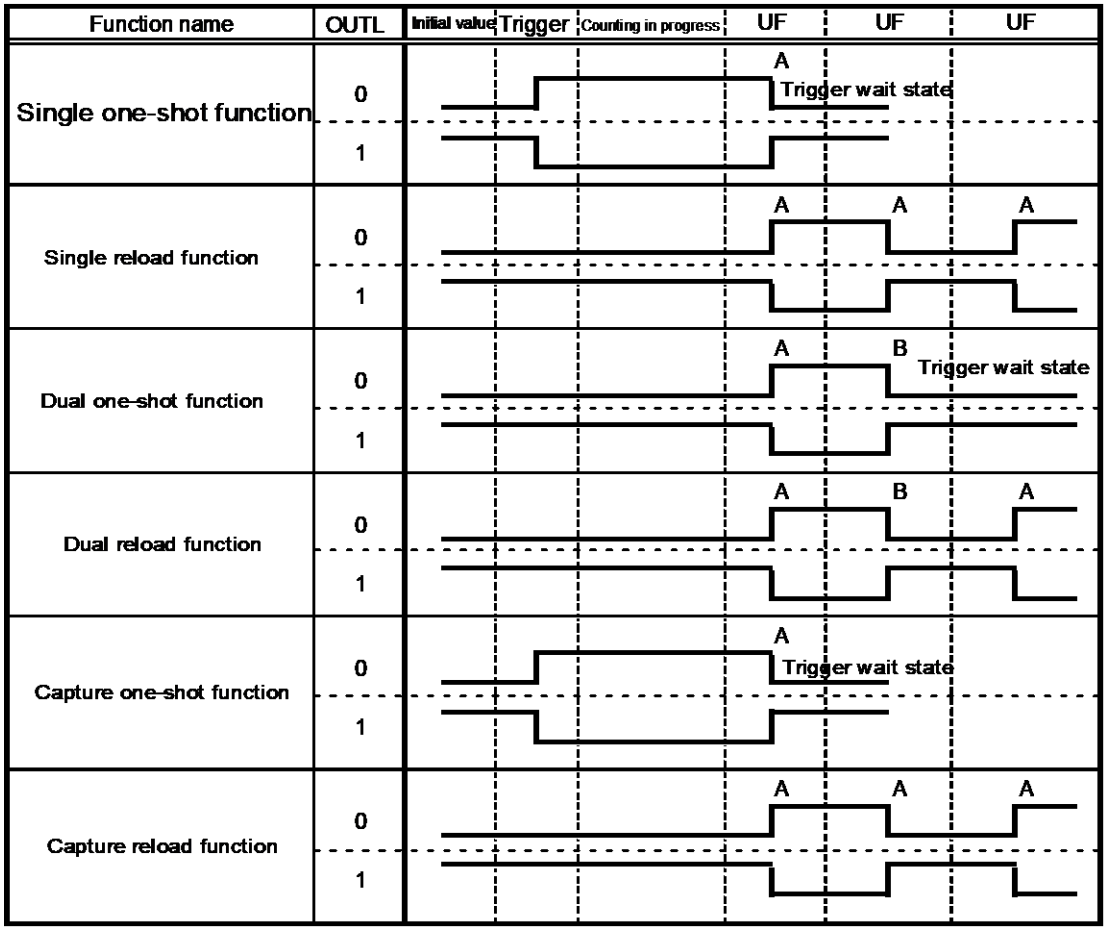


Figure 20-4. TOUT Output Change in Each Event (2 / 3)

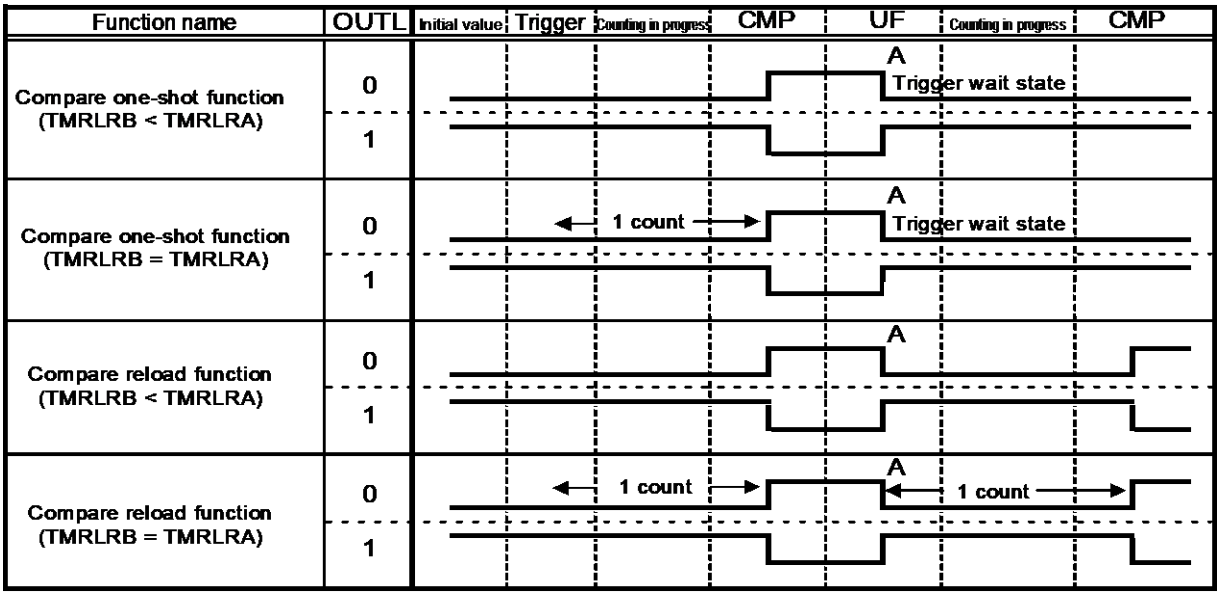
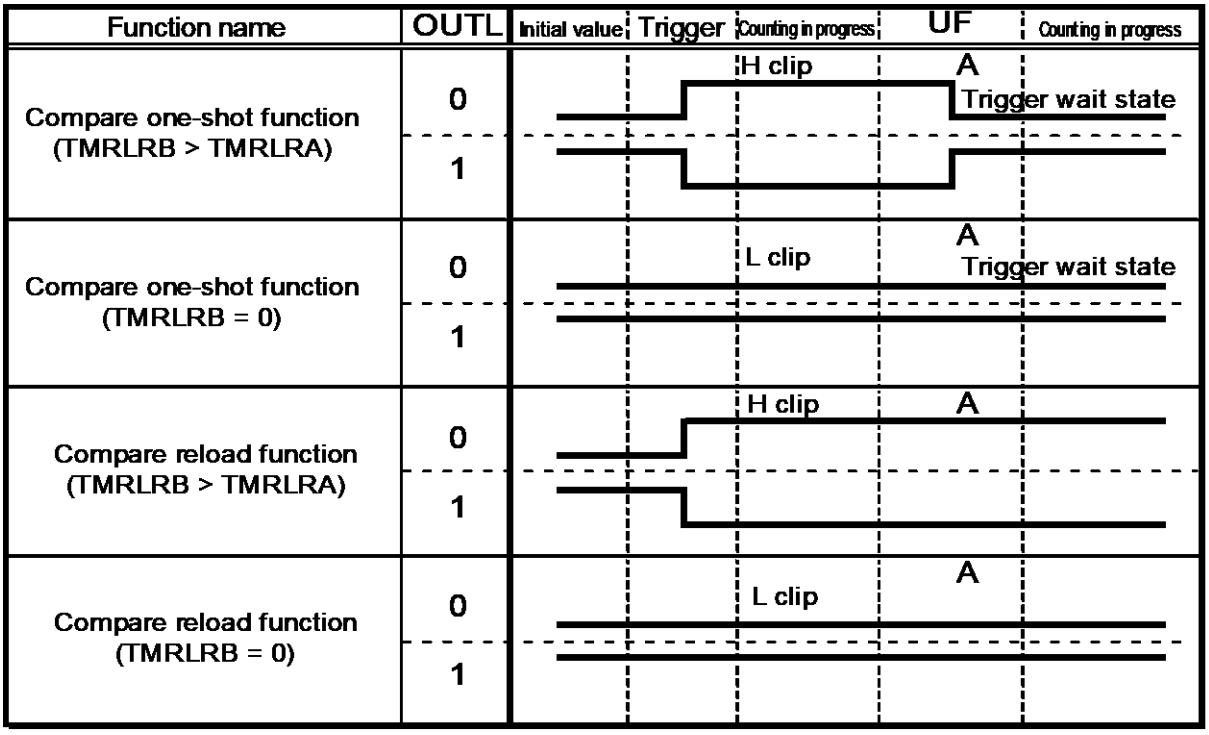


Figure 20-5. TOUT Output Change in Each Event (3 / 3)



20.5.2 Operation Procedure

Operation procedures are shown.

20.5.2.1 Activation
20.5.2.2 Retrigger
20.5.2.3 Underflow/Reload
20.5.2.4 Generation of Interrupt Requests
20.5.2.5 Concurrent Operation of Register Write and a Timer Activation

20.5.2.1 Activation

Activation is shown below.

Writing "1" into the bit1:CNTE bit of the TMCSR register changes the counter state to activation trigger waiting.

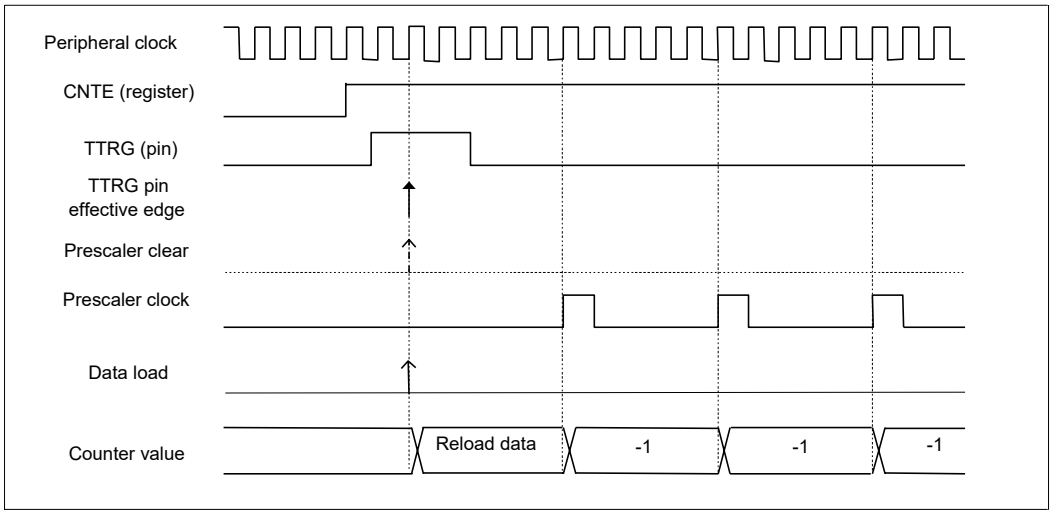
- TTRG input during trigger input functioning

  If writing "1" to the bit0:TRG bit of the TMCSR register or inputting external trigger through TTRG 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 TTRG, input pulse of 2 × T (T indicates the peripheral clock (PCLK) cycle) or more.

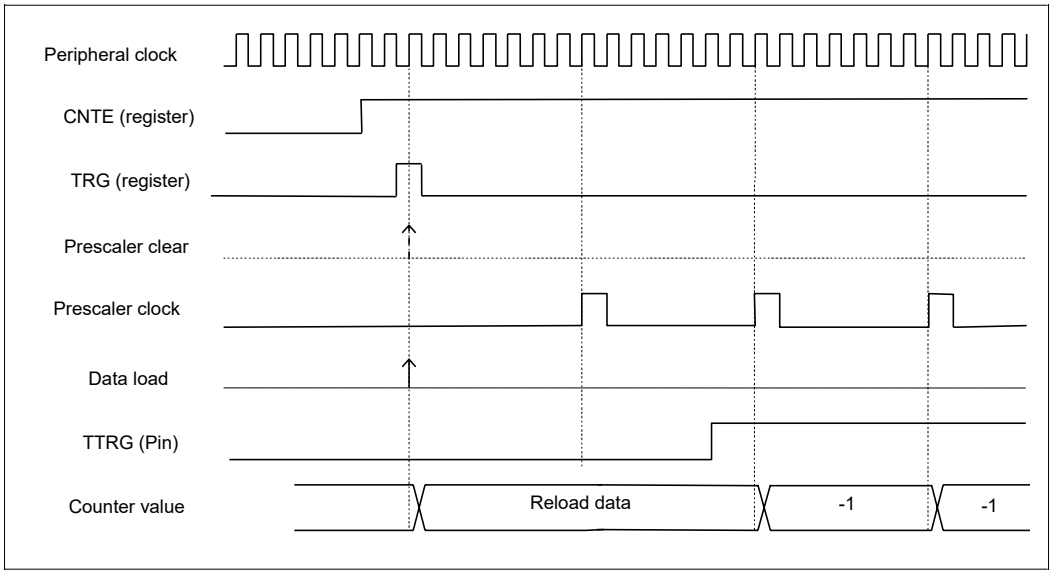
- TTRG input during gate input functioning

  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 TTRG input in the effective input polarity waiting, the timer initiates down count operation. For TTRG, input pulse of 2 × T (T indicates the peripheral clock (PCLK) cycle) or more.

Figure 20-6. Timer Activation



Timer Activation (when the trigger input function and the rising edge trigger are selected)



Timer Activation (when in the gate input function)

20.5.2.2 Retrigger

The retrigger is explained.

The trigger which is generated during timer counting is called "retrigger". In this case, the following actions are taken:

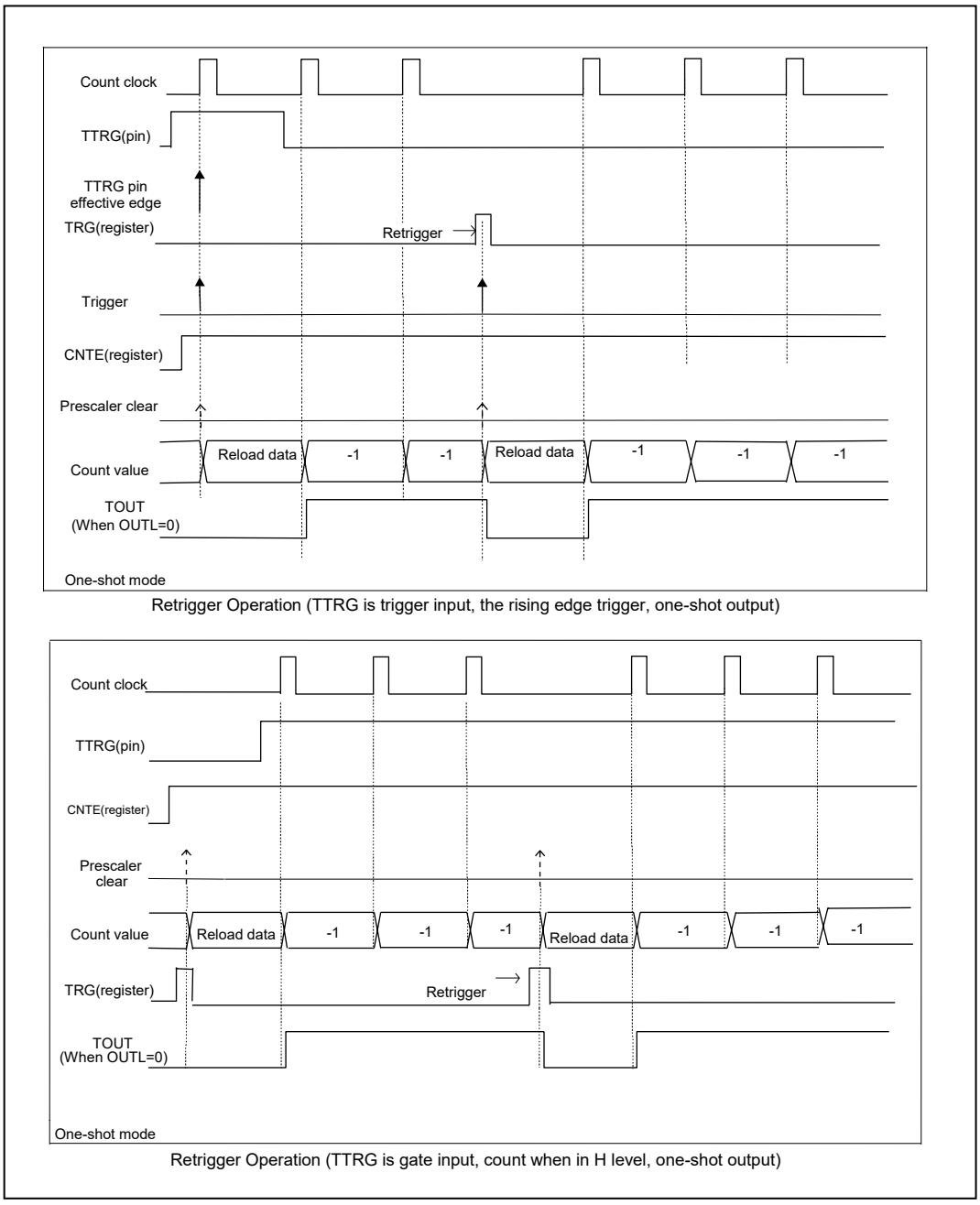
1. Initialize TOUT
2. Load the reload register value to the counter
3. Clear the 6-bit prescaler
4. Continue counting

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 20-7. Retrigger Operation



20.5.2.3 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.

20.5.2.4 Generation of Interrupt Requests

Generation of interrupt requests is shown below.

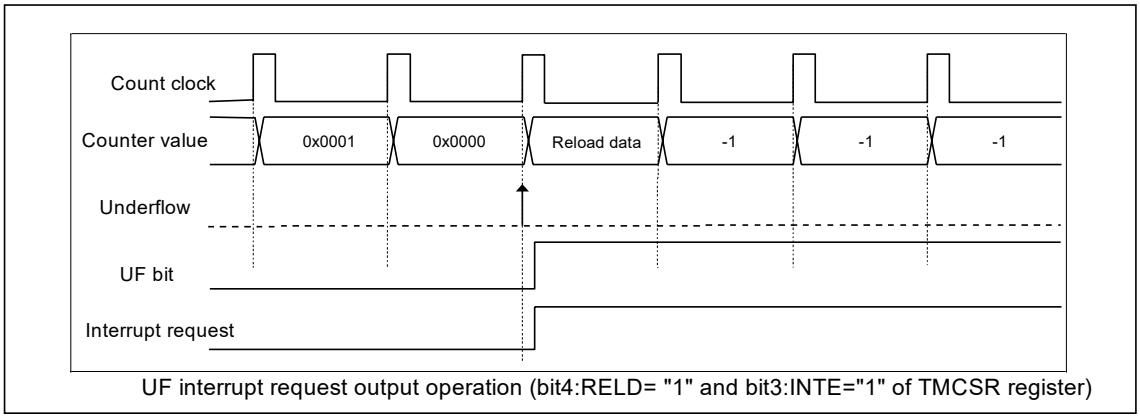
When bit3:INTE bit of the TMCSR register is "1", if bit2:UF bit/bit7:EF bit are set, an interrupt request is generated. In interval timer mode, the UF bit and the EF bit will be set under the following conditions.

- 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 20-8. Example of UF Interrupt Request Output Operation



UF interrupt request output operation (bit4:RELD= "1" and bit3:INTE="1" of TMCSR register)

20.5.2.5 Concurrent Operation of Register Write and a Timer Activation

The concurrent operation of register write and a timer activation is shown below.

The following table shows the operation when a register write by a user and the timer operation occurred simultaneously.

Figure 20-9. 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)

20.5.3 Operations of Each Counter

Operations of each counter are shown.

20.5.3.1 Single One-shot Operation
20.5.3.2 Single Reload Operation
20.5.3.3 Dual One-shot Operation
20.5.3.4 Dual Reload Operation
20.5.3.5 Compare One-shot Operation
20.5.3.6 Compare Reload Operation
20.5.3.7 Capture Mode

20.5.3.1 Single One-shot Operation

The single one-shot operation is shown below.

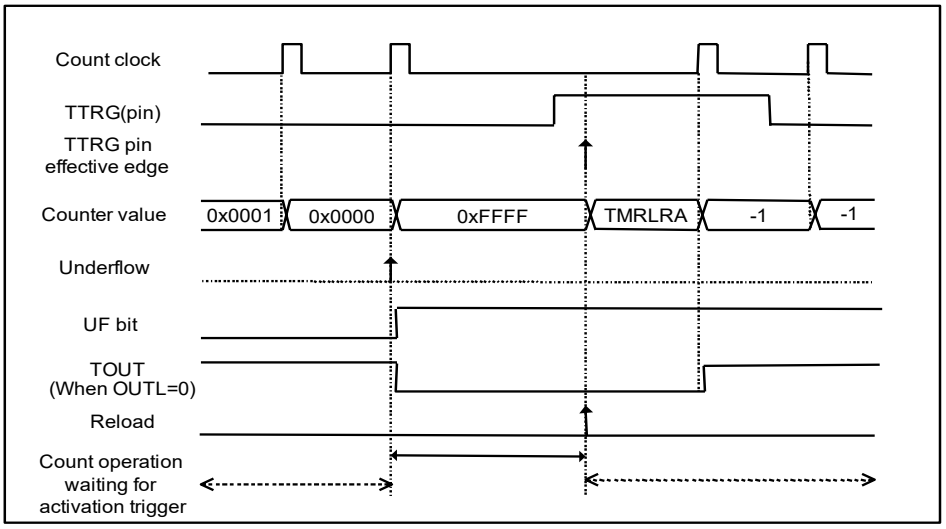
When bit15, bit14:MOD[1:0]=00 and bit4:RELD of the TMCSR register =0, the single one-shot operation will be performed in which the timer stops with 0xFFFF by an occurrence of an underflow.

In the single one-shot configuration, if an underflow occurs, the following operation will be performed.

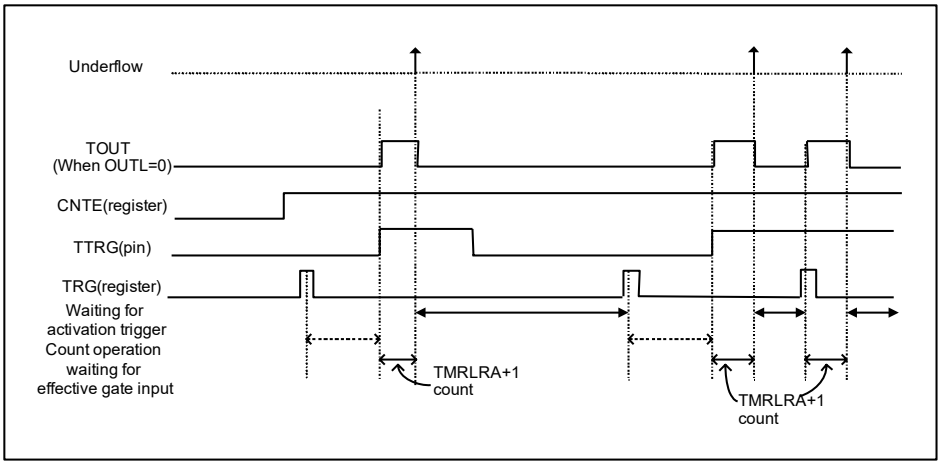
- Sets the UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE=1 of TMCSR register), an interrupt occurs.
- Stops the count with 0xFFFF.
- Initializes TOUT output.
- Timer is waiting for a trigger.

For the single one-shot timer, TMRLRA turns to the initial value of the counter when a reload took place. TMRLRB is not used.

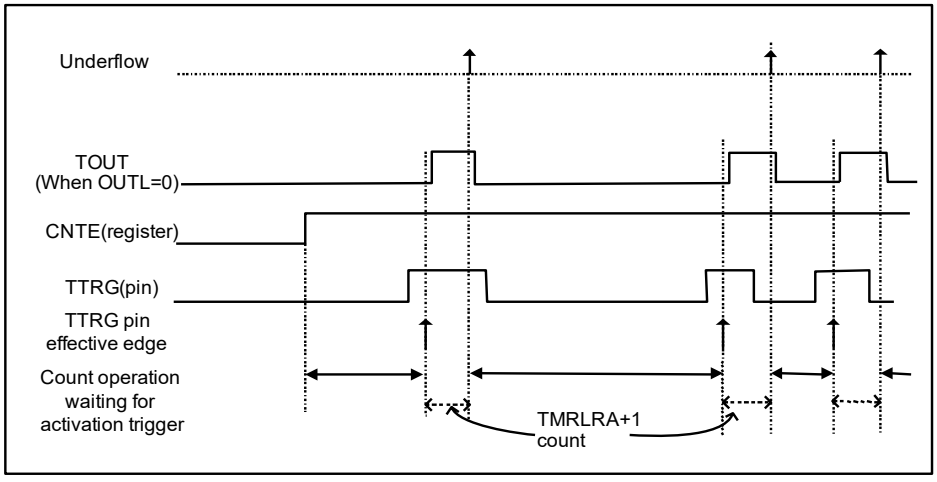
Figure 20-10. Single One-shot Operation



Details of Underflow operation
(When the trigger input and rising edge trigger are selected)



Single one-shot timer
(GATE="0": When the trigger input and rising edge trigger are selected)



Single one-shot timer (GATE="1": gate input, TRGM:H input interval count)

20.5.3.2 Single Reload Operation

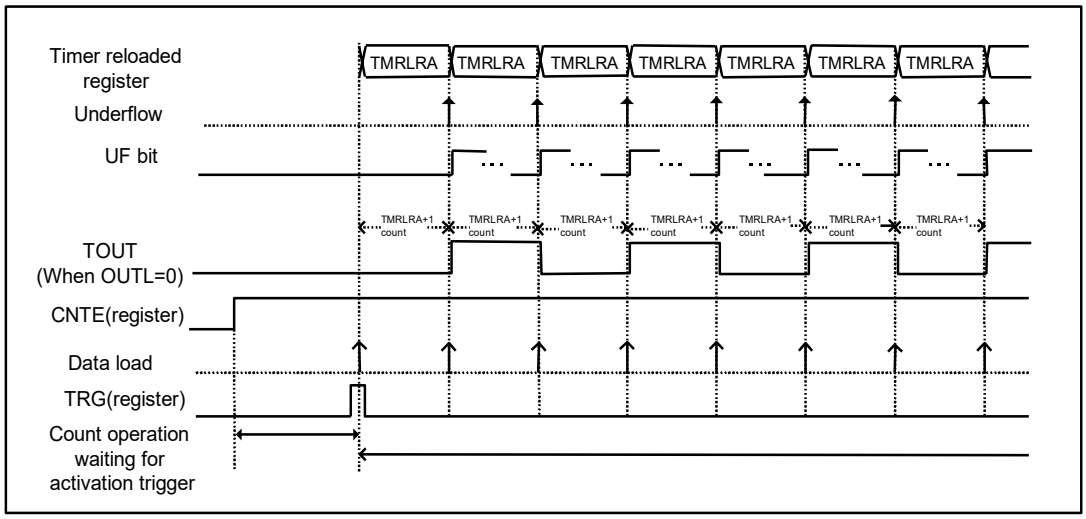
The single reload operation is shown below.

When bit15, bit14:MOD[1:0] of the TMCSR register =00, and bit4:RELD of the TMCSR register =1, the single reload operation will be performed.

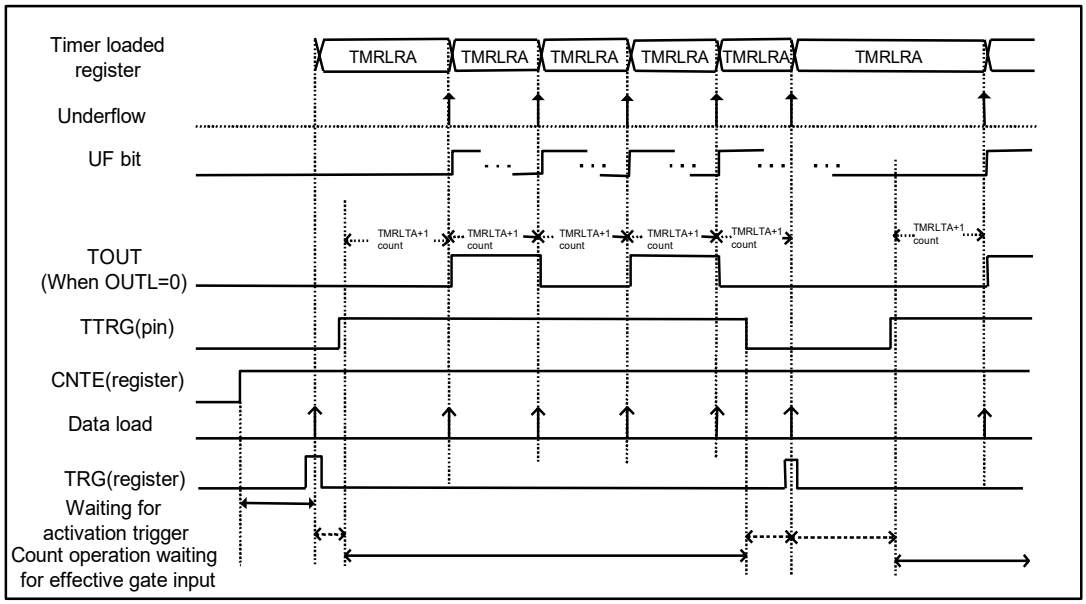
In single reload operation, a value will be loaded from TMRLRA to the timer by trigger input, a down count (decrementing the count) will start. When an underflow occurs, the value is reloaded from TMRLRA again and the down count operation continues. The value of TMRLRA represents the time the timer will 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 20-11. Single Reload Operation



Single reload function (GATE="0": trigger input)



Single reload function (GATE="1": gate input, TRGM: H input interval count)

20.5.3.3 Dual One-shot Operation

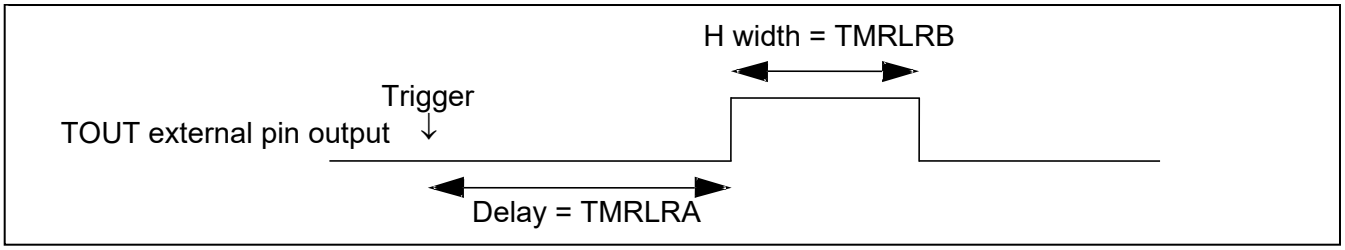
The dual one-shot operation is shown below.

When bit15, bit14:MOD[1:0] of the TMCSR register =01, and bit4:RELD of the TMCSR register =0, 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, the loaded values decrements the counter for each load. The counter will stop by the second underflow.

When bit5:OUTL=0 of the TMCSR register, 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 20-12. TOUT Pulse Width



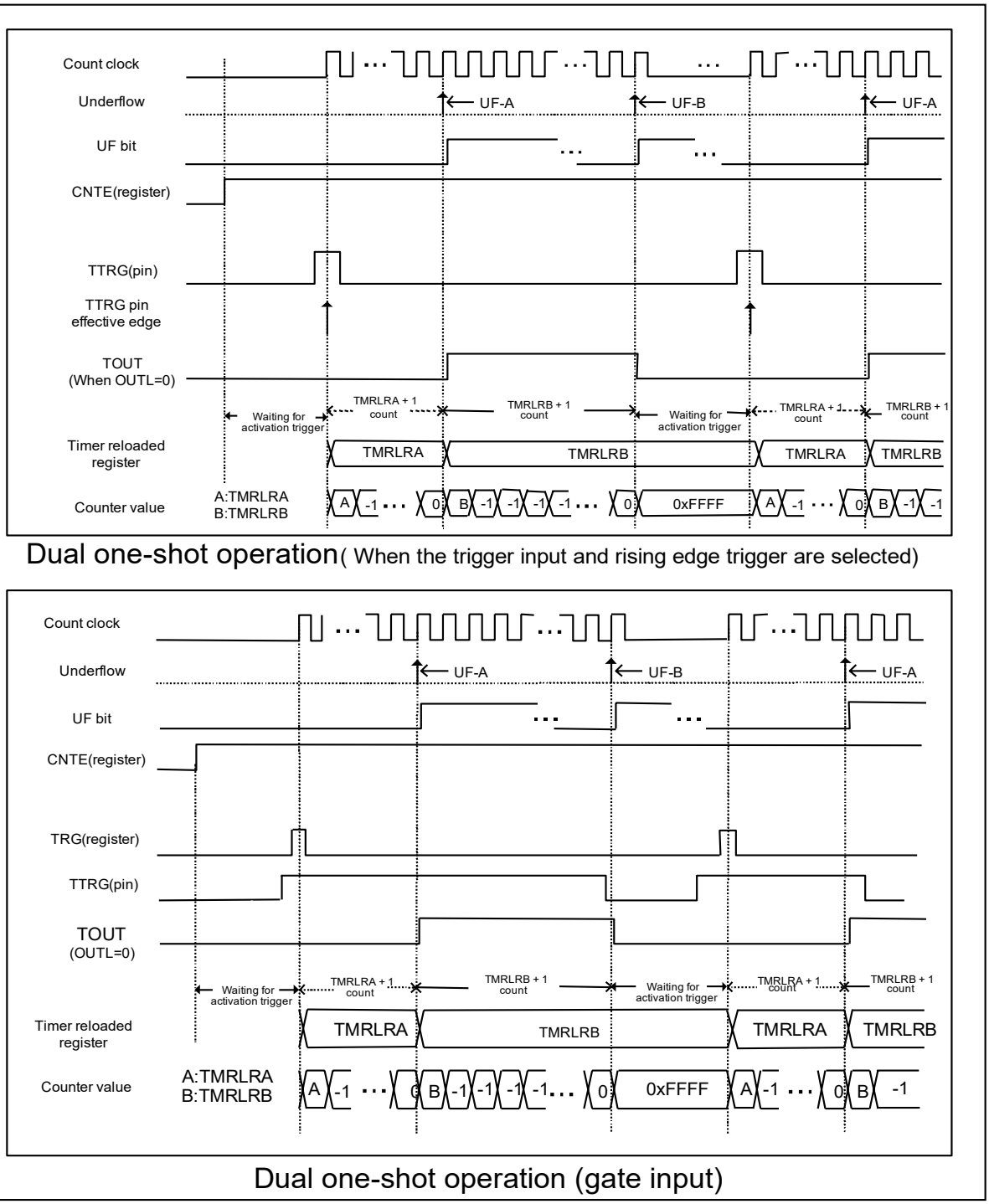
When the first underflow occurs (UF-A), the following operation will take place.

- 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 20-13. Dual One-shot Operation



20.5.3.4 Dual Reload Operation

The dual one-shot operation is shown below.

When bit15, bit14:MOD[1:0] of the TMCSR register =01, and bit4:RELD of the TMCSR register =1, 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=0 of the TMCSR register, 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.

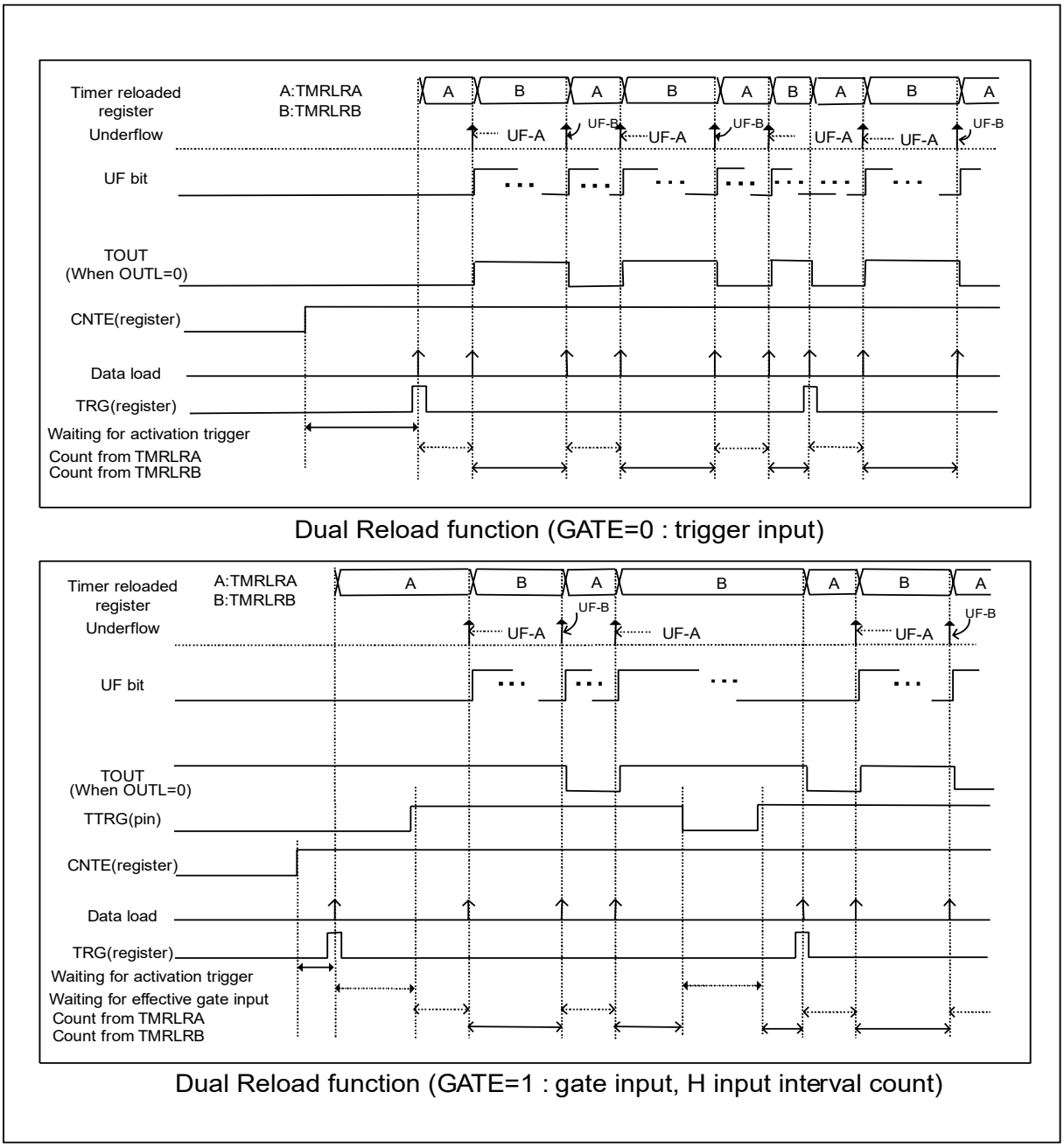
If an underflow (UF-A) occurs at the down count after loading a value from the TMRLRA, the following operation will be performed.

- 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 20-14. Dual Reload Operation



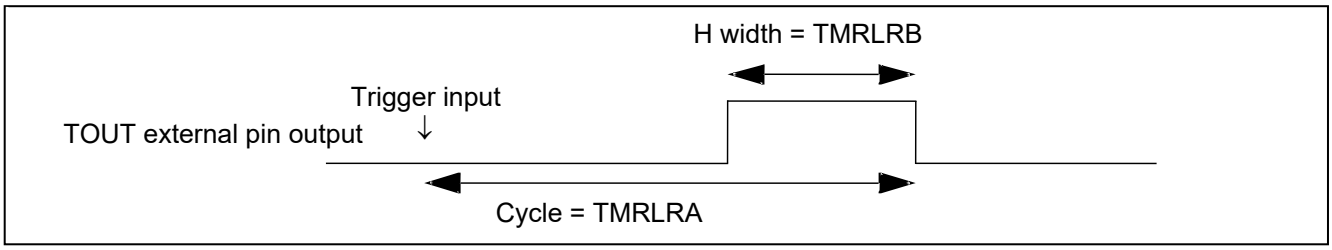
20.5.3.5 Compare One-shot Operation

The compare one-shot operation is shown below.

When bit15, bit14:MOD[1:0] of the TMCSR register =10, and bit4:RELD of the TMCSR register =0, 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 a compare matched (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 20-15. 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.

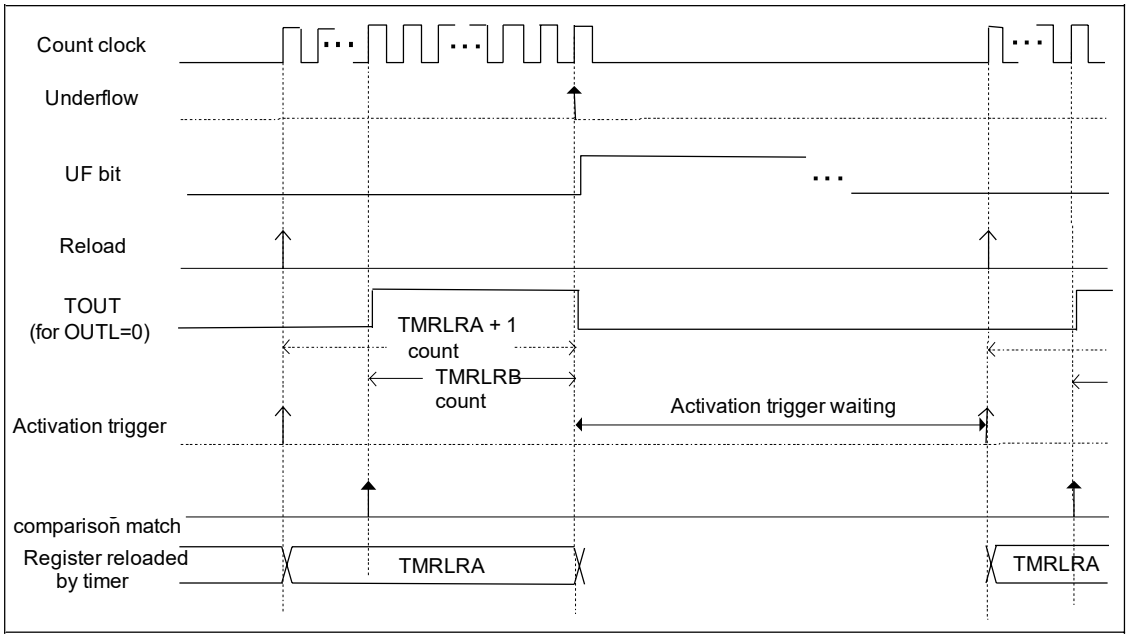
- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE=1 of TMCSR register), an interrupt occurs.
- Initializes TOUT output.
- The timer stops with 0xFFFF.
- Timer is waiting for an activation trigger.

The operation of the compare function changes depending on the setting relation between TMRLRA and TMRLRB.

Figure 20-16. Compare One-shot Operation (1 / 2)

• Sets TMRLRB < TMRLRA

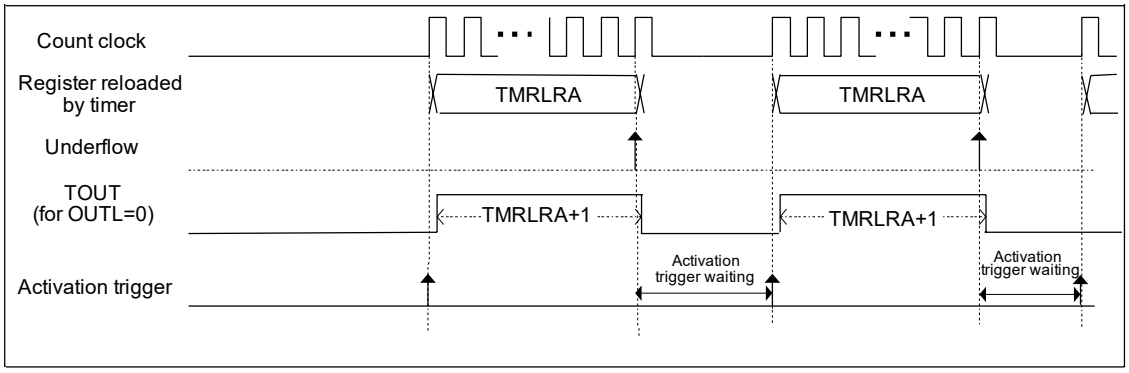
When the register relation is as described above, the TOUT output is the L level until TMR and TMRLRB match after loading to the timer. When down counting from the comparison match (TMR = TMRLRB), the level is H until the TOUT output is inverted and an underflow occurs. When an underflow occurs the TOUT 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").

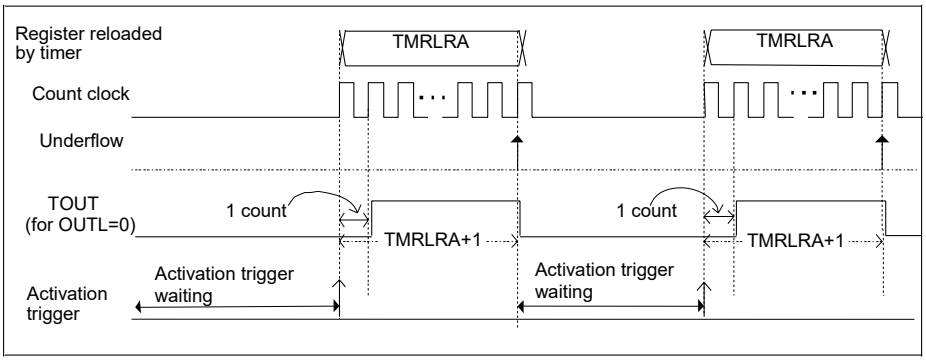


Compare one-shot function (TMRLRB > TMRLRA)

Figure 20-17. Compare One-shot Operation (2 / 2)

• Sets TMRLRB = TMRLRA

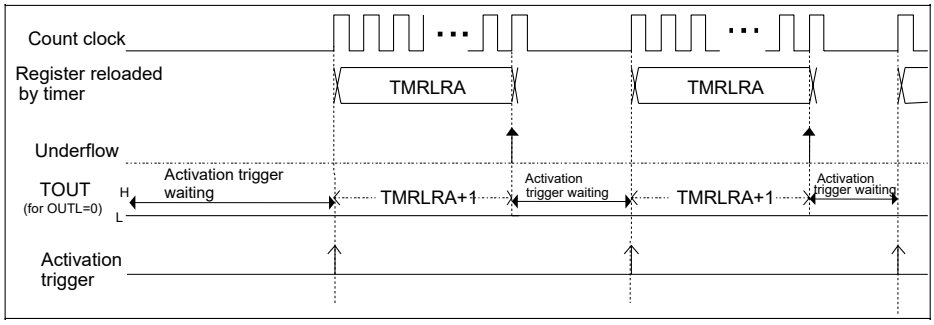
When the register relation is as described above, TMRLRB will become bigger than TMR after 1 count. Thus the TOUT output is the L level for 1 down count and then the H level until an underflow occurs. When an underflow occurs, the timer will turn into the activation trigger waiting state and the TOUT 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 TMRLRB is always smaller than TMR. The level will remain to be L even when an underflow occurs (for OUTL="0").



Compare one-shot function (TMRLRB="0")

20.5.3.6 Compare Reload Operation

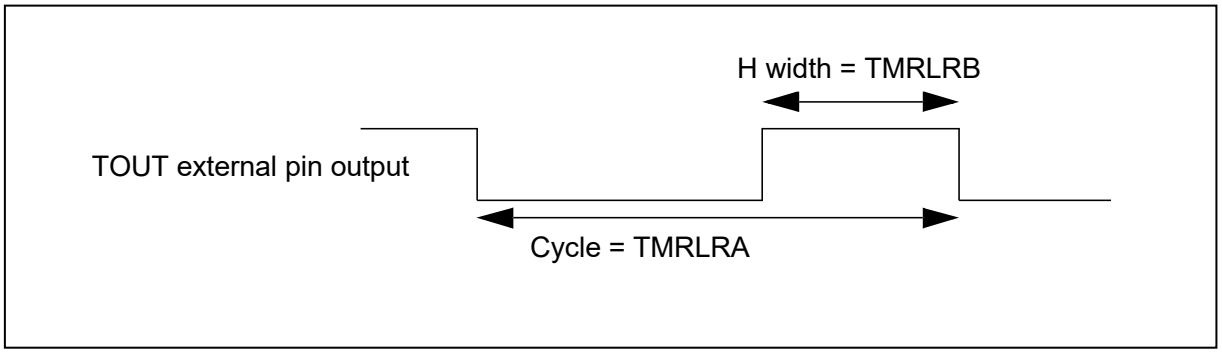
The compare reload operation is shown below.

When bit15, bit14:MOD[1:0] of the TMCSR register =10, and bit4:RELD of the TMCSR register =1, the timer compares a counter value (TMR) to the value of TMRLRB for each down count and if a compare matched (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 20-18. 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.

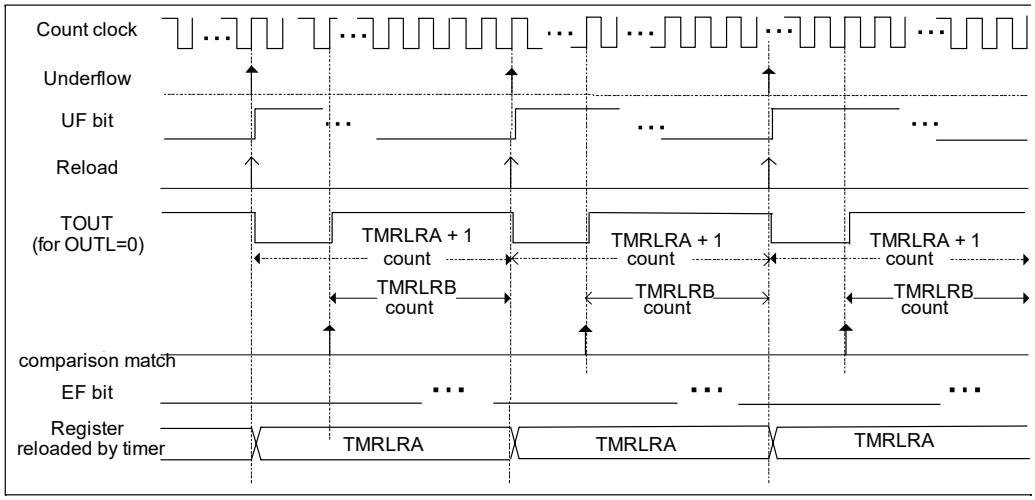
- Sets bit2:UF bit of the TMCSR register.
- When interrupts are enabled (bit3:INTE=1 of TMCSR register), an interrupt occurs.
- Initializes TOUT output.
- Reloads a value from TMRLRA.
- The timer continues to count.

The operation of a compare feature depends on the relationship between TMRLRA and TMRLRB.

Figure 20-19. Compare Reload Operation (1 / 2)

• Sets TMRLRB < TMRLRA

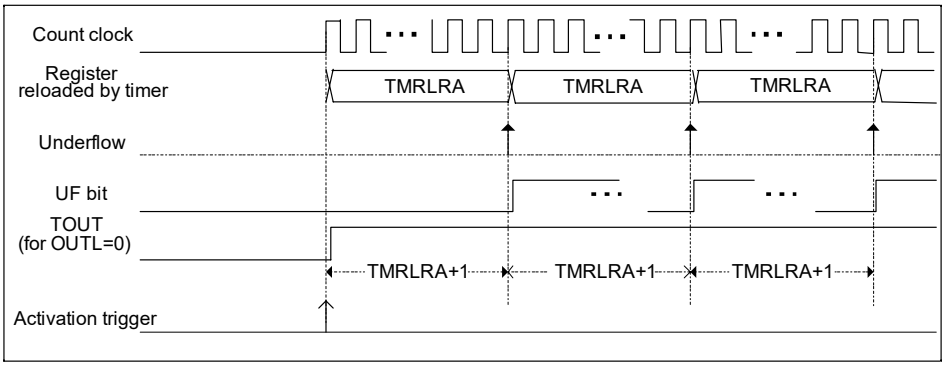
When the register relation is as described above, the TOUT output is the L level until TMR and TMRLRB match after loading to the timer. When down counting from the comparison match (TMR=TMRLRB), the level is H until the TOUT output is inverted and an underflow occurs. When an underflow occurs the TOUT 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").

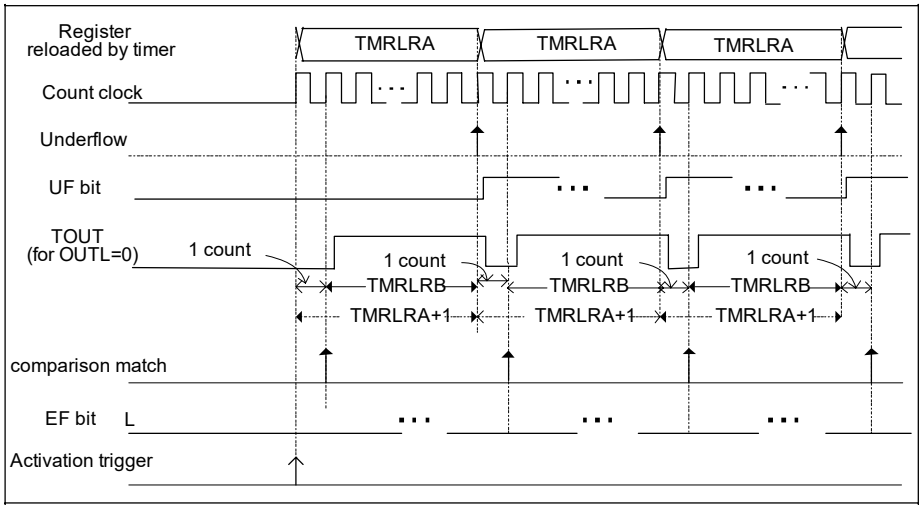


Compare reload function (TMRLRB > TMRLRA) trigger input

Figure 20-20. Compare Reload Operation (2 / 2)

• Sets TMRLRB = TMRLRA

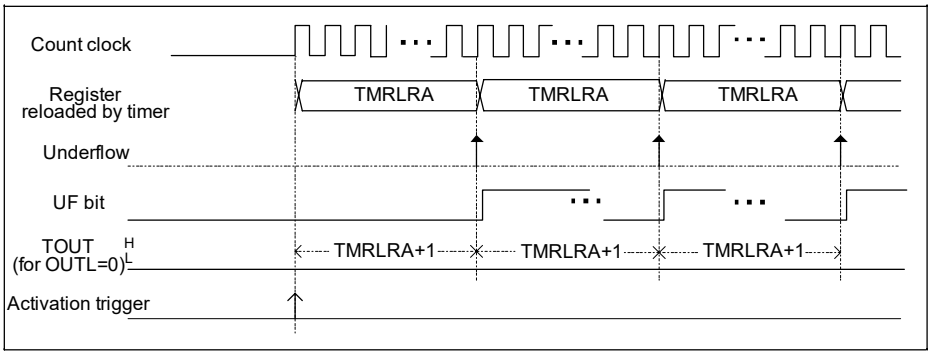
When the register relation is as described above, 1 count up after loading to the timer makes TMRLRB become bigger than TMR. Thus the TOUT output is the L level for 1 down count and then the H level until an underflow occurs. When an underflow occurs, the timer will reload from TMRLRA and continue counting 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 TMRLRB is smaller than TMR. The level will remain to be L even when an underflow occurs.



Compare reload function (TMRLRB = "0") trigger input

20.5.3.7 Capture Mode

The capture mode is shown below.

When bit15, bit14:MOD[1:0] of the TMCSR register =11, 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 TTRG input as the gate input (when bit8:GATE=1 of the TMCSR register), generate a retrigger by bit0:TRG of the TMCSR register.

In modes other than capture mode, a capture will not be performed at a retrigger. The EF bit interrupt will also not be generated.

The timer operation and the TOUT output will be the same for the single one-shot feature and the single reload feature.

Note:

TOUT is not initialized in the one shot mode at retrigger.

Figure 20-21. Operation of Capture

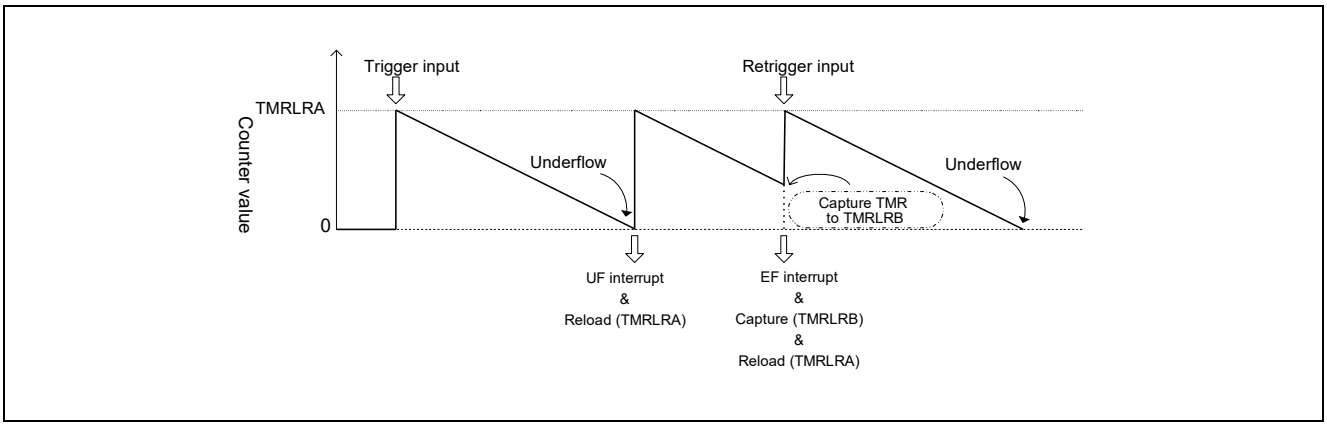


Figure 20-22. Flowchart of Trigger Input Features in Interval Timer Mode

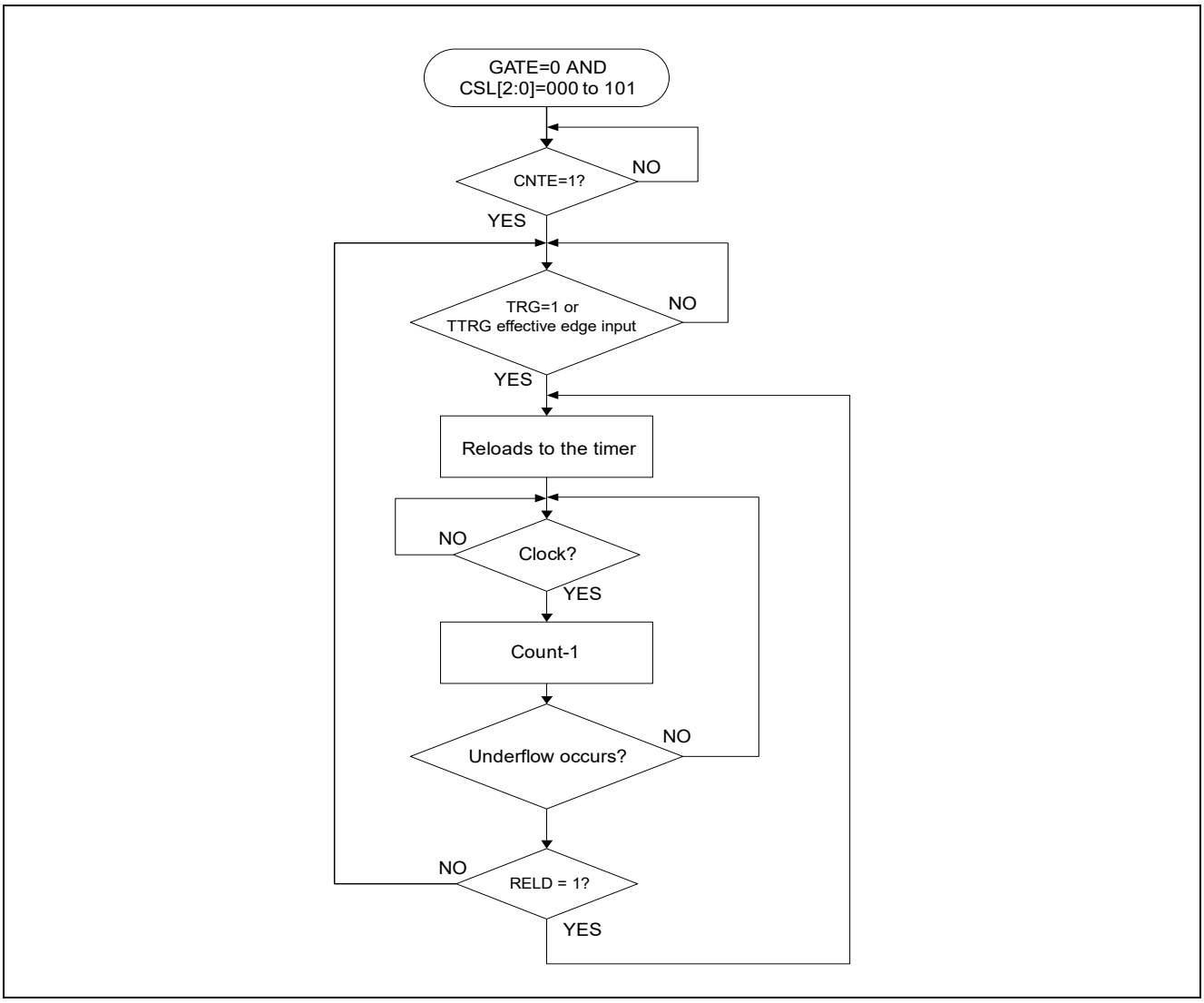
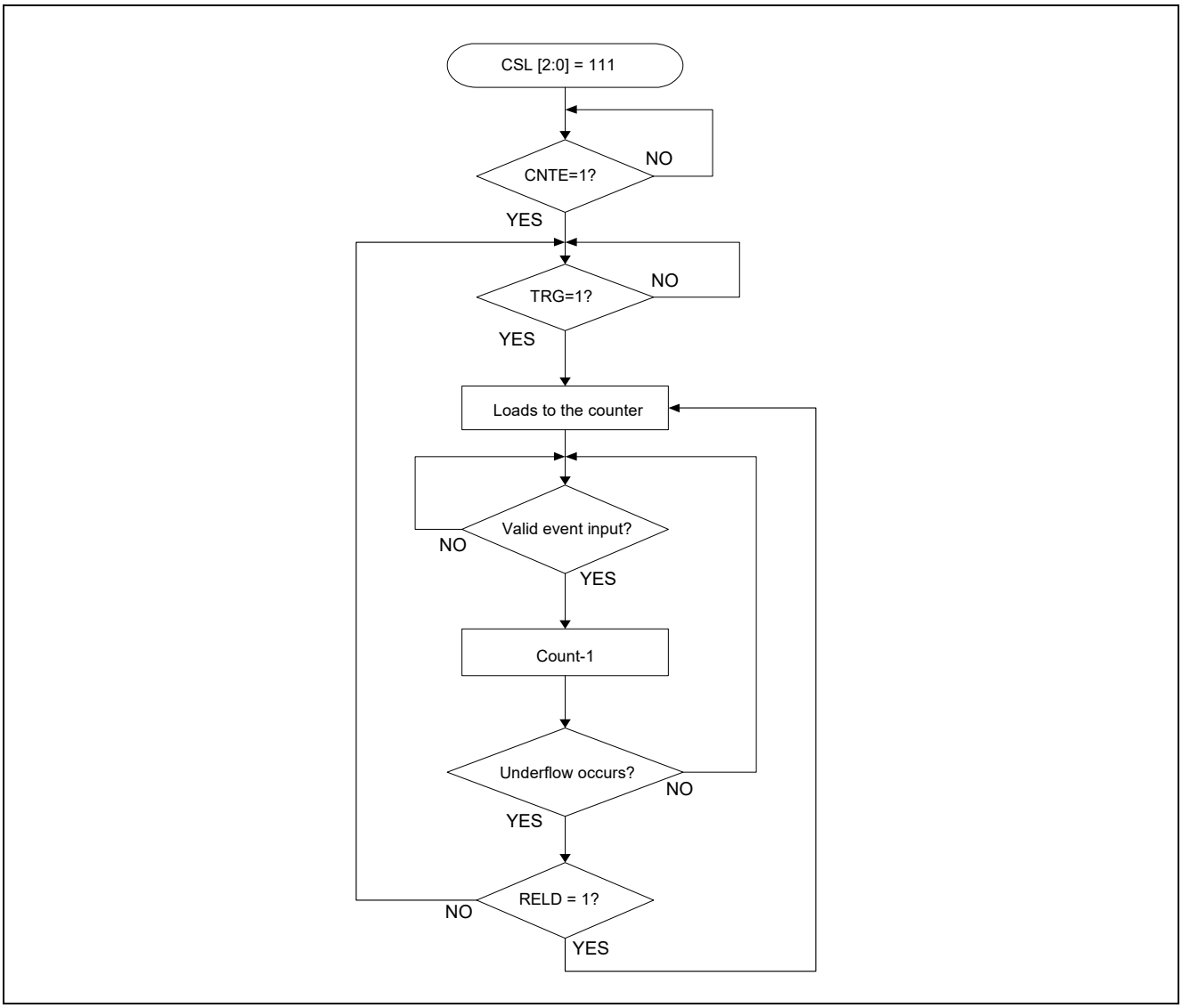


Figure 20-23. Flowchart in Event Counter Mode



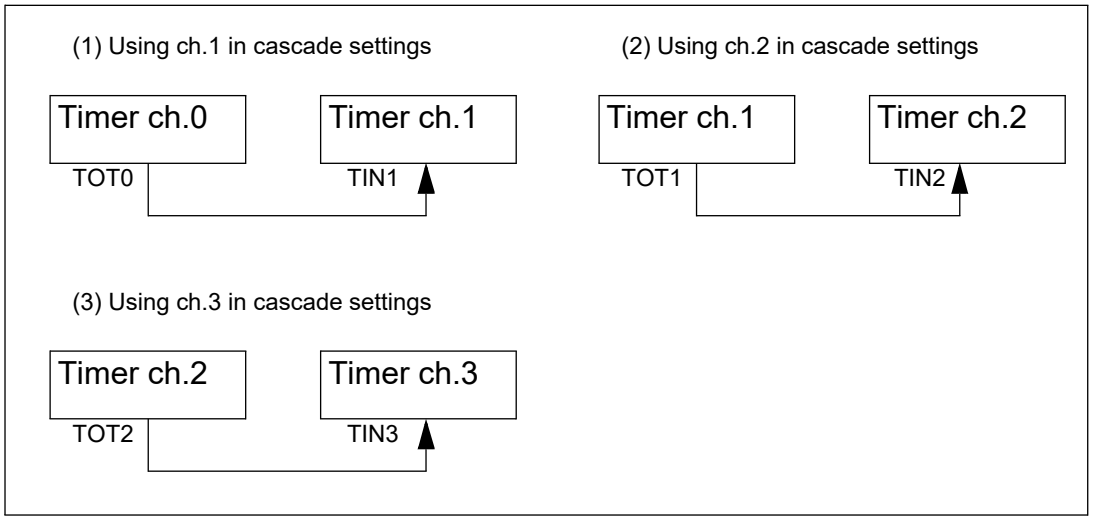
20.5.4 Cascade Input

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's ch.0 output (TOUT0) for the input for ch.1 (TTRG1), ch.1 output (TOUT1) for the input for ch.2 (TTRG2), and ch.2 output (TOUT2) for the input for ch.3 (TTRG3).

ch.4 to ch.6 are also similar to the above.

Figure 20-24. Timer Input/Output in Cascade Input Configuration



20.5.5 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:

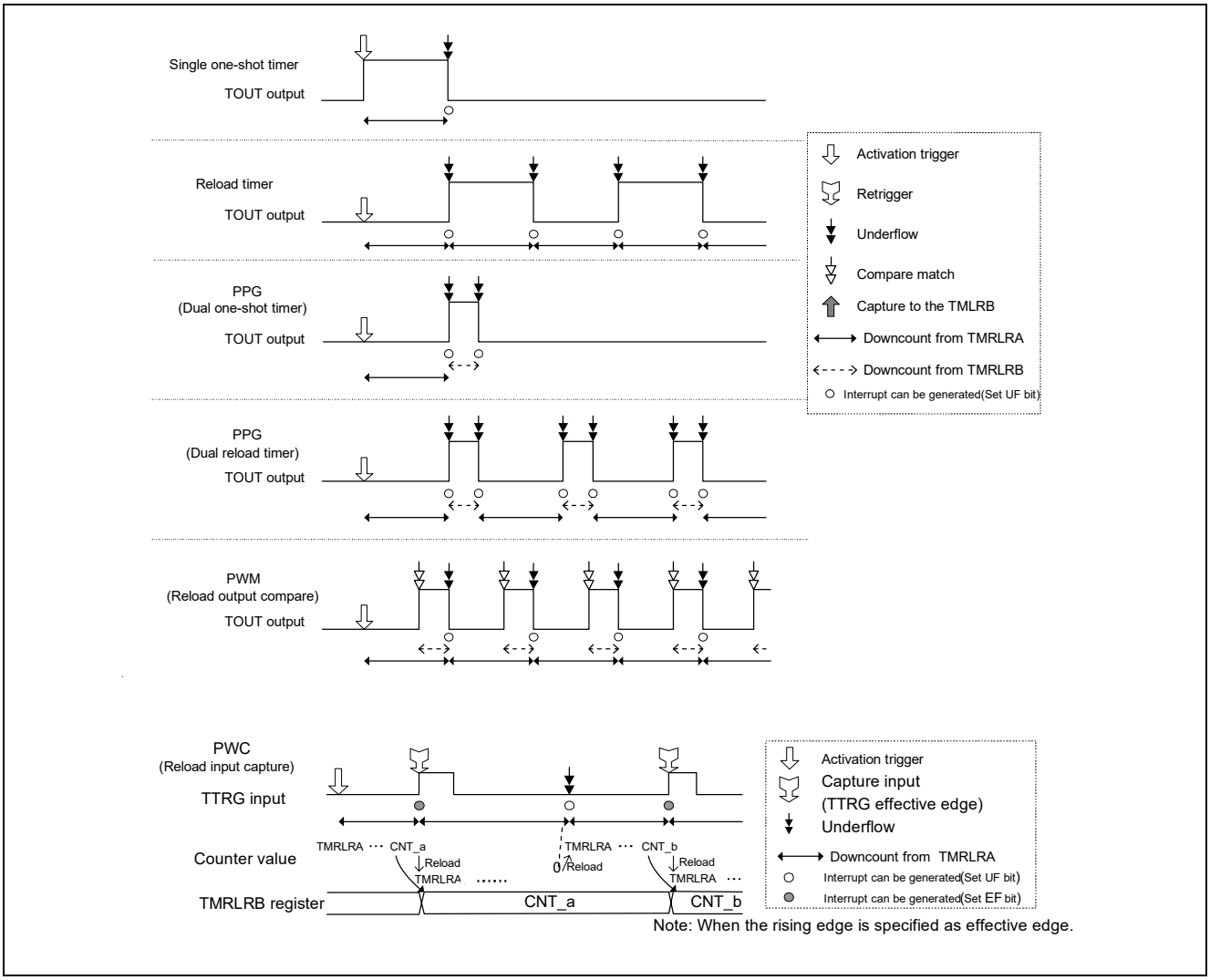
1. Setting flag by the timer operation
2. Writing to a register for a clear of flag to the UF bit/EF bit

20.6 Application Note

An application note is shown below.

This section shows the typical functions which can be realized with this timer.

Figure 20-25. Example



Note: When the rising edge is specified as effective edge.

Following are some configurations for use of example figure above.

Table 20-6. 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	-

20.6.1 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 TTRG 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 TTRG 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]: TTRG 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 TTRG 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]: TTRG 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

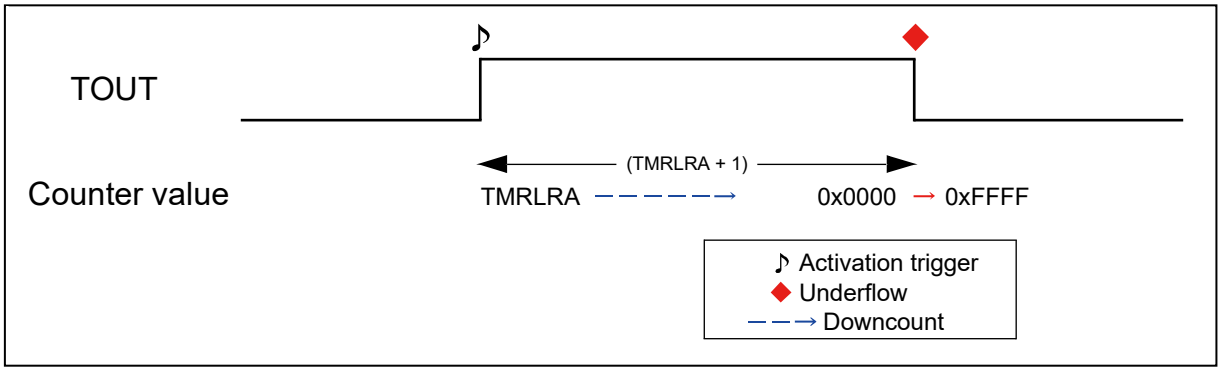
[3]: OUT output polarity setting
OUTL= 0------Initial value L=> Count starts H=> Underflow occurs L
OUTL= 1------Initial value H=> Count starts L=> Underflow occurs H

[4]: Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer:

1. Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TTRG pin)
2. Input an effective level when you use TTRG pin input as the gate input

Figure 20-26. Example of Operation (OUTL = 0)



20.6.2 Reload Timer

The reload time 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 going on from the activation trigger to the occurrence of the first underflow, then the output will be inverted to the "H level" at the timing of the occurrence of the first underflow, inverting the outputs whenever an underflow occurs. 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 TTRG 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 TTRG 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]: TTRG effective level setting
TRGM[1:0]=x0------Count only for TTRG=L input interval
TRGM[1:0]=x1------Count only for TTRG=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 TTRG 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]: TTRG 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

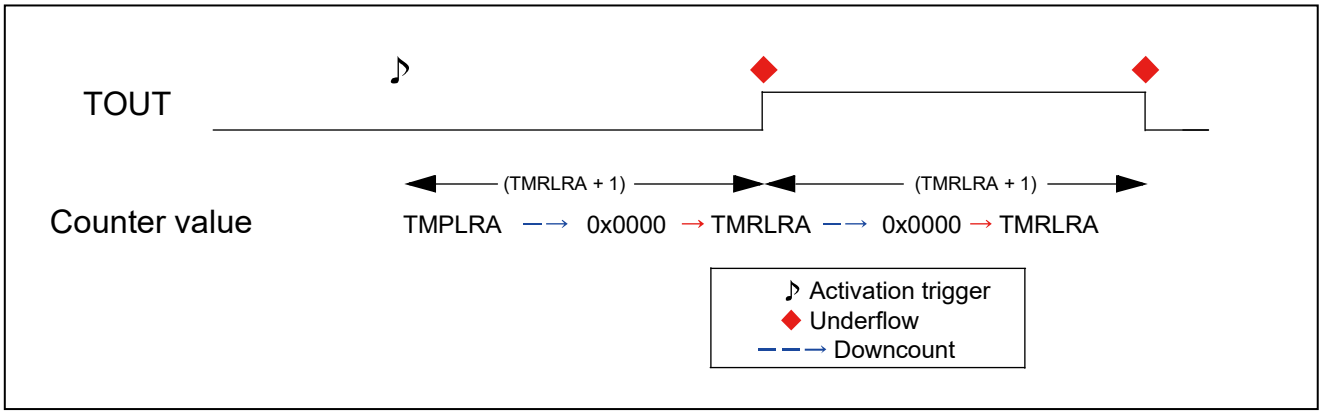
[3]: OUT output polarity setting
OUTL= 0------Initial value L=> Count starts L=> Invert whenever an underflow occurs
OUTL= 1------Initial value H=> Count starts H=> Invert whenever an underflow occurs

[4]: Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer:

1. Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TTRG pin)
2. Input an effective level when you use TTRG pin input as the gate input

Figure 20-27. Example of Operation (OUTL=0)



20.6.3 PPG

PPG is shown below.

PPG is the feature 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 the operation switches to load the value from TMRLRB and executes a down count when an underflow occurs.

When RELD=0, "Activation trigger => TMRLRA load => Down count=> Underflow => TMRLRB load => Down count => Underflow," then stops the down count.

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 TTRG 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 TTRG 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]: TTRG effective level setting
TRGM[1:0]= x0------Count only for TTRG=L input interval
TRGM[1:0]= x1------Count only for TTRG=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 TTRG 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]: TTRG 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

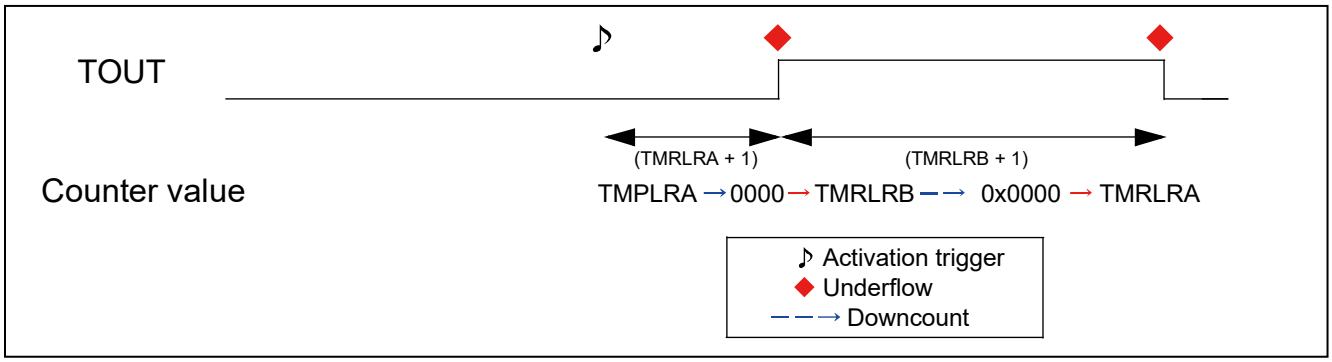
[4]: Reload setting when an underflow occurs
RELD=0------One-shot mode
RELD=1------Reload mode

[5]: Interrupt request enable setting
INTE=0------Interrupt disabled
INTE=1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer:

1. Input an activation trigger (a write of "1" to the TRG bit or an input of effective external edge from TTRG pin)
2. Input an effective level when you use TTRG pin input as the gate input

Figure 20-28. Example of Operation (OUTL=0)



20.6.4 PWM

PWM is shown below.

PWM is the feature 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 TTRG 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 TTRG 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]: TTRG 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 TTRG 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]: TTRG 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

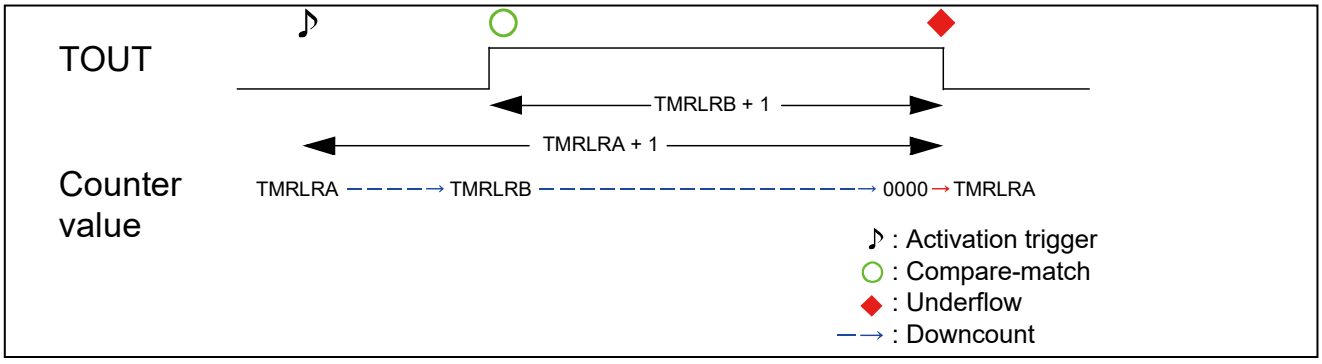
[5]: Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

[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:

1. Input an activation trigger (a write of "1" to TRG bit or an input of effective external edge from TTRG pin).
2. Input an effective level when you use TTRG pin input as the gate input.

Figure 20-29. Example of Operation (OUTL=0)



20.6.5 PWC

PWC is shown below.

PWC is the feature 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)
S: Use at timer activation
-: Does not influence operation

[1]: TTRG 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

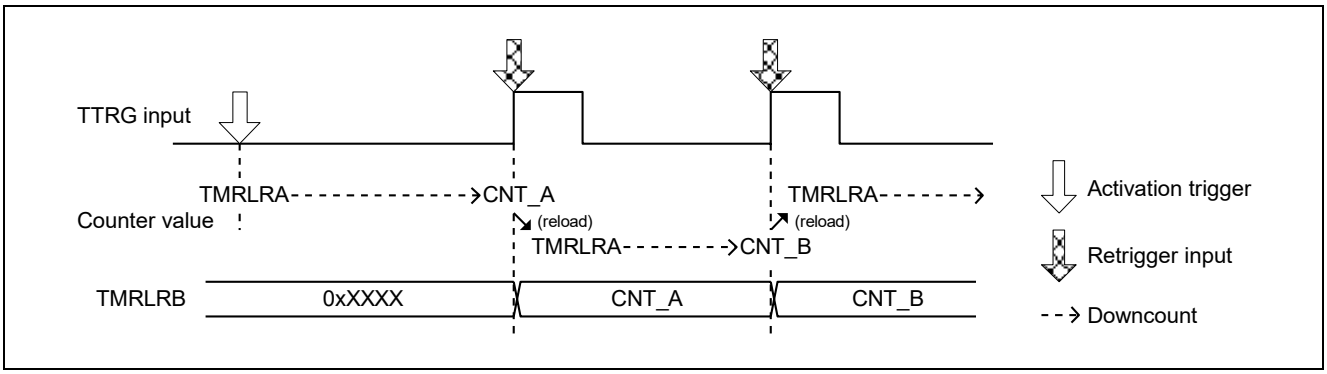
[5]: Interrupt request enable setting
INTE= 0------Interrupt disabled
INTE= 1------Interrupt enabled

[Timer activation] Follow the steps below to activate the timer:

1. Input an activation trigger (a write of "1" to TRG bit or an input of effective external edge from TTRG 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) × Peripheral clock (PCLK) cycle × Division ratio set with CSL

Figure 20-30. Example of Operation (TRGM=01)



21. Free-run Timer

This chapter explains the free-run timer.

21.1 Overview

21.2 Features

21.3 Configuration

21.4 Registers

21.5 Operation

21.6 Setting

21.7 Q&A

21.8 Sample Program

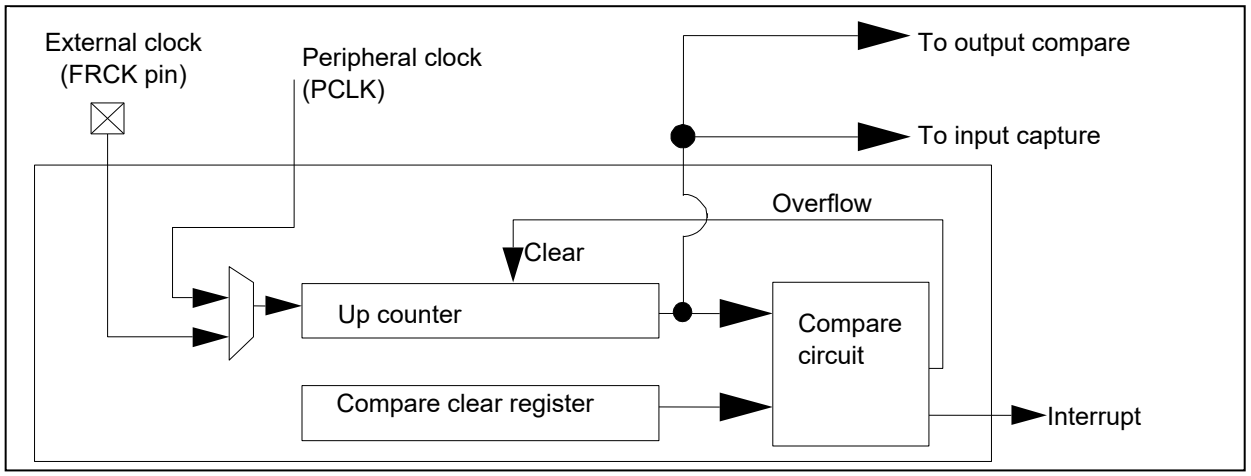
21.9 Notes

21.1 Overview

This section explains the overview of the free-run timer.

The free-run timer consists of a 32-bit up counter and a control circuit. The free-run timer can be used in combination with input capture and output compare.

Figure 21-1. Block Diagram (Overview)



21.2 Features

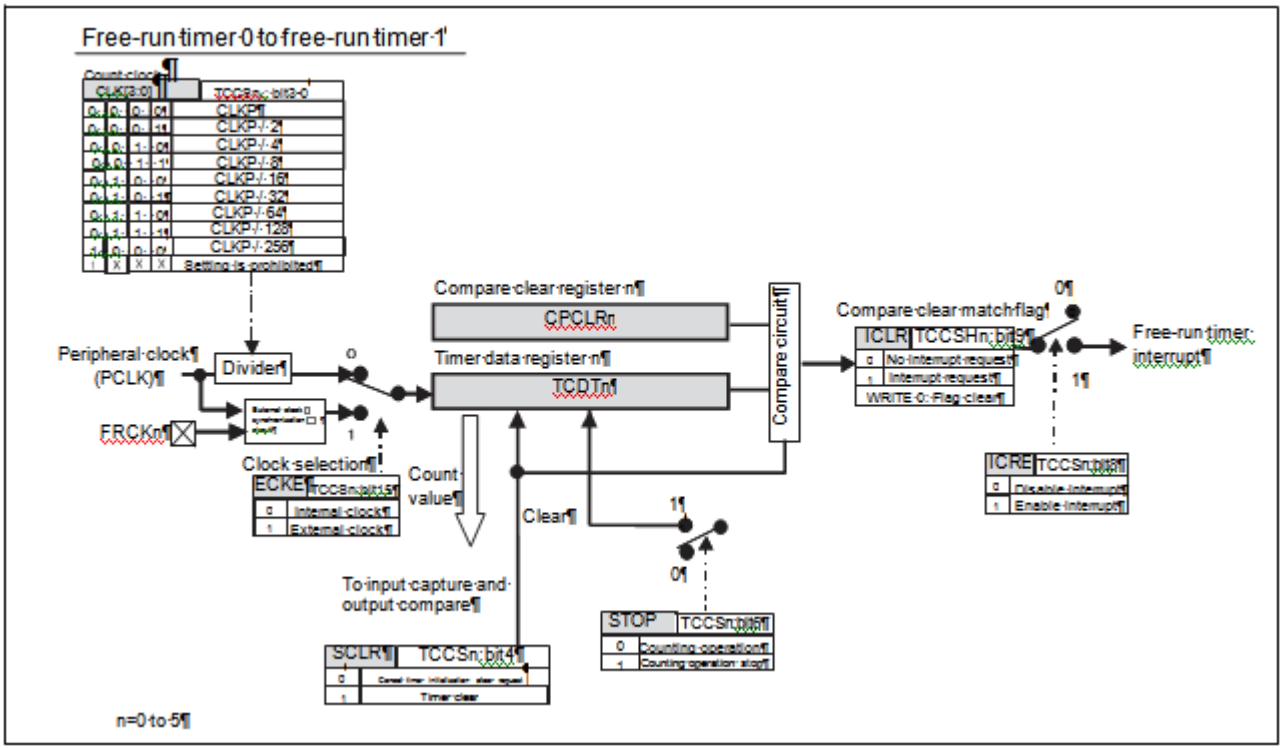
This section shows the features of the free-run timer.

- Format: 32-bit up counter
- Number of units: 6
- Clock source: One of 9 internal clocks (peripheral clock (PCLK)/1, /2, /4, /8, /16, /32, /64, /128, /256) or one of two external clocks (FRCK)
- Count clear factors :
  - ☐ Software
  - ☐ Reset
  - ☐ Compare match (count value of the free-run timer matches the compare clear register)
- Operation start/stop: The operation can be started and stopped by software.
- Interrupt: Compare clear interrupt
- Count value: Read/write enabled (writing is only enabled while counting is inactive)
- The 32-bit free-run timer consists of a 32-bit up counter, control register, 32-bit compare clear register, and prescaler.
- A compare clear interrupt will be generated when a compare clear register matches the 32-bit free-run timer upon comparison of the two.
- If there is a compare match with reset, software clear or compare clear register, the counter value will be reset to "00000000_H".
- It is used as the reference count for output compare and input capture.

21.3 Configuration

This section explains configuration of the free-run timer.

Figure 21-2. Configuration Diagram (Detailed)



21.4 Registers

This section explains the registers of the free-run timer.

Base Address (Base_addr) · External Pin Table

Channel	Base_addr	External Pin
		FRCK
0	0x0240	FRCK0_0/FRCK0_1
1	0x024C	FRCK1_0/FRCK1_1
2	0x0FA0	FRCK2_0/FRCK2_1
3	0x0FAC	FRCK3_0/FRCK3_1
4	0x0FB8	FRCK4_0
5	0x0FC4	FRCK5_0

Table 21-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0240	CPCLR0				Compare clear register 0
0x0244	TCDT0				Timer data register 0
0x0248	TCCSH0	TCCSL0	Reserved		Timer control register (Upper Bit) 0 Timer control register (Lower Bit) 0
0x024C	CPCLR1				Compare clear register 1
0x0250	TCDT1				Timer data register 1
0x0254	TCCSH1	TCCSL1	Reserved		Timer control register (Upper Bit) 1 Timer control register (Lower Bit) 1
0x0FA0	CPCLR2				Compare clear register 2
0x0FA4	TCDT2				Timer data register 2
0x0FA8	TCCSH2	TCCSL2	Reserved		Timer control register (Upper Bit) 2 Timer control register (Lower Bit) 2

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0FAC	CPCLR3				Compare clear register 3
0x0FB0	TCDT3				Timer data register 3
0x0FB4	TCCSH3	TCCSL3	Reserved		Timer control register (Upper Bit) 3 Timer control register (Lower Bit) 3
0x0FB8	CPCLR4				Compare clear register 4
0x0FBC	TCDT4				Timer data register 4
0x0FC0	TCCSH4	TCCSL4	Reserved		Timer control register (Upper Bit) 4 Timer control register (Lower Bit) 4
0x0FC4	CPCLR5				Compare clear register 5
0x0FC8	TCDT5				Timer data register 5
0x0FCC	TCCSH5	TCCSL5	Reserved		Timer control register (Upper Bit) 5 Timer control register (Lower Bit) 5

21.4.1 Timer Control Register (Upper Bit): TCCSH

The bit configuration of the timer control register (Upper bit) is shown.

This register is used to control the operation of the free-run timer.

TCCSH0-5 (Free-run timer 0-5): Address Base_addr+08$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ECKE	-	-	-	-	-	ICLR	ICRE
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R(RM1),W	R/W

[bit15] ECKE: Clock selection

ECKE	Count Clock Selection
0	Internal clock
1	External clock (FRCK)

- When this bit is set to "0": Internal clock is selected. To select the count clock frequency, you will also need to select the clock frequency selection bits (CLK3 to CLK0:bit3 to bit0) of the TCCSL register.
- When this bit is set to "1": External clock is selected. The external clock is input from the "FRCK" pin. Therefore, enable external clock input by writing "0" to the bit of the port direction register (DDR) corresponding to the FRCK input pin and writing "0" to the bit of the corresponding port function register (PFR) to switch to port input state. If external clock is selected by the ECKE bit, clock count will detect both edges. Set the pulse width of the external clock to 4/F_{PCLK} or more.

Note:

The setting change for the count clock selection bit must be performed while other peripheral modules using the free-run timer output (output compare and input capture) are inactive.

[bit14 to bit10] - : Undefined

The read value is always "0". This does not affect the writing operation.

[bit9] ICLR: Compare clear interrupt flag

ICLR	State	
	Read	Write
0	No compare clear match	Clear the flag (ICLR)
1	Compare clear match	No effect on operation

- This bit will be set to "1" when the compare clear value matches the 32-bit free-run timer value.

[bit8] ICRE: Compare clear interrupt request enabled

ICRE	Operation
0	Interrupt disabled
1	Interrupt enabled

- When the ICRE bit and compare clear interrupt flag bit (ICLR) are set to "1", an interrupt request for CPU will be generated.

21.4.2 Timer Control Register (Lower Bit): TCCSL

The bit configuration of timer control register (Lower bit) is shown.

This register is used to control the operation of the free-run timer.

TCCSL0-5 (Free-run timer 0-5): Address Base_addr+09$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	STOP	-	SCLR	CLK3	CLK2	CLK1	CLK0
Initial value	0	1	0	0	0	0	0	0
Attribute	R0,WX	R/W	R0,WX	R0,W	R/W	R/W	R/W	R/W

[bit7] - : Undefined

The read value is always "0". This does not affect the writing operation.

[bit6] STOP: Timer enabled

STOP	Operation
0	Count enabled (operation)
1	Count disabled (stop)

- The STOP bit is used to start/stop counting of the 32-bit free-run timer.
- When the STOP bit is "0": Counter of the 32-bit free-run timer is started.
- When the STOP bit is "1": Counter of the 32-bit free-run timer is stopped.

Note:

If output compare is in use, the output compare operation will stop when the free-run timer stops.

[bit5] - : Undefined

The read value is always "0". This does not affect the writing operation.

[bit4] SCLR: Timer clear

SCLR	State	
	Read	Write
0	The read value is always "0".	Writing "0" has no effect on operation.
1		Clears the free-run timer.

- When this bit is set to "1", the count value of the free-run timer is cleared to "00000000_H". The prescaler within the macro is also cleared at this time.
- The value read out is always "0".

Note:

If you set this bit to "1", timer clear will be performed at the next internal clock timing.

[bit3 to bit0] CLK3 to CLK0 : Clock frequency selection (when internal clock is selected)

CLK3	CLK2	CLK1	CLK0	Clock Frequency Selection (F_{PCLK}: Peripheral Clock (PCLK))				
				Count clock	F_{PCLK} =16 MHz	F_{PCLK} =8 MHz	F_{PCLK} =4 MHz	F_{PCLK} =1 MHz
0	0	0	0	$1/F_{PCLK}$	62.5ns	125ns	0.25 µs	1 µs
0	0	0	1	$2/F_{PCLK}$	125ns	0.25 µs	0.5 µs	2 µs
0	0	1	0	$4/F_{PCLK}$	0.25 µs	0.5 µs	1 µs	4 µs
0	0	1	1	$8/F_{PCLK}$	0.5 µs	1 µs	2 µs	8 µs
0	1	0	0	$16/F_{PCLK}$	1 µs	2 µs	4 µs	16 µs
0	1	0	1	$32/F_{PCLK}$	2 µs	4 µs	8 µs	32 µs
0	1	1	0	$64/F_{PCLK}$	4 µs	8 µs	16 µs	64 µs
0	1	1	1	$128/F_{PCLK}$	8 µs	16 µs	32 µs	128 µs
1	0	0	0	$256/F_{PCLK}$	16 µs	32 µs	64 µs	256 µs
Other settings				-	Prohibit			

- The frequency is changed at the same time as the setting change to the clock frequency selection bit. If internal clock is selected as the count clock of the free-run timer (clock selection bit (ECKE= 0)), change the setting while other peripheral modules (output compare and input capture) using the free-run timer output are inactive.
- When the free-run timer is used as compare data for the output compare, the free-run timer clock frequency cannot be set as CLK[3:0]= 0000_B.

21.4.3 Compare Clear Register: CPCLR

The bit configuration of the compare clear register (CPCLR) is shown.

Compare clear register is a 32-bit register to be used for comparison with the free-run timer.

CPCLR0-5 (Free-run timer 0-5): Address Base_Addr+00_H (Access: Word)

	bit31							bit0
	CL[31:0]							
Initial value	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
Attribute				R/W				

[bit31 to bit0] CL[31:0]: Compare clear

- The compare clear register is used for comparison with the count value of the 32-bit free-run timer. If the count value of this register matches that of the free-run timer, the 32-bit free-run timer will be reset to "00000000_H" and an interrupt will be generated when the value set to this register matches the counter value. However, the value needs to be written while the timer is inactive (the STOP bit of timer state control register lower (TCCSL) = 1).
- Writing to this register during operation will have no meaning.
- When accessing this register, use a word access instruction.

21.4.4 Timer Data Register: TCDT

The bit configuration of the timer data register (TCDT) is shown.

The timer data register is used for reading the count value of the 32-bit free-run timer.

TCDT0-5 (Free-run timer 0-5): Address Base_addr+04H (Access: Word)

	bit31							bit0
	T[31:0]							
Initial value	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Attribute				R,W				

- The count value of the 32-bit free-run timer can be read by reading the timer data register.
- Timer value can be written to the free-run timer by writing to the timer data register. Always write to this register while the free-run timer is inactive (timer control register lower (STOP of TCCSL = 1)).
- When accessing this register, use a word access instruction.
- The 32-bit free-run timer will be initialized as soon as any of the following occurs.
  - ☐ Reset
  - ☐ The Clear bit (SCLR = 1) of the timer state control register (TCCSL)
  - ☐ The timer count value matches the compare clear register
- Writing to this register while it is in operation will have no meaning.

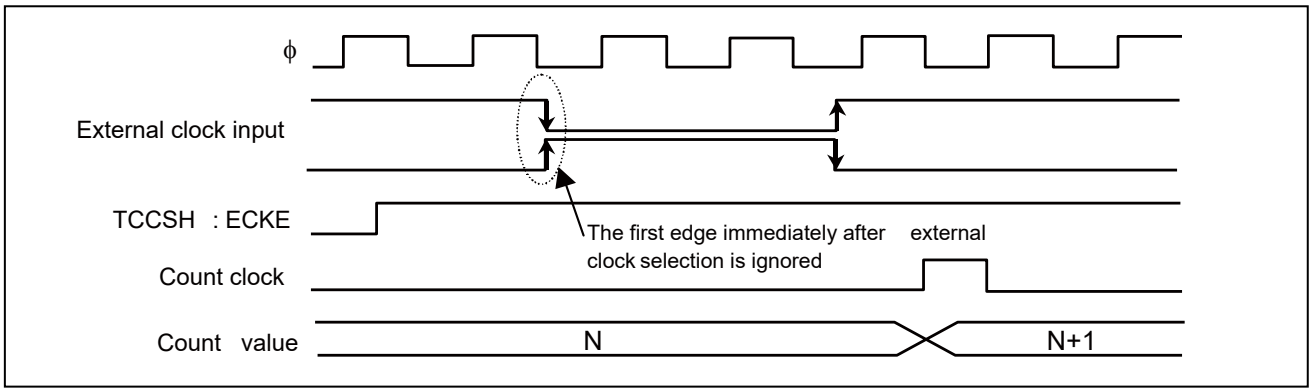
21.5 Operation

This section explains the operations of the free-run timer.

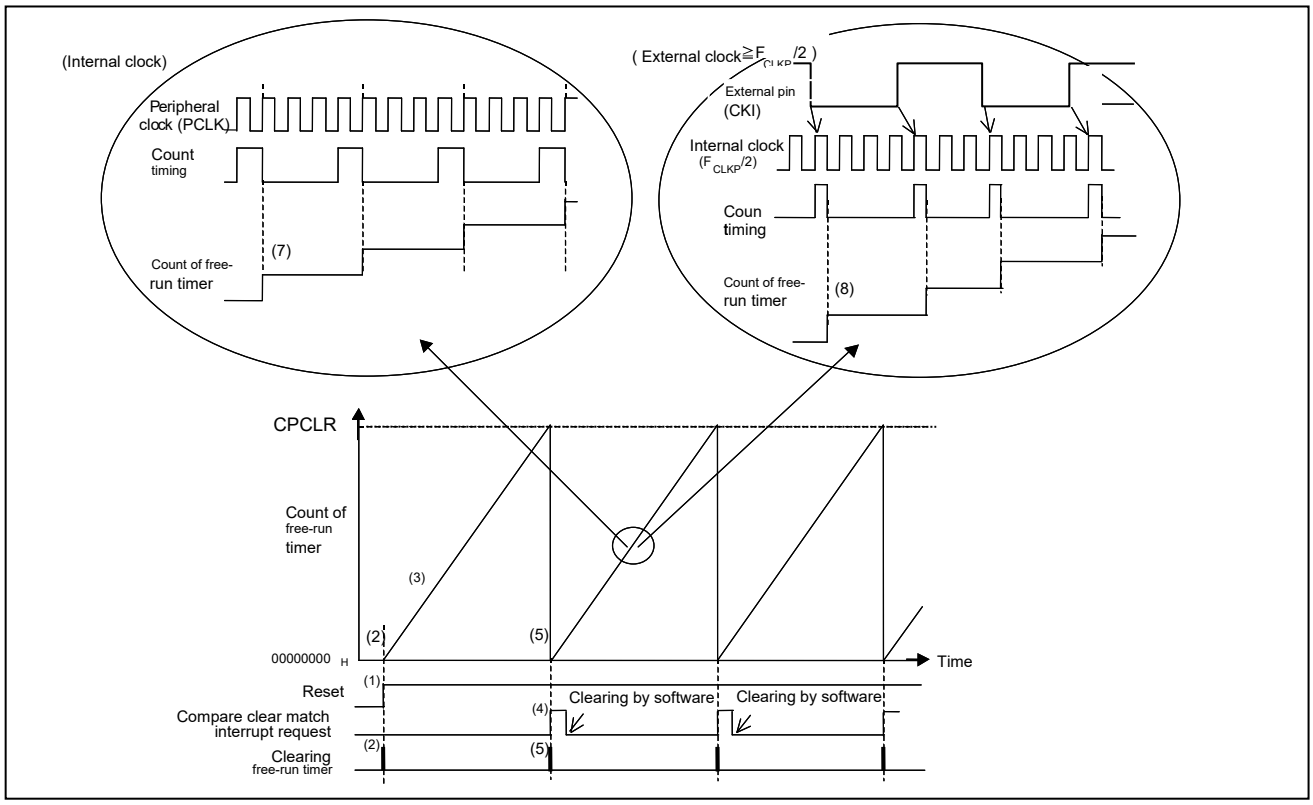
21.5.1 Count Operation of the Free-run Timer
21.5.2 Counting Up
21.5.3 Timer Clear
21.5.4 Each Clear Operations of the Free-run Timer
21.5.5 Timer Interrupt

21.5.1 Count Operation of the Free-run Timer

This section explains the count operation of the free-run timer.



The free-run timer will be incremented based on the input clock (internal clock or external clock). If the external clock mode (TCCSH:ECKE = 1) is selected, the free-run timer starts counting up by the rising and falling edges of the external input clock.
The first rising and falling edges of the external clock immediately after the selection of external clock mode will be ignored. This means that the first falling edge will be ignored if the initial value of the external clock input is "1", and the first rising edge will be ignored if the initial value is "0".



(1) Reset

(2) Clearing of the free-run timer by reset (Count value "00000000H")

(3) Count up operation by the free-run timer

(4) Compare clear match of the free-run timer and interrupt generation

(5) Clearing of the free-run timer by compare clear match (Count value "00000000H")

(6) Repetition of step (3) to (5)

(7) The free-run timer counts up in the clock obtained by dividing the internal clock (count clock).

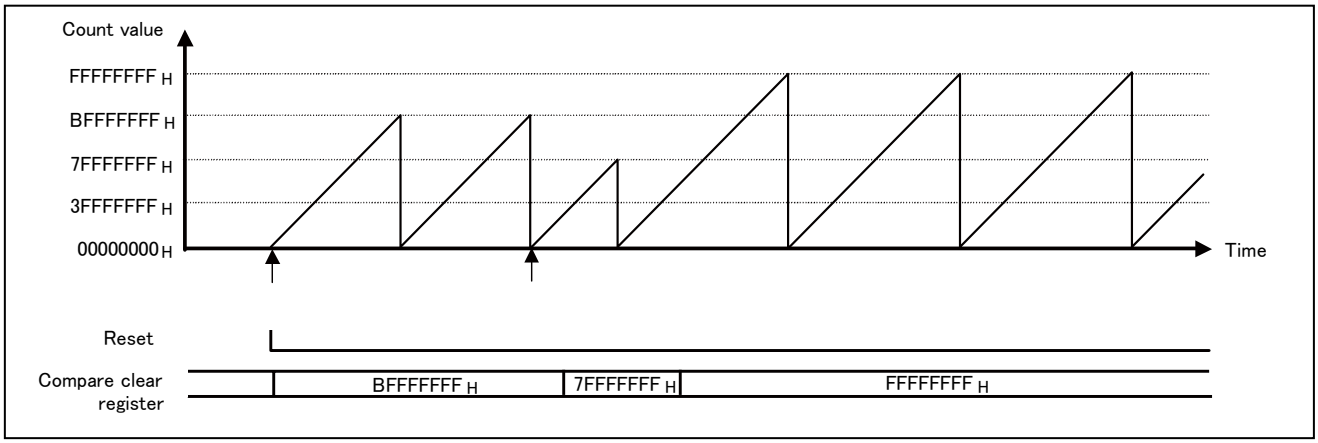
(8) The free-run timer counts up in the count clock obtained by synchronizing the external clock with the internal clock.

21.5.2 Counting Up

This section explains counting up of the free-run timer.

32-bit free-run timer is an up counter. The counter starts counting up 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 then be cleared to "00000000_H" and start counting up again.

Figure 21-3. Up Counter Operation



21.5.3 Timer Clear

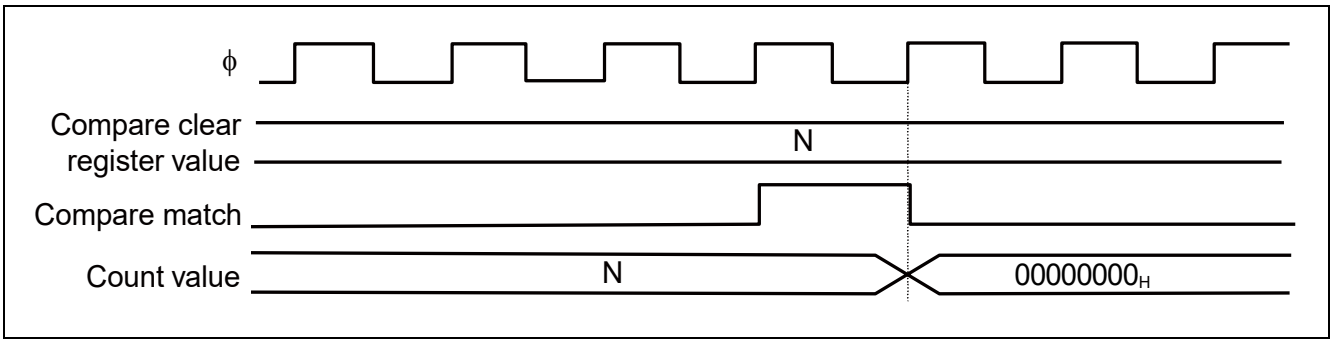
This section explains timer clear of the free-run timer.

The count value of the free-run timer will be cleared in any of the followings:

- When there is a match with the compare clear register
- When "1" is written to the SCLR bit of the TCCSL register while it is in operation
- When "00000000_H" is written to the TCDT register while it is in stop
- When it has been reset.

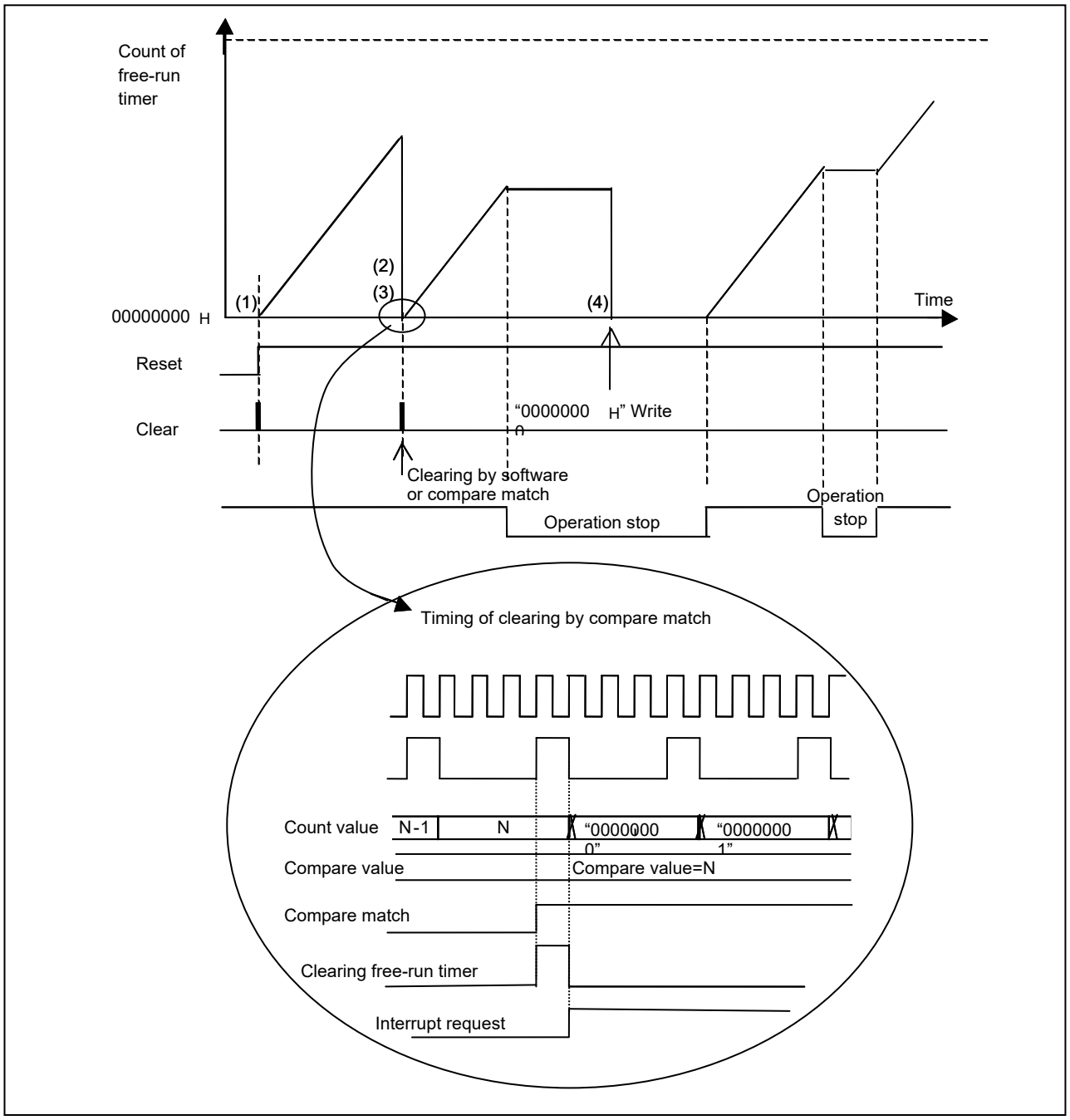
The counter will be cleared as soon as it has been reset. When there is a match with the compare clear register, the counter will be cleared in synchronization with the count timing.

Figure 21-4. Clear Timing of the Free-run Timer



21.5.4 Each Clear Operations of the Free-run Timer

This section explains each clear operations of the free-run timer.



Clearing of the free-run timer (4 types)

(1) When it has been reset
(2) When "1" is written to SCLR: bit4 of the TCCSL register while it is in operation
(3) When there is a match with the compare clear register
(4) When "00000000_H" is written to the TCDT register while it is in stop

21.5.5 Timer Interrupt

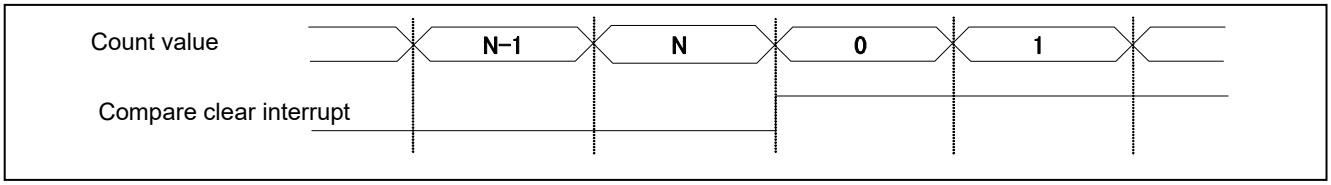
This section explains timer interrupt of the free-run timer.

For the free-run timer, you will be able to generate the following type of interrupt.

- Compare clear interrupt

The compare clear interrupt will be generated when the timer value matches the value of the compare clear register (CPCLR).

Figure 21-5. Interrupt



21.6 Setting

This section explains setting of the free-run timer.

Table 21-2. Settings Required for Using the Free-run Timer

Configuration	Configured Register	Setting Method
Timer initialization condition setting	Timer control registers (TCCSHn,TCCSLn)	See 21.7.4
Count clock setting Internal clock selection		See 21.7.1
External clock selection		See 21.7.2
Count operation start		See 21.7.3
For external clock, set the clock input pins (FRCK) for input.	Set the pins for peripheral input. See "Chapter: I/O Ports".	

Table 21-3. Settings Required for Performing Free-run Timer Interrupt

Configuration	Configured Register	Setting Method
Free-run timer interrupt vector Free-run timer interrupt level setting	See "Chapter: Interrupt Control".	See 21.7.5
Free-run timer interrupt setting Interrupt request clear Interrupt request enable	Timer control registers (TCCSHn)	See 21.7.6

Table 21-4. Settings Required for Stopping the Free-run Timer

Configuration	Configured register	Setting method
Free-run timer stop bit setting	Timer control registers (TCCSLn)	See 21.7.7

21.7 Q&A

This section explains Q&A of the free-run timer.

21.7.1 How to Select Internal Clock Dividers
21.7.2 How to Select the External Clock
21.7.3 How to Enable/Disable the Count Operation of the Free-run Timer
21.7.4 How to Clear the Free-run Timer
21.7.5 About Interrupt Related Registers
21.7.6 How to Enable Compare Clear Interrupt
21.7.7 How to Stop the Free-run Timer Operation

21.7.1 How to Select Internal Clock Dividers

This section explains how to select internal clock dividers.

There are nine types of internal clock dividers. You can configure it using the clock selection bit (TCCSHn.ECKE) , and count clock bit (TCCSLn.CLK[3:0]). (n=0 to 7 is channel number)

Internal Clock	Configuration	
	Clock Selection bit (ECKE)	Count Clock Bits (CLK[3:0])
To select F_{PCLK}	Set "0".	Set "0000".
To select $2/F_{PCLK}$	Set "0".	Set "0001".
To select $4/F_{PCLK}$	Set "0".	Set "0010".
To select $8/F_{PCLK}$	Set "0".	Set "0011".
To select $16/F_{PCLK}$	Set "0".	Set "0100".
To select $32/F_{PCLK}$	Set "0".	Set "0101".
To select $64/F_{PCLK}$	Set "0".	Set "0110".
To select $128/F_{PCLK}$	Set "0".	Set "0111".
To select $256/F_{PCLK}$	Set "0".	Set "1000".

21.7.2 How to Select the External Clock

This section explains how to select the external clock.

You can configure it using the clock selection bit (TCCSHn.ECKE), data direction bits and port function bits. (n=0 to 7 is channel number)

To set to External Clock Input	Configuration		Pin	Pulse Width (H Width, L Width)
Free-run timer 0	Set the clock selection bit (ECKE) to "1".	Set the FRCK0 pin for peripheral input. (See "Chapter: I/O Ports".)	FRCK0	$4/F_{PCLK}$ or higher
Free-run timer 1		Set the FRCK1 pin for peripheral input. (See "Chapter: I/O Ports".)	FRCK1	

Ch.2 to 5 are similar to the above.

21.7.3 How to Enable/Disable the Count Operation of the Free-run Timer

This section explains how to enable/disable the count operation of the free-run timer.

Set the count operation bit (TCCSLn.STOP). (n=0 to 7 is channel number)

Operation	Count Operation Bit (STOP)
To operate the free-run timer	Set "0".
To stop the free-run timer	Set "1".

21.7.4 How to Clear the Free-run Timer

This section explains how to clear the free-run timer.

You can clear the free-run timer using the following method.

- Set using the clear bit (TCCSLn:SCLR). (n=0 to 7 is channel number)

Operation	Clear Bit (SCLR)
To clear the free-run timer	Write "1".

- Perform a reset.

  When a reset is performed (RSTX pin input, watchdog reset, software reset, etc.), the free-run timer will be cleared to its initial state.
- Write "00000000_H" while the free-run timer is inactive.

  If "00000000_H" is written while the free-run timer is inactive, the count value will be "00000000_H".
- Overflow of the free-run timer will result in the count value returning to "00000000_H".
- It will be cleared if there is a match with the compare clear register.

21.7.5 About Interrupt Related Registers

This section explains interrupt related registers.

Free-run timer interrupt vector and free-run timer interrupt level settings

The relationship between free-run timer numbers, interrupt levels and interrupt vectors is as shown in "A.3 List of Interrupt Vector" in "Appendix".

For details of the interrupt levels and interrupt vectors, see "Chapter: Interrupt Control (Interrupt Controller)".

Number	Interrupt Vector (Default)	Interrupt Level Setting Bit (ICR[4:0])
Free-run timer 0	#50 Address: $0FFF34_H$	Interrupt level register (ICR34) Address: 00462_H
Free-run timer 1	#51 Address: $0FFF30_H$	Interrupt level register (ICR35) Address: 00463_H
Free-run timer 2	#50 Address: $0FFF34_H$	Interrupt level register (ICR34) Address: 00462_H
Free-run timer 3	#51 Address: $0FFF30_H$	Interrupt level register (ICR35) Address: 00463_H

Refer “Appendix A.3 List of Interrupt Vector” for ch.4 and ch.5.

Since interrupt request flags (TCCSHn.ICLR) will not be cleared automatically, clear the flags using software before returning from interrupt processing. (Write "0" to the ICLR bit) (n=0 to 7 is channel number)

21.7.6 How to Enable Compare Clear Interrupt

This section explains how to enable compare clear interrupt.

Enable interrupt request, interrupt request flag

Interrupt enable setting can be performed using interrupt request enable bit (TCCSHn:ICRE). (n=0 to7 is channel number)

Operation	Compare Clear Interrupt Request Enable Bit (ICRE)
Interrupt disabled	Set "0".
Interrupt enabled	Set "1".

Clearing of the interrupt request can be configured using interrupt flag bit (TCCSHn:ICLR). (n=0 to 7 is channel number)

Operation	Compare Clear Interrupt Flag Bit (ICLR)
Interrupt request clear	Write "0".

21.7.7 How to Stop the Free-run Timer Operation

This section explains how to stop the free-run timer operation.

Set using the count operation bits (TCCS0:STOP), (TCCS1:STOP), (TCCS2:STOP), (TCCS3:STOP), ...

See "21.7.3 How to Enable/Disable the Count Operation of the Free-run Timer".

21.8 Sample Program

This section explains sample program of the free-run timer.

Setting procedure example 1

Free-run timer 0, Clock=PCLK/2^6,
Count the number of compare matches using interrupt processing.

< Initial setting>

-Free-run timer ch.0 control	Register name.Bit name
Control register setting Clock selection≫	TCCSH0/TCCSL0 .ECKE
Compare interrupt request flag≫ Compare interrupt request enabled≫	.ICLR .ICRE
Counting Operation>>	.STOP
TCDT clear>> Count clock≫	.CLR .CLK3-0
Timer data value setting	TCDT0

-Interrupt-related	Register name.Bit name
Sets an interrupt level.	ICR34
I flag setting	(CCR)

-Variable setting

<Activation>

-Free-run timer ch.0 activation	Register name.Bit name
Count operation activation	TCCS0 .STOP

<Interrupt >

-Interrupt processing	Register name.Bit name
Clearing of interrupt request flag	TCCS0.ICLR
(Any process)	
Variable counting	

<Interrupt vector>
Vector table setting

(Note)
Clock-related settings and the setting of __set_il (numeric value) need to be configured in advance. See “Chapter: Clock” and “Chapter: Interrupt Control (Interrupt Controller)”

Program example 1

```
void FREE_RUN_TIMER0_sample(void)
{
        FREERUN0_initial();
        FREERUN0_start();
}

void FREERUN0_initial(void)
{
        IO_TCCS1.word = 0x0041; /* Setting value=0000_0000_0100_0001 */
                                /* bit15 = 0       ECKE internal clock source */
                                /* bit14 -10= 00000 Reserved bit */
                                /* bit9 = 0        ICLR compare interrupt request flag */
                                /* bit8 = 0        ICRE compare interrupt disabled */
                                /* bit7 = 0        Reserved bit */
                                /* bit6 = 1        STOP count disabled */
                                /* bit5 = 0        Reserved bit */
                                /* bit4 = 0        SCLR Initialization of SCLR free-run timer value
                                                      (no) */
                                /* bit3-0 = 0001   CLK3-0 Count clock PCLK/2=32MHz/2 */

        IO_TCDT0 = 0x0000;      /* Initialization of timer data value */

        IO_ICR[34].byte = 0x10; /* Free-run timer 0 interrupt level setting (any value) */
        __EI();                 /* Interrupt enabled */
        count = 0;
}

void FREERUN0_start(void)
{
        IO_TCCS0.bit.STOP = 0;  /* bit6 = 0 STOP count enabled */
}

__interrupt void
FREE_RUN_TIMER0_int(void)
{
        IO_TCCS0.bit.ICLR = 0;  /* bit9 = 0 Clearing of ICLR compare match flag */

        count++;
}

Specification of interrupt routine required in vector table
#pragma intvect FREE_RUN_TIMER0_int 50
```

21.9 Notes

This section explains notes of the free-run timer.

- Clear Timing of the Free-run Timer
  - When a reset is performed (RSTX pin input, watchdog reset, software reset, etc.) , the counter will stop counting after initializing to "00000000_H".
  - A software clear (TCCSL:SCLR=1) clears the counter at the next cycle after the clear request is generated. However, in the case of compare match, the counter is cleared in the same timing as the counting up.
  - Counter clear operation (software, compare match) will only be enabled while the free-run timer is in operation. To clear the counter while the free-run timer is in stop, you need to write "00000000_H" to the timer count data register.
- Writing to the Timer Data Register
  - Always write a value to the free-run timer while the free-run timer is inactive (STOP = "1"), using a word access instruction.
- External Clock Operation
  - The timings of the compare match output and generation of interrupt of the external clock will be the next count clock timing after the compare match. Therefore, in order to the generate compare match output and interrupt, 1 clock (external clock) must at least be input after the compare match.
- Read-modify-write
  - Compare clear interrupt flag bits of the timer control register are "1" when read using a read-modify-write instruction.

22. Output Compare

This chapter explains the output compare.

22.1 Overview

22.2 Features

22.3 Configuration Diagram

22.4 Registers

22.5 Operation

22.6 Setting

22.7 Q&A

22.8 Sample Program

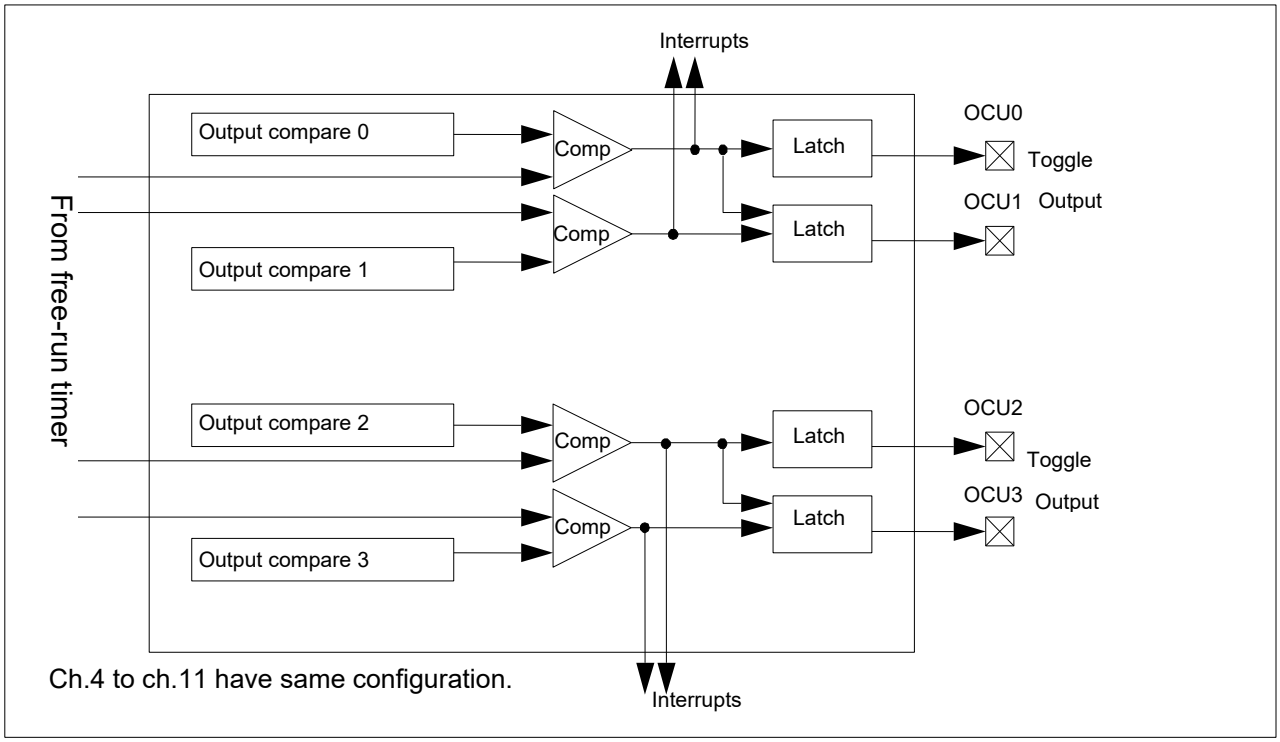
22.9 Notes

22.1 Overview

This section explains the overview of the output compare.

The output compare consists of a 32-bit compare register, a compare output latch, and a compare control register. When the 32-bit free-run timer value matches the compare register value, the output level is inverted and an interrupt also can be generated.

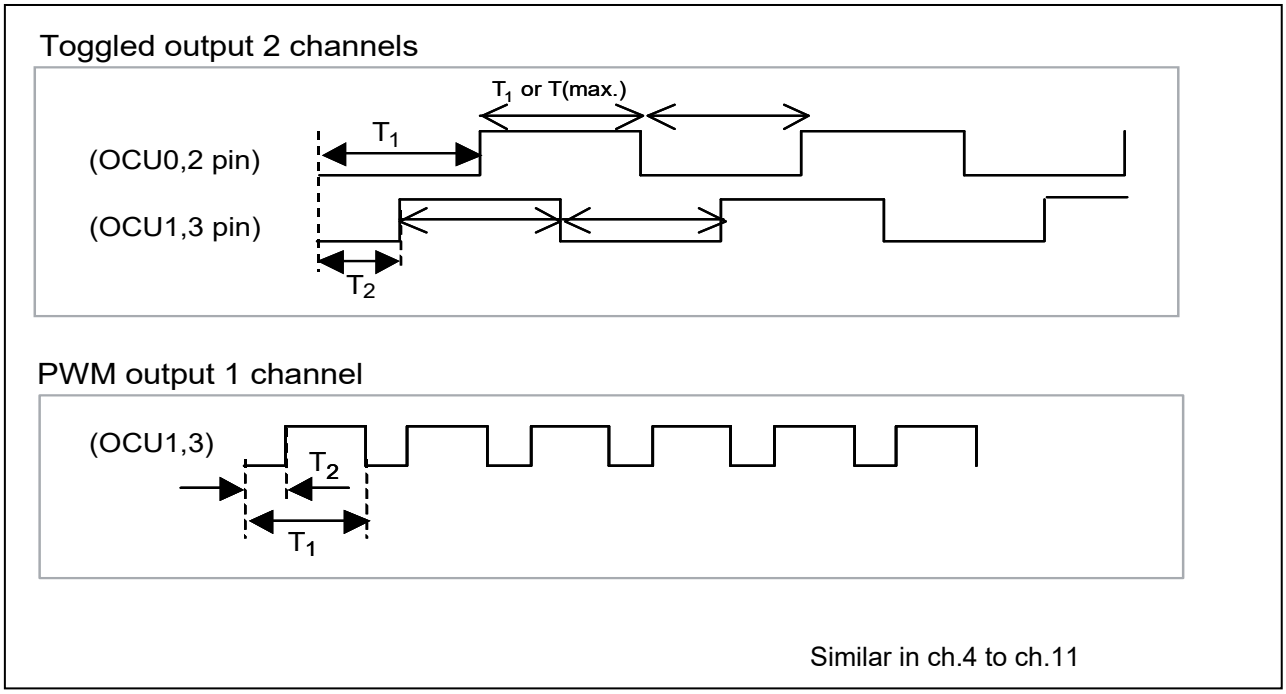
Figure 22-1. Block Diagram (Overview)



22.2 Features

This section explains the features of the output compare.

Figure 22-2. Output Waveform



- Type: 32-bit compare register × 4 + compare circuit
- Corresponding timer: Free-run timer is used

  Output compare 0 to 3: Free-run timer ch.0 or ch.1 is used.
  Output compare 6 to 9: Free-run timer ch.2 or ch.3 is used.
  Output compare 4, 5, 10 and 11: Free-run timer ch.4 or ch.5 is used.
- Number: 12 channels
- Operation by compare match
- Pin output value invert (toggle output)
- Interrupt occurrence
- Count accuracy: Peripheral clock (PCLK/2, PCLK/4, PCLK/8, PCLK/16, PCLK/32, PCLK/64, PCLK/128, PCLK/256) (Dependent on the free-run timer)

  Note:

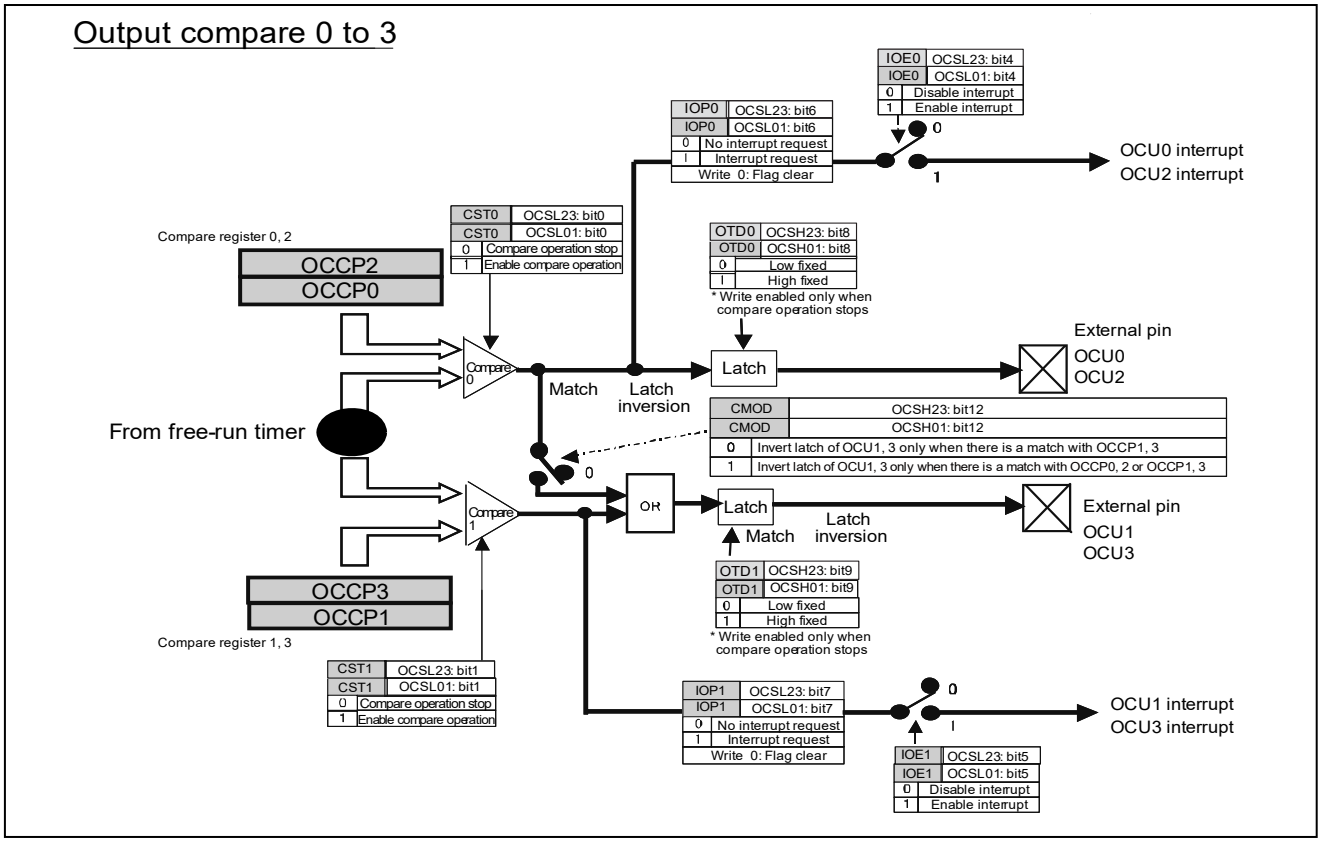
  The setting of the peripheral clock (PCLK) divided by 1 is prohibited.
- Toggle change width (T): 1 × count accuracy to 100000000_H × count accuracy
- Interrupt: Compare match interrupt

- Others:
  - Output level initial value setting is enabled. ("H"/"L")
  - Unused pins as OCU output can be used as general-purpose ports.
  - 12 compare registers can be used for independence.
  - Output pins and interrupt flags correspond to the compare register.
  - Output pins can be inverted with the use of two compare registers. (Function only for OCU1, 3, 5, 7, 9 and 11)
  - The initial value of each output pin can be set.
  - When the output compare register matches the 32-bit free-run timer, an interrupt can be generated.

22.3 Configuration Diagram

This section explains the configuration diagram of the output compare.

Figure 22-3. Configuration Diagram (Detail)



Ch.4 to ch.11 have same configuration.

22.4 Registers

This section explains the registers of the output compare.

Channel	Base_addr	External Pin
		OCU
0	0x02E8	OCU0_0, OCU0_1
1	0x02E8	OCU1_0, OCU1_1
2	0x02F4	OCU2_0, OCU2_1
3	0x02F4	OCU3_0, OCU3_1
4	0x0F90	OCU4_0, OCU4_1
5	0x0F90	OCU5_0, OCU5_1
6	0x0120	OCU6_0, OCU6_1
7	0x0120	OCU7_0, OCU7_1
8	0x012C	OCU8_0, OCU8_1
9	0x012C	OCU9_0, OCU9_1
10	0x0138	OCU10_0, OCU10_1
11	0x0138	OCU11_0, OCU11_1

Table 22-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0120	OCCP6				Compare register 6
0x0124	OCCP7				Compare register 7
0x0128	OCFS67	Reserved	OCSH67	OCSL67	Free-run timer selection register 67 Output control register 67 upper Output control register 67 lower
0x012C	OCCP8				Compare register 8
0x0130	OCCP9				Compare register 9
0x0134	OCFS89	Reserved	OCSH89	OCSL89	Free-run timer selection register 89 Output control register 89 upper Output control register 89 lower
0x0138	OCCP10				Compare register 10
0x013C	OCCP11				Compare register 11
0x0140	OCFS1011	Reserved	OCSH1011	OCSL1011	Free-run timer selection register 1011 Output control register 1011 upper Output control register 1011 lower
0x02E8	OCCP0				Compare register 0
0x02EC	OCCP1				Compare register 1
0x02F0	OCFS01	Reserved	OCSH01	OCSL01	Free-run timer selection register 01 Output control register 01 upper Output control register 01 lower
0x02F4	OCCP2				Compare register 2
0x02F8	OCCP3				Compare register 3
0x02FC	OCFS23	Reserved	OCSH23	OCSL23	Free-run timer selection register 23 Output control register 23 upper Output control register 23 lower
0x0F90	OCCP4				Compare register 4
0x0F94	OCCP5				Compare register 5
0x0F98	OCFS45	Reserved	OCSH45	OCSL45	Free-run timer selection register 45 Output control register 45 upper Output control register 45 lower

22.4.1 Free-run Timer Selection Register: OCFS

The bit configuration of the free-run timer selection register is shown below.

The free-run timer to compare is selected.

x: Channel number 0, 2, 4, 6, 8, and 10.
y: Channel number 1, 3, 5, 7, 9, and 11.

OCFSxy (Free-run timer selection xy): Address Base_addr+08_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	-	-	SELy	SELx
Initial value	-	-	-	-	-	-	1	1
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W

[bit7 to bit2] - : Undefined

Writing to these bits does not affect the operation of the output compare.

[bit1] SELy: Free-run timer selection (Output compare y)
[bit0] SELx: Free-run timer selection (Output compare x)

SEL{0,1,2,3}	Operating Mode
0	Free-run timer 0
1	Free-run timer 1

SEL{4,5}	Operating Mode
0	Free-run timer 4
1	Free-run timer 5

SEL{6,7,8,9}	Operating Mode
0	Free-run timer 2
1	Free-run timer 3

SEL{10,11}	Operating Mode
0	Free-run timer 4
1	Free-run timer 5

22.4.2 Output Control Register (Upper Bit): OCSH

The bit configuration of the output control register (Upper bit) is shown below.

This register is to control operations of the output compare.

x: Channel number 0, 2, 4, 6, 8, and 10.
y: Channel number 1, 3, 5, 7, 9, and 11.

OCSHxy (Output compare xy): Address Base_addr+0A$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-	-	-	CMOD	Reserved	Reserved	OTDy	OTDx
Initial value	-	-	-	0	0	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R/W	R/W0	R/W0	R,W	R,W

[bit15 to bit13] - : Undefined

Writing to these bits does not affect the operation of the output compare.

[bit12] CMOD: Compare mode

CMOD	Operating Mode
0	Independent operation (OCU0 to OCU11 pins output level invert operation is independent.) OCU0, 2, 4, 6, 8, 10 pins: When the free-run timer value corresponds to the compare register 0, 2, 4, 6, 8, 10 (OCCP0, 2, 4, 6, 8, 10) value, the output is inverted. OCU1, 3, 5, 7, 9, 11 pins: When the free-run timer value corresponds to the compare register 1, 3, 5, 7, 9, 11 (OCCP1, 3, 5, 7, 9, 11) value, the output is inverted. The comparison target free-run timer is selected by OCFS registers.
1	Coordinated operation OCU0, 2, 4, 6, 8, 10 pins: When the free-run timer value corresponds to the compare register 0, 2, 4, 6, 8, 10 (OCCP0, 2, 4, 6, 8, 10), the output is inverted. OCU1, 3, 5, 7, 9, 11 pins: When the free-run timer value corresponds to either the compare register (0 or 1), (2 or 3), ... , (10 or 11) the output is inverted. The comparison target free-run timer is selected by OCFS registers.

- When the compare register 0, 1 and 2, 3 have the same value, the operation is the same one as when only one compare register is used. The compare register 4 to 7 and 8 to 11 behave as same as the 1 to 3.

[bit11, bit10] Reserved

Always set these bits to "0".

[bit9,8] OTDy, OTDx: Port Level Configuration

It can configure initial output port level for OCUn and OCUm. OTDy is for OCUn, and OTDx for OCUm. (n: odd channel number, m: even channel number). GPIO configuration is necessary for OCU output during compare operation stop. Port level will be read out at reading operation.

OTDy	Operation
0	Initial level "L" at OCUn
1	Initial level "H" at OCUn

OTDx	Operation
0	Initial level "L" at OCUm
1	Initial level "H" at OCUm

22.4.3 Output Control Register (Lower Bit): OCSL

The bit configuration of the output control register (Lower bit) is shown below.

This register is to control operations of the output compare.

x: Channel number 0, 2, 4, 6, 8, and 10.
y: Channel number 1, 3, 5, 7, 9, and 11.

OCSLxy (Output compare xy): Address Base_addr+0B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IOPy	IOPx	IOEy	IOEx	-	-	CSTy	CSTx
Initial value	0	0	0	0	1	1	0	0
Attribute	R(RM1),W	R(RM1),W	R/W	R/W	R1,WX	R1,WX	R/W	R/W

[bit7] IOPy: Interrupt request flag (output compare y)

[bit6] IOPx: Interrupt request flag (output compare x)

IOP	State	
	Read	Write
0	Without interrupt request	Flag (IOP) is cleared.
1	With interrupt request	No effect on operations

- This bit becomes "1" when the count value of free-run timer (TCDT) corresponds to the output compare compare register (OCCP).
- The interrupt request becomes enabled when the interrupt enable bit (IOE) is "1".

[bit5] IOEy: Interrupt request enable (Output compare y)

[bit4] IOEx: Interrupt request enable (Output compare x)

IOE	State
0	Output compare interrupt request is disabled.
1	Output compare interrupt request is enabled.

- This bit is used to "enable" the output compare interrupt for the compare register.
- While "1" is written to this bit, if the compare match interrupt flag bit (IOP) is set, the output compare interrupt is generated.

[bit3, bit2] - : Undefined

Writing to these bits does not affect the operation of the output compare.

[bit1] CSTy: Operation enable (Output compare y)

[bit0] CSTx: Operation enable (Output compare x)

CST	Operation
0	Operation of the output compares is stopped.
1	Operation of the output compares is enabled.

- This bit enables the compare operation for the count value of free-run timer (TCDT) and the output compare register.
- The compare registers (OCCP) must be set with values before the compare operation is enabled
- Because the output compare is synchronized with the free-run timer, when the free-run timer is stopped, the output compare operation also is stopped.

22.4.4 Compare Register: OCCP

The bit configuration of the compare register is shown below.

These registers set the values to be compared with the 32-bit free-run timer count value.
x: Channel number 0, 2, 4, 6, 8, and 10.
y: Channel number 1, 3, 5, 7, 9, and 11.

OCCPx (Output compare x): Address Base_addr+00_H (Access: Word)

OCCPy (Output compare y): Address Base_addr+04_H (Access: Word)

	bit31							bit0
	OP[31:0]							
Initial value	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
Attribute	R/W							

- The compare registers OCCP is compared with the count value of free-run timer (TCDT).
- When the OCCP register values correspond to the 32-bit free-run timer value, a compare signal is generated and an output compare interrupt flag is set. The compare value is reflected after the write instruction is completed. Therefore, the compare value change during operation might generate an interrupt twice per one free-run counting if the newly written compare value is larger than the previous compare value.
- In addition, when the corresponding OCU of the port function register (PFR) is set and output is enabled, the output level corresponding to the compare register is inverted.
- For access to this register, use a word access instruction.

22.5 Operation

This section explains the operations of the output compare.

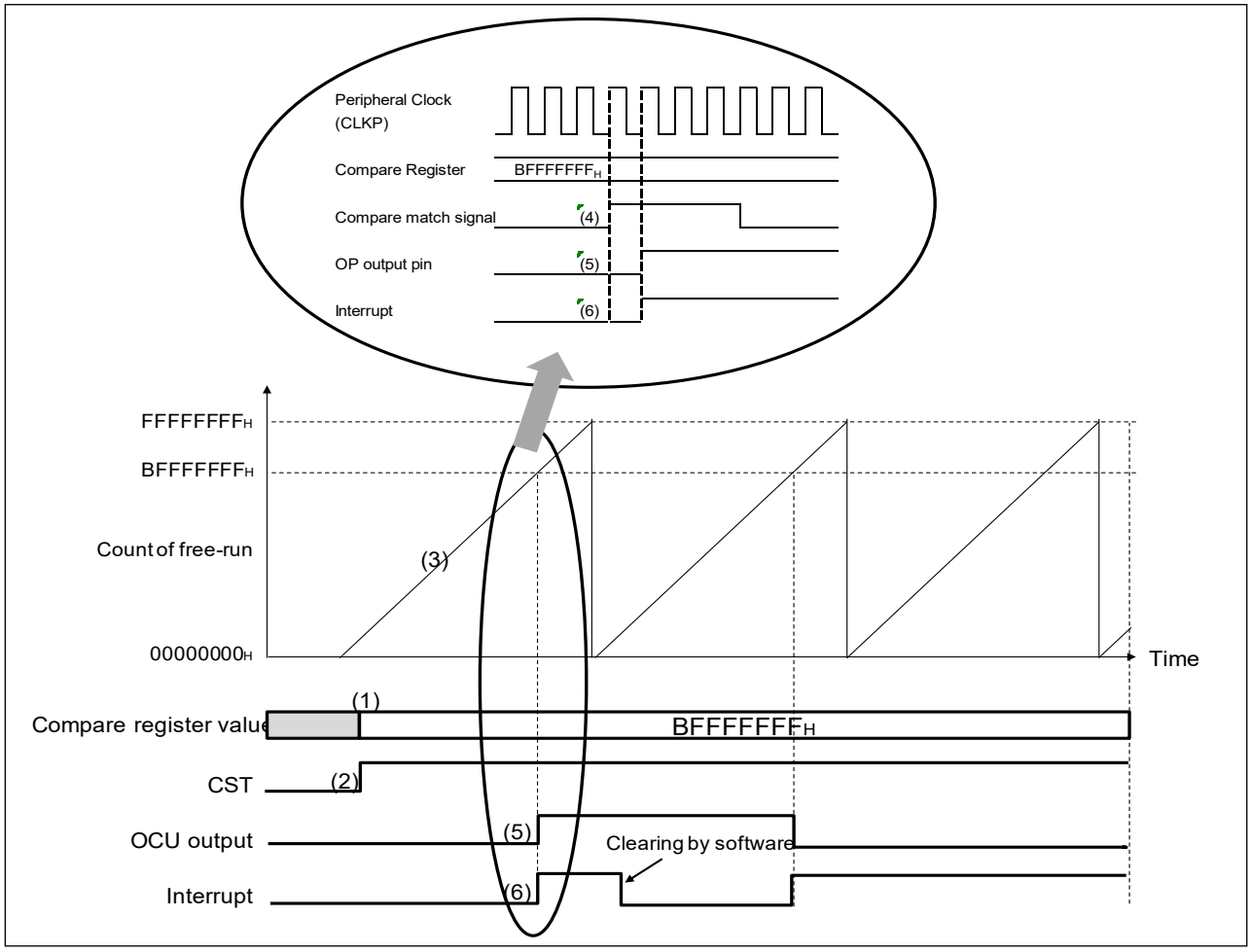
22.5.1 Output Compare Output (Independent Invert) CMOD = "0"

22.5.2 Output Compare Output (Coordinated Invert) CMOD = "1"

22.5.3 Output Compare Operation Timing

22.5.1 Output Compare Output (Independent Invert) CMOD = "0"

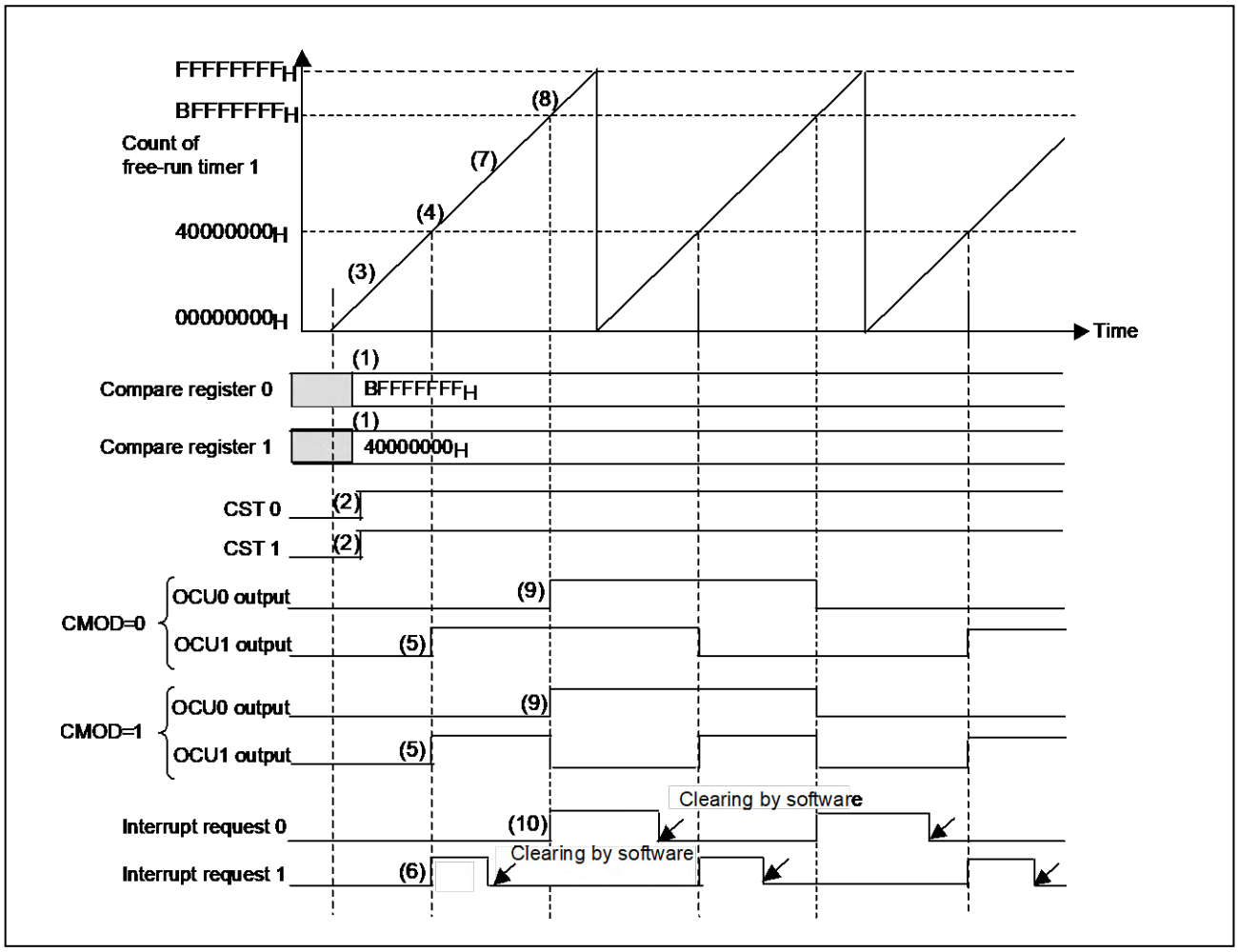
This section explains the output compare output (independent invert).



(1) A compare value is set.
(2) Compare operation is enabled (CST = 1)
(3) Free-run timer count up (example of one count per four clocks)
(4) A free-run timer value is compared with a compare value and they match (Compare match)
(5) OCU output level is inverted.
(6) A compare match interrupt request is generated.

22.5.2 Output Compare Output (Coordinated Invert) CMOD = "1"

This section explains the output compare output (coordinated invert).



(1) Values of Compare 0 and Compare 1 are set.
(2) Compare operation is enabled.
(3) Free-run timer count up
(4) Compare 1 match
(5) OCU1 output level is inverted.
(6) Compare1 match interrupt
(7) Free-run timer count up
(8) Compare 0 match
(9) OCU0 output level is inverted.
When CMOD = 1, OCU1 output level also is inverted.
(10) Compare 0 match interrupt

22.5.3 Output Compare Operation Timing

This section explains the output compare operation timing.

With the use of two pairs of compare registers, the output level can be changed. (For CMOD = 1)

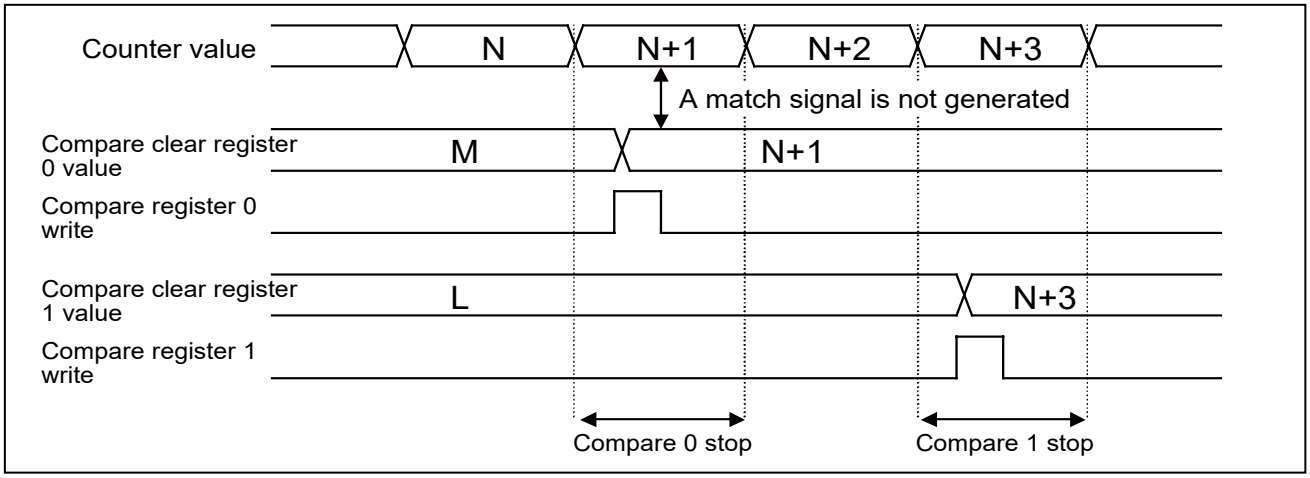
The output compare can invert the output as well as generate an interrupt when the free-run timer value matches the specified compare register value and a compare match signal is generated. The output invert timing on compare match is synchronized with the counter count timing.

22.5.3.1 Compare Register Write

Compare register write is shown below.

The compare operation with the counter value is not performed on compare register rewrite.

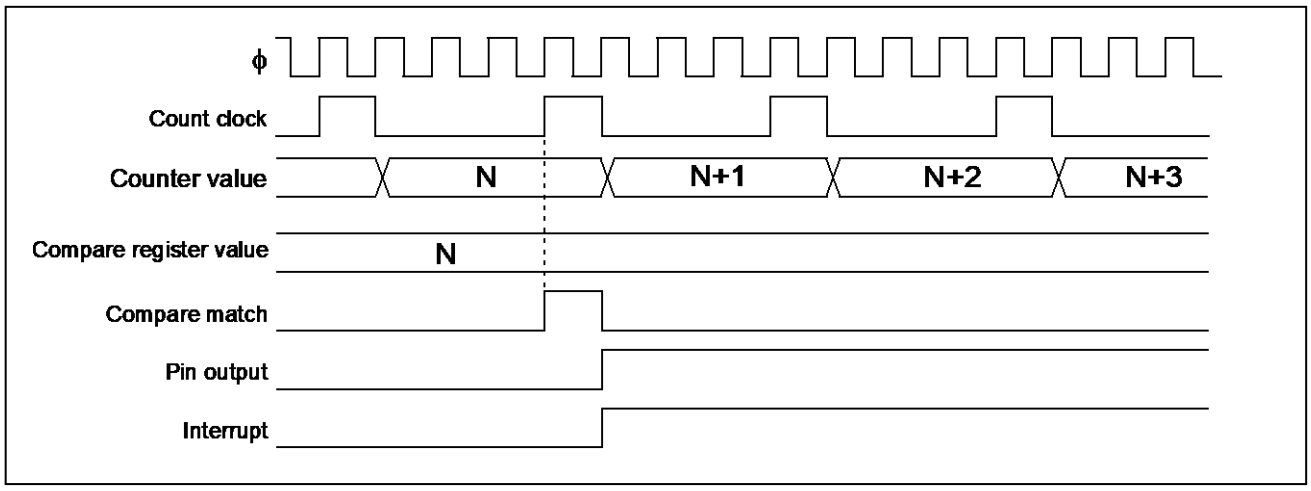
Figure 22-4. Compare Register Write Timing



22.5.3.2 Compare match, Interrupt

Compare match, interrupt are shown below.

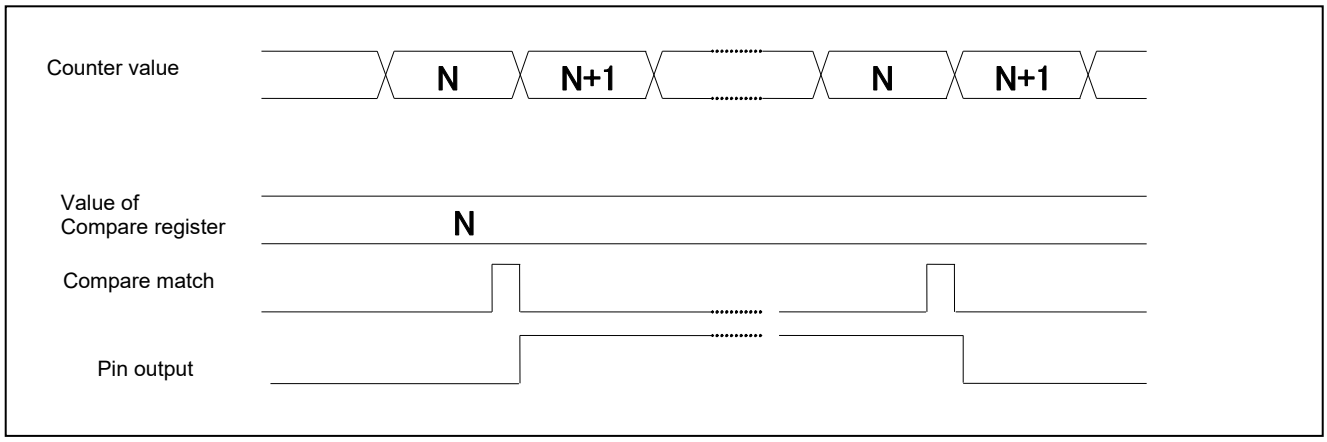
Figure 22-5. Compare match, Interrupt Timing



22.5.3.3 Pin Output

This section shows the pin output.

Figure 22-6. Pin Output Timing



22.6 Setting

This section explains settings of the output compare.

Table 22-2. Configuration Necessary for Use of Output Compare

Configuration	Register to be configured	Setting Method
Setting of the free-run timer	See "Chapter: Free-run Timer".	-
Setting of the compare value	Compare register: (OCCPx)	See 22.7.1
Setting of the compare mode	Output control register (OCSHxx, OCSLxx)	See 22.7.2
Compare operation stop		See 22.7.3
Setting of the compare pin output initial level		See 22.7.4
Setting of OCU0 to OCU11 pins to output	Set each pin for peripheral output. See "Chapter: I/O Ports", for the setting method.	
The free-run timer clear	Timer control register (TCCSL) See "Chapter: Free-run Timer".	See 22.7.6
Compare operation enable (activation)	Output control register (OCSLxx)	See 22.7.7

Table 22-3. Items Necessary for Interrupt Execution

Configuration	Register to be configured	Setting Method
Setting of output compare interrupt vector and output compare interrupt level	See "Chapter: Interrupt Control (Interrupt Controller)".	See 22.7.8
Setting of output compare interrupt Interrupt request clear Interrupt request enable	Output control register (OCSHxx, OCSLxx)	See 22.7.10

22.7 Q&A

This section explains Q&A of the output compare.

22.7.1 Set the Compare Value
22.7.2 Set the Compare Mode (Example with OCU1)
22.7.3 Enable/Disable the Compare Operation (Example with OCU0, OCU1)
22.7.4 Set the Compare Pin Output Initial Level (Example with OCU0, OCU1)
22.7.5 Set the Compare Pin OCU0, OCU1 for Output
22.7.6 Clear the Free-run Timer
22.7.7 Enable the Compare Operation
22.7.8 Interrupt Related Register
22.7.9 Interrupt Type
22.7.10 Enable the Interrupt
22.7.11 Calculation Method for the Compare Value

22.7.1 Set the Compare Value

This section explains how to set the compare value.

Write the compare value to the compare register OCCPx.

22.7.2 Set the Compare Mode (Example with OCU1)

This section explains how to set the compare mode.

Set with the compare mode bit (OCSH01:CMOD)

Operation	Compare mode bit
To invert the OCU1, OCU3 pins output when the free-run timer value matches the compare register 1 (OCCP1)	Set (OCSH01:CMOD) to "0".
To invert the OCU1 pin output when the free-run timer value matches either the compare register 0 (OCCP0) or the compare register 1 (OCCP1)	Set (OCSH01:CMOD) to "1".

Regardless of the CMOD bit, the operation is as follows:

- Regardless of the compare mode bit (OCSH01:CMOD) setting, the OCU0 output is inverted when the free-run timer value matches the compare register (OCCP0).

22.7.3 Enable/Disable the Compare Operation (Example with OCU0, OCU1)

This section explains how to enable/disable the compare operation.

Set the compare operation enable bit (OCSL01:CST0), (OCSL01:CST1).

Operation	Compare	Compare Operation Enable Bit
To stop (disable) the compare operation	Compare 0	Set (OCSL01:CST0) to "0".
	Compare 1	Set (OCSL01:CST1) to "0".
To enable the compare operation	Compare 0	Set (OCSL01:CST0) to "1".
	Compare 1	Set (OCSL01:CST1) to "1".

22.7.4 Set the Compare Pin Output Initial Level (Example with OCU0, OCU1)

This section explains how to set the compare pin output initial level.

Set the compare pin output specification bit (OCSH01:OTD0), (OCSH01:OTD1).

Operation	Compare Pin Output Specification Bit
To set the compare 0 pin to "L"	Set (OCSH01:OTD0) to "0".
To set the compare 0 pin to "H"	Set (OCSH01:OTD0) to "1".
To set the compare 1 pin to "L"	Set (OCSH01:OTD1) to "0".
To set the compare 1 pin to "H	Set (OCSH01:OTD1) to "1".

22.7.5 Set the Compare Pin OCU0, OCU1 for Output

This section explains how to set the compare pin OCU0, OCU1 for output.

Set the pin for peripheral output. For setting method, see "Chapter: I/O Ports".

22.7.6 Clear the Free-run Timer

This section explains how to clear the free-run timer.

Set the clear bit (TCCSL:SCLR) of the free-run timer used.

Operation	Clear Bit (SCLR)
To clear the free-run timer	Write "1".

For other methods, see "Chapter: Free-run Timer".

22.7.7 Enable the Compare Operation

This section explains how to enable the compare operation.

Set the compare operation enable bit (OCSL01:CST0, OCSL01:CST1, OCSL23:CST2, OCSL23:CST3).

See "22.7.3 Enable/Disable the Compare Operation (Example with OCU0, OCU1)".

22.7.8 Interrupt Related Register

This section explains the interrupt related register.

Both the output compare interrupt vector and the output compare interrupt level are set.

The relation among the output compare number, interrupt level, and interrupt vector is shown in the table below:

For the interrupt level and interrupt vector, see "Chapter: Interrupt Control (Interrupt Controller)".

Number	Interrupt Vector (Default)	Interrupt Level Setting Bit (ICR[4:0])
Output compare 0/1/6/7/10/11	#58 Address: $0FFF14_H$	Interrupt level register (ICR42) Address: $0046A_H$
Output compare 2/3/4/5/8/9	#59 Address: $0FFF10_H$	Interrupt level register (ICR43) Address: $0046B_H$

The interrupt request flag (OCSL01:IOP0, OCSL01:IOP1, OCSL23:IOP2, OCSL23:IOP3, OCSL45:IOP4, OCSL45:IOP5, OCSL67:IOP6, OCSL67:IOP7, OCSL89:IOP8, OCSL89:IOP9, OCSL1011:IOP10, OCSL1011:IOP11) are not cleared automatically.

22.7.9 Interrupt Type

This section explains the interrupt type.

The interrupt has one type only. It is generated by a compare match.

22.7.10 Enable the Interrupt

This section explains how to enable the interrupt.

Set the interrupt request enable bit (OCSLxy: IOEx, OCSLxy: IOEy) for the interrupt enable. (x, y: channel number)

Operation	Interrupt request enable bit (OCSLxy: IOEx, OCSLxy: IOEy)
Interrupt disable	Set "0".
Interrupt enable	Set "1".

Set the interrupt request flag bit (OCSLxy: IOPx, OCSLxy: IOPy) for the interrupt request clear. (x, y: channel number)

Operation	Interrupt request flag bit (OCSLxy: IOPx, OCSLxy: IOPy)
Interrupt request clear	Write "0".

22.7.11 Calculation Method for the Compare Value

This section explains the calculation method for the compare value.

22.7.11.1 Toggle Output Pulse

22.7.11.2 PWM Output

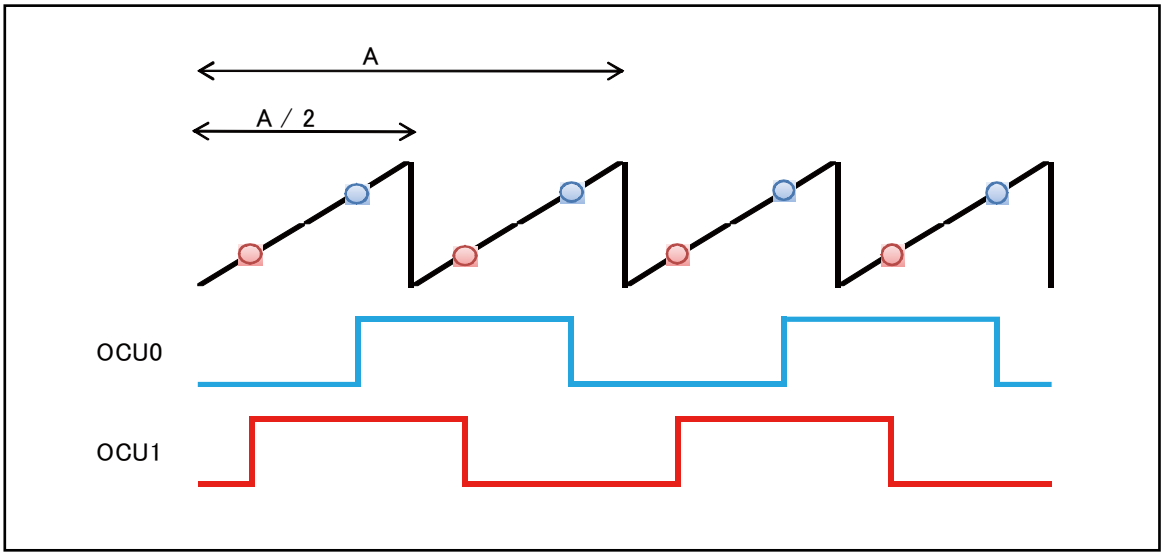
22.7.11.1 Toggle Output Pulse

This section explains the toggle output pulse.

(Example) To calculate a two-phase pulse with OCU0, OCU1, cycle: A, and one-fourth phase difference

- FreeRunTimer.CPCLR = (A/2) -1
- Output Compare.OCCP0 = (A/2)×(3/4) -1
- Output Compare.OCCP1 = (A/2)×(1/4) -1
- Output Compare.OCSH01.CMOD = 0

are setting.



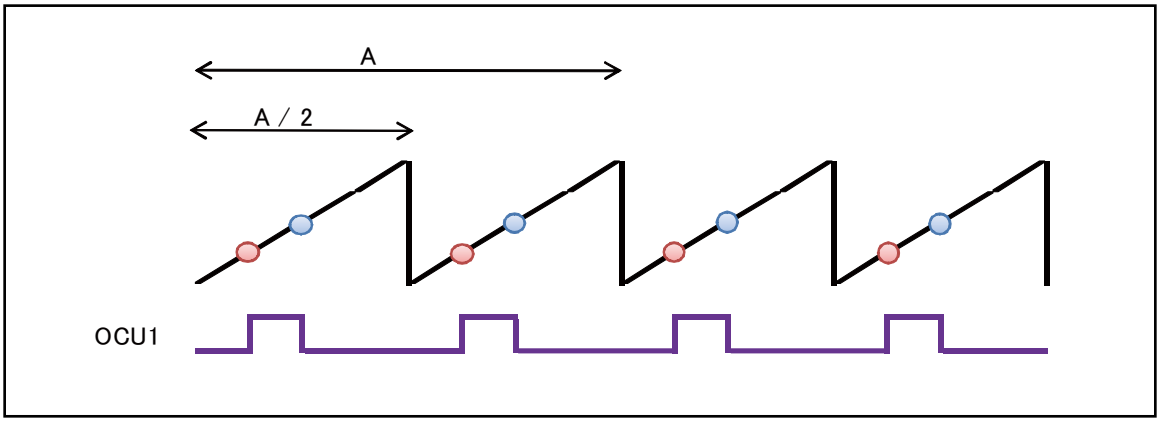
22.7.11.2 PWM Output

This section explains the PWM output.

(Example) To calculate the PWM with OCU0, OCU1, cycle: A, and duty 1/4

- FreeRunTimer.CPCLR = (A/2)-1
- Output Compare.OCCP0 = (A/2)×(1/2) -1
- Output Compare.OCCP1 = (A/2)×(1/4) -1
- Output Compare.OCSH01.CMOD = 1

are setting.



22.8 Sample Program

This section explains a sample program.

Configuration procedure example 1

.2 channels independent output
Compare operation (7FFF, BFFF)
Interrupt occurrence compare no clear

1. Initial setting
- Free-run timer ch.1 control Register name.bit name

Setting of control register	TCCSH1,TCCSL11
Clock selection>>	.ECKE
Compare interrupt request flag>>	.ICLR
Compare interrupt request enable>>	.ICRE
Counting Operation>>	.STOP
TCDT clear>>	
Count clock>>	.SCLR .CLK3-0
Setting of the timer data value	TCDT1

- Port Register name.bit name

Port OCU0 output setting	See "Chapter: I/O Ports"
Port OCU1 output setting	

- Output compare control Register name.bit name

Free-run timer selection	OCFS01
Setting of control register	OCSH01,OCSL01
Pin output level invert operation>>	.CMOD
Pin output level specification>>	.OTD1,OTD0
Interrupt request flag>>	.IOP1,IOP0
Interrupt request enable>>	.IOE1,IOE0
Operation enable setting>>	.CST1,CST0
Setting of compare value ch.0	OCCP0
Setting of compare value ch.1	OCCP1

- Interrupt relation Register name.bit name

Setting of an interrupt level.	ICR42 ICR43
Setting of I flag	(CCR)

2. Activation
- Output compare activation Register name.bit name

Interrupt control	OCSL01.IOE1
Compare operation activation	OCSL01.CST1 OCSL01.CST0

- Free-run timer ch.1 activation Register name.bit name

Counting operation activation	TCCSL1.STOP

3. Interrupt
- Interrupt process Register name.bit name

Clearing of interrupt request flag	OCSL01.IOP0
(any process)	
.....	
Clearing of interrupt request flag	OCSL01.IOP1
(any process)	
.....	

4. Interrupt vector
- Setting of the vector table

(Note)
Clock-related setting and setting of __set_il(numerical value) in advance are required. See “Chapter: Clock” and “Chapter: Interrupt Control (Interrupt Controller)”.

Program example 1

```
void OUTPUT01_sample(void)
{
        freerun1_initial();
        OUTPUT01_initial();
        OUTPUT01_start();
        freerun1_start();
}

void freerun1_initial(void)
{
        IO_TCCS1.word = 0x0041;         /* Setting value =0000_0000_0100_0001 */
                                        /* bit15 = 0     ECKE internal clock source */
                                        /* bit14 -10 =0   Reserved Bit */
                                        /* bit9 = 0      ICLR compare interrupt flag clear */
                                        /* bit8 = 0      ICLR interrupt disable */
                                        /* bit7 = 0      Reserved Bit */
                                        /* bit6 = 1      STOP Counting disable */
                                        /* bit5 = 0      Reserved Bit */
                                        /* bit4 = 0      SCLR free-run timer value (no) initialization */
                                        /* bit3-0 = 0001  CLK3-0  count clock PCLK/2=32MHz/2 */
        IO_TCDT1 = 0x0000;              /* timer data value initialization */
}

void OUTPUT01_initial(void)
{
        PORT_SETTING_OCU0_OUT();  /* Set the OCU0 pin for peripheral input. */
        PORT_SETTING_OCU1_OUT();  /* Set the OCU1 pin for peripheral input. */

        IO_OCFS01.hword = 0x0003;       /* Select the free-run timer 1. */
        IO_OCS01.hword = 0xEC0C;        /* Setting value =1110_1100_0000_1100 */
                                        /* bit15-13 = 111   Undefined bit*/
                                        /* bit12 = 0       CMOD ch.0, ch.1 level invert */
                                        /* bit11-10 = 11    Undefined bit*/
                                        /* bit9-8 = 00      OTD1,OTD0 Compare pin output 0 */
                                        /* bit7-6 = 00      IOP1,IOP0 Output compare no match */
                                        /* bit5-4 = 00      IOE1,IOE0 Output compare interrupt disable */
                                        /* bit3-2 = 11      Undefined bit*/
                                        /* bit1-0 = 00      CST1,CST0 Compare operation disable */
        IO_OCCP0 = BFFF                 /* Setting of compare register ch.0 */
        IO_OCCP1 = 7FFF                 /* Setting of compare register ch.1 */

        IO_ICR[42].byte = 0x10;         /* Output compare ch.0 interrupt level setting (any value) */
        IO_ICR[43].byte = 0x10;         /* Output compare ch.1 interrupt level setting (any value) */
        __EI();                         /* Interrupt enable */
}

void OUTPUT01_start(void)
{
        IO_OCS01.hword = 0xEC3C;        /* bit5-4 = 11 IOE1,IOE0 Output compare interrupt enable */
        IO_OCS01.hword = 0xEC3F;        /* bit1-0 = 11 CST1,CST0 Compare operation enable */
}

void freerun1_start(void)
{
        IO_TCCSLL1.bit.STOP = 0;        /* bit4 = 0 STOP Counting enable */
}

__interrupt void INPUT0_int(void)
{
        IO_OCSL01.byte & = 0xBF;        /* bit6 = 0 IOP0 Clearing of interrupt flag */
        ……
}
__interrupt void INPUT0_int(void)
{
        IO_OCSL01.byte & = 0x7F;        /* bit7 = 0 IOP1 Clearing of interrupt flag */
        ……
}

Interrupt routine specification with the vector table is required.
#pragma intvect OUTPUT0_int 58
#pragma intvect OUTPUT1_int 59
```

Configuration procedure example 2

.Compare for two pairs
Output of ch.1
Compare operation (7FFF, BFFF)
Compare is cleared with a cycle of a larger compare value.
Interrupt occurrence

1. Initial setting

- Control of free-run timer ch.1 Register name.bit name

Setting of control register	TCCSH1,TCCSL1
Clock selection>>	.ECKE
Compare interrupt request flag>>	.ICLR
Compare interrupt request enable>>	.ICRE
Counting Operation>>	.STOP
TCDT clear>>	
Count clock>>	.SCLR .CLK3-0
Setting of the timer data value	TCDT1

- Port Register name.bit name

Port OCU1 output setting	See "Chapter: I/O Port".

- Output compare control Register name.bit name

Free-run timer selection	OCFS01
Setting of control register	OCSH01,OCSL01
Pin output level invert operation>>	.CMOD
Pin output level specification>>	.OTD1,OTD0
Interrupt request flag>>	.IOP1,IOP0
Interrupt request enable>>	.IOE1,IOE0
Operation enable setting>>	.CST1,CST0
Setting of the compare value ch.0	OCCP0
Setting of the compare value ch.1	OCCP1

- Interrupt relation Register name.bit name

Setting of an interrupt level.	ICR42 ICR43
Setting of I flag	(CCR)

2. Activation

- Output compare activation Register name.bit name

Interrupt control	OCSL01.IOE1
Compare operation activation	OCSL01.CST1 OCSL01.CST0

- Free-run timer ch.1 activation Register name.bit name

Counting operation activation	TCCS1.STOP

3. Interrupt

- Interrupt process Register name.bit name

Clearing of interrupt request flag	OCSL01.IOP0
(any process)	
.....	

4. Interrupt vector

- Setting of the vector table

(Note)
Clock-related setting and setting of __set_il(numerical value) in advance are required. See “Chapter: Clock” And “Chapter: Interrupt Control (Interrupt Controller)”.

Program example 2

```
void OUTPUT23_sample(void)
{
        freerun1_initial();
        OUTPUT01_initial();
        OUTPUT01_start();
        freerun1_start();
}

void freerun1_initial(void)
{
        IO_TCCS1.word = 0x0041;         /* Setting value =0000_0000_0100_0001 */
                                        /* bit15 = 0     ECKE internal clock source */
                                        /* bit14 -10 =0   Reserved Bit */
                                        /* bit9 = 0      ICLR interrupt flag clear */
                                        /* bit8 = 0      ICLR interrupt disable */
                                        /* bit7 = 0      Reserved Bit */
                                        /* bit6 = 1      STOP Counting disable */
                                        /* bit5 = 0      Reserved Bit */
                                        /* bit4 = 0      SCLR free-run timer value (no) initialization */
                                        /* bit3-0 = 0001  CLK3-0 count clock PCLK/2=32MHz/2 */
        IO_TCDT1 = 0x0000;              /* timer data value initialization */
}

void OUTPUT01_initial(void)
{
        PORT_SETTING_OCU0_OUT();  /* Set the OCU1 pin for peripheral input. */

        IO_OCFS01.hword = 0x0003;       /* Select the free-run timer 1. */
        IO_OCS01.hword = 0xEC0C;        /* Setting value =1110_1100_0000_1100 */
                                        /* bit15-13 = 111   Undefined bit */
                                        /* bit12 = 0       CMOD ch.0,ch.1 Level invert */
                                        /* bit11-10 = 11    Undefined bit */
                                        /* bit9-8 = 00     OTD1,OTD0 Compare pin output 0*/
                                        /* bit7-6 = 00     IOP1,IOP0 Output compare no match */
                                        /* bit5-4 = 00     IOE1,IOE0 Output compare interrupt disable */
                                        /* bit3-2 = 11     Undefined bit*/
                                        /* bit1-0 = 00     CST1,CST0 Compare operation disable */
        IO_OCCP0 = BFFF                 /* Setting of compare register ch.0 */
        IO_OCCP1 = 7FFF                 /* Setting of compare register ch.1 */

        IO_ICR[42].byte = 0x10;         /* Output compare ch.0 interrupt level setting (any value) */
        IO_ICR[43].byte = 0x10;         /* Output compare ch.1 interrupt level setting (any value) */
        __EI();                         /* Interrupt enable */
}

void OUTPUT01_start(void)
{
        IO_OCS01.hword = 0xEC3C;        /* bit5-4 = 11 IOE1,IOE0 Output compare interrupt enable */
        IO_OCS01.hword = 0xEC3F;        /* bit1-0 = 11 CST1,CST0 Compare operation enable */
}

void freerun1_start(void)
{
        IO_TCCS1.bit.STOP = 0;          /* bit4 = 0 STOP Counting enable */
}


__interrupt void INPUT0_int(void)
{
        IO_OCSL01.byte & = 0xBF;   /* bit6 = 0 IOP0 Clearing of interrupt flag */
        ……
        IO_OCSL01.byte & = 0x7F;   /* bit7 = 0 IOP1 Clearing of interrupt flag */
        ……
}
```

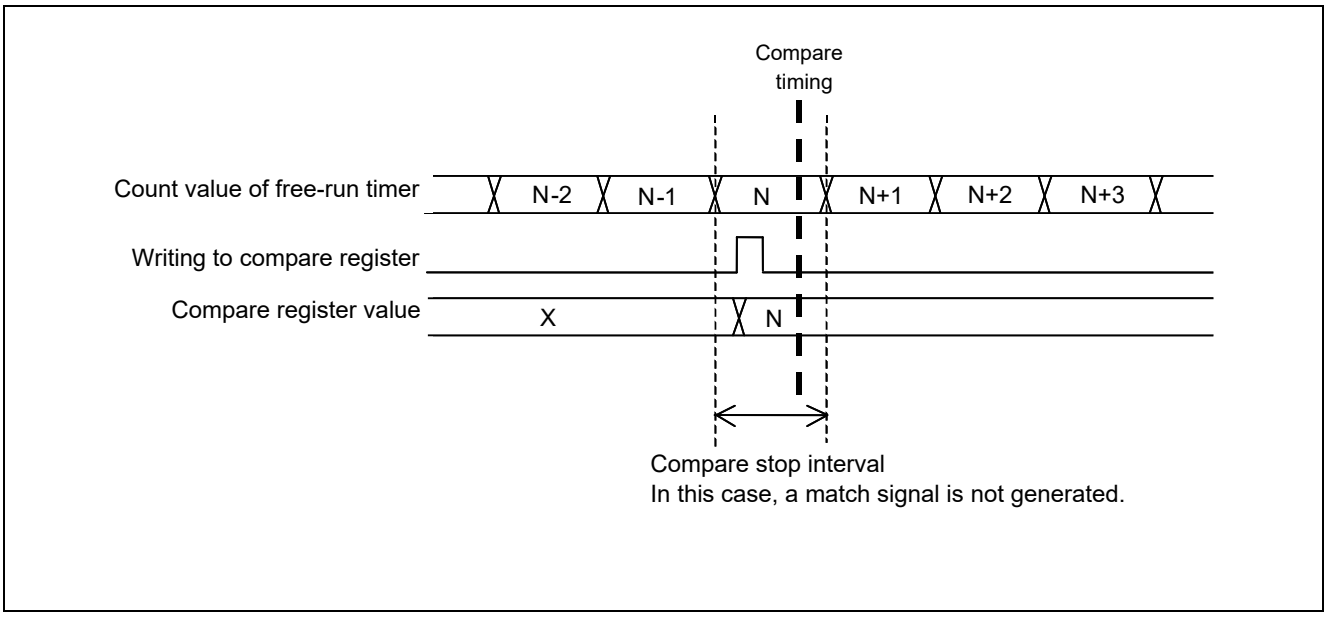
Interrupt routine specification with the vector table is required.
#pragma intvect OUTPUT1_int 59

22.9 Notes

This section explains the notes of the output compare.

- About the Compare Stop Interval During Compare Operation

  For one count right after the writing of a compare value to the compare register, there is no compare operation as shown below.



  - ☐ For the setting of CMOD= "1" and OCCP0 = OCCP1, OCCP2 = OCCP3, when compare match occurs, the port inverts only once. (Similar in ch.4 to ch.11)
  - ☐ When the output level of compare pins (OCU0, OCU1, OCU2, OCU3, ...) is specified, first stop the compare operation, and then specify it.
  - ☐ Because the output compare is synchronized with the free-run timer, when the free-run timer is stopped, the compare operation also is stopped.
  - ☐ When the compare mode bit is set to CMOD = "1" also, the interrupt operation occurs for each OCU0, OCU1, OCU2, OCU3, ... , OCU11 independently.
  - ☐ When the free-run timer is used as the compare data of the output compare, the setting of "0000_B"($1/F_{PCLK}$) is disabled for the free-run timer clock frequency TCCSL:CLK[3:0].

- Read-modify-write

  When the interrupt request flag bits (IOP0), (IOP1), (IOP2), (IOP3), ... , (IOP11) are read with read-modify-write (RMW) instruction, "1" is read.

23. Up/Down Counter

This chapter explains the up/down counter.

23.1 Overview

23.2 Features

23.3 Configuration

23.4 Registers

23.5 Interrupt

23.6 Operation and Setting Procedure Examples

23.1 Overview

This section explains the overview of the up/down counter.

The up/down counter counts up or down depending on the setting.

The 16-bit up/down counter can be used as an 8-bit up/down counter by using its low-order byte only.

The 8-bit up/down counter can count up or down in the range of "00_H" to "FF_H". The 16-bit up/down counter can count up or down in the range of "0000_H" to "$FFFF_H$".

This product incorporates 2 channels of the 16-bit up/down counter. However, only the low-order byte can be used as the 8-bit up/down counter. So, the number of channels usable for 8 and 16 bits is 2 in total.

23.2 Features

This section explains the features of the up/down counter.

- Counter mode: You can select one of the following two:
  - 8-bit up/down counter (8-bit mode)
  - 16-bit up/down counter (16-bit mode)
- Operating mode: You can select one of the following three (four types):
  - Timer mode

    The time is counted down in synchronization with the count clock.

    As the count clock, the internal clock is used which is generated by dividing the peripheral clock (PCLK) by 2 or 8 using the prescaler.
  - Up/down count mode

    Signals entered from the two external signal input pins are counted up or down. The edge to be counted can be selected from among the rising edge, falling edge, and both edges.
  - Phase difference count mode

    The phase difference of signals entered from the two external signal input pins are counted up or down.

    The phase difference count mode is suitable for counting of encoders such as motors. This mode enables high-precision counting of rotation angles, number of rotations and the like, by inputting outputs of phases A, B, and Z of the encoder.

    There are two types of phase difference count mode: the two-time multiplication mode and four-time multiplication mode. The counting differs between the two mode types.

Table 23-1 lists the up/down counter operating modes.

Table 23-1. Up/Down Counter Operating Modes

Operation Mode	Count Timing	Count Direction
Timer mode	Internal clock	Count down
Up/down count mode	External clock	Count up/Count down
Phase difference count mode (multiply-by 2/ multiply-by 4)	Phase of the input signal from an external signal input pin	Count up/Count down

- Reload compare function: You can select one of the following three:
  - Compare function

    The compare function clears the counter and continues counting when counting reaches the preset value.
  - Reload function

    The reload function loads the reload value and continues counting if an underflow occurs.
  - Reload compare function

    Both the compare function and reload function can be combined for use.

- Counting direction: The last counting direction (count up/count down) can be checked.
- Interrupt request: An interrupt request can be generated in one of the following events:
  - The counting direction was inverted.
  - The counter value matches the preset value.
  - An overflow occurs.
  - An underflow (reload) occurs.

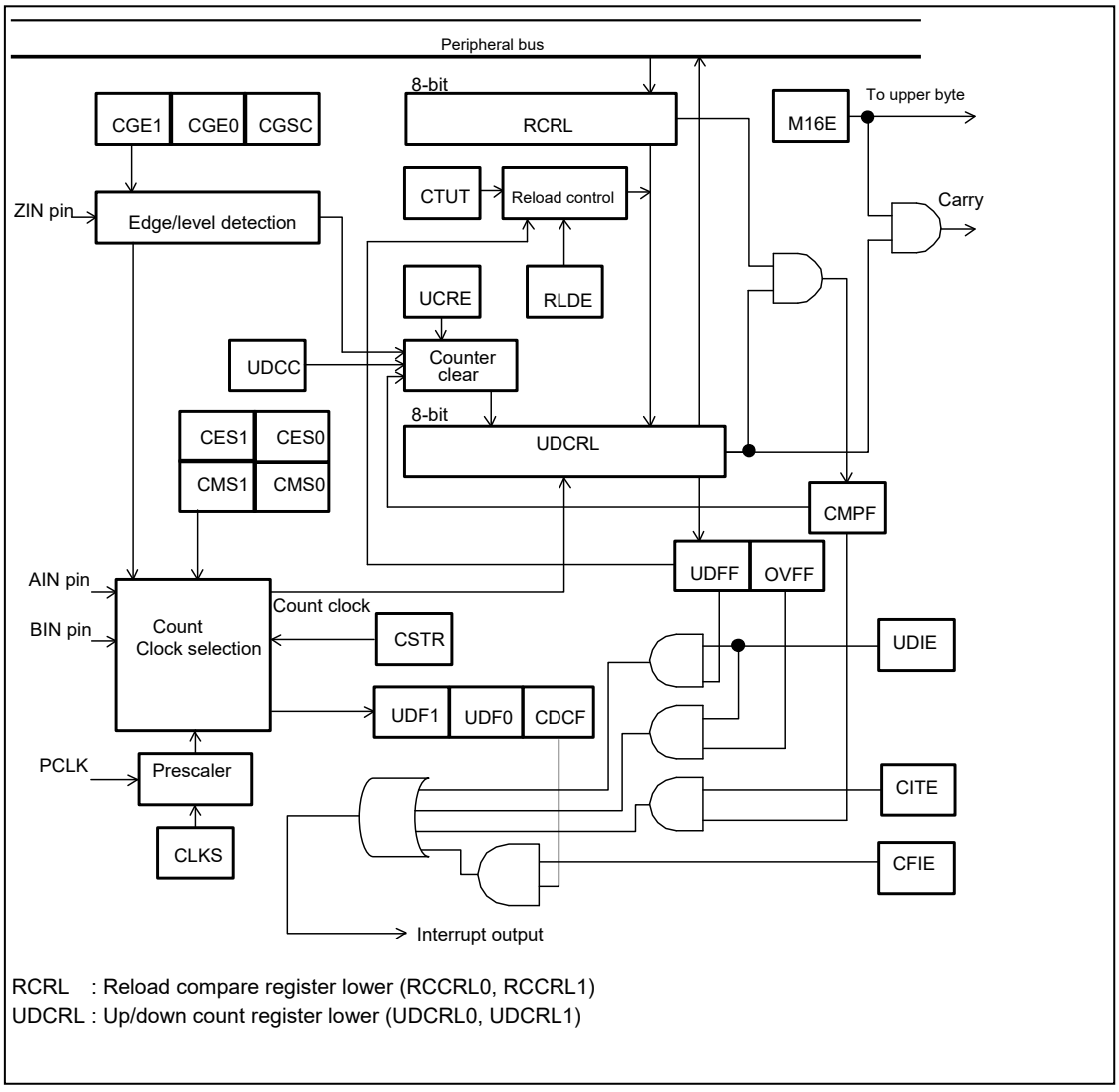
23.3 Configuration

This section explains the configuration of the up/down counter.

Block Diagram of the Up/Down Counter

Figure 23-1 shows the block diagram of the up/down counter using ch.0 as an example.

Figure 23-1. Block Diagram of the Up/Down Counter



RCRL : Reload compare register lower (RCCRL0, RCCRL1)
UDCRL : Up/down count register lower (UDCRL0, UDCRL1)

- Reload compare register (RCR)

  This register sets reload and compare values of the up/down counter.
  As shown below, this counter consists of upper 8 bits and lower 8 bits.
  To use the register in 8-bit mode, use the lower side.

  - ☐ Reload compare register upper (RCRH)
  - ☐ Reload compare register lower (RCRL)

- Up/down count register (UDCR)

  This register operates as the counter for the up/down counter.
  As shown below, this counter consists of upper 8 bits and lower 8 bits.
  To use the register in 8-bit mode, use the lower side.

  - ☐ Up/down count register upper (UDCRH)
  - ☐ Up/down count register lower (UDCRL)

- Counter control register (CCR)

  This register controls the up/down counter.

- Counter status register (CSR)

  This register checks the up/down counter status or controls an interrupt request.

- Count clock selection circuit

  This circuit is used to select a count clock of the up/down counter.

- Prescaler

  In using the up/down counter in the timer mode, this prescaler is used to select a division ratio of the peripheral clock (PCLK).

Clock

Table 23-2 lists the clocks used for the up/down counter.

Table 23-2. Clocks Used for the Up/Down Counter

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Internal clock (peripheral clock)	Generated by dividing the peripheral clock (PCLK)
	Counting of inputs from an external pin	Input from AIN and BIN pins

23.4 Registers

This section explains the registers of the up/down counter.

Correspondence between Pins and Channels

Table 23-3 shows the correspondence between channels and pins.

Table 23-3. Correspondence between Pins and Channels

Channel Number	External Signal Input Pins		
0	AIN0_0, AIN0_1, AIN0_2	BIN0_0, BIN0_1, BIN0_2	ZIN0_0, ZIN0_1, ZIN0_2
1	AIN1_0, AIN1_1	BIN1_0, BIN1_1	ZIN1_0, ZIN1_1

Ch.0 and ch.1 select the external pin used by the IO relocation function.

Registers Map

Table 23-4 lists the up/down counter register map.

Table 23-4. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0F70	RCRH0	RCRL0	UDCRH0	UDCRL0	Reload compare register upper 0 Reload compare register lower 0 UP/down count register upper 0 UP/down count register lower 0
0x0F74	CCR0		Reserved	CSR0	Counter control register 0 Counter Status register 0
0x0F80	RCRH1	RCRL1	UDCRH1	UDCRL1	Reload compare register upper 1 Reload compare register lower 1 UP/down count register upper 1 UP/down count register lower 1
0x0F84	CCR1		Reserved	CSR1	Counter control register 1 Counter Status register 1

23.4.1 Reload Compare Register (RCR0, RCR1)

The bit configuration of the reload compare register is shown below.

This register sets reload and compare values of the up/down counter.

The reload value is the one from which counting starts at counting down; the compare value is compared with the value counted at counting up (in other words, this value indicates that counting continues until this value is reached). The reload and compare values are the same.

RCRH0: Address 0F70$_H$ (Access: Half-word, Word)

RCRH1: Address 0F80$_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D9	D8
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

RCRL0: Address 0F71$_H$ (Access: Byte, Half-word, Word)

RCRL1: Address 0F81$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
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

As shown below, this register consists of a high-order byte and a low-order byte.

- Reload compare register high-order (RCRH0, RCRH1)
- Reload compare register low-order (RCRL0, RCRL1)

In the 16-bit mode, both byte values are used. In the 8-bit mode, the low-order value is used.

When the value written in this register is transferred to the up/down count register (UDCR), the up/down counter performs counting in the range from "0000$_H$" ("00$_H$" for 8 bits) to that value set in this register.

Notes:

- When "1" is written to the CTUT bit of the counter control register (CCR), a value set in this register can be transferred to the up/down count register (UDCR). However, write the value in this CTUT bit of the counter control register (CCR) while the up/down counter stops.
- If the 16-bit mode is set with the M16E bit of the counter control register (CCR) (M16E=1), this register must always be written by half-word access.
- If the 8-bit mode is set with the M16E bit of the counter control register (CCR) (M16E=0), this register must always be written in the reload compare register low-order side (RCRL) by byte access.

23.4.2 Up/Down Count Register (UDCR0, UDCR1)

The bit configuration of the up/down count register is shown below.

This register operates as the counter for the up/down counter. The counter value can be checked by reading these registers.

UDCRH0: Address 0F72H (Access: Half-word, Word)

UDCRH1: Address 0F82H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D15	D14	D13	D12	D11	D10	D9	D8
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

UDCRL0: Address 0F73H (Access: Byte, Half-word, Word)

UDCRL1: Address 0F83H (Access: Byte, Half-word, Word)

	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

As shown below, this register consists of a high-order byte and a low-order byte.

- Up/down count register high-order (UDCRH0, UDCRH1)
- Up/down count register low-order (UDCRL0, UDCRL1)

In the 8-bit mode, the high-order byte value is invalid.
The low-order byte of the up/down count register (UDCRL) must be read.

Notes:

- This is a read-only register. To set a value in this register, transfer the reload compare register (RCR) value to this register in the following procedure:
  1. Write a value in the reload compare register (RCR)
  2. Write the CSTR bit of the counter status register (CSR) to "0"
  3. Write the CTUT bit of the counter control register (CCR) to "1"
- If the 16-bit mode is set with the M16E bit of the counter control register (CCR) (M16E=1), this register must always be read by half-word access.
- If the 8-bit mode is set with the M16E bit of the counter control register (CCR) (M16E=0), the low-order side of the up/down count register (UDCRL) must be read.

23.4.3 Counter Control Register (CCR0, CCR1)

The bit configuration of the counter control register is shown below.

This register controls the up/down counter operations.

CCR0: Address 0F74H (Access: Byte, Half-word)

CCR1: Address 0F84H (Access: Byte, Half-word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0
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/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0
Initial value	0	0	0	0	1	0	0	0
Attribute	R0,W0	R0,W	R/W	R/W	R1,W	R/W	R/W	R/W

[bit15] M16E: 16-bit mode selection bit

This bit specifies that the up/down counter is used in 8-bit mode or in 16-bit mode.

Write Value	Description
0	Uses the counter in the 8-bit mode (1 channel).
1	Uses the counter in the 16-bit mode (1 channel).

[bit14] CDCF: Count direction change flag bit

This bit indicates that the counting direction has changed from counting down to counting up or from counting up to counting down once or more.

When this bit is "1" and the CFIE bit is set to "1", a counting direction change interrupt request is generated.

CDCF	Read	Write
0	The counting direction is not changed.	This bit is cleared to "0".
1	The counting direction was changed once or more.	Ignored

Notes:

- If the counter is reset, the counting down direction is set. Therefore, if the counting up is set immediately after the reset, this bit is changed to "1".
- If the counting direction is continuously changed in a short time, the counting direction may be returned to the original direction and the UDF1 and UDF0 bits of counter status register (CSR) may not change.

[bit13] CFIE: Counting direction change interrupt enable bit

Sets whether or not to generate a counting direction change interrupt request when the counting direction is changed (CDCF=1).

Write Value	Description
0	Disables to generate a counting direction change interrupt request.
1	Enables to generate a counting direction change interrupt request.

[bit12] CLKS: Internal clock division selection bit

This bit specifies that the peripheral clock (PCLK) divided by the division ratio (set by this bit) is used as the count clock when the timer mode is selected.

Write Value	Description
0	Peripheral clock (PCLK) divided by 2
1	Peripheral clock (PCLK) divided by 8

Note:

This bit is valid only if the timer mode has been set with the CMS1 and CMS0 bits (CMS1, CMS0=00). This bit setting is ignored if another operating mode has been selected.

[bit11, bit10] CMS1, CMS0: Operating mode selection bits

Select an operating mode of the up/down counter as follows.

- Timer mode

  The time is counted down in synchronous with the count clock.

- Up/down count mode

  Input signals entered from the two external signal input pins are counted up or down.

- Phase difference count mode

  A phase difference at the two external signal input pins is counted up or down. There are two types of phase difference count mode: the two-time multiplication mode and four-time multiplication mode. The counting differs between the two mode types.

CMS1	CMS0	Operation Mode
0	0	Timer mode
0	1	Up/down count mode
1	0	Phase difference count mode (multiply-by-2)
1	1	Phase difference count mode (multiply-by-4)

[bit9, bit8] CES1, CES0: Count clock edge selection bits

Select a detection edge of the AIN and BIN pins.

If the up/down count mode is selected, the signal is counted each time a signal edge selected by these bits is detected.

CES1	CES0	Detection Edge
0	0	Disables signal edge detection
0	1	Falling edge
1	0	Rising edge
1	1	Both edges

Note:

These bits are valid only if the up/down count mode has been set by the CMS1 and CMS0 bits (CMS1, CMS0=01). This bit setting is ignored if another operating mode has been selected.

[bit7] Reserved bit

Write	This bit must always be written to "0".
Read	"0" is read.

[bit6] CTUT: Counter write bit

This bit transfers a value being set in the reload compare register (RCR) to the up/down count register (UDCR).

CTUT	Read	Write
0	"0" is read.	Ignored
1		The value is transferred.

Note:

When this bit is written to "1", the reload compare register (RCR) value is transferred. Therefore, if the CSTR bit of counter status register (CSR) is "1" (the counter is operating), this bit must not be rewritten to "1".

[bit5] UCRE: Counter clear enable bit

This bit enables or disables to use the compare function.

The compare function clears the counter value to "0000H" and continues counting if the counter value matches the value being set in the reload compare register (RCR).

Write Value	Description
0	Disables to use the compare function.
1	Enables to use the compare function.

Note:

This bit can only clear the counter value using the compare function.

This bit cannot control the following clearing operations.

- Clear the counter when this device is reset.
- Clear the counter when an effective edge signal is input from the ZIN pin (if CGSC bit is 0).
- Clear the counter by writing the UDCC bit to "0". (Software-triggered clear)

[bit4] RLDE: Reload enable bit

This bit enables or disables to use the reload function.

The reload function continues counting by reloading the value, being set in the reload compare register (RCR), onto the counter when the counter has underfollowed during counting down.

Write Value	Description
0	Disables to use the reload function.
1	Enables to use the reload function.

[bit3] UDCC: Counter clear bit

Clears the counter value to "0000_H".

UDCC	Read	Write
0	"1" is read.	This bit is cleared to "0".
1		Ignored

[bit2] CGSC: Counter clear/gate selection bit

This bit selects a function to be assigned to the ZIN pin as follows.

- Counter clear function

  Clears the counter value to "0000H" when an effective edge signal is entered from the ZIN pin.
- Gate function

  Operates the counter only when an effective level of signal is being entered from the ZIN pin.

Write Value	Description
0	Counter clear function
1	Gate function

Note:

The ZIN pin functions if a combination of this bit and CGE1 and CGE0 bits is set.
Therefore, the CGE1 and CGE0 bits must always be set.

[bit1, bit0] CGE1, CGE0: Edge/level selection bits

These bits select an effective edge or an effective level of signal at the ZIN pin. The meaning of these bits depends on the CGSC bit setting as follows.

- If the counter clear function is selected by the CGSC bit (if CGSC=0)

  An effective edge of signal is selected.
  When a signal edge, selected by this bit, is detected at the ZIN pin, the counter value is cleared to "0000_H".

- If the gate function is selected by the CGSC bit (if CGSC=1)

  An effective level of signal is selected.
  The counter operates only when a signal having the level, selected by this bit, is being entered from the ZIN pin.

CGE1	CGE0	If the Counter Dlear Function is Selected (CGSC=0)	If the Gate Function is Selected (CGSC=1)
0	0	Disables signal edge detection.	Disables signal level detection (disabled counting)
0	1	Falling edge	"L" level
1	0	Rising edge	"H" level
1	1	Setting disabled	Setting disabled

23.4.4 Counter Status Register (CSR0, CSR1)

The bit configuration of the counter status register is shown below.

This register is used to check the status of the up/down counter and control interrupt requests.

CSR0: Address 0F77H (Access: Byte)

CSR1: Address 0F87H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R(RM1),W	R(RM1),W	R(RM1),W	R,WX	R,WX

[bit7] CSTR: Count activation bit

This bit starts and stops the up/down counter.

Write Value	Description
0	Stops the counting.
1	Starts the up/down counter.

[bit6] CITE: Compare result match interrupt enable bit

This bit sets whether or not to generate a compare result match interrupt request when the counter value matches the value set in the reload compare register (RCR) (CMPF=1).

Write Value	Description
0	Disables compare result match interrupt requests.
1	Enables compare result match interrupt requests.

[bit5] UDIE: Overflow/underflow interrupt enable bit

This bit sets whether or not to generate an overflow/underflow interrupt request when the up/down counter overflows/underflows (OVFF/UDFF=1).

Write Value	Description
0	Disables overflow/underflow interrupt requests.
1	Enables overflow/underflow interrupt requests.

[bit4] CMPF: Compare result match detection flag bit

This bit indicates that the counter value has matched the value set in the reload compare register (RCR).
When this bit is "1" and the CITE bit is set to "1", a compare result match interrupt request is generated.

CMPF	Read	Write
0	The value did not match.	This bit is cleared to "0".
1	The value matched.	Ignored

Note:

This bit changes to "1" in the following cases:

- The value matched in counting up.
- The value of the reload compare register (RCR) is reloaded to the counter.
- The value has already matched when the up/down counter is started.

[bit3] OVFF: Overflow detection flag bit

This bit indicates that the up/down counter has overflowed.
When this bit is "1" and the UDIE bit is set to "1", an overflow interrupt request is generated.

OVFF	Read	Write
0	No overflow has occurred.	This bit is cleared to "0".
1	An overflow has occurred.	Ignored

An overflow occurs if counting up is attempted when the counter value is "$FFFF_H$".

[bit2] UDFF: Underflow detection flag bit

This bit indicates that the up/down counter has underfollowed.
When this bit is "1" and the UDIE bit is set to "1", an underflow interrupt request is generated.

UDFF	Read	Write
0	No underflow has occurred.	This bit is cleared to "0".
1	An underflow has occurred.	Ignored

An underflow occurs if counting down is attempted when the counter value is "0000_H".

[bit1, bit0] UDF1, UDF0: Up/down flag bits

These bits indicate the last counting direction.
These bits are updated every time the up/down counter counts.

UDF1	UDF0	Description
0	0	No input
0	1	Count down
1	0	Count up
1	1	Count up/count down at the same time

23.5 Interrupt

This section shows the interrupt of the up/down counter.

An interrupt request is generated in one of the following events:

- The counting direction is inverted (Counting direction change interrupt request).
- The counter value matches the value set in the reload compare register (RCR) (Compare result match interrupt request).
- An overflow occurs (Overflow interrupt request).
- An underflow occurs (Underflow interrupt request).

Different interrupt requests are generated depending on the up/down counter operating mode.

Table 23-5 shows the correspondence between operating modes and interrupt requests.

Table 23-5. Correspondence between Operating Modes and Interrupt Requests

Interrupt Request	Timer Mode	Up/down Count Mode	Phase Difference Count Mode (Multiply-by-2/ Multiply-by-4)
Counting direction change interrupt request	×	○	○
Compare result match interrupt request	○	○	○
Overflow interrupt request	×	○	○
Underflow interrupt request	○	○	○

Table 23-6 shows interrupts that can be used for the up/down counter.

Table 23-6. Up/Down Counter Interrupts

Interrupt request	Interrupt Request Flag	Interrupt Request Enable	Clearing of Interrupt Request
Counting direction change interrupt request	CDCF=1 in CCR	CFIE=1 in CCR	Writing of CDCF bit to "0" in CCR.
Compare result match interrupt request	CMPF=1 in CSR	CITE=1 in CSR	Writing of CMPF bit to "0" in CSR.
Overflow interrupt request	OVFF=1 in CSR	UDIE=1 in CSR	Writing of OVFF bit to "0" in CSR.
Underflow interrupt request	UDFF=1 in CSR	UDIE=1 in CSR	Writing of UDFF bit to "0" in CSR.

CCR: Counter control register

CSR: Counter status register

Notes:

- Once an interrupt request is generated, the up/down counter stops operation until the interrupt request flag is cleared.
- The CMPF bit in the counter control register (CCR) changes to "1" if the value matches in counting up, if the value of the reload compare register (RCR) is reloaded, or if the value has already matched when the up/down counter is started.
- For the clearing of the counter and the reload timing, see "Clear Events" and "Reload Event" in "Operation and Setting Procedure Examples".
- 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.
- For interrupt vector numbers used for issuing an interrupt request, see "Appendix A.3. List of Interrupt Vector".
- 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 the Chapter of "Interrupt Control (Interrupt Controller)".

23.6 Operation and Setting Procedure Examples

This section explains the operation of the up/down counter. An example is also given to set operating state.

Overview

- Counter Mode

  Depending on the setting, the up/down counter can be used as a 16-bit up/down counter or an 8-bit up/down counter.

  Set the counter mode in the M16E bit in the counter control register (CCR).

  - 8-bit mode (M16E=0)

    Only the up/down count register low-order bit (UDCRL) is used. Write the reload and compare values only to the reload compare register low-order bit (RCRL) using byte access.

  - 16-bit mode (M16E=1)

    Both the high-order and low-order bytes of the up/down count register (UDCR) are used. Write the reload and compare values to the reload compare register (RCR) using half-word access.

- Operating Mode

  One of the following three modes (four types) can be selected as the operating mode of the up/down counter using the CMS1 and CMS0 bits of the counter control register (CCR).

  - Timer mode (CMS1, CMS0=00)

    The counter decrements from a preset value in synchronization with the count clock.

    The count clock is generated by dividing the peripheral clock (PCLK) by 2 or 8 using the prescaler.

  - Up/down count mode (CMS1, CMS0=01)

    The counter increments or decrements based on signals supplied from the external signal input pin.

  - Phase difference count mode (multiply-by-two) (CMS1, CMS0=10)/Phase difference count mode (multiply-by-four) (CMS1, CMS0=11)

    The counter increments or decrements based on phase differences of signals supplied from the external signal input pin. This mode is suitable for counting of encoders such as motors because it enables high-precision counting of rotation angles and number of rotations and detection of the rotation direction by entering the encoder A-phase to the AIN pin, B-phase to the BIN pin, and Z-phase to the ZIN pin.

Available Functions

- Reload/Compare Functions

  For the 8/16-bit up/down counter, the reload and compare functions can be enabled and disabled using the RLDE and UCRE bits of the counter control register (CCR).

  - Reload function

    When an underflow occurs during countdown, the value set in the reload compare register (RCR) is reloaded and counting down is restarted. For the operations, see "23.6.1 Operation in Timer Mode".

  - Compare function

    If the up/down counter value matches the value set in the reload compare register (RCR) (compare result match) and further counting up is attempted, the value of the up/down counter is cleared to "0000_H" and counting up is restarted. For the operations, see "23.6.2 Operation in Up/Down Count Mode".
    This function is not available in timer mode.

  - Reload compare function

    This function is a combination of the reload and compare functions. The counter decrements and increments between "0000_H" and a value set in the reload compare register (RCR), enabling counting in any range. See "Counting" in "23.6.2 Operation in Up/Down Count Mode".
    This function is not available in timer mode.

Table 23-7 shows the setting method for the reload/compare functions.

Table 23-7. Setting Method for Reload/Compare Functions

RLDE	UCRE	Description
0	0	Disables the reload and compare functions.
0	1	Disables the reload function. Enables the compare function
1	0	Enables the reload function. Disables the compare function.
1	1	Enables the reload and compare functions.

- Functions of ZIN Pin

  One of the following functions can be selected as the function of the ZIN pin using the CGSC bit of the counter control register (CCR).

  - ☐ Counter clear function (CGSC=0)

    If an effective edge is input from the ZIN pin during counting, the counter value is cleared to "0000_H".

  - ☐ Gate function (CGSC=1)

    Operates the counter only when an effective level of signal is being entered from the ZIN pin.

  Using the CGE1 and CGE0 bits of the counter control register (CCR), select either the effective edge if the counter clear function is selected or the effective level if the gate function is selected.

CGE1	CGE0	If the counter clear function is selected (CGSC=0)	If the gate function is selected (CGSC=1)
0	0	Disables signal edge detection.	Disables signal level detection (disabled counting)
0	1	Falling edge	"L" level
1	0	Rising edge	"H" level
1	1	Disables setting	Disables setting

Clear Events

The counter value is cleared to "0000_H" in one of the following events.

- This device is reset.
- An effective edge is entered from the ZIN pin.

  (If the CGSC bit of the counter control register (CCR) is set to make the ZIN pin work for the counter clear function (CGSC=0).)
- Software-triggered clear

  The UDCC bit of the counter control register (CCR) is written to "0".
- Clear due to the compare function

  The counter value matches the value set in the reload compare register (RCR) and an attempt is made to increment the counter.

  (The counter is not cleared if an attempt is made to decrement or stop the counter.)
- Clear due to overflow

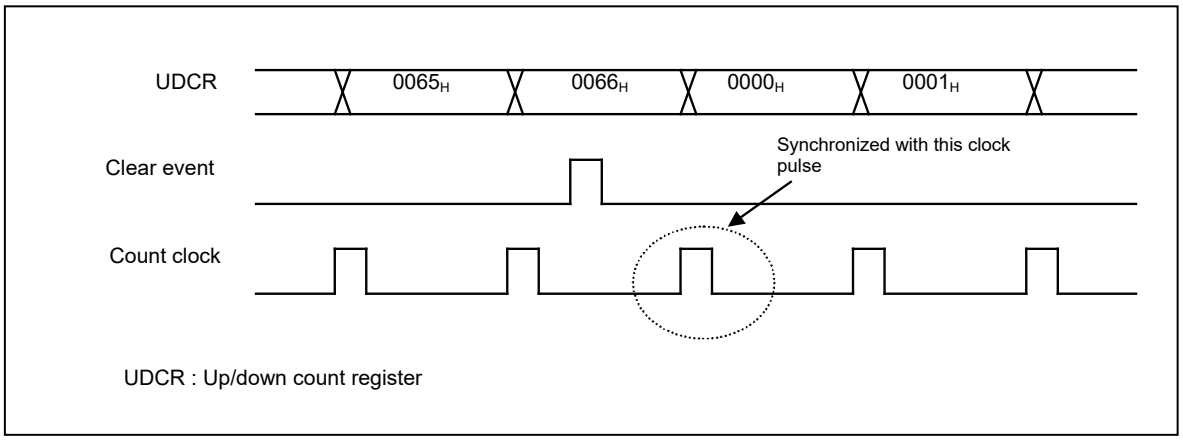
  Count up/down timing after the counter reaches "$FFFF_H$" (or "FF_H" in 8-bit mode).

The time the counter is cleared to "0000_H" depends on the up/down counter operating status as follows.

- If a clear event occurs during counting, the counter will be cleared in synchronization with the count clock.

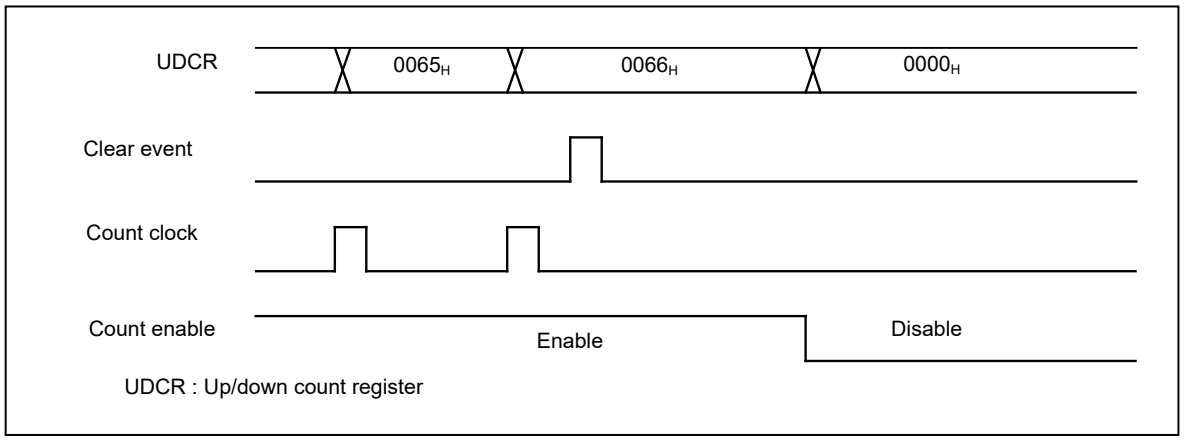
Figure 23-2 shows clear event occurrence timing.

Figure 23-2. Clear Event Occurrence Timing



- If a clear event occurs during counting and the counting stops before the next count clock pulse is entered (the CSTR bit of the counter status register (CSR) is "0"), the value will be cleared when the up/down counter stops.

Figure 23-3. Clear Event Occurrence Timing



Reload Event

The up/down counter value is reloaded in one of the following events.

- The CTUT bit of the counter control register (CCR) is written to "1".
- The reload function is activated to reload the value:

The timing the up/down counter value is reloaded depends on the up/down counter operating status as follows.

- If a reload event occurs during counting:

  The value will be reloaded in synchronization with the count clock.
- If a reload event occurs except during counting:

  The value will be reloaded when a reload event occurs.

Notes:

- During counting, do not write "1" to the CTUT bit of the counter control register (CCR).
- If a reload event and a clear event occur at the same time, the clear event takes precedence.

23.6.1 Operation in Timer Mode

This section explains the operation in timer mode.

Overview

In this mode, the up/down counter counts down from the value set in the reload compare register (RCR). The frequency of the peripheral clock (PCLK) is divided by the prescaler to ensure that the result can be used as the count clock.

It is also possible to use the reload function in order to reload the value of the reload compare register (RCR) when the counter underflows, so that counting-down can be restarted from the reloaded value.

Counting

- Normal Operation
  - ☐ The reload/compare value is set in the reload compare register (RCR).
  - ☐ When "1" is written to the CTUT bit of the counter control register (CCR), the set value is transferred to the up/down count register (UDCR).
  - ☐ When "1" is written to the CSTR bit of the counter status register (CSR) to enable up/down counter operation, the counter begins to count down from the value set in the reload compare register (RCR).

  When the counter underflows, the UDFF bit of the counter status register (CSR) changes to "1". At this time, an underflow interrupt request occurs if the UDIE bit of the counter status register is set to "1".

  If the CGSC bit of the counter control register (CCR) is set to make the ZIN pin work for the gate function (CGSC=1), the counter will only count while the effective level specified by the CGE1 and CGE0 bits is entered from the ZIN pin.

  For information on effective level setting, see "23.4.3 Counter Control Register (CCR0, CCR1)".

  Note:

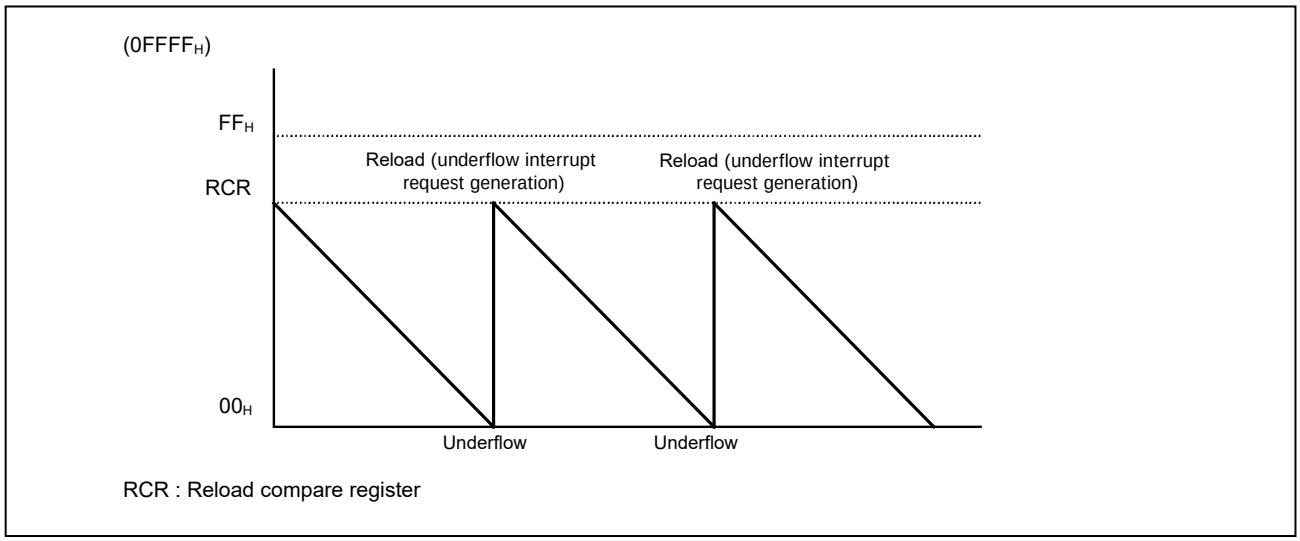
  The minimum pulse width required at the ZIN pin is 2T (T is the cycle of the peripheral clock (PCLK)).

- Operation Performed When the Reload Function is in Use

  When the counter underflows during counting down, the UDFF bit of the counter status register (CSR) changes to "1". At the time of the next count-down operation after the occurrence of underflow, the value of the reload compare register (RCR) is reloaded to the counter, which then resumes counting down. At this time, an underflow interrupt request occurs if the UDIE bit of the counter status register (CSR) is set to "1".

  Figure 23-4 shows the operation performed when the reload function is in use.

Figure 23-4. Operation Performed When the Reload Function is in Use



Note:

The value of the reload compare register (RCR) serves as both the reload value and compare value. Therefore, when a value is reloaded to the reload compare register (RCR), the CMPF bit of the counter status register (CSR) also changes to "1".

23.6.2 Operation in Up/Down Count Mode

This section explains the operation in up/down count mode.

Overview

In this mode, the up/down counter counts up/down with count clocks that are external signals entered from the AIN and BIN pins.

When the external signal is entered from the AIN pin, the up/down counter counts up. When the external signal is entered from the BIN pin, the up/down counter counts down.

Which edge of the external signal is used to trigger counting is determined by the CES1 and CES0 bits of the counter control register (CCR) as follows:

- Falling edge (CES1, CES0=01)
- Rising edge (CES1, CES0=10)
- Both edges (CES1, CES0=11)

In up/down count mode, the following three functions can be used:

- Reload function
- Compare function
- Reload compare function

Counting

- Normal Operation

  When the effective edge is entered from the AIN pin while the counter is enabled to operate, the counter counts up. When it is entered from the BIN pin while the counter is enabled to operate, the counter counts down.

  When the counter changes its counting direction from counting up to counting down or vice versa, the CDCF bit of the counter control register (CCR) changes to "1". At this time, a counting direction change interrupt request occurs if the CFIE bit of the counter control register (CCR) is set to "1".

  If the CGSC bit of the counter control register (CCR) is set to make the ZIN pin work for the gate function (CGSC=1), the counter will only count while the effective level specified by the CGE1 and CGE0 bits is entered from the ZIN pin.

  For information on effective level setting, see "Counter Control Register (CCR0".

  Note:

  The minimum pulse width required at the AIN, BIN, and ZIN pins is 2T (T is the cycle of the peripheral clock (PCLK)).

- Operation Performed When the Reload Function is in Use

  The operation is similar to that in timer mode. See "Counting" in "23.6.1 Operation in Timer Mode".

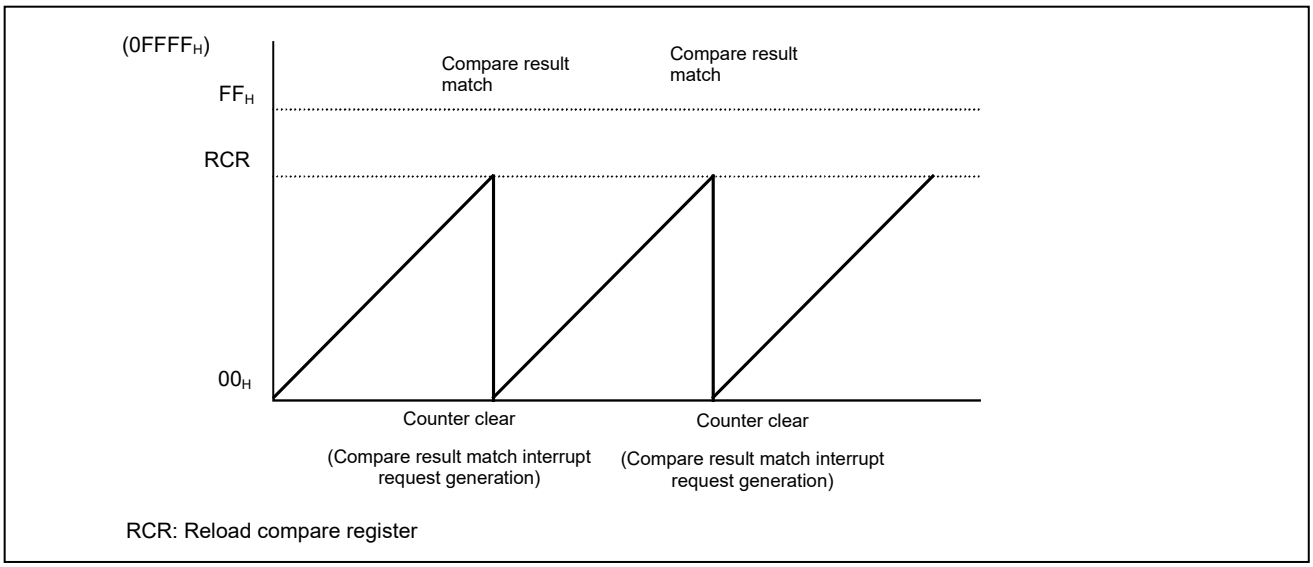
- Operation Performed When the Compare Function is in Use

  When the up/down counter value matches the value set in the reload compare register (RCR), the CMPF bit of the counter status register (CSR) changes to "1". At this time, a compare result match interrupt request occurs if the CITE bit of the counter status register (CSR) is set to "1".

  If an attempt is made to further increment the counter in this condition, the up/down counter value is cleared to "0000_H" and counting-up restarts.

  Figure 23-5 shows the operation performed when the compare function is in use.

Figure 23-5. Operation Performed When the Compare Function is in Use



Notes:

If the compare function is in use, the up/down counter value will be cleared to "0000_H" when one of the following conditions is fulfilled.

- The up/down counter value matches the value set in the reload compare register (RCR) (compare result match) and further, the next counting up operation is performed.

  However, a comparison result match does not cause clearing of the up/down counter value if one of the following conditions is fulfilled:

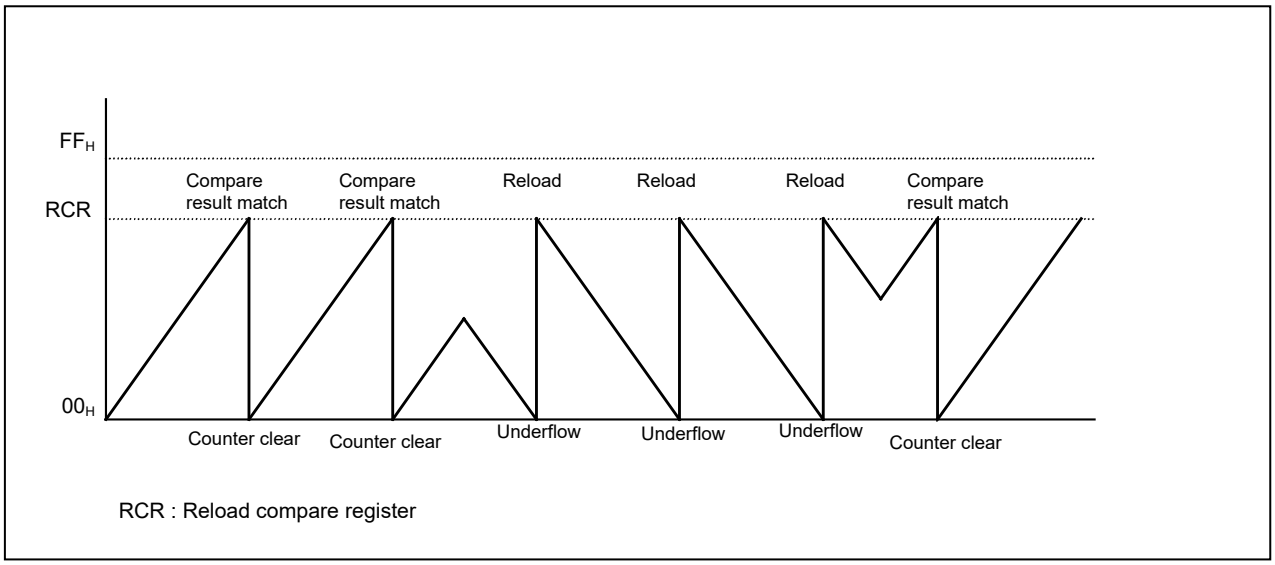
- The next operation is counting down.
- The up/down counter is inactive.

- Operation Performed When the Reload Compare Function is in Use

  The reload function is used at counting down and the compare function is used at counting up.

  Figure 23-6 shows operation performed when the reload compare function is in use.

Figure 23-6. Operation Performed When the Reload Compare Function is in Use



- Checking Counting Direction

  This mode involves both the counting up and counting down. So, the counting direction can be confirmed with the UDF1 and UDF0 bits of the counter status register (CSR). These bits are rewritten each time counting occurs, so enabling the current counting direction to be checked. These bits are useful to know the rotation direction during motor control or the like.

  Table 23-8 lists the counting directions indicated with the UDF1 and UDF0 bits.

Table 23-8. Correspondence between UDF1 and UDF0 Bits and Counting Directions

UDF1	UDF0	Count Direction
0	0	No input
0	1	Counting down
1	0	Counting up
1	1	Concurrent generation of counting up and counting down

If the counting direction is inverted one or more times from the counting down to counting up or vice versa, the CDCF bit of the counter control register (CCR) changes to "1". In this case, a direction change interrupt request can also be generated. So, using the CDCF bit and the direction change interrupt request, you can check whether the counting direction has been inverted.

Note:

If the counting direction is continuously changed in a short period of time, the counting direction is restored and so the direction indicated with the UDF1 and UDF0 bits of the counter status register (CSR) may be the same as the direction set before the CDCF bit changes to "1".

23.6.3 Operation in the Phase Difference Count Mode (Multiply-by-Two)

This section explains the operation in the phase difference count mode (multiply-by-two).

Overview

This mode involves counting the phase difference of the signal input from two external signal input pins. This mode is suitable to count the phase difference of phases A and B of encoder outputs.

When a rising edge or falling edge is detected from the BIN pin, the input level of the AIN pin is verified to count up or down the phase difference of the BIN and AIN pins. If phase A advances faster than phase B, their phase difference is counted up. If the former is delayed more than the latter, their phase difference is counted down.

Counting up or counting down is determined depending on the BIN pin detection edge and AIN pin input level.

Table 23-9 lists the count methods.

Table 23-9. Count Methods

BIN Pin	AIN Pin	Count Direction
Rising edge	"H" level	Counting up
	"L" level	Counting down
Falling edge	"H" level	Counting down
	"L" level	Counting up

Moreover, the following three types of functions can be used in the phase difference count mode (multiply-by-two).

- Reload function
- Compare function
- Reload compare function

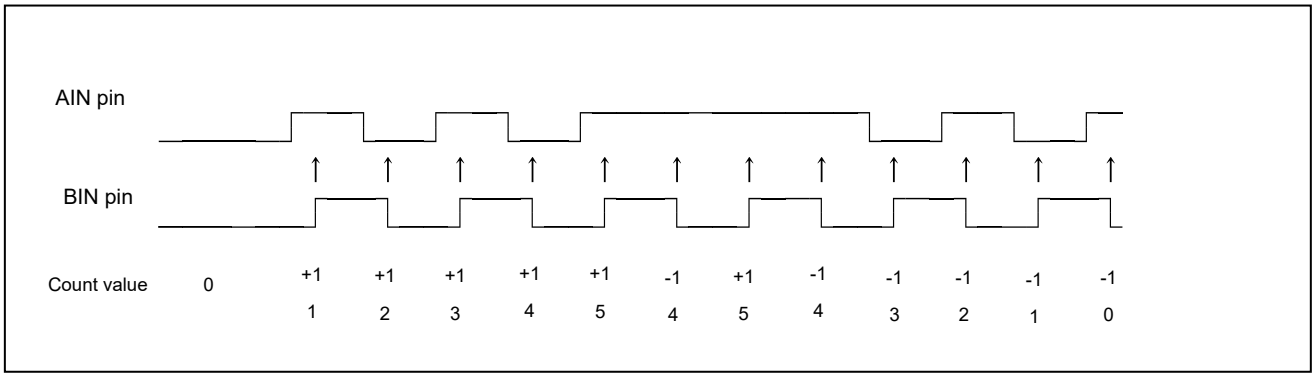
Count Operation

- Normal Operation

  If the counter is operable and the rising or falling edge is input from the BIN pin, the input level of the AIN pin is detected and the counter counts up or down.

  Figure 23-1 shows the operation in the phase difference count mode (multiply-by-two).

Figure 23-7. Operation in the Phase Difference Count Mode (Multiply-by-two)



If, however, the ZIN pin is set as the gate function (CGSC=1) with the CGSC bit of the counter control register (CCR), counting occurs only while the effective level set with the CGE1 and CGE0 bits is input from the ZIN pin.

For information on effective level setting, see "Counter Control Register (CCR0".

Note:

The minimum pulse width required at the AIN, BIN, and ZIN pins is 2T (T is the cycle of the peripheral clock (PCLK)).

- Operation Performed When the Reload Function is in Use

  The operation is similar to that in timer mode. See "Counting" in "23.6.1 Operation in Timer Mode".

- Operation Performed When the Compare Function is in Use

  The operation is similar to that in up/down count mode. See "Counting" in "23.6.2 Operation in Up/Down Count Mode".

- Operation Performed When the Reload Compare Function is in Use

  The operation is similar to that in up/down count mode. See "Counting" in "23.6.2 Operation in Up/Down Count Mode".

- Checking Counting Direction

  The operation is similar to that in the up/down count mode. See "Checking Counting Direction" in "23.6.2 Operation in Up/Down Count Mode".

23.6.4 Operation in the Phase Difference Count Mode (Multiply-by-four)

This section explains the operation in the phase difference count mode (multiply-by-four).

Overview

This mode involves counting the phase difference of the signal input from two external signal input pins. This mode is suitable to count the phase difference of phases A and B of encoder outputs.

When a rising or falling edge is detected from the AIN or BIN pin, the input level from the other pin is verified to count up or down the phase difference of the AIN and BIN pins.

Counting up or counting down is determined depending on the combination of the edge to be detected and the input level.

Table 23-10 lists the count methods.

Table 23-10. Count Methods

Edge Detection Pin	Detection Edge	Level Check Pin	Input Level	Count Direction
BIN pin	Rising edge	AIN pin	"H" level	Counting up
			"L" level	Counting down
	Falling edge		"H" level	Counting down
			"L" level	Counting up
AIN pin	Rising edge	BIN pin	"H" level	Counting down
			"L" level	Counting up
	Falling edge		"H" level	Counting up
			"L" level	Counting down

Moreover, the following three types of functions can be used in the phase difference count mode (multiply-by-four).

- Reload function
- Compare function
- Reload compare function

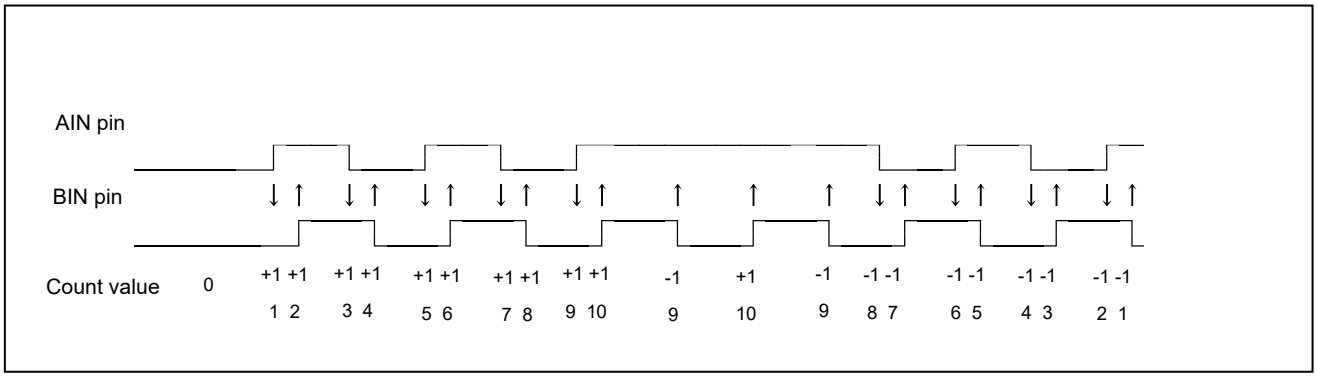
Count Operation

- Normal Operation

  If the counter is operable and the rising or falling edge is input from the AIN or BIN pin, the input level of the other pin is detected and the counter counts up or down.

  Figure 23-8 shows the operation in the phase difference count mode (multiply-by-four).

Figure 23-8. Operation in the Phase Difference Count Mode (Multiply-by-four)



If, however, the ZIN pin is set as the gate function (CGSC=1) with the CGSC bit of the counter control register (CCR), counting occurs only while the effective level set with the CGE1 and CGE0 bits is input from the ZIN pin.

For information on effective level setting, see "Counter Control Register (CCR0".

Note:

The minimum pulse width required at the AIN, BIN, and ZIN pins is 2T (T is the cycle of the peripheral clock (PCLK)).

- Operation Performed When the Reload Function is in Use

  The operation is similar to that in timer mode. See "Counting" in "23.6.1 Operation in Timer Mode".

- Operation Performed When the Compare Function is in Use

  The operation is similar to that in up/down count mode. See "Counting" in "23.6.2 Operation in Up/Down Count Mode".

- Operation Performed When the Reload Compare Function is in Use

  The operation is similar to that in up/down count mode. See "Counting" in "23.6.2 Operation in Up/Down Count Mode".

- Checking Counting Direction

  The operation is similar to that in the up/down count mode. See "Checking Counting Direction" in "23.6.2 Operation in Up/Down Count Mode".

24. Input Capture

This chapter explains the input capture.

24.1 Overview

24.2 Features

24.3 Configuration

24.4 Registers

24.5 Operation

24.6 Setting

24.7 Q&A

24.8 Sample Program

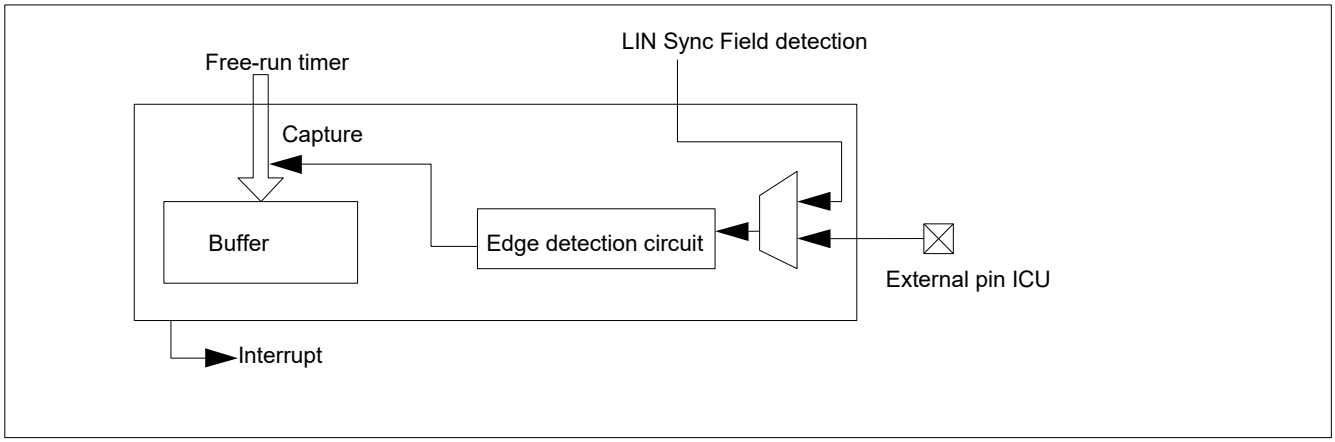
24.9 Notes

24.1 Overview

This section explains the overview of the input capture.

The input capture stores the count value of the 32-bit free-run timer at the timing when the signal from the external source is detected. The time between signals can then be calculated from the count values that have been recorded repeatedly. An interrupt can be generated when an effective edge from the external input pin is detected.

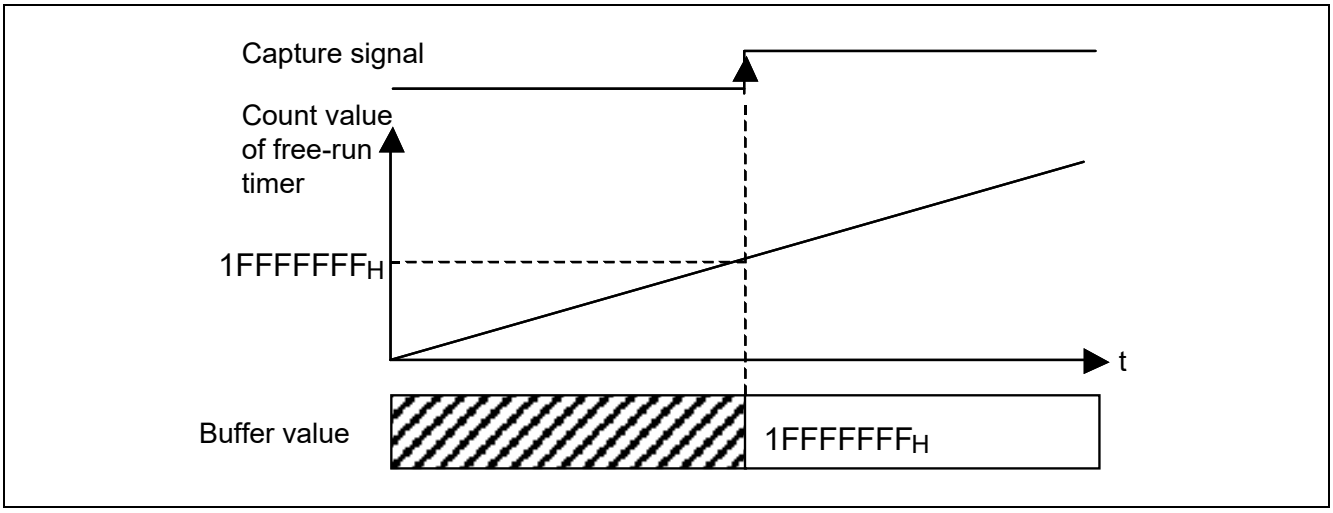
Figure 24-1. Block Diagram



24.2 Features

This section explains features of the input capture.

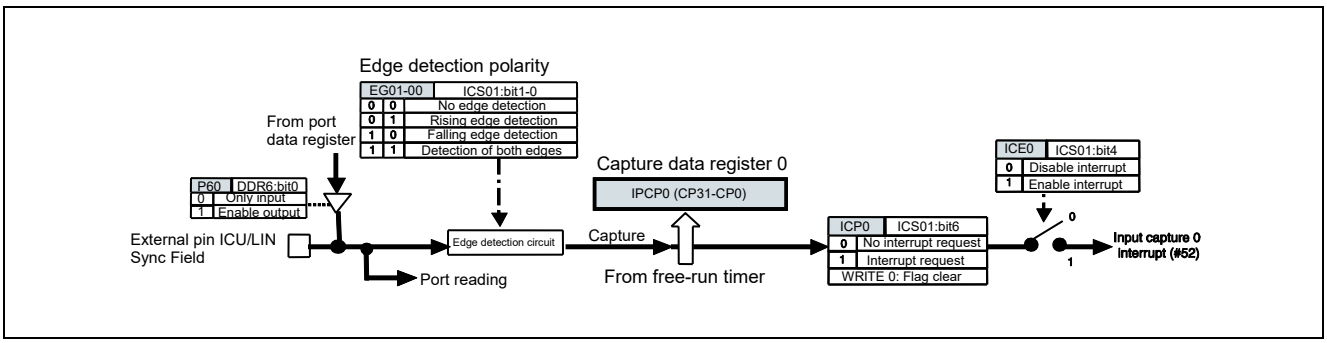
- Format: Edge detection circuit + 32-bit buffer (capture register)
- Number of units: 12
- Edge detection: Rising/falling/both edges
- Interrupt: Edge detection interrupt
- Capture value: Timer count value (00000000_H to $FFFFFFFF_H$)
- Timer:
  Input capture 0 to 5 : Use free-run timer 0 or 1.
  Input capture 6 to 11 : Use free-run timer 2 or 3.
- Precision: Peripheral clocks (PCLK)/1,/2, /4, /8, /16, /32, /64, /128, /256) (count clock of the free-run timer)



24.3 Configuration

This section explains the configuration of the input capture.

Figure 24-2. Block Diagram (Detailed; per Channel)



24.4 Registers

This section explains registers of the input capture.

Table of Base Addresses (Base_addr) and External Pins

Table 24-1. Table of Base Addresses (Base_addr) and External Pins

Channel	Base_addr	External Pin
		ICU input
0	0x02C4	ICU0/ICU0_1/ICU0_2
1	0x02C4	ICU1/ICU1_1/ICU1_2
2	0x02D0	ICU2/ICU2_1/ICU2_2
3	0x02D0	ICU3/ICU3_1/ICU3_2
4	0x02DC	ICU4/ICU4_1/ICU4_2
5	0x02DC	ICU5/ICU5_1/ICU5_2
6	0x0FD0	ICU6_0/ICU6_1/ICU6_2
7	0x0FD0	ICU7_0/ICU7_1/ICU7_2
8	0x0FDC	ICU8_0/ICU8_1/ICU8_2
9	0x0FDC	ICU9_0/ICU9_1/ICU9_2
10	0x0FE8	ICU10_0/ICU10_1/ICU10_2
11	0x0FE8	ICU11_0/ICU11_1/ICU11_2

Table 24-2. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x02C4	IPCP0				Input capture data register 0
0x02C8	IPCP1				Input capture data register 1
0x02CC	ICFS01	Reserved	LSYNS0	ICS01	Free-run timer selection register 01 LIN SYNCH FIELD switching register 0 Input capture control register 01
0x02D0	IPCP2				Input capture data register 2
0x02D4	IPCP3				Input capture data register 3
0x02D8	ICFS23	Reserved		ICS23	Free-run timer selection register 23 Input capture control register 23
0x02DC	IPCP4				Input capture data register 4
0x02E0	IPCP5				Input capture data register 5
0x02E4	ICFS45	Reserved		ICS45	Free-run timer selection register 45 Input capture control register 45
0x0FD0	IPCP6				Input capture data register 6
0x0FD4	IPCP7				Input capture data register 7
0x0FD8	ICFS67	Reserved	LSYNS1	ICS67	Free-run timer selection register 67 LIN SYNCH FIELD switching register 1 Input capture control register 67
0x0FDC	IPCP8				Input capture data register 8
0x0FE0	IPCP9				Input capture data register 9
0x0FE4	ICFS89	Reserved		ICS89	Free-run timer selection register 89 Input capture control register 89
0x0FE8	IPCP10				Input capture data register 10
0x0FEC	IPCP11				Input capture data register 11
0x0FF0	ICFS1011	Reserved		ICS1011	Free-run timer selection register 1011 Input capture control register 1011

24.4.1 Input Capture Data Register: IPCP

This section explains the bit configuration for the input capture data register (IPCP).

This register can be used to hold and read the count value of the free-run timer using a change in the input signal from the external source as a trigger.
x: Channel number 0, 2, 4, 6, 8,10
y: Channel number 1, 3, 5, 7, 9, 11

IPCPx (Input capture x): Address Base_addr+00_H (Access: Word)

IPCPy (Input capture y): Address Base_addr+04_H (Access: Word)

	bit31							bit0
	CP[31:0]							
Initial value	X X X X	X X X X	X X X X	X X X X	X X X X	X X X X	X X X X	X X X X
Attribute	R,WX							

Note:

When accessing this register, use a word access instruction. No data can be written to this register.

24.4.2 Free-run Timer Selection Register: ICFS

This section explains the bit configuration for the free-run timer selection register (ICFS).

This register selects the capture source free-run timer.
x: Channel number 0, 2, 4, 6, 8,10
y: Channel number 1, 3, 5, 7, 9, 11

ICFSxy (Free-run timer selection xy): Address Base_addr+08_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	-	-	SELy	SELx
Initial value	-	-	-	-	-	-	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W

[bit7 to bit2] - : Undefined

This does not affect the writing operation.

[bit1, bit0] SELy, SELx: Free-run timer selection

SEL{0,1,2,3,4,5}	Operation
0	Free-run timer 0
1	Free-run timer 1

SEL{6,7,8,9,10,11}	Operation
0	Free-run timer 2
1	Free-run timer 3

24.4.3 Input Capture Control Register: ICS

This section explains the bit configuration the input capture control register (ICS).

This register is used to control the input capture.
x: Channel number 0, 2, 4, 6, 8,10
y: Channel number 1, 3, 5, 7, 9, 11

ICSxy (Input capture x, y): Address Base_addr+0B$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICPy	ICPx	ICEy	ICEx	EGy1	EGy0	EGx1	EGx0
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] ICPy, ICPx: Input capture interrupt request flag

ICPn	State	
	Read	Write
0	No interrupt request	Clear the flag
1	Interrupt request present (edge detected)	No effect on operation

ICPn: n corresponds to the input capture channel numbers

- This flag will be set to "1" when the signal change (edge) selected in the capture effective edge selection bit (EG[n1:n0) is detected in the input signal from the external pin.
- To enable the CPU interrupt request, you need to enable interrupt request enable setting (ICEn= 1).

[bit5, bit4] ICEy, ICEx: Input capture interrupt request enabled

ICEn	Operation
0	Interrupt disabled
1	Interrupt enabled

ICEn: n corresponds to the input capture channel numbers

An input capture interrupt is generated when the input capture interrupt request flag is set to "1" while the input capture interrupt request enable bit is set to "1".

[bit3 to bit0] EGn1, EGn0: Input capture n effective edge selection

EGn1	EGn0	Edge Selection
0	0	Input capture stopped
0	1	Rising edge
1	0	Falling edge
1	1	Both edges (rising and falling edges)

EGn1, EGn0: n corresponds to the input capture channel numbers

- These bits are used to select the capture effective edge(s) for the input capture signal from the external pin.
- The input capture will be in stop if the effective edge selection bit is "00_B".

24.4.4 LIN SYNCH FIELD Switching Register: LSYNS

This section explains the bit configuration for the LIN SYNCH FIELD switching register (LSYNS).

LSYNS0 (Input capture 0-5): Address 02CE$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	LSYN5	LSYN4	LSYN3	LSYN2	LSYN1	LSYN0
Initial value	-	-	0	0	0	0	0	0
Attribute	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit7, bit6] - : Undefined

The read value is always "1". This does not affect the writing operation.

[bit5 to bit0] LSYN5 to LSYN0: Input capture 5 to 0 input selection

LSYNn (n=0 to 5)	Input Selection
0	External pin input (ICUn)
1	LIN synch field detection signal input from LIN-UART ch.(n +2)

Note:

The input for the input capture must be switched while the capture is inactive (ICS:EG[n1:n0]=00).

When the capture operation is enabled (ICS:EG[n1:n0] is other than "00") and input is switched while the signal level of the external pin input and the state of the LIN synch field detection signal (level) are different, edges will be detected and will operate as capture effective edges.

LSYNS1 (Input capture 6-9): Address 0FDA$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	LSYN9	LSYN8	LSYN7	LSYN6
Initial value	-	-	-	-	0	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W	R/W	R/W

[bit7 to bit4] - : Undefined

The read value is always "1". This does not affect the writing operation.

[bit3, bit2] LSYN9, LSYN8: Input capture 9, 8 input selection

LSYNn (n=8, 9)	Input Selection
0	External pin input (ICUn)
1	LIN synch field detection signal input from multi-function serial interface ch.(n)

[bit1, bit0] LSYN7, LSYN6: Input capture 7, 6 input selection

LSYNn (n=6, 7)	Input Selection
0	External pin input (ICUn)
1	LIN synch field detection signal input from multi-function serial interface ch.(n-6)

Note:

The input for the input capture must be switched while the capture is inactive (ICS:EG[n1:n0]= 00).

24.5 Operation

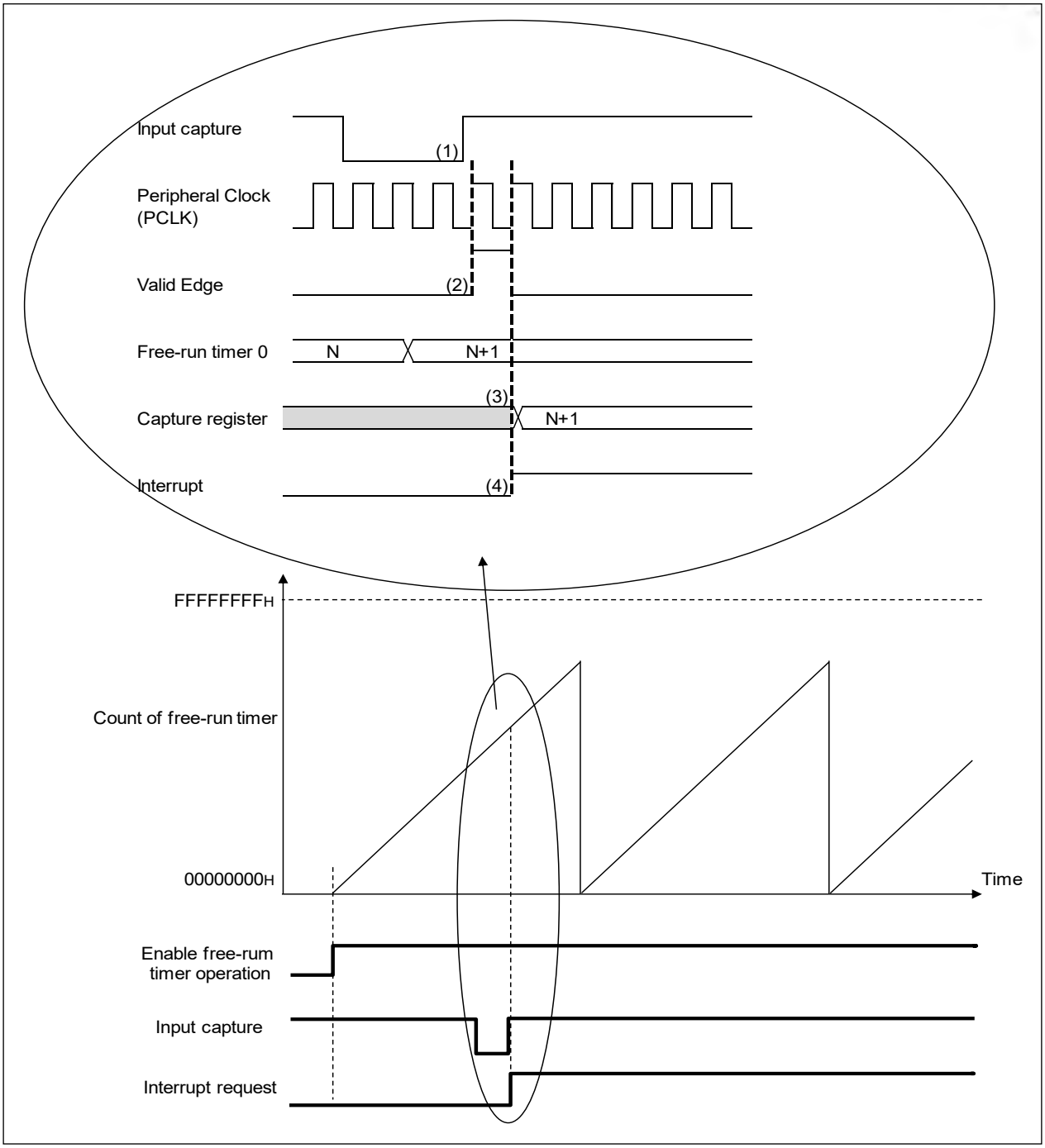
This section explains the operation of the input capture.

When a set effective edge is detected, the 32-bit input capture can retrieve the value of the 32-bit free-run timer into the capture register and generate an interrupt.

This section explains the input capture operation.

24.5.1 Capture and Interrupt Timings

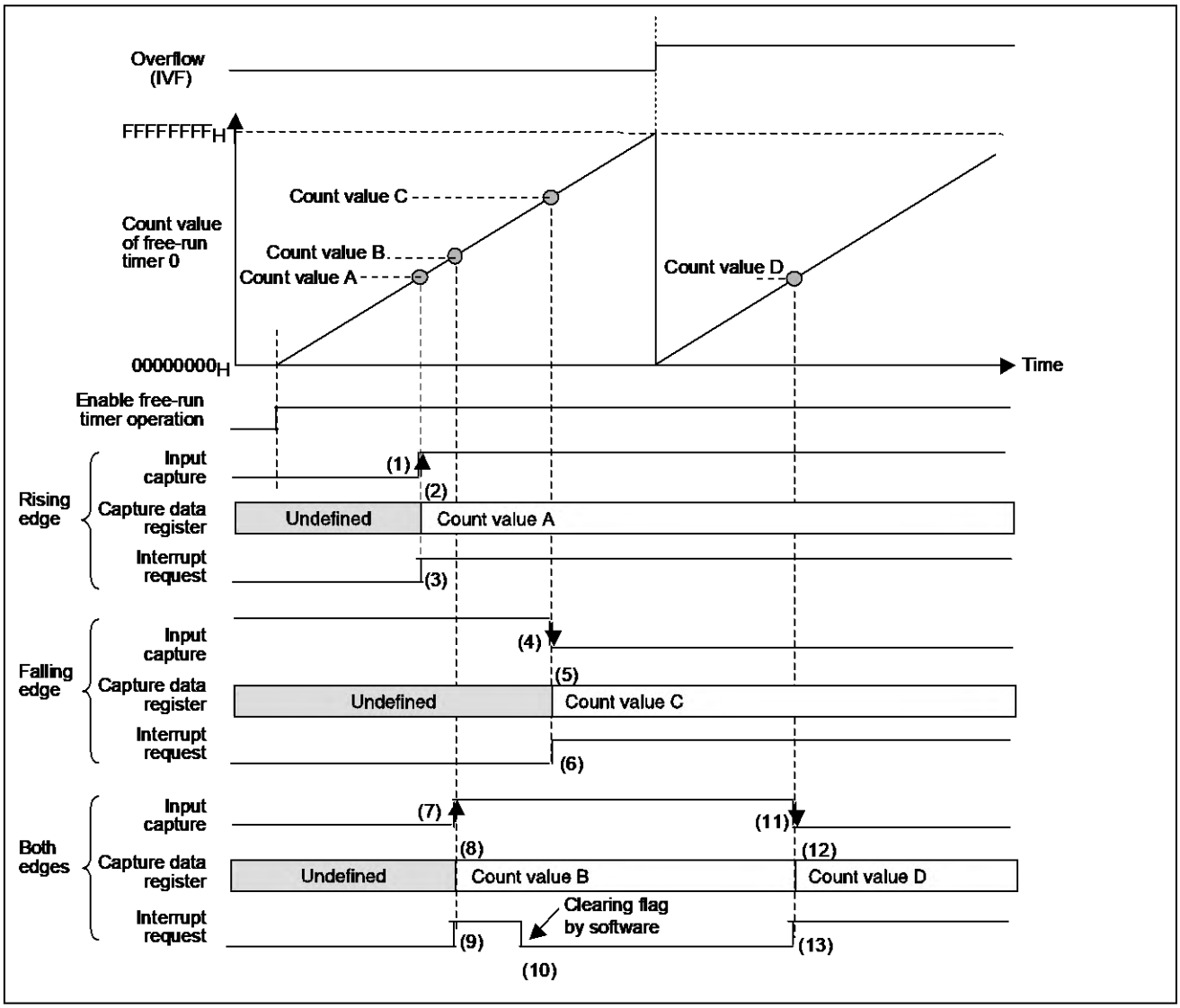
This section explains capture and interrupts timings of the input capture.



(1) Rising edge of the input signal
(2) Internal signal generated by edge detection (synchronized to the peripheral clock)
(3) Free-run timer value is recorded to the capture register (capture).
(4) Input capture interrupt is generated (ICP(0 to 11)=1).

24.5.2 Edge Detection Specifications for Input Capture and their Operations

This section explains edge detection specifications for the input capture and their operations.



- When rising edge is selected

  (1) Rising edge of the input signal is detected.
  (2) Free-run counter value is recorded to the capture register (capture).
  (3) Input capture interrupt is generated.

- When falling edge is selected

  (4) Falling edge of the input signal is detected.
  (5) Free-run counter value is recorded to the capture register (capture).
  (6) Input capture interrupt is generated.

- Both edges

  (7) Rising edge of the input signal is detected.
  (8) Free-run counter value is recorded to the capture register (capture).
  (9) Input capture interrupt is generated.
  (10) Interrupt request flag ((ICS01:ICP0), (ICS01:ICP1), (ICS23:ICP2), (ICS23:ICP3), (ICS45:ICP4), (ICS45:ICP5), ...) is cleared using software.
  (11) Falling edge of the input signal is detected.
  (12) Free-run counter value is recorded to the capture register (capture).
  (13) Input capture interrupt is generated.

24.6 Setting

This section explains setting of the input capture.

Table 24-3. Settings Required for Using Input Capture

Configuration	Configured Register	Setting Method
Free-run timer setting Free-run timer activation	See "Chapter: Free-run Timer".	-
Setting for switching inputs between input pins ICU0 to ICU11 and input capture	If the linkage function for LIN_UART or multi-function serial interface is used: LIN SYNCH FIELD switching register (LSYNS0), (LSYNS1) External input: ■ Settings of the LIN SYNCH FIELD switching register (LSYNS0), (LSYNS1) ■ ICU pins (See "Chapter: I/O Ports").	See 24.7.2
Effective edge polarity selection for external input	Input capture control registers (ICS01), (ICS23),....	See 24.7.1

Table 24-4. Settings Required for Performing Input Capture Interrupt

Configuration	Configured Register	Setting Method
Input capture interrupt vector and input capture interrupt level settings	See "Chapter: Interrupt Control (Interrupt Controller)".	See 24.7.3
Input capture interrupt setting Interrupt request clear Interrupt request enable	Input capture control registers (ICS01), (ICS23), (ICS45), (ICS67), (ICS89), (ICS1011)	See 24.7.5

24.7 Q&A

This section explains Q&A of the input capture.

24.7.1 Effective Edge Polarity of External Input: Types and How to Select
24.7.2 How to Enable External Input Pins (ICU0, ICU1, ICU2, ICU3, ICU4, ICU5, …)
24.7.3 About Interrupt Related Registers
24.7.4 About Interrupt Types
24.7.5 How to Enable Interrupt
24.7.6 How to Measure the Pulse Width of the Input Signal

24.7.1 Effective Edge Polarity of External Input: Types and How to Select

This section explains types of the effective edge polarity of external input and the selection method.

There are 3 types of the effective edge polarity: rising, falling and both edges.

You can configure it using the effective edge polarity bits of the external input (ICS01:EG[01:00]), (ICS01:EG[11:10]), (ICS23:EG[21:20]), (ICS23:EG[31:30]), (ICS45:EG[41:40]), (ICS45:EG[51:50]), (ICS67:EG[61:60]), (ICS67:EG[71:70]), ...

Operation	Effective Edge Polarity Bits of the External Input (EG[01:00]), (EG[11:10]), (EG[21:20]), (EG[31:30]), (EG[41:40]), (EG[51:50]), (EG[61:60]), (EG[71:70]), …
To select rising edge	Select "01".
To select falling edge	Select "10".
To select both edges	Select "11".

24.7.2 How to Enable External Input Pins (ICU0, ICU1, ICU2, ICU3, ICU4, ICU5, …)

This section explains how to enable external input pins (ICU0 to ICU11).

Set the LSYNS0/1 register for external pin input. Also, set the ICU0 to ICU11 pins for peripheral input. For information on the setting method, see "Chapter: I/O Ports".

24.7.3 About Interrupt Related Registers

This section explains interrupt related registers.

Input capture interrupt vector and input capture interrupt level settings

See "C. List of Interrupt Vector" in "Appendix" for interrupt number.

For details of the interrupt levels and interrupt vectors, see "Chapter: Interrupt Control (Interrupt Controller)".

Interrupt request flags ((ICS01:ICP0), (ICS01:ICP1), (ICS23:ICP2), (ICS23:ICP3), (ICS45:ICP4), (ICS45:ICP5), (ICS67:ICP6) and (ICS67:ICP7), ...) are not cleared automatically. Therefore, clear the input capture interrupt request flags (ICP0, ICP1, ICP2, ICP3, ICP4, ICP5, ICP6, ICP7, ...) by writing "0" using software before returning from interrupt processing.

24.7.4 About Interrupt Types

This section explains interrupt types.

There is only 1 type of interrupt. It is generated when an edge is detected in the input signal.

24.7.5 How to Enable Interrupt

This section explains how to enable interrupt.

Enable interrupt request, interrupt request flag
You can configure the interrupt enable setting using the following interrupt request enable bits:
(ICS01:ICE0), (ICS01:ICE1), (ICS23:ICE2), (ICS23:ICE3), (ICS45:ICE4), (ICS45:ICE5), (ICS67:ICE6), (ICS67:ICE7), ...

Operation	Interrupt Request Enable Bits (ICE0), (ICE1), (ICE2), (ICE3), (ICE4), (ICE5), (ICE6), (ICE7), …
Interrupt disabled	Set "0".
Interrupt enabled	Set "1".

You can clear the interrupt request using the following interrupt request flags:
(ICS01:ICP0), (ICS01:ICP1), (ICS23:ICP2), (ICS23:ICP3), (ICS45:ICP4), (ICS45:ICP5), (ICS67:ICP6), (ICS67:ICP7), ...

Operation	Interrupt Request Flag Bits (ICP0), (ICP1), (ICP2), (ICP3), (ICP4), (ICP5), (ICP6), (ICP7), …
Interrupt request clear	Write "0".

24.7.6 How to Measure the Pulse Width of the Input Signal

This section explains how to measure the pulse width of the input signal.

- "H" Width Measurement

  Enable detection of both edges.

  Ensure that the rising edge is detected first, followed by the falling edge.

  Pulse width = {Value recorded at falling edge (input capture register value)
  + "100000000_H " × No. of overflows
  - Value recorded at rising edge (input capture register value)}
  × Count clock width of the free-run timer

  Example: Value recorded at falling edge = 23200000_H, Value recorded at rising edge = $A6350000_H$,
  No. of overflows = 1, Count clock = 125 ns
  ==> Pulse width = (23200000_H + 100000000_H - $A6350000_H$) × 125ns = 261.972s

- Interval Measurement

  Enable rising (or falling) edge detection.

  The specified edge is detected twice.

  Cycle = {2nd recorded value (input capture register value)
  + "100000000_H" × No. of overflows
  - {1st recorded value (input capture register value)
  × Count clock width of the free-run timer

Note:

This calculation formula is an example that is not use compare match clear function.

24.8 Sample Program

This section explains the sample program of the input capture.

Setting procedure example 1

Detect the rising edge of the pulse for input to ICU0 and record the value of free-run timer.
This process is repeated twice to measure the time from one trigger to another. However, reading and calculation of the capture value are to be handled as interrupt processes.

1. Initial setting

-Free-run timer ch.0 control Register name.Bit name

Control register setting	TCCS0
Clock selection≫	.ECKE
Compare interrupt request flag≫	.ICLR
Compare interrupt request enable≫	.ICRE
Counting operation≫	.STOP
TCDT clear	.SCLR
Count clock≫	.CLK3-0
Timer data value setting	TCDT0

-Port Register name.Bit name

Port ICU0 input setting	See "Chapter: I/O Ports".

-Input capture control Register name.Bit name

Control register setting	ICS0
Interrupt request flag≫	.ICP1,ICP0
Interrupt request enabled≫	.ICE1,ICE0
ch.1 Effective edge polarity selection≫	.EG11,EG10
ch.0 Effective edge polarity selection≫	.EG01,EG00

-Interrupt-related Register name.Bit name

Sets an interrupt level.	ICR36
I flag setting	(CCR)

-Variable setting

2. Activation

-Input capture ch.0 activation Register name.Bit name

Interrupt control	ICS01.ICE0

-Free-run timer ch.0 activation Register name.Bit name

Count operation activation	TCCS0.STOP

3. Interrupt

-Interrupt processing Register name.Bit name

Clearing of interrupt request flag	ICS01.ICP0
(Any process)	
.....	

4. Interrupt vector

-Vector table setting

(Note)
Clock-related settings and the setting of __set_il (numeric value) need to be configured in advance. See “Chapter: Clock” and “Chapter: Interrupts Control (Interrupts Controller)”.

Program example 1

```
void INPUT0_sample_1(void)
{
        freerun0_initial();
        INPUT0_initial();
        INPUT0_start();
        freerun0_start();
}

void freerun0_initial(void)
{
        IO_TCCS0.word = 0x0041;     /* Setting value=0000_0000_0100_0001 */
                                    /* bit15 = 0    ECKE internal clock source */
                                    /* bit14 -10 =0  Reserved bit */
                                    /* bit9 = 0     Interrupt flag clear */
                                    /* bit8 = 0     Interrupt disabled */
                                    /* bit7 = 0     Reserved bit */
                                    /* bit6 = 1     */
                                    /* bit5 = 0     Reserved bit */
                                    /* bit4 = 0     */
                                    /* bit3-0 = 0001 */
        IO_TCDT0 = 0x0000;          /* Initialization of timer data value */
}

void INPUT0_initial(void)
{
        PORT_SETTING_ICU0_IN();  /* Set the ICU0 pin for peripheral input. */

        IO_ICS01.byte = 0x01;       /* Setting value=0000_0001 */
                                    /* bit7 to 6 = 00 ICP1, 0 No effective edge detected */
                                    /* bit5 to 4 = 00 ICE1, 0 Interrupt disabled */
                                    /* bit3 to 2 = 00 EG11, EG10 ch.1 No edge detected */
                                    /* bit1 to 0 = 01 EG01, EG00 ch.0 Rising edge detected */

        IO_ICR[36].byte = 0x10;     /* Input capture ch.0 interrupt level setting (any value) */
        __EI();                     /* Interrupt enabled */
        count = 0;
}

void INPUT0_start(void)
{
        IO_ICS01.bit.ICE0 = 1;      /* bit4 = 1 ICE0 ch.0 Interrupt enabled */
}

void freerun0_start(void)
{
        IO_TCCS0.bit.STOP = 0;      /* bit6 = 0 STOP count enabled */
}

__interrupt void INPUT0_int(void)
{
        IO_ICS01.bit.ICP0 = 0;      /* bit6 = 0 Clearing of ICP0 effective edge detection flag */
        if(count==0)
        data1 = IO_IPCP0;           /* Free-run timer value is recorded. (1st time) */
        else if(count==1) {
        data2 = IO_IPCP0;           /* Free-run timer value is recorded. (2nd time) */
        cycle = (data2-data1)*125;  /* Time is measured. */
        count = 0;
        }
} count++;

Specification of interrupt routine required in vector table
#pragma intvect INPUT0_int 52
```

24.9 Notes

This section explains notes of the input capture.

- Input Capture Register

  The input capture register value is undefined after a reset.

  Reading of the input capture register must be performed in word (32-bit mode) access.

- Read-modify-write

  The input capture interrupt request bits (ICP0), (ICP1), (ICP2), (ICP3), (ICP4), (ICP5), (ICP6), ..., (ICP11) are "1" when read using a read-modify-write (RMW) instruction.

25. Real-time Clock (RTC)

This chapter explains the real-time clock (RTC).

25.1 Overview

25.2 Features

25.3 Configuration

25.4 Registers

25.5 Operation

25.6 Setting

25.7 Q&A

25.8 Sample Program

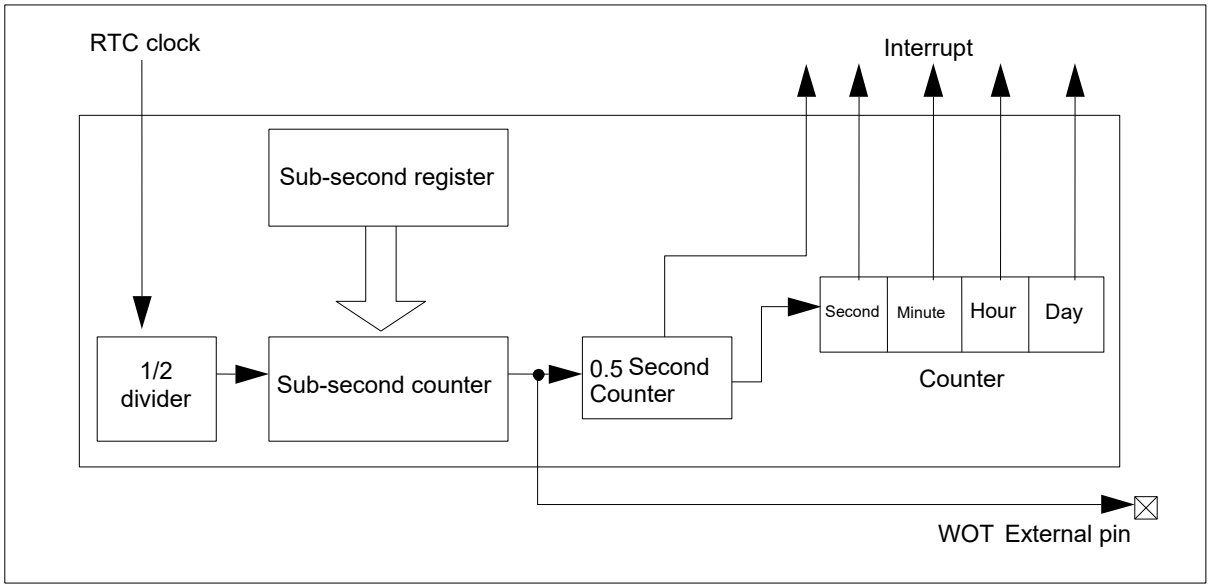
25.9 Notes

25.1 Overview

This section explains the overview of the real-time clock (RTC).

The real-time clock (watch timer) consists of the timer control register, sub-second register, Second/ Minute/ Hour/ Day registers, 1/2 clock frequency divider, sub-second counter(22-bit down counter) and Second/ Minute/ Hour/ Day counters. The real-time clock operates as the real-world timer and provides the real-world timer information.

Figure 25-1. Block Diagram (Overview)



25.2 Features

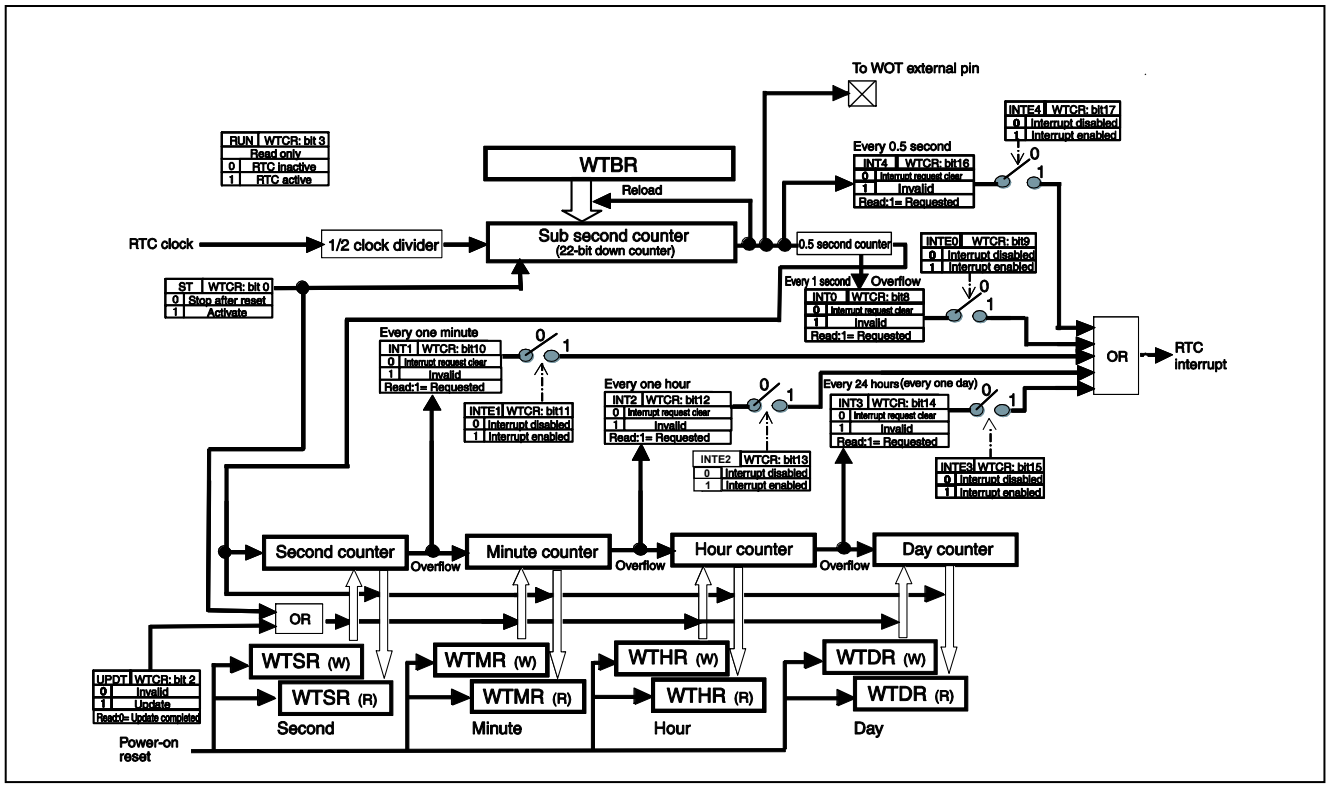
This section explains features of the real-time clock (RTC).

- Function: Counts the number of days and time (day/ hour/ minute/ second) (operations are kept on in the watch mode too.)
  The default values of the number of days and time can be modified.
- Operation clock: RTC clock (See "Chapter: Clock" for the selection of the clock source of the RTC clock.
  See "Chapter: RTC/WDT1 Calibration" for the correction when a sub-clock(only dual clock product) is selected as a source.)
- Interrupt: Interrupts can be generated based on five intervals: 0.5 second, 1 second, 1 minute, 1 hour, and 1 day.
  In addition, interrupts at any interval (from short interval to long interval) can be generated by changing the sub-second value.

25.3 Configuration

This section explains the configuration of the real-time clock (RTC).

Figure 25-2. Configuration Diagram



25.4 Registers

This section explains registers of the real-time clock (RTC).

Table 25-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x055C	Reserved	Reserved	WTDR		Day/Hour/Minute/Second Registers(day)
0x0560	Reserved	WTCR			RTC control register
0x0564	Reserved	WTBR			Sub-second register
0x0568	WTHR	WTMR	WTSR	Reserved	Day/Hour/Minute/Second registers(hour) Day/Hour/Minute/Second registers(minute) Day/Hour/Minute/Second registers(second)

25.4.1 RTC Control Register: WTCR

The bit configuration of the RTC control register (WTCR) is shown below.

This register controls the operations of the real-time clock module.

WTCRH: Address 0561$_H$ (Access: Byte)

WTCRM: Address 0562$_H$ (Access: Byte, Half-word)

WTCRL: Address 0563$_H$ (Access: Byte, Half-word)

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	-	-	-	-	-	-	INTE4	INT4
Initial value	-	-	-	-	-	-	0	0
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R(RM1), W

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	INTE3	INT3	INTE2	INT2	INTE1	INT1	INTE0	INT0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R(RM1), W	R/W	R(RM1), W	R/W	R(RM1), W	R/W	R(RM1), W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	RUN	UPDT	Reserved	ST
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W0	R/W0	R/W0	R/W0	R,WX	R(RM0),W	R/W0	R/W

This register will be initialized by all reset source without the return reset from watch mode (power-shutdown).

[bit23 to bit18] - : Undefined

The read value is always "1". The data writing does not affect the operation.

[bit17] INTE4: 0.5 second interrupt request enable

INTE4	Operation
0	0.5 second interrupt request disabled
1	0.5 second interrupt request enabled

[bit16] INT4: 0.5 second interrupt request flag

INT4	State	
	Read	Write
0	0.5 second interrupt request not generated	Flag clear
1	0.5 second interrupt request generated	This does not affect the operations

When the frequency division output of the borrow signal of the sub-second counter (22-bit down counter) is enabled, the flag will be set to "1".

[bit15] INTE3: 1 day interrupt request enable

INTE3	Operation
0	1 day (24 hours) interrupt request disabled
1	1 day (24 hours) interrupt request enabled

[bit14] INT3: 1 day interrupt request flag

INT3	State	
	Read	Write
0	1 day (24 hours) interrupt reques not generated	Flag clear
1	1 day (24 hours) interrupt request generated	This does not affect the operations

When overflow occurs in the hour counter, the flag will be set to "1".

[bit13] INTE2: 1 hour interrupt request enable

INTE2	Operation
0	1 hour interrupt request disabled
1	1 hour interrupt request enabled

[bit12] INT2: 1 hour interrupt request flag

INT2	State	
	Read	Write
0	1 hour interrupt request not generated	Flag clear
1	1 hour interrupt request generated	This does not affect the operations

When overflow occurs in the minute counter, the flag will be set to "1".

[bit11] INTE1: 1 minute interrupt request enable

INTE1	Operation
0	1 minute interrupt request disabled
1	1 minute interrupt request enabled

[bit10] INT1: 1 minute interrupt request flag

INT1	Operation	
	Read	Write
0	1 minute interrupt request not generated	Flag clear
1	1 minute interrupt request generated	This does not affect the operations

When overflow occurs in the second counter, the flag will be set to "1".

[bit9] INTE0: 1 second interrupt request enable

INTE0	Operation
0	1 second interrupt request disabled
1	1 second interrupt request enabled

[bit8] INT0: 1 second interrupt request flag

INT0	State	
	Read	Write
0	1 second interrupt request not generated	Flag clear
1	1 second interrupt request generated	This does not affect the operations

When overflow occurs in the 0.5 second counter, the flag will be set to "1".

[bit7 to bit4] Reserved

These bits must always be written to "0".

[bit3] RUN: Operation state

RUN	State
0	Real-time clock module is stopped
1	Real-time clock module is running

[bit2] UPDT: Update

UPDT	State/Operation	
	Read	Write
0	Update completed	This does not affect the operations
1	Updating	The counter values of the Hour/ Minute/ Second counters are updated to Day/ Hour/Minute/ Second register values respectively.

Before writing "1" to the update bit (UPDT), set the value to be updated in the Day/ Hour/ Minute/ Second registers.

Update for Day/ Hour/ Minute/ Second registers will be performed when reload occurs at the sub-second counter (22-bit down counter).

When the counter value is updated, the UPDT bit will be cleared by hardware. However, when update is completed at the same time as writing "1", the UPDT bit will not be cleared to "0".

[bit1] Reserved

This bit must always be written to "0".

[bit0] ST: Start

ST	Operation
0	Real-time clock module is stopped. All the counters are cleared.
1	Values set at Day/Hour/Minute/Second registers are loaded into Day/Hour/Minute/Second counters, and the real-time clock starts to run.

Note:

- When writing "1" to the start bit (ST) from RTC stop state (ST=0) (RTC operation start), do not write "1" to the update bit (UPDT) at the same time as the start bit.
  (While ST=0, writing "1" as byte immediate value to the ST bit and the UPDT bit at the same time is prohibited.)
- To write "1" to the update bit (UPDT), do it while RTC is working (ST=1).
- While the update bit (UPDT) is "1", writing "0" to the start bit (ST) (RTC stop) is prohibited.

25.4.2 Sub-second Register: WTBR

The bit configuration of the sub-second register (WTBR) is shown below.

This register contains the reload value of the sub-second counter (22-bit down counter).

WTBRH: Address 0565H (Access: Byte)

WTBRM: Address 0566H (Access: Byte)

WTBRL: Address 0567H (Access: Byte)

WTBRH

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	D21	D20	D19	D18	D17	D16
Initial value	-	-	X	X	X	X	X	X
Attribute	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W

WTBRM

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D15	D14	D13	D12	D11	D10	D9	D8
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

WTBRL

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
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

The sub-second register contains the reload value used in the sub-second counter(22-bit down counter).

This value will be reloaded as soon as the sub-second counter (22-bit down counter) becomes "0". To modify the sub-second register, confirm that no reload operations are being performed during the writing instruction. Otherwise, the sub-second counter (22-bit down counter) will load a wrong value that combines both new and old data bytes. Generally, it is recommended to perform update while the ST bit is "0". While the sub-second register is set to "0", the sub-second counter (22-bit down counter) will not run at all.

The sub-second register settings for counting 0.5 second are as follows:

Table 25-2. WTBR Setting Example

RTC Clock Frequency	WTBR Setting Value
32 kHz	0x001F3F
4 MHz	0x0F423F

25.4.3 Day/Hour/Minute/Second Register: WTDR/WTHR/WTMR/WTSR

The bit configuration of the Day/Hour/Minute/Second register (WTDR/WTHR/WTMR/WTSR) is shown below.

These registers indicate the time information of the real-time clock (Day/ Hour/ Minute/ Second).

WTDR (day register): Address 055E$_H$ (Access: Half-word)

WTHR (hour register): Address 0568$_H$ (Access: Byte, Half-word)

WTMR (minute register): Address 0569$_H$ (Access: Byte, Half-word)

WTSR (second register): Address 056A$_H$ (Access: Byte)

WTDR

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	N15	N14	N13	N12	N11	N10	N9	N8
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
	N7	N6	N5	N4	N3	N2	N1	N0
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

WTHR

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	H4	H3	H2	H1	H0
Initial value	-	-	-	0	0	0	0	0
Attribute	R1,WX	R1,WX	R1,WX	R,W	R,W	R,W	R,W	R,W

WTMR

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	M5	M4	M3	M2	M1	M0
Initial value	-	-	0	0	0	0	0	0
Attribute	R1,WX	R1,WX	R,W	R,W	R,W	R,W	R,W	R,W

WTSR

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	S5	S4	S3	S2	S1	S0
Initial value	-	-	0	0	0	0	0	0
Attribute	R1,WX	R1,WX	R,W	R,W	R,W	R,W	R,W	R,W

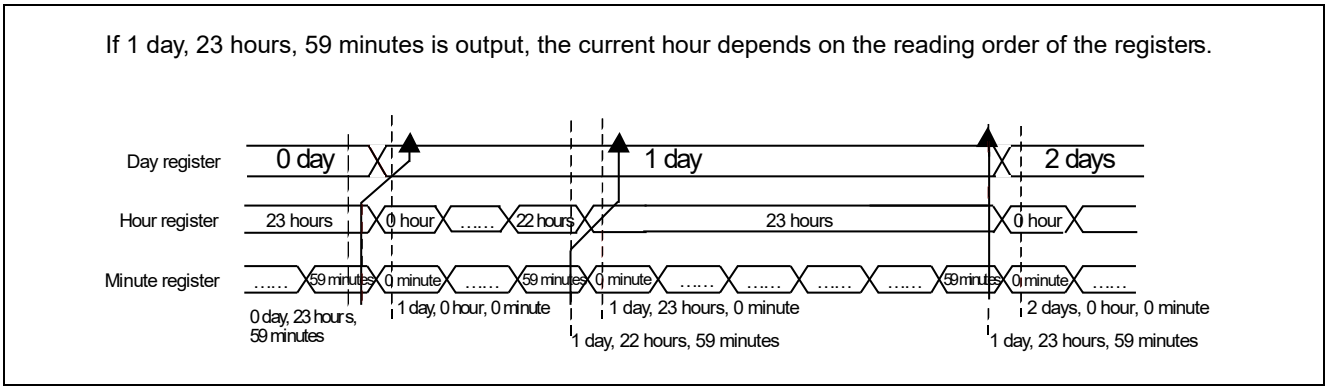
This register will be initialized by power-on reset source.

- The Second/ Minute/ Hour/ Day registers contain day and time information. Binary-coded notation is used for second, minute, hour, and day.
- When the register is read out, the counter value will be read out. The written data will be loaded to the counter after the UPDT bit is set to "1".
- As word access is not available, perform access for the respective registers.
- Word access is not available for the number of days register either. In addition, be sure to perform halfword access for the number of days register as the number of days is counted using a 16-bit counter. As byte access may cause carry during read, having the possibility of getting an inappropriate read value, byte access and word access are prohibited.
- Set the Hour/Minute/Second registers within the following ranges:

  Hour (WTHR): 0 to 17_H (0 hour to 23 hours)
  Minute (WTMR): 0 to $3B_H$ (0 minute to 59 minutes)
  Second (WTSR): 0 to $3B_H$ (0 second to 59 seconds)

- Confirm that there are no contradictions among the values output from the four registers: Day/Hour/Minute/Second registers. The following example may occur.
- [Example] Output value "1 day, 23 hours, 59 minutes, 59 seconds", "0 day, 23 hours, 59 minutes, 59 seconds". "1 day, 0 hour, 0 minute, 0 second", "1 day, 22 hours, 59 minutes, 59 seconds", 1 day, 23 hours, 0 minute, 0 second, "2 days, 0 hour, 0 minute, 0 second"

Figure 25-3. Diagram of Day, Hour, Minute and Second Register Transitions

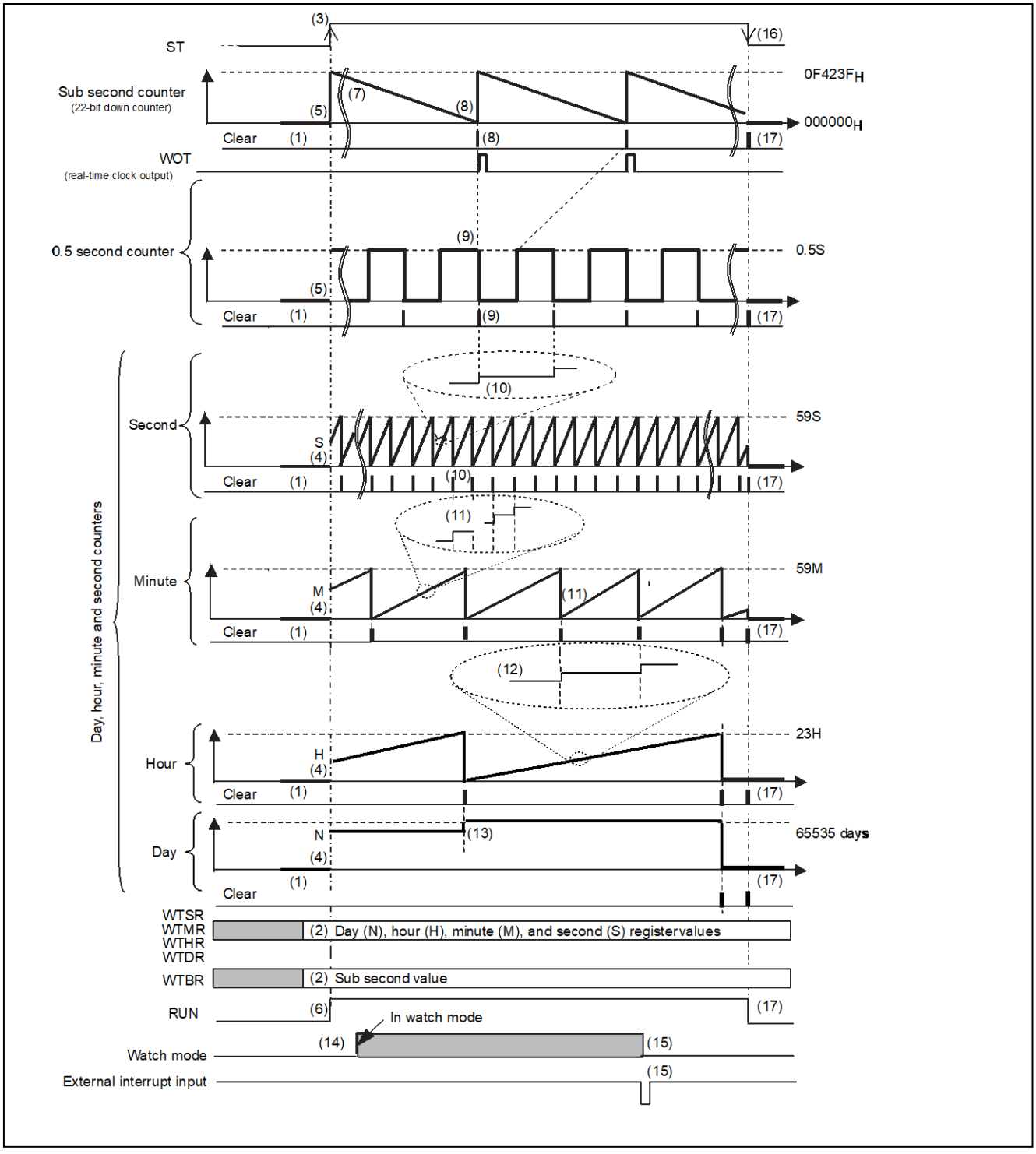


- When the operation clock frequency is obtained by dividing the frequency of the main clock by 2 (while PLL is stopped), the wrong values may be read out from the Hour/Minute/Second registers. This is caused due to synchronization adjustment between reading operations and count operations. Therefore, it is recommended to use second interrupts in the trigger for reading instructions.
- To restart operations with the duration the counter has stopped as the initial value, read the Day/Hour/Minute/Second registers prior to restart and write these values to the Day/Hour/Minute/Second registers to start.
- As this series does not provide the RTC detection reset function, the Day/Hour/Minute/Second registers are cleared only in case of power-on reset. Therefore, when the microcomputer internal low-voltage detection flag is set, the Day/Hour/Minute/Second registers are recommended to be cleared.

25.5 Operation

This section explains the operation of the real-time clock (RTC).

Figure 25-4. Operation Descriptions for the Real-time Clock



1. Use the start bit (ST="0") to reset the sub-second counter (22-bit down counter) and Day/Hour/Minute/Second timers (0), and then stop them.
2. a. Write the values of Day/Hour/Minute/Second to Day/Hour/Minute/Second registers: WTDR, WTHR, WTMR, WTSR by software.
   b. Write "0FH", "42H", "3FH" to sub-second registers: WTBRH, WTBRM, WTBRL by software.
   c. Initialize the interrupt request bits (INT0, INT1, INT2, INT3, INT4), and set the interrupt request enable bits (INTE0, INTE1, INTE2, INTE3, INTE4) (enable interrupts to be used).
3. Set the start bit (ST) to "1".
4. Use the start bit (ST="1") to load the values in the Day/Hour/Minute/Second registers: WTDR, WTHR, WTMR, WTSR to the Day/ Hour/ Minute/ Second timers.
5. Moreover, as the count value of the sub-second counter (22-bit down counter) is "000000H", load the values in second registers: WTBRH, WTBRM, WTBRL to the sub-second counter (22-bit down counter).
6. The operation flag (RUN) becomes "1".
7. The sub-second counter (22-bit down counter) starts to count using a clock obtained by dividing the main clock frequency by 2 (4/2 MHz).
8. When the sub-second counter (22-bit down counter) becomes "000000H", load the sub-second register value "0F423H" to the sub-second counter (22-bit down counter).
   In addition, an interrupt request of 0.5 second counter occurs.
   Moreover, when the real-time clock output enable is set (WOT pin output enable), an "H" level with a width twice as long as that of the main clock is output to the WOT pin.
   (Example: For main clock 4 MHz, "H" output with a width of 500 ns)
9. After the 0.5 second counter is counted up, it is cleared at the next count up, the second counter of the Day/Hour/Minute/Second counters is counted up, and a second interrupt request occurs.
10. The second counter of the Day/Hour/Minute/Second counters is counted up, it is cleared at the next count up when the value is "59", the minute counter is counted up, and the minute interrupt request occurs at this time.
11. The minute counter of Day/Hour/Minute/Second counters is counted up, it is cleared at the next count up when the value is "59", the hour counter is counted up, and the hour interrupt request occurs at this time.
12. The hour counter of the Day/Hour/Minute/Second counters is counted up, it is cleared at the next count up when the value is "23", the day counter is counted up, and the day interrupt request occurs at this time.
13. The day counter of the Day/Hour/Minute/Second counters is counted up, it is cleared at the next count up when the value is "65535".
14. Move to the watch mode by software.
    The real-time clock will continue to run in the watch mode.
15. Input a signal from an interrupt pin (INTxx) to restore from the watch mode and restart CPU.
16. Set the start bit (ST) to "0".
17. Use the start bit ST="0" to clear(reset) the sub-second counter (22-bit down counter) and the Day/Hour/Minute/Second counters, and then stop them.

25.6 Setting

This section explains setting of the real-time clock (RTC).

Table 25-3. Settings Required for Starting the Real-time Clock

Settings	Setting Registers	Setting Procedure
Setting of the reload value (sub-second register)	Sub-second register (WTBRH, WTBRM, WTBRL)	See 25.7.1
Initialization of the real-time clock	RTC Control Register (WTCR)	See 25.7.2
Setting of number of days, time (Day/Hour/Minute/Second)	Day/ Hour/ Minute/ Second registers (WTDR,WTHR, WTMR, WTSR)	See 25.7.3
Startup of the real-time clock	RTC Control Register (WTCR)	See 25.7.4

Table 25-4. Settings Required for Knowing the Time

Settings	Setting Registers	Setting Procedure
Reading of number of days and time	Day/ Hour/ Minute/ Second registers (WTDR,WTHR, WTMR, WTSR)	See 25.7.6

Table 25-5. Settings Required for Stopping the Real-time Clock

Settings	Setting Registers	Setting Procedure
Stop of the real-time clock	RTC Control Register (WTCR)	See 25.7.7

Table 25-6. Settings Required for Performing Real-time Clock Interrupts

Settings	Setting Registers	Setting Procedure
Setting of the RTC interrupt vector and the RTC interrupt level	See "Chapter: Interrupt Control (Interrupt Controller)".	See 25.7.10
RTC interrupt setting Interrupt request clear Interrupt request enable	RTC Control Register (WTCR)	See 25.7.11

25.7 Q&A

This section explains Q&A of the real-time clock (RTC).

25.7.1 Set the 0.5 Second Count Interval
25.7.2 Initialize the Real-time Clock
25.7.3 Set/Update Number of Days (Day) and Time (Hour/Minute/Second)
25.7.4 Start/Stop the Count of the Real-time Clock
25.7.5 Confirm That the Real-time Clock is Running
25.7.6 Know the Number of Days and Time
25.7.7 Stop the Real-time Clock
25.7.8 Calibrate the Real-time Clock
25.7.9 Interrupt Related Registers
25.7.10 Interrupt Types and Selection Me
25.7.11 Enable Interrupts

25.7.1 Set the 0.5 Second Count Interval

This section explains how to set the 0.5 second count interval.

Stop the real-time clock, and set the value indicated in Table 25-2 to the sub-second register (WTBR) according to the RTC clock frequency.

25.7.2 Initialize the Real-time Clock

This section explains how to initialize the real-time clock.

Perform initialization using the start bit (WTCR:ST).

Write "0" instead of "1" to the start bit to reset all the bits of the Hour/ Minute/ Second counters and the subsecond counter (22-bit down counter) to "0" (initialization) and to stop counting.

25.7.3 Set/Update Number of Days (Day) and Time (Hour/Minute/Second)

This section explains how to set/update number of days (day) and time (hour/minute/second).

Write the values in Day/ Hour/ Minute/ Second registers(WTDR, WTHR, WTMR, WTSR), and then update them using the update bit (UPDT).

Operation	Update Bit (UPDT)
To update the Day/ Hour/ Minute/ Second counters	Set to "1"

25.7.4 Start/Stop the Count of the Real-time Clock

This section explains how to start/stop the count of the real-time clock.

Use the start bit (WTCR:ST) to set.

Operation	Start Bit (ST)
To stop the count of the real-time clock	Set to "0"
To start the count of the real-time clock	Set to "1"

25.7.5 Confirm That the Real-time Clock is Running

This section explains how to confirm that the real-time clock is running.

Confirm using the operation flag (WTCR:RUN).

Operation	Operation Flag (RUN)
The real-time clock has stopped	"0" can be read
The real-time clock is running	"1" can be read

25.7.6 Know the Number of Days and Time

This section explains how to know the number of days and time.

They can be known by reading Day/Hour/Minute/Second registers: WTDR, WTHR, WTMR, WTSR.

However, as word access is not available, access to the respective registers is required. As the time may be misread when the value is read in the boundary of the hour/minute count, perform multiple reads and use the logically correct time.

Example:

When read from second:
1 day 2 hours 59 minutes 59 seconds => 1 day 3 hours 59 minutes 59 seconds => 1 day 3 hours 0 minute 0 second

When read from hour:
1 day 2 hours 59 minutes 59 seconds => 1 day 2 hours 0 minute 0 second => 1 day 3 hours 0 minute 0 second

25.7.7 Stop the Real-time Clock

This section explains how to stop the real-time clock.

See "25.7.4 Start/Stop the Count of the Real-time Clock".

25.7.8 Calibrate the Real-time Clock

This section explains how to calibrate the real-time clock.

When the sub clock(only dual clock product) is selected as the RTC clock, the ratio of main clock: sub clock can be used for calibration. See "Chapter: RTC/WDT1 Calibration".

25.7.9 Interrupt Related Registers

This section explains interrupt related registers.

Setting of RTC interrupt vector and the RTC interrupt level

The following table shows the relationship between interrupt levels and interrupt vectors.

For details on interrupt levels and interrupt vectors, see "Chapter: Interrupt Control (Interrupt Controller)".

Interrupt Vector (Default)	Interrupt level Setting Bit(ICR[4:0])
#37 ($0FFF68_H$)	Interrupt level register ICR21 (00455_H)

The interrupt request flags (INT0, INT1, INT2, INT3, INT4) are not automatically cleared. Therefore, use software to clear the flags prior to restoration from interrupt processing. (Write "0" to INT0, INT1, INT2, INT3, INT4 bits)

25.7.10 Interrupt Types and Selection Method

This section explains the interrupt types and selection method.

There are five interrupt factors as follows:

Interrupt Factor	Interrupt Request Bit	Interrupt Request Enable Bit
Time (1second) count timing	INT0	INTE0
Time (minute) count timing	INT1	INTE1
Time (hour) count timing	INT2	INTE2
1 day count timing	INT3	INTE3
Time(0.5 second) count timing	INT4	INTE4

As interrupt occurs by OR of these five factors, select using the interrupt request enable bit.

25.7.11 Enable Interrupts

This section explains how to enable interrupts.

Use the interrupt request enable bits (WTCR:INTE0, WTCR:INTE1, WTCR:INTE2, WTCR:INTE3, WTCR:INTE4) to perform the operation.

Operation	Setting Procedure
	Interrupt request enable bits (INTE0, INTE1, INTE2, INTE3, INTE4)
To disable interrupts	Set to "0"
To enable interrupts	Set to "1"

Use the interrupt request bits (WTCR:INT0, WTCR:INT1, WTCR:INT2, WTCR:INT3, WTCR:INT4) to clear interrupt requests.

Operation	Setting Procedure
	Interrupt request bits (INT0, INT1, INT2, INT3, INT4)
To clear interrupt requests	Write "0"

25.8 Sample Program

This section explains the sample program of the real-time clock.

Setting Procedure Example 1

Start to count the real-time clock from 10 days 10 hours 10 minutes 00 second, enable the external interrupt (INT0) for "H" level detection, and move to the watch mode.
Restore from the watch mode in case of external interrupt detection, and read the time of the real-time clock.

RTC initialization
RTC startup, interrupt level setting
External interrupt settings
Move to the watch mode
Reading RTC after restoration from the watch mode

<RTC Initialization Settings>

・RTC

	Register name. bit name
Register initialization	WTCR.ST
Setting of interval time (1second)	WTBR
Setting of the time initialization values	WTSR WTMR WTHR WTDR
Initialization setting for RTC interrupts	WTCRM,WTCRL WTCRH

<RTC startup, interrupt level setting>

	Register name. bit name
RTC startup	WTCR.ST
Setting of interrupt level (RTC)	ICR21
Setting of interrupt level (INT0)	ICR00
Setting of the I flag	(CCR)

<RTC time reading preparation (interrupt settings) >

	Register name. bit name
RTC interrupt setting	WTCR .INT0 .INTE0

<RTC interrupt>

	Register name. bit name
Time reading	WTHR WTMR WTSR WTDR
Interrupt disable	WTCR.INTE0

<External interrupt>

	Register name. bit name
Clearing of interrupt request flag	EIRR.ER0

<Interrupt Vector>

Setting of the vector table

<Other>

(Note)

Clock related settings and __set_il (number) setting are required to be performed in advance. See “Chapter: Clock” And ‘Chapter: Interrupt Control (Interrupt Controller)”.

Program Example 1

```
void RTC_sample1(void)
{
        RTC_initial();
        RTC_start();
        EX_INT0_initial();                  /* Subroutine for external interrupt setting*/
        STOP_Hiz_hold_with_clock();  /* Subroutine for moving to the watch mode*/
        RTC_read();
}



void RTC_initial(void)
{
        IO_WTCR.bit.ST = 1;                                 /* Initialization preparation*/
        IO_WTCR.bit.ST = 0;                                 /* Stop (register initialization)*/
        IO_WTBR.word = 0x0F423F;                            /* Count value setting 4MHz/2 × 0x0F423F=0.5 second */
        IO_WTSR.byte = 0x00;                                /* Second setting */
        IO_WTMR.byte = 0x0A;                                /* Minute setting */
        IO_WTHR.byte = 0x0A;                                /* Hour setting */
        IO_WTDR.hword = 0x000A;                             /* Day setting */
        IO_WTCRL.hword = IO_WTCRL.hword & 0x0000;   /* Interrupt flag clear, interrupt disable*/
        IO_WTCRH.byte = 0x00                                /* Interrupt flag clear, interrupt disable*/
}

void RTC_start(void)
{
        IO_WTCR.bit.ST = 1;         /* RTC startup*/
        IO_ICR[21].bit.ICR = 18;    /* The value is arbitrary */
        IO_ICR[00].bit.ICR = 20;    /* The value is arbitrary */
        __EI();                     /* Interrupt enable */
}

RTC_read(void)
{
        IO_WTCR.bit.INT0 = 0;       /* RTC second interrupt request flag clear*/
        IO_WTCR.bit.INTE0 = 1;      /* RTC second interrupt request enable */
}

__interrupt void RTC_read_int(void) /* RTC interrupt */
{
        JIKAN(char) = IO_WTHR.byte & 0x1F;      /*Hour*/
        FUNN(char) = IO_WTMR.byte & 0x3F;       /*Minute*/
        BYOU(char) = IO_WTSR.byte & 0x3F;       /*Second*/
        HI(char) = IO_WTDR.hword ;              /*Day*/
                                                /*Multiple reads*/
        IO_WTCR.bit.INTE0 = 0;                  /* RTC interrupt disable */
}

__interrupt void INT0_int() /* External interrupt */
{
        IO_EIRR0.bit.ER0= 0; /* ER0 second interrupt request flag clear*/
}
```

25.9 Notes

This section explains notes of the real-time clock.

- The interrupt request flags (WTCR:INT0, WTCR:INT1, WTCR:INT2, WTCR:INT3, WTCR:INT4) will be set to "1" when they are written to "0" at the same time when they are set to "1" in case of overflow. (Flag setting takes precedence)
- When reload occurs while update on the sub-second register (WTBRH, WTBRM, WTBRL) is in progress, an unexpected value may be reloaded to the sub-second counter (22-bit down counter). Therefore, it is recommended to update the sub-second register (WTBR) while the start bit (WTCR:ST) is "0".
- When all the bits of the sub-second register (WTBRH, WTBRM, WTBRL) are set to "0", the sub-second counter (22-bit down counter) will not run. Therefore, the real-time clock will not run.
- Carry may occur while Day/Hour/Minute/Second registers (WTDR, WTHR, WTMR, WTSR) are being read, leading to inappropriate read values. Therefore, it is recommended to use interrupt (INT0) to read the number of days and time (Day/Hour/Minute/Second).
- As word access is not available for Day/Hour/Minute/Second registers (WTDR, WTHR, WTMR, WTSR), access to the respective registers is required. Therefore, as the time may be misread when the value is read in the boundary of the hour/minute count, perform multiple reads and use the logically correct time.

  Example:
  When read from second:
  1 day 23 hours 59 minutes 59 seconds= >2 days 0 hour 59 minutes 59 seconds=>2 days 0 hour 0 minute 0 second
  When read from hour:
  1 day 23 hours 59 minutes 59 seconds= >2 days 23 hours 0 minute 0 second=>2 days 0 hour 0 minute 0 second
  When read from day:
  1 day 23 hours 59 minutes 59 seconds=>1 day 0 hour 0 minute 0 second=>2 days 0 hour 0 minute 0 second

  This case is judged as 2 days 0 hour.
- Day/Hour/Minute/Second registers are not cleared by internal reset, while Day/Hour/Minute/Second counters are cleared by internal reset. After internal reset occurs, the ST flag is cleared, and the RTC macro is in the stop state. In addition, counter values prior to internal reset are set to Day/Hour/Minute/Second registers. To use Day/Hour/Minute/Second in case of internal reset, set the values read from the Day/Hour/Minute/Second counters to the Day/Hour/Second registers.
- The number of days register has a built-in function for counting the number of days from "0 day" to "65535 days".
- Notes on Setting the RTC Control Register
  - ☐ When writing "1" to the start bit (ST) from RTC stop state (ST=0) (RTC operation start), do not write "1" to the update bit (UPDT) at the same time as the start bit.
    (While ST=0, writing "1" as byte immediate value to the ST bit and the UPDT bit at the same time is prohibited.)
  - ☐ To write "1" to the update bit (UPDT), do it while RTC is running (ST=1).
  - ☐ While the update bit (UPDT) is "1", writing "0" to the start bit (ST) (RTC stop) is prohibited.
- When returning from the standby watch mode (power shutdown), the register of RTC is not initialized.
- The internal reset is issued at the return from the standby watch mode (power shutdown). Therefore, only the reset factors (power-on reset, internal low-voltage reset, and simultaneous assert of RSTX and NMIX) are accepted. At this time, the register of the RTC is not initialized. If the reset input from the RSTX pin input or the external low-voltage detection flag is set after the start-up, initialize the register of RTC before using.

26. RTC/WDT1 Calibration

This chapter explains the RTC/WDT1 calibration.

26.1 Overview

26.2 Features

26.3 Configuration

26.4 Registers

26.5 Operation

26.1 Overview

This section explains the overview of the RTC/WDT1 calibration.

This module calculates the values for frequency calibrations in CR oscillation circuit built in real-time clock, WDT1 and CSV.

26.2 Features

This section explains features of the RTC/WDT1 calibration.

- RTC Clock source select register

  See "Chapter: Clock" for the selection method. The main clock or sub clock (only dual clock product) can be selected.

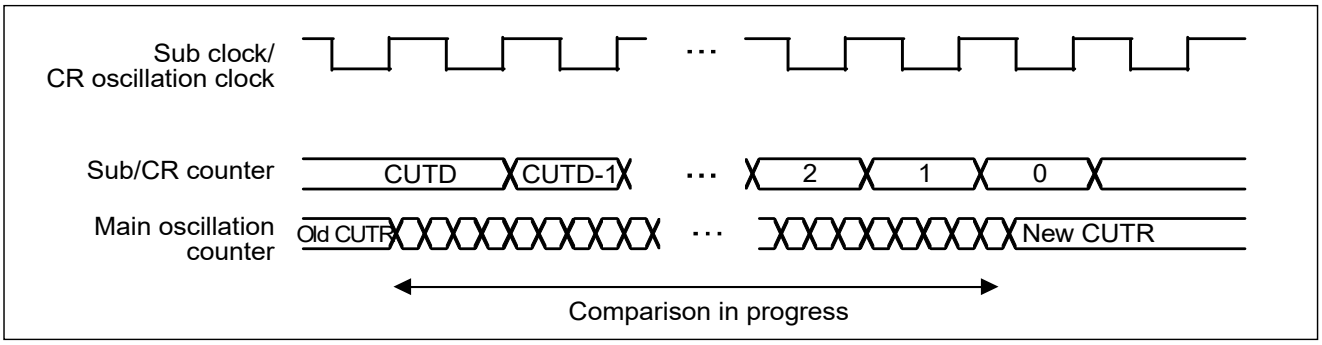
- Real-Time Clock (RTC) Calibration (Only dual clock product. A function that is effective only when the sub clock is used.)

  Operates the main clock driven counter and the sub clock driven counter concurrently (Figure 26-1), and calculates the sub clock frequency from the main clock frequency to set the prescaler value of RTC.

- WDT1(CR clock) calibration

  Operates the main clock driven counter and the CR clock driven counter concurrently (Figure 26-1), and calculates the CR clock frequency from the main clock frequency to set the CR clock trimming value.

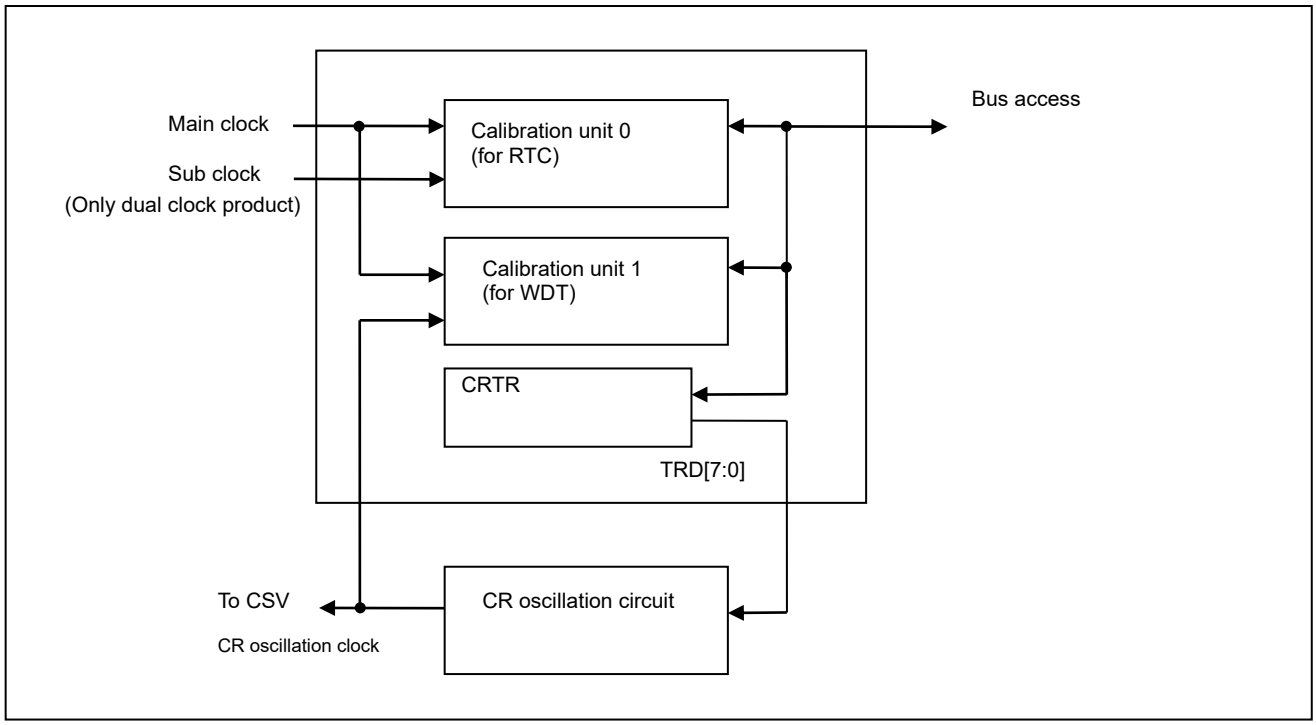
Figure 26-1. Comparison for Counters Driven by Different Clocks



26.3 Configuration

This section explains the configuration of the RTC/WDT1 calibration.

Figure 26-2. Block Diagram



26.4 Registers

This section explains the registers of the RTC/WDT1 calibration.

Figure 26-3. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x04B8	CUCR0		CUTD0		Calibration unit control register 0 Sub clock timer data register
0x04BC	CUTR0				Main oscillation timer data register 0
0x04C0	Reserved	Reserved	Reserved	Reserved	Reserved
0x04C4	CUCR1		CUTD1		Calibration unit control register 1 CR oscillation timer data register
0x04C8	CUTR1				Main oscillation timer data register 1
0x04CC	CRTR	Reserved	Reserved	Reserved	CR oscillation trimming setting register

26.4.1 Calibration Unit Control Register 0: CUCR0 (Calibration Unit Control Register 0)

The bit configuration of the calibration unit control register 0 (CUCR0) is explained.

This register configures calibration start and interrupts for RTC calibration unit.

CUCR0: Address 04B8H (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,W	R/W

[bit7] Reserved

"0" should be written to this bit.

[bit4] STRT (Calibration Start): Calibration start

This bit starts counters driven by main clock and sub clock. The INT bit will be set at the end of comparison.

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" at the end of comparison.

[bit1] INT (Calibration Interrupt): Interrupt

The INT bit will be set to "1" at the end of comparison. If the INTEN bit has been 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

26.4.2 Sub Clock Timer Data Register: CUTD0 (Calibration Unit Timer Data register 0)

The bit configuration of the sub clock timer data register (CUTD0) is explained.

This register configures the time interval for driving sub clock driven counter.

CUTD0: Address 04BA$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	TDD[15:8]							
Initial value	1	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
	TDD[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] TDD[15:0] (Timer Data Data field): Timer data

These bits configure the comparison time interval in number of sub clocks.

26.4.3 Main Oscillation Timer Result Register 0: CUTR0 (Calibration Unit Timer Result register 0)

The bit configuration of the main oscillation timer result register 0 (CUTR0) is explained.

This register displays the number of the main clock driven counter within the interval set using CUTD0.

CUTR0: Address 04BC$_H$ (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

[bit 31 to bit24] Reserved

The read value is always “0”. Data writing is ineffective.

	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

[bit23 to bit0] TDR[23:0] (Timer Data Register): Timer data

These bits display the value of the count in the comparison interval. Read the results at the end of comparison results. The read value during comparison is undefined. Writing has no effect on operation.

26.4.4 Calibration Unit Control Register 1: CUCR1 (Calibration Unit Control Register 1)

The bit configuration of the calibration unit control register 1 (CUCR1) is explained.

This register configures calibration start and interrupts for the WDT calibration unit.

CUCR1: Address 04C4H (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

[bit7] Reserved

"0" should be written to this bit.

[bit4] STRT (Calibration Start): Calibration start

This bit starts counters driven by main clock and CR clock. The INT bit will be set at the end of comparison.

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" at the end of comparison.

[bit1] INT (Calibration Interrupt): Interrupt

The INT bit will be set to "1" at the end of comparison. If the INTEN bit has been 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

26.4.5 CR Clock Timer Data Register: CUTD1 (Calibration Unit Timer Data register 1)

The bit configuration of the CR clock timer data register (CUTD1) is explained.

This register sets the CR clock driven counter drive duration.

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 number of CR clocks.

26.4.6 Main Oscillation Timer Result Register 1: CUTR1 (Calibration Unit Timer Result register 1)

The bit configuration of the main oscillation timer result register 1 (CUTR1) is explained.

This register displays the number of the main clock driven counter in the interval set using 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

[bit31 to bit24] Reserved

The read value is always “0”. Data writing is ineffective.

	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

[bit23 to bit0] TDR[23:0] (Timer Data Register): Timer data

These bits display the value of the count in the comparison interval. Read the results at the end of comparison. The read value during comparison is undefined. Writing is disabled.

26.4.7 CR Oscillation Trimming Setting Register: CRTR (CR oscillator calibration Trimming Register)

The bit configuration of the CR oscillation trimming setting register (CRTR) is explained.

This register sets the trimming value for the CR oscillation circuit.

CRTR: Address 04CC$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	TRD[7:0]							
Initial value	0	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7 to bit0] TRD[7:0] (Trimming Data)

Trimming Value (in steps of about 0.4%)

TRD7	TRD6	TRD5	TRD4	TRD3	TRD2	TRD1	TRD0	Trimming Value[1]	n Value
0	0	0	0	0	0	0	0	-48.01%	0
0	0	0	0	0	0	0	1	-47.61%	1
0	0	0	0	0	0	1	0	-47.23%	2
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
0	1	1	1	1	1	1	1	0% (Initial value)	127
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
1	1	1	1	1	1	0	1	+45.62%	253
1	1	1	1	1	1	1	0	+45.98%	254
1	1	1	1	1	1	1	1	+46.37%	255

[1]: As changes take place according to conditions such as temperature, it is necessary to perform using the procedure explained at "26.5.2 WDT1 Calibration (CR Clock Calibration)".

26.5 Operation

This section explains an operation of the RTC/WDT1 calibration.

26.5.1 Real-Time Clock (RTC) Calibration

26.5.2 WDT1 Calibration (CR Clock Calibration)

26.5.3 Notes

26.5.1 Real-Time Clock (RTC) Calibration

This section explains real-time clock (RTC) calibration.

The calibration procedure is as follows:

1. Sets CUTD0
2. Sets CUCR0:INTEN
3. Sets CUCR0:STRT
4. Interrupt Waiting Loop
5. Interrupt Occurrence
6. CUTR0 Reading
7. Comparison of CUTR0 and CUTD0 can be used to calculate the ratio of the main clock frequency and the frequency of sub clock.
8. Sets the prescaler value in RTC using the value calculated at (7).

26.5.2 WDT1 Calibration (CR Clock Calibration)

This section explains WDT1 calibration (CR clock calibration).

Calculate the trimming value as follows:

1. Set TRD[7:0] to 00000000 to start the calibration unit, and get the CUTR value. Take the CR oscillation frequency calculated from this CUTR value as Fmin.
2. Set TRD[7:0] to 11111111 to start the calibration unit, and get the CUTR value. Take the CR oscillation frequency calculated from this CUTR value as Fmax.
3. Substitute 0 to 255 for n in the following formula, and find n which makes Fer the minimum, which is the trimming value.
   Fstep = (Fmax - Fmin) / 255
   Fer = | (100kHz) - (Fmin + Fstep × n) |

26.5.3 Notes

This section explains notes of the RTC/WDT1 calibration.

The counter value becomes invalid if factors, such as standby mode transition have been included. Write "0" to the STRT bit to stop, and write "1" again to redo.

TOSC32/OSC100 > 2 x TOSC4 + 3 x TCLKP should be fullfilled

TOSC4: Main clock period

TOSC32: Sub clock period

TOSC100: CR oscillation period

TCLKP: Peripheral clock oscillation period

27. Power Consumption Control

This chapter explains the power consumption control.

27.1 Overview
27.2 Features
27.3 Configuration
27.4 Registers
27.5 Operation
27.6 Example of Use

27.1 Overview

This section explains the overview of the power consumption control.

This series have variety of low-power consumption modes and can perform the power consumption control feature accordingly for situations.

27.2 Features

This section explains features of the power consumption control.

- Clock Control for Low-power
  - ☐ Clock division

    The clock division can change the division ratio for each running clock and lower the running frequency accordingly. See "Chapter: Clock".
- Sleep Mode
  - ☐ CPU sleep mode

    By setting this mode, operation of the CPU core are stopped. But, peripherals continue to run. .
  - ☐ Bus sleep mode

    By setting this mode, operation of the CPU core and on-chip buses are stopped.
- Standby Mode
  - ☐ Watch mode

    By setting this mode, all operations are stopped except for some clock oscillations and the timer operations.
  - ☐ Stop mode

    By setting this mode, all clock oscillation and operations are stopped.
- Standby Mode with Power-shutdown
  - ☐ Watch mode with power-shutdown

    By setting this mode, the device can be controlled to set power-shutdown state and all operations are stopped except for some clock oscillations and the timer.
  - ☐ Stop mode with power-shutdown

    By setting this mode, the device can be controlled to set power-shutdown state and all clock oscillations and operations are stopped.

27.3 Configuration

This section explains the configuration of the power consumption control.

Figure 27-1. Block Diagram of Overall Control

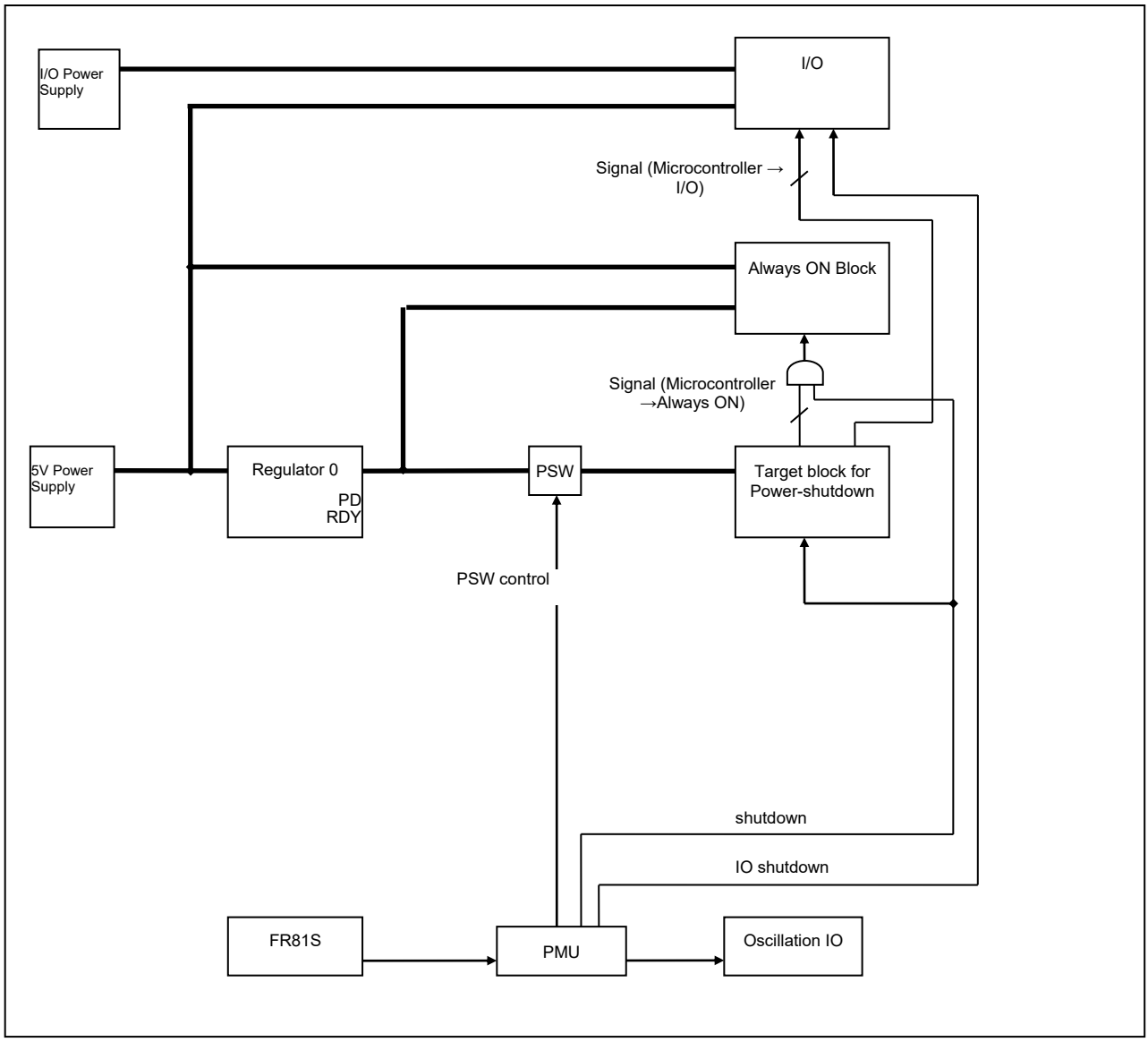
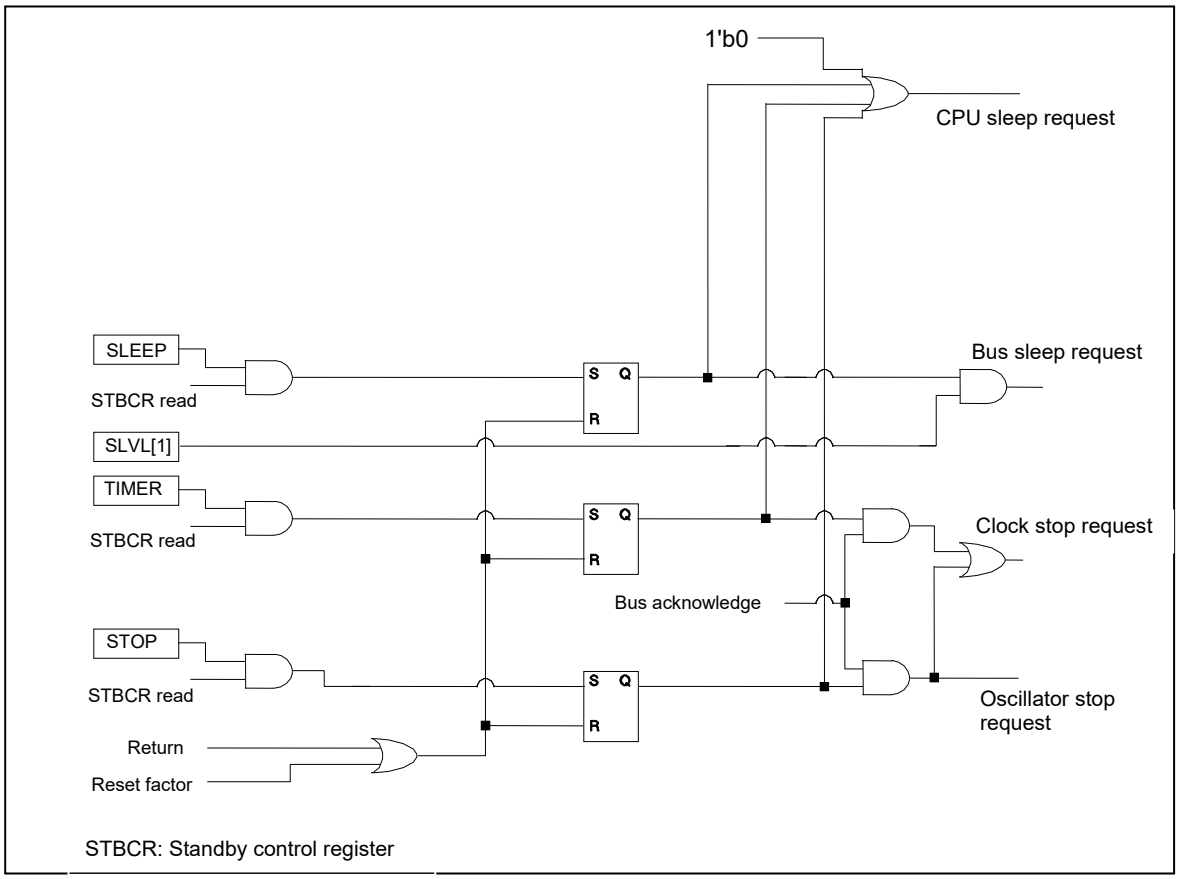


Figure 27-2. Block Diagram of Microcontroller Internal Control



27.4 Registers

This section explains registers of the power consumption control.

Table 27-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0480	Reserved	Reserved	STBCR	Reserved	Standby control register
0x0590	Reserved	PMUCTLR	PWRTMCTL	Reserved	PMU control register Power on timing control register
0x0594	PMUINTF0	PMUINTF1	PMUINTF2	Reserved	PMU interrupt flag register 0 to 2
0x0598	Reserved	Reserved	Reserved	Reserved	Reserved
0x059C	Reserved	Reserved	Reserved	Reserved	Reserved

Notes:

The address of 0x0480 to 0x0481 and 0x0590 is allocated for the register of "Reset". (See "Chapter: Reset".)

A group of registers (except for STBCR) is initialized only when one or some of the following resources occur:

1. Power-on reset
2. Internal low-voltage detection
3. Simultaneous assert of RSTX and NMIX external pins
4. Hardware watchdog reset

   Registers are not initialized by reset of the INIT level and RST level. (exception for STBCR)

27.4.1 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.

Note:

Writing to this register by DMA is prohibited.

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

[bit7] STOP (STOP mode): Stop mode enabled

[bit6] TIMER (TIMER mode): Watch mode enabled

[bit5] SLEEP (SLEEP mode): Sleep mode enabled

Transition to each standby mode; stop, watch and sleep are specified and enabled by those 3 bit. CPU goes into each standby mode by reading STBCR after writing the values shown below to those 3 bit.

STOP	TIMER	SLEEP	Transition to Each Standby Mode Enabled
0	0	0	No transition (initial value)
0	0	1	Transit to sleep mode by reading STBCR
0	1	X	Transit to watch mode by reading STBCR
1	X	X	Transit to stop mode by reading STBCR

The read value of each bit is as follows regardless of the writing value.

STOP	TIMER	SLEEP	Transition to Each Standby Mode Enabled
0	0	0	No transition
0	0	1	Transit to sleep mode
0	1	0	Transit to watch mode
1	0	0	Transit to stop mode

These bits are cleared to an initial value by generating the wake up factor from each low-power consumption mode.

[bit4] Reserved

The read value is always "0". This bit must always be written to "0".

[bit3, bit2] Reserved

The read value is always "0". These bits must always be written to "0".

[bit1, bit0] SLVL[1:0] (Standby Level): Standby level setting

These bits control the operations in standby mode and sleep mode as follows.

Mode	SLVL[1:0]	Operation Control
Stop mode	0x	Pins are not used for high impedance.
	1x	Pins are used for high impedance.
Watch mode	0x	Pins are not used for high impedance.
	1x	Pins are used for high impedance.
Sleep mode	0x	CPU sleep mode (stop only CPU)
	1x	Bus sleep mode (stop CPU and on-chip bus)[1]

[1]: On-chip bus will run when DMA transfer is in progress.

For information on pins with high impedance, see "Appendix".

27.4.2 PMU Control Register: PMUCTLR (Power Management Unit Control register)

The bit configurations of the PMU control register are shown below.

This register controls PMU.

PMUCTLR: Address 0591$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SHDE	Reserved	IOCTMD	IOCT	Reserved			
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R0,W0	R/W	R/W	R0,W0	R0,W0	R0,W0	R0,W0

This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assert of RSTX and NMIX, and hardware watchdog timer reset.

[bit7] SHDE (Shut Down Enable)

This setting is for whether you establish shutdown mode when the CPU mode transits to standby (watch/stop).

SHDE	SHDE Mode Enable
0	When transiting to standby, you must not execute shutdown process.
1	When transiting to standby, you must execute shutdown process.

[bit6] Reserved

The read value is always "0". This bit must always be written to "0".

[bit5] IOCTMD (I/O Clear Timing Mode)

This bit selects timing to maintain the I/O state when returning from standby (ShutDown) mode. (Hardware process)

IOCTMD	I/O Maintain Cancellation Request Mode
0	I/O state is maintained until returning from standby (WATCH and STOP) mode.
1	I/O state is maintained until IOCT register is cleared.

[bit4] IOCT (I/O Clear Timing)

By setting this bit to “1” when IOCTMD=1, I/O state maintaining are cancelled.

IOCT	I/O Maintain Cancellation Request
0	No request
1	Request is on-going

This bit is cleared to "0" automatically after cancellation of I/O maintaining by I/O state maintaining cancellation request is accepted.

Writing at times other than when I/O is maintained is invalid.

Writing this bit to "0" is invalid.

[bit3 to bit0] Reserved

The read value is always "0". These bits must always be written to "0".

27.4.3 Power on Timing Control Register: PWRTMCTL

The bit configurations of the Power on Timing control register are shown below.

This register controls timing for power-on reset and so on.

PWRTMCTL: Address 0592$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved					PTC[2:0]		
Initial value	0	0	0	0	0	0	1	1
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W

This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assert of RSTX and NMIX, and hardware watchdog timer reset.

[bit7 to bit3] Reserved

The read value is always "0". These bits must always be written to "0".

[bit2 to bit0] PTC[2:0] (Power on Timing Cycle setting)

These bits set the rising time for PSW.

PTC[2:0]	Rising Time	Remarks of the Case that PMUCLK=32 kHz
000	1×(1/PMUCLK)	30 µS
001	3×(1/PMUCLK)	90 µS
010	5×(1/PMUCLK)	150 µS
011	9×(1/PMUCLK)	270 µS
100	Prohibit	-
101	2×(1/PMUCLK)	60 µS
110	4×(1/PMUCLK)	120 µS
111	7×(1/PMUCLK)	210 µS

27.4.4 PMU Interrupt Flag Register 0: PMUINTF0 (Power Management Unit Interrupt Flag0 register)

The bit configurations of the PMU interrupt flag register 0 are shown below.

This register indicates the interrupt request by external input at shutdown.

PMUINTF0: Address 0594$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EIF15	EIF14	EIF13	EIF12	EIF11	EIF10	EIF9	EIF8
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

This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assert of RSTX and NMIX, and hardware watchdog timer reset.

[bit7 to bit0] EIF15 to EIF8 (External Interrupt Flag15 to 8)

These flags indicate the interrupt request by external input at shutdown.

EIFn	External Interrupt Rrequest
0	No request
1	Request

n -> The number from 15 to 8 is assigned.

These bits are enabled only at shutdown.

These bits are cleared by writing to "0". Writing to "1" is invalid.

27.4.5 PMU Interrupt Flag Register 1: PMUINTF1 (Power Management Unit Interrupt Flag1 register)

The bit configurations of the PMU interrupt flag register 1 are shown below.

This register indicates the interrupt request by external input at shutdown.

PMUINTF1: Address 0595H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EIF7	EIF6	EIF5	EIF4	EIF3	EIF2	EIF1	EIF0
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

This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assert of RSTX and NMIX, and hardware watchdog timer reset.

[bit7 to bit0] EIF7 to EIF0 (External Interrupt Flag7 to 0)

These flags indicate the interrupt request by external input at shutdown.

EIFn	External Interrupt Request
0	No request
1	Request

n -> The number from 7 to 0 is assigned.

These bits are enabled only at shutdown.

These bits are cleared by writing to "0". Writing to "1" is invalid.

27.4.6 PMU Interrupt Flag Register 2: PMUINTF2 (Power Management Unit Interrupt Flag2 register)

The bit configurations of the PMU interrupt flag register 2 are shown below.

This register indicates the interrupt request at shutdown.

PMUINTF2: Address 0596H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RIF	NIF	MTIF	STIF	Reserved			
Initial value	0	0	0	0	0	0	0	0
Attribute	R(RM1), W	R(RM1), W	R(RM1), W	R(RM1), W	R0,W0	R0,W0	R0,W0	R0,W0

This register will be initialized by power-on reset, internal low-voltage reset, reset by simultaneous assert of RSTX and NMIX, and hardware watchdog timer reset.

[bit7] RIF (RTC Interrupt Flag)

This flag indicates the interrupt request by RTC at shutdown.

RIF	RTC Interrupt Request
0	No request
1	Request

This bit is enabled only at shutdown.

This bit is cleared by writing to "0". Writing to "1" is invalid.

[bit6] NIF(NMI Flag)

This flag indicates the interrupt request by NMI at shutdown.

NIF	NMI Interrupt Request
0	No request
1	Request

This bit is valid only at shutdown.

This bit is cleared by writing to "0". Writing to "1" is invalid.

[bit5] MTIF (Main Timer Interrupt Flag)

This flag indicates the interrupt request by Main Timer at shutdown.

MTIF	Main Timer Interrupt Request
0	No request
1	Request

This bit is enabled only at shutdown.

This bit is cleared by writing to "0". Writing to "1" is invalid.

MTIF is not set during return from the standby mode (shut-down) because the internal reset is generated.

[bit4] STIF (Sub Timer Interrupt Flag)

This flag indicates the interrupt request by Sub Timer at shutdown.

STIF	Sub Timer Interrupt Request
0	No request
1	Request

This bit is enabled only at shutdown.

This bit is cleared by writing to "0". Writing to "1" is invalid.

STIF is not set during return from the standby mode (shut-down) because the internal reset is generated.

[bit3 to bit0] Reserved

The read value is always "0". These bits must always be written to "0".

27.5 Operation

The power consumption control features of this series are explained as follows.

27.5.1 Clock Control
27.5.2 List of Clock Supply in Low-power Consumption Mode
27.5.3 Sleep Mode
27.5.4 Standby Mode: Watch Mode
27.5.5 Standby Mode: Watch Mode with Power-shutdown
27.5.6 Standby Mode: Stop Mode
27.5.7 Standby Mode: Stop Mode with Power-shutdown
27.5.8 Stop State of Microcontroller
27.5.9 Transition to Illegal Standby Mode
27.5.10 Restrictions on Power-shutdown and Normal Standby Control

27.5.1 Clock Control

This section explains the clock control of the power consumption control.

This series can perform optimization of power consumption and processing ability by adjusting each operating clock.

Division Setting

See "Chapter: Clock".

Stopping of Unused Clocks

Following clocks have the setting to stop separately.

- External bus clock (TCLK): Can select to supply/stop in bus sleep mode

For details on the setting method, see "Chapter: Clock".

27.5.2 List of Clock Supply in Low-power Consumption Mode

The list of clock supply in low-power consumption mode is shown below.

Table 27-2. List of Clock Supply 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)	○	○	×	×
External bus I/F clock (TCLK)	○	○	[2]	×
PLL clock (PLLCLK)	○	○	×	×
Main clock (MCLK)	○	×	×	×
Sub clock (SBCLK)	○	×	×	×
CR oscillation	○[4]	○[4]	×[3]	×[3]

○: Stops

×: Does not stop.
(If the main/sub/PLL let them stop in each clock setting register, follow those.)

[1]: When on-chip bus clock (HCLK) is selected as CAN prescaler clock, this clock stops. When PLL output is selected, it depends on PLL output. Otherwise, this clock does not stop.

[2]: This clock is set by the DIVR1:TSTP bit. See "Chapter: Clock".

[3]: During sleep mode, the CR oscillation does not stop, but the watchdog timer 1 (HWWDT) stops.

[4]: It is necessary to set it beforehand to stop the CR oscillation at the standby.

See the description of CSVCR:RCE in "Chapter: Clock Supervisor".

27.5.3 Sleep Mode

This section explains sleep mode.

Sleep mode is the mode which CPU and on-chip bus stop and only the peripherals run. In sleep mode, there are several modes for different stop ranges.

- CPU sleep mode: Stops CPU only.
- Bus sleep mode: Stops CPU and on-chip bus.

The stop state is continued until a wake up request occurs. A return to the program operation within several clock times is possible by generating a wake up request.

Following are the explanation of an operation of each mode.

CPU Sleep Mode

CPU sleep mode is the mode to stop the CPU operation.

In this mode, the DMA controller and on-chip bus can continue their operation, while more power will be consumed than that of bus sleep mode.

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, stop 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.

Configuration of Sleep Mode

Before activating sleep mode, select whether to supply/stop external bus clock in sleep mode with the values set to bit7:TSTP in the DIVR1 register.

- When setting bit7:TSTP="0" in the DIVR1 register, the external bus clock does not stop.
- When setting bit7:TSTP="1" in the DIVR1 register, the external bus clock stops.

When 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.

Activation of Sleep Mode

To activate sleep mode, follow the steps below:

1. Write "001" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
2. 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
```

Wake Up from the Sleep Mode

To stop sleep mode, follow the conditions below:

- 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 wake up from interrupt request, CPU does not always have to accept this interrupt request. When an interrupt request is not accepted, the program starts from next instruction which activates 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.

Effect of Sleep Mode

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. While, a return to the program operation within several clock times is possible by generating a wake up request.

27.5.4 Standby Mode: Watch Mode

This section explains 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. When the sub clock (SBCLK) is selected as the clock source, only the sub clock oscillates and only the sub timer counts.

Note:

- Enter the standby mode only when main RUN or sub RUN is in progress. For the operation of transition to standby mode from PLL run, see "27.5.9 Transition to Illegal Standby Mode".
- Transition to the standby state during FLASH program/erase is prohibited.

Configuration of Watch Mode

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".

Activation of Watch Mode

To activate watch mode, follow the steps below:

1. "0" is written in bit7:SHDE of the PMUCTLR register.
2. When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to watch mode.)
3. Write "010" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
4. 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
```

Wake Up from the Watch Mode

To stop watch mode, follow the conditions below.

- 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 wake up from an interrupt request, CPU does not always have to accept this interrupt request. When an interrupt request is not accepted, the program starts from next instruction which activates watch mode.

Effect of Watch Mode

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.[1]

[1]: When continue to run program with activate clocks.

27.5.5 Standby Mode: Watch Mode with Power-shutdown

This section explains standby mode: watch mode with power-shutdown.

Watch mode with power-shutdown is the mode to continue oscillation only for the specific clock and to continue counting the clock timer corresponding to that clock while the power is shut off. When the sub clock (SBCLK) is selected as the clock source, only the sub clock oscillates and only the sub timer counts.

Note:

- Enter the standby mode only when main RUN and sub RUN is in progress. For the operation of transition to standby mode from PLL run, see "27.5.9 Transition to Illegal Standby Mode".
- Transition to the standby state during FLASH program/erase is prohibited.

Configuration of Watch Mode with Power-shutdown

Before activating watch mode with power-shutdown, set and control the followings.

1. Set the state of external pins in watch mode with power-shutdown 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".

Activation of Watch Mode with Power-shutdown

To activate watch mode with power-shutdown, follow the steps below:

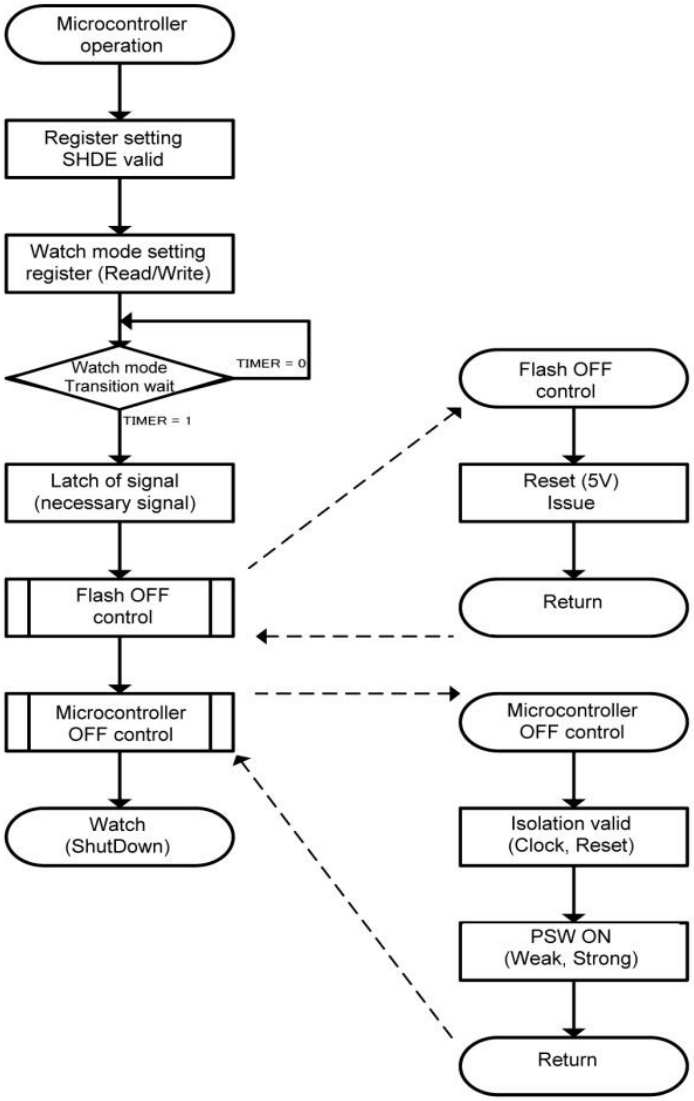
1. "1" is written in bit7:SHDE of the PMUCTLR register.
2. When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to watch mode with power-shutdown.)
3. Write "010" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
4. Read the STBCR 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 with power-shutdown.

[Example] Sample program of watch mode activation (power-shutdown)

```
LDI     #value_of_PMU, R0     ; SHDE bit ="1", IOCTMD/IOCT bit setting
LDI     #_PMUCTLR, R12        ;
STB     R0, @R12              ; Write
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 with power-shutdown)
MOV     R0, R0                ; Dummy processing for pipeline adjustment
NOP                           ; Dummy processing for pipeline adjustment
```

Figure 27-3. Transition Sequence to Watch Mode with Power-shutdown



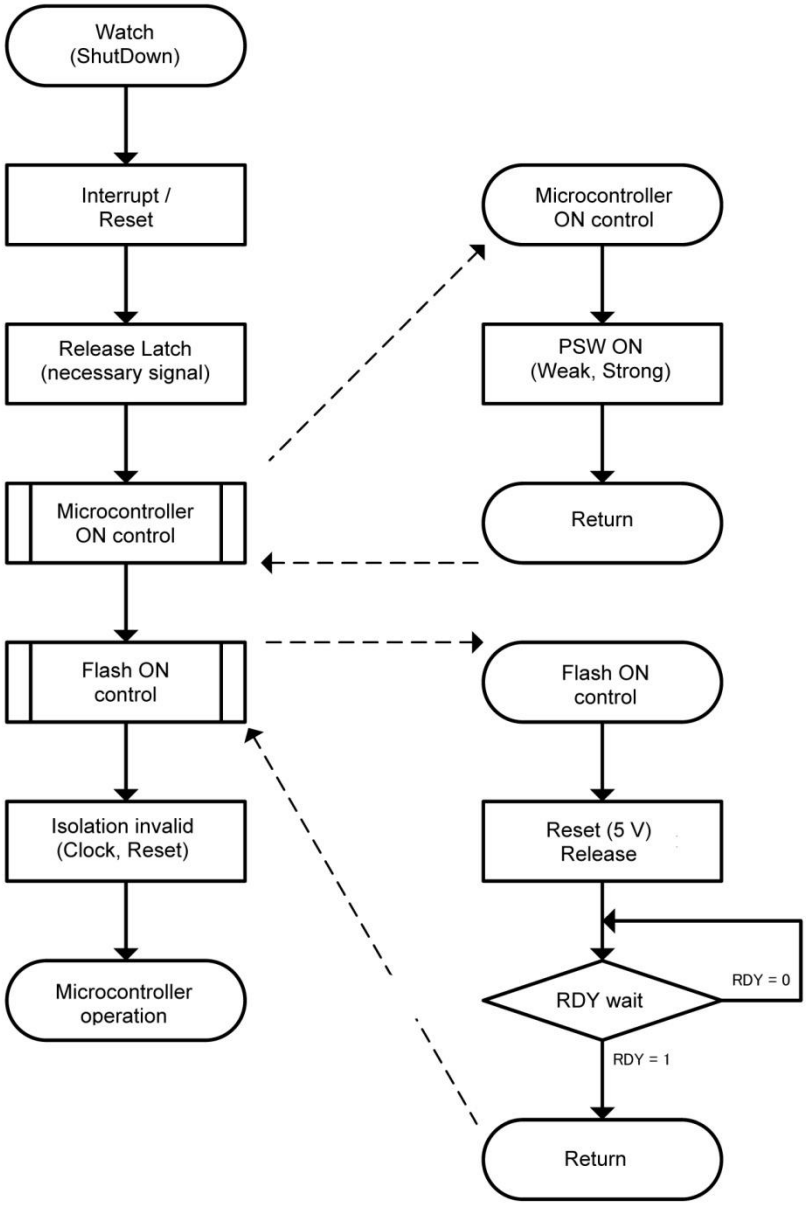
Wake Up from the Watch Mode with Power-shutdown

To stop watch mode with power-shutdown, follow the conditions below:

- Reset
- Generation of external interrupt request
- Generation of NMI request
- Generation of RTC interrupt request
- Generation of main/sub timer interrupt request

For wake up by interrupt request, CPU and interrupt controller do not always have to accept this interrupt request. CPU always starts operation from the reset state.
The register of RTC, LCD controller and external interrupt input (IOCTMD=1) is not initialized.
Only the reset factors (power-on reset, internal low-voltage reset, and simultaneous assert of RSTX and NMIX) are accepted during wake-up. At this time, the register of the RTC, LCD controller and external interrupt input (IOCTMD=1) is not initialized. If the reset input from RSTX pin input or the external low-voltage detection flag are set after the start-up, initialize the RTC/external interrupt input register before using.

Figure 27-4. Restore Sequence from Watch Mode with Power-shutdown



Effect of Watch Mode with Power-shutdown

You can reduce wait current for unnecessary circuit greatly by watch mode with power-shutdown. 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 without clock oscillation stabilization wait is possible by generating a wake up request.

27.5.6 Standby Mode: Stop Mode

This section explains standby mode: stop mode.

Stop mode is the mode to stop all clock oscillations and minimize power consumption of this series.

Note:

- Enter the standby mode only when main RUN or sub RUN is in progress. For the operation of transition to standby mode from PLL run, see “27.5.9 Transition to Illegal Standby Mode”.
- Transition to the standby state during FLASH program/erase is prohibited.

Configuration of Stop Mode

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".

Activation of Stop Mode

To activate stop mode, follow the steps below:

1. "0" is written in bit7:SHDE of the PMUCTLR register.
2. When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to stop mode.)
3. Write "100" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
4. 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
```

Wake Up from the Stop Mode

To stop stop mode, follow the conditions below:

- 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 wake up from interrupt request, CPU does not always have to accept this interrupt request. When an interrupt request is not accepted, the program starts from the next instruction which activates stop mode.

Effect of Stop Mode

You can minimize power consumption on the external input event wait state by using stop mode. While, a return to the program operation after generating a wake up request needs the oscillation stabilization wait time.

27.5.7 Standby Mode: Stop Mode with Power-shutdown

This section explains standby mode: stop mode with power-shutdown.

Stop mode with power-shutdown is the mode to stop all clock oscillations and minimize power consumption of this series.

Note:

- Enter the standby mode only when main RUN or sub RUN is in progress. For the operation of transition to standby mode from PLL run, see "27.5.9 Transition to Illegal Standby Mode".
- Transition to the standby state during FLASH program/erase is prohibited.

Configuration of Stop Mode with Power-shutdown

Before activating stop mode with power-shutdown, set and control the followings.

1. Set the state of external pins in stop mode with power-shutdown 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".

Activation of Stop Mode with Power-shutdown

To activate stop mode with power-shutdown, follow the steps below:

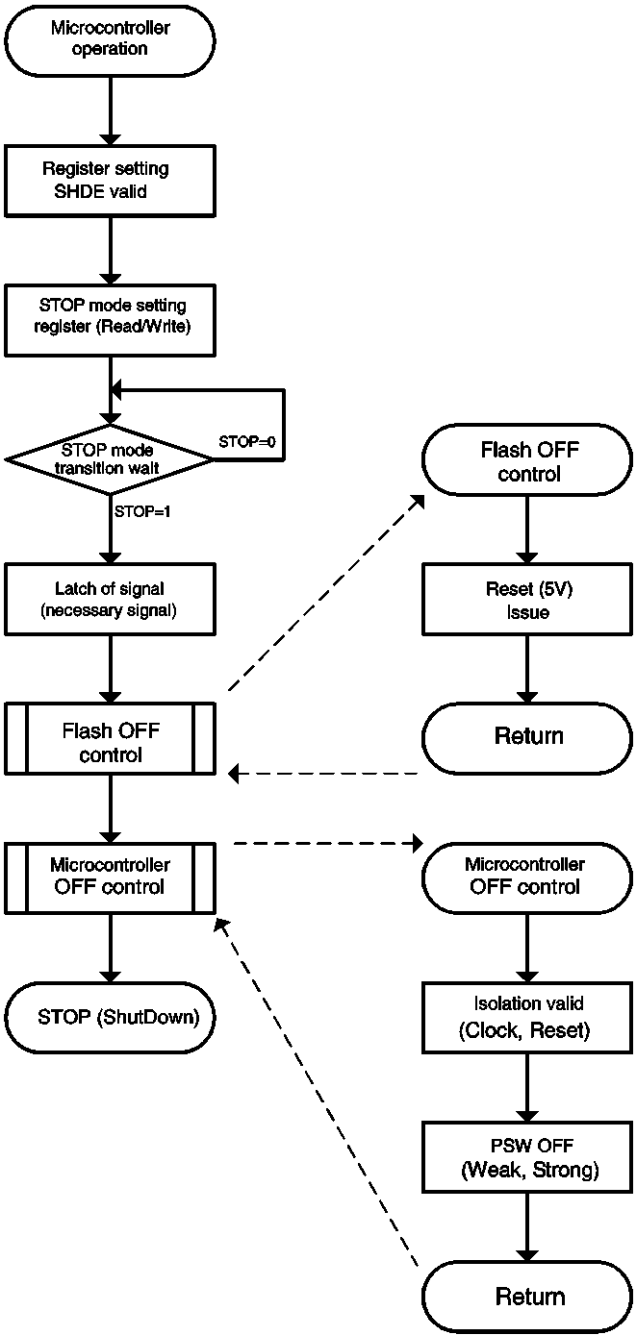
1. "1" is written in bit7:SHDE of the PMUCTLR register.
2. When performing PLL RUN, CPU must go into main RUN state first. (When performing sub RUN state, it transits directly to stop mode with power-shutdown.)
3. Write "100" to bit7:STOP, bit6:TIMER, bit5:SLEEP in the STBCR register.
4. Read the STBCR 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 with power-shutdown.

[Example] Sample program of stop mode with power-shutdown activation

```
LDI     #value_of_PMU, R0     ;SHDE bit ="1", IOCTMD/IOCT bit setting
LDI     #_PMUCTLR, R12        ;
STB     R0, @R12              ; Write
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 with power-shutdown)
MOV     R0, R0                ; Dummy processing for pipeline adjustment
NOP                           ; Dummy processing for pipeline adjustment
```

Figure 27-5. Transition Sequence to Stop Mode with Power-shutdown



Wake Up from the Stop Mode with Power-shutdown

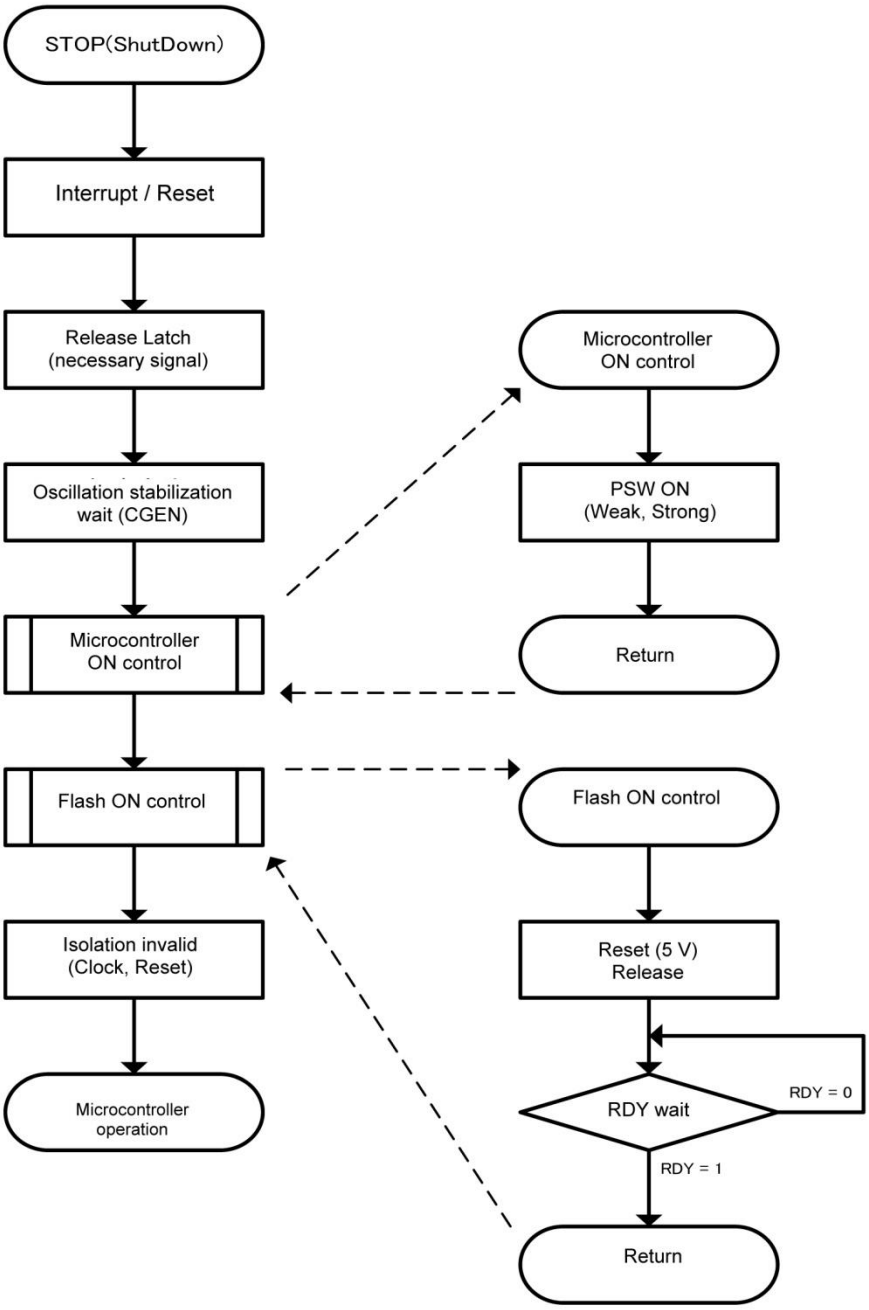
To stop the stop mode with power-shutdown, follow the conditions below:

- Reset
- Generation of external interrupt request
- Generation of NMI request

For wake up by an interrupt request, CPU and the interrupt controller do not always have to accept this interrupt request. CPU always starts operation from the reset state.

The register of the external interrupt input (IOCTMD=1) is not initialized.
Only the reset factors (power-on reset, internal low-voltage reset and simultaneous assert of RSTX and NMIX) are accepted during wake-up. At this time, the register of the external interrupt input (IOCTMD=1) is not initialized. If the reset input from RSTX pin input or the external low-voltage detection flag are set after the start-up, initialize the register before using.

Figure 27-6. Return Sequence from Stop Mode with Power-shutdown



Effect of Stop Mode with Power-shutdown

You can minimize wait current for unnecessary circuit by stop mode with power-shutdown. While, a return to the program operation after generating a wake up request needs the oscillation stabilization wait time.

27.5.8 Stop State of Microcontroller

The stop state of microcontroller is shown below.

When the transition from the state of the standby mode (watch mode/watch mode with power-shutdown/stop mode/stop mode with power-shutdown)transition prohibition to the standby is controlled, the standby transition is not concluded.

< State of standby transition prohibition >

- Connecting OCD
- Operating PLL

<Standby control not done by microcontroller stop condition>

- Flash memory power saving control
- Oscillation stop (At the stop mode stop mode with power-shutdown)

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. See "27.5.9 Transition to Illegal Standby Mode" for the illegal standby mode transition.

27.5.9 Transition to Illegal Standby Mode

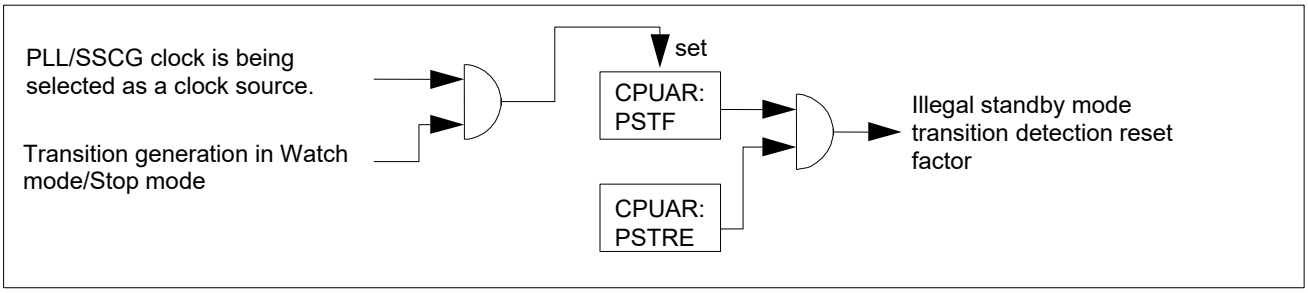
Transition to illegal standby mode is shown below.

If transit from the PLL run state to standby mode (watch mode/watch mode (power-shutdown)/stop mode/stop mode (power-shutdown)) is executed, 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 of clock).

The PSTF flag of the CPUAR register is set concurrently with transition to the standby mode. When the PSTRE bit in the CPUAR register is set, reset occurs by illegal standby mode transition detection reset source. For the CPUAR register, see "7.4.3 CPU Abnormal Operation Register: CPUAR" in "Chapter: Reset".

Figure 27-7. Generation Diagram of Illegal Standby Mode Transition Detection Reset Source



27.5.10 Restrictions on Power-shutdown and Normal Standby Control

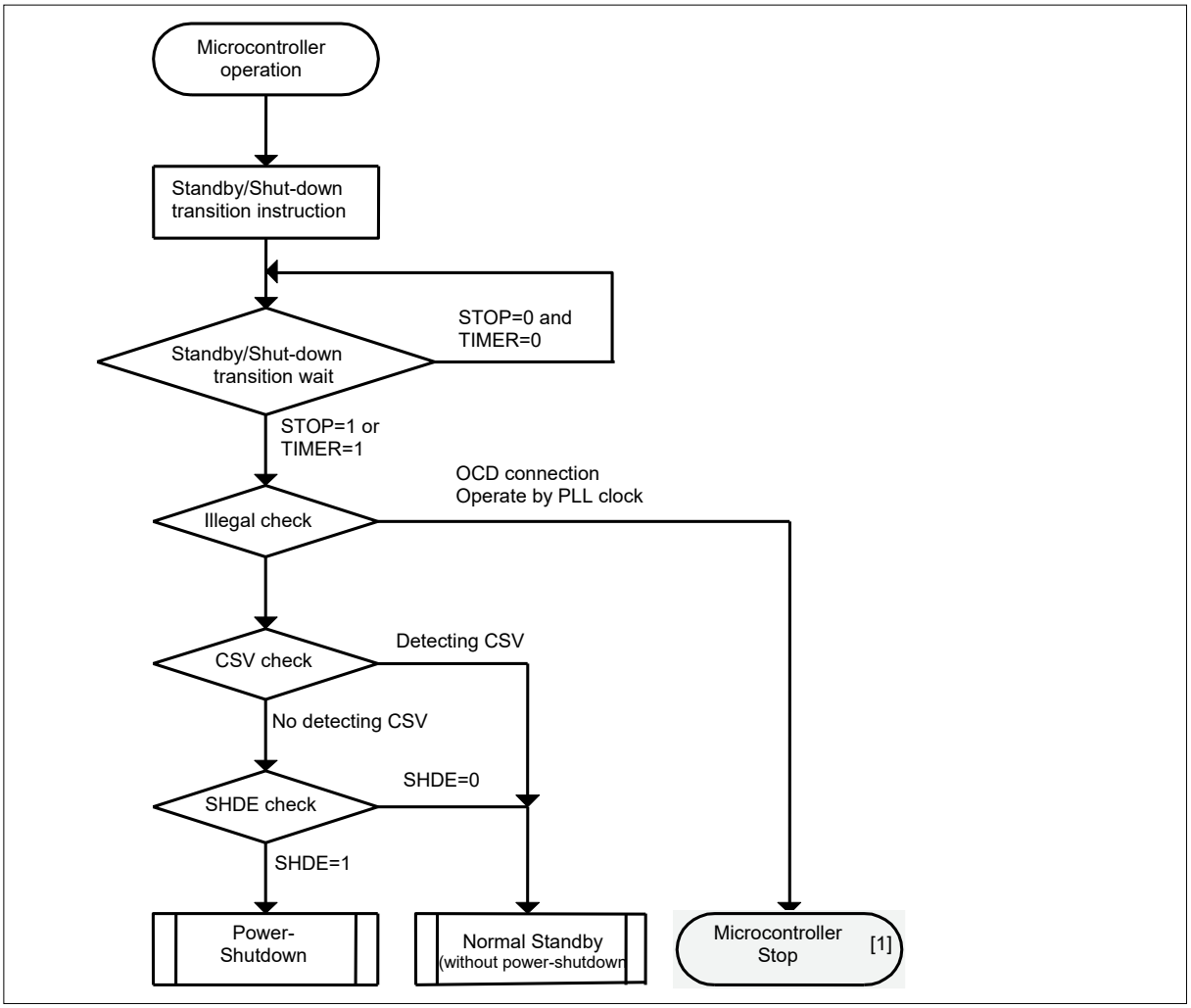
Restrictions on shutdown and normal standby control are explained below.

This microcontroller does not perform standby control on the following conditions:

- When CPU operate with PLL
- When enabling OCD operation
- When missing the clock by CSV function [2],[3].

The standby control does not operate in the states above, but CPU is in the standby state.

Figure 27-8. Restriction on Power-shutdown and Normal Standby Control



[1]: This state is not recognized as power-shutdown and the state that CPU transits to standby mode.

[2]: It is the case when stop of operating clock source are detected by CSV circuit. For instance, in the case that CPU operate with the CR clock after main clock stop are detected, CPU does not perform standby control. However, it is not the limitation case, when stop detection of sub-clock is done while CPU run with the main clock.

[3]: When standby (power-shutdown) transition is directed after the operating clock source is missing, it usually becomes standby processing. Moreover, it is necessary to note it because the function of CSV stops when the power-shutdown permission is enabled with the operating clock source is not missing.

Only a part of registers is maintained at returning, because the power supply is not supplied to almost all blocks inside in standby mode with power-shutdown Table 27-3 shows the list of registers that are stored at return from standby mode with power-shutdown).

Table 27-3. List of Registers that are Stored at Return from Standby Mode with Power-shutdown

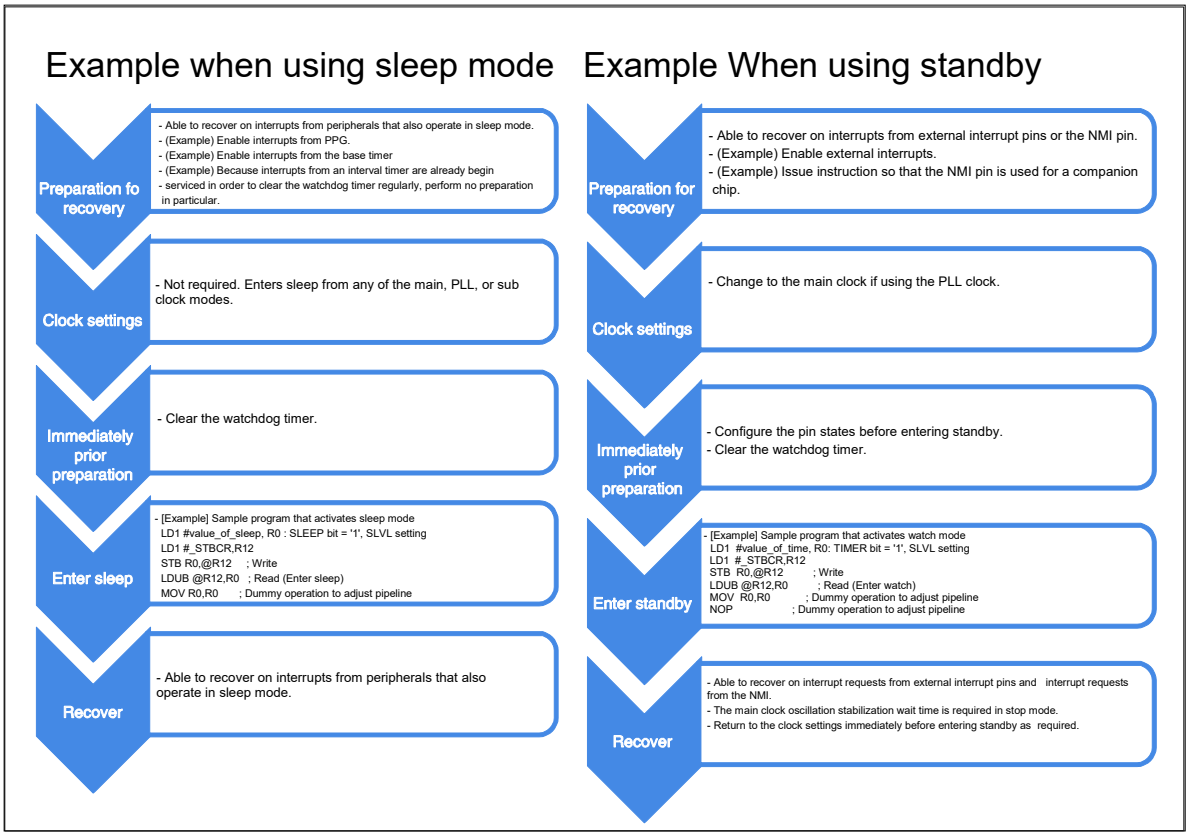
Register Group	Register, Flag Name	Type	Address	Remarks
PMU register	PMUSTR.PMUST	Flag	0590_H bit7	
	PMUSTR.PONR_F	Flag	0590_H bit1	
	PMUSTR.RSTX_F	Flag	0590_H bit0	
	PMUCTLR	Register	0591_H	
	PWRTMCTL	Register	0592_H	
	PMUINTF0	Flag	0594_H	
	PMUINTF1	Flag	0595_H	
	PMUINTF2	Flag	0596_H	
Reset source register	CPUAR.PMDF	Flag	$051A_H$ bit2	
	CPUAR.PSTF	Flag	$051A_H$ bit1	
	CPUAR.HWDF	Flag	$051A_H$ bit0	
	LVD5R.LVD5R_F	Flag	0584_H bit0	
	LVD5F.LVD5F_F	Flag	0585_H bit0	
	LVD.LVD_F	Flag	0586_H bit0	
Low-voltage detection register	LVD5F.LVD5F_PD	Register	0585_H bit7	
	LVD5F.LVD5F_OE	Register	0585_H bit3	
	LVD.LVD_PD	Register	0586_H bit7	
	LVD.LVD_OE	Register	0586_H bit3	
CSV register	CSVCR	Register	$056D_H$	
LCD controller register	All		$05A8_H$-$05BF_H$	[1]
External interrupt register	EIRR0/1	Register	0550_H/0554_H	[3]
	ENIR0/1	Register	0551_H/0555_H	[3]
	ELVR0/1	Register	0552_H/0556_H	[3]
RTC register	WTDR	Register	$055E_H$-$055F_H$	
	WTCR	Register	0561_H-0563_H	[1]
	WTBR	Register	0565_H-0567_H	
	WTHR	Register	0568_H	
	WTMR	Register	0569_H	
	WTSR	Register	$056A_H$	
Clock selection register	CSELR.SCEN	Register	0510_H bit7	[1], [2]
	CMONR.SCRDY	Flag	0511_H bit7	[1], [2]
	CCRTSELR.CST	Flag	0530_H bit7	[1], [2]
	CCRTSELR.CSC	Register	0530_H bit0	[1], [2]

[1]: These registers are initialized at return from stop mode with power-shutdown.
[2]: These registers are for the dual clock products.
[3]: It is initialized at PMUCTLR:IOCTMD=0.

27.6 Example of Use

The example of activation of sleep mode and standby mode is shown below.

Figure 27-9. Example of Activation of Sleep Mode and Standby Mode



28. Low-voltage Detection (Internal Low-voltage Detection)

This chapter explains the low-voltage detection (internal low-voltage detection).

28.1 Overview

28.2 Features

28.3 Configuration

28.4 Registers

28.5 Operation

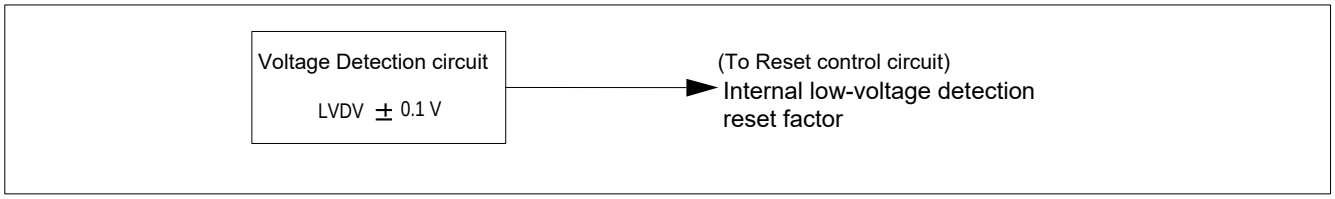
28.6 Notes

28.1 Overview

This section explains the overview of the low-voltage detection (internal low-voltage detection).

The internal low-voltage detection is a function that monitors the internal voltage and detects when the internal voltage falls below the detection voltage value. When the internal low-voltage detection sets the detection flag, the device goes to the reset state by low-voltage detection reset.

Figure 28-1. Block Diagram of Low-Voltage Detection (Internal Low-voltage Detection) (Overview)



28.2 Features

This section explains the features of the low-voltage detection (internal low-voltage detection).

Low-voltage detection circuit

- Method: Generates settings initialization reset if a voltage lower than LVDV ± 0.1 V is detected. (LVDV: 0.9 V)
- Number of units: 1
- Operation: Continues to operate in sleep mode, stop mode, and watch mode.
- Voltage comparison circuit: Compares the internal voltage to the detection voltage,
  and changes output from "H" to "L" if low-voltage is detected.
  Operates constantly after the power is turned on.

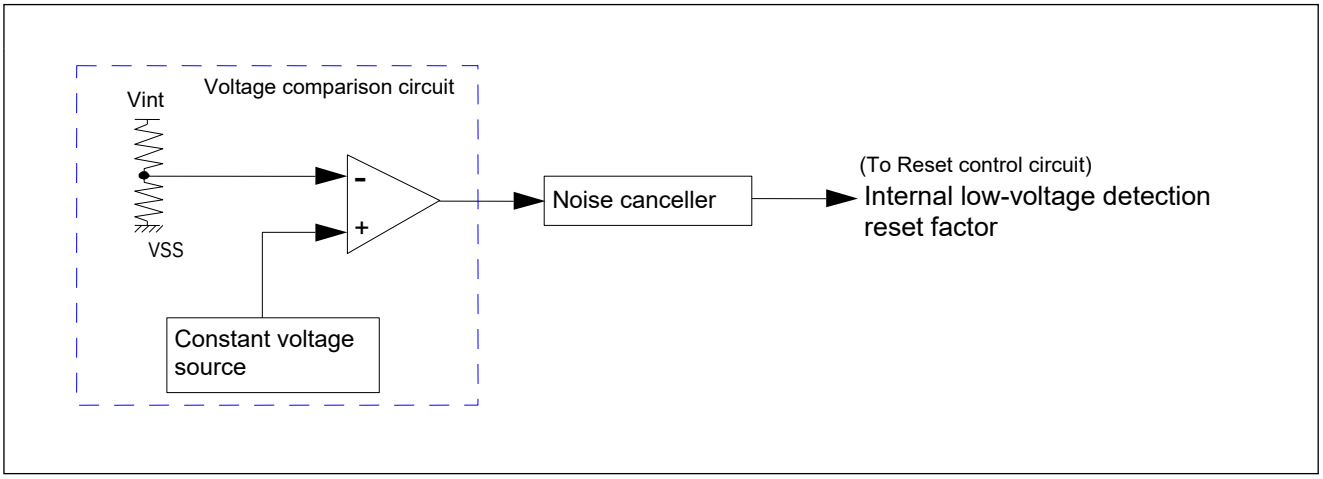
28.3 Configuration

This section shows the configuration of the low-voltage detection (internal low-voltage detection).

Configuration Diagram of low-voltage detection (internal low-voltage detection)

Figure 28-2 shows Configuration diagram.

Figure 28-2. Configuration Diagram



28.4 Registers

This section explains the registers of the low-voltage detection (internal low-voltage detection).

Table 28-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0584	LVD5R	LVD5F	LVD	Reserved	Internal low-voltage detection register

28.4.1 Internal Low-Voltage Detection Register: LVD (Low-Voltage Detect Internal Power Fall Register)

The bit configuration of the internal low-voltage detection register is explained.

This register has the internal low-voltage detection flag (LVD_F) and 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 is used to set whether to detect a fall in internal voltage or not.

LVD_PD	Internal Voltage Fall Power Down Setting
0	Disabled (Detection is executed.)
1	Enabled (Detection is stopped.)

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 set it before 100 µs, some detection flag setting will be occur.

[bit6 to bit4] LVD_SEL [2:0] (Low-voltage Detect Internal Power fall Select)

These bits are a selection signal of detection level of the internal voltage fall detection.

LVD_SEL[2:0]	Internal Voltage Fall Detection Voltage Setting
100	0.9 V ± 0.1 V
Other than the above	Setting is prohibited

These bits can be rewritten only when LVD_OE="1".

[bit3] LVD_OE (Low-voltage Detect Internal Power Fall Output Enable)

This bit is an output enable signal of the internal voltage fall detection.

LVD_OE	Internal Voltage Fall Detection Output Enable Setting
0	Enable
1	Disable

This bit is initialized by only power-on reset.

[bit2, bit1] Reserved

[bit0] LVD_F (Low-Voltage Detect internal power fall Flag): Internal low-voltage detection flag

This is an internal voltage fall detection flag.

LVD_F	Internal Voltage Fall Detection Flag	
	Read	Write
0	Not detected	Clear the flag
1	Detected	No effect on operation

If a power-on reset or a drop in the internal voltage is detected, the LVD_F bit is set to "1".

It will be initialized only at the external reset input.

28.5 Operation

This section explains the operations of the low-voltage detection (internal low-voltage detection).

28.5.1 Internal Low-voltage Detection

28.5.1 Internal Low-voltage Detection

The internal low-voltage detection is explained.

The internal low-voltage detection monitors the internal voltage and detects when the internal voltage falls below the detection voltage value and sets the detection flag. It generates settings initialization reset if it detects low-voltage and sets the flag.

If the internal voltage falls below the detection voltage, it takes the oscillation stabilization wait time after the internal low-voltage detection voltage recovered. For details, see "Chapter: Reset".

Oscillation stabilization wait time	2^{15} × Main clock cycle

28.6 Notes

This section explains the notes on the low-voltage detection (internal low-voltage detection).

- Operation of Internal Low-voltage Detection

  If the internal voltage falls and the internal low-voltage detection flag is set (LVD:LVD_F="1"), internal reset is generated by the low-voltage detection reset function. Thus, writing and reading of the internal low-voltage detection register (LVD) 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.

  Internal low-voltage detection circuit can be operate/stopped by the user setting.

- 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, it takes the oscillation stabilization wait time after the internal voltage recovered. For details, see "Chapter: Reset".

- Hysteresis of Detection/Release

  The release voltage becomes set value +0.05V so that detection/release may have the hysteresis of 0.05 V. For example, when LVD:1.0 V ± 0.1 V is set, the release voltage becomes 1.05 V ± 0.1 V.

29. Low-voltage Detection (External Low-voltage Detection)

This chapter explains the low-voltage detection (external low-voltage detection).

29.1 Overview

29.2 Features

29.3 Configuration

29.4 Registers

29.5 Operation

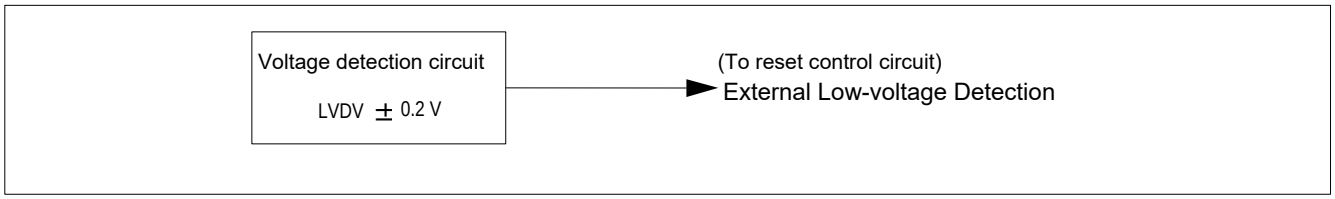
29.6 Notes

29.1 Overview

This section explains the overview of the low-voltage detection (external low-voltage detection).

The external low-voltage detection is a function that monitors the external voltage and detects when the voltage falls below the detection voltage value.

Figure 29-1. Block Diagram



Note: Rising LVDV: 2.3 V
Falling LVDV: 3.7 to 4.3 V (variable in units of 0.2 V)

29.2 Features

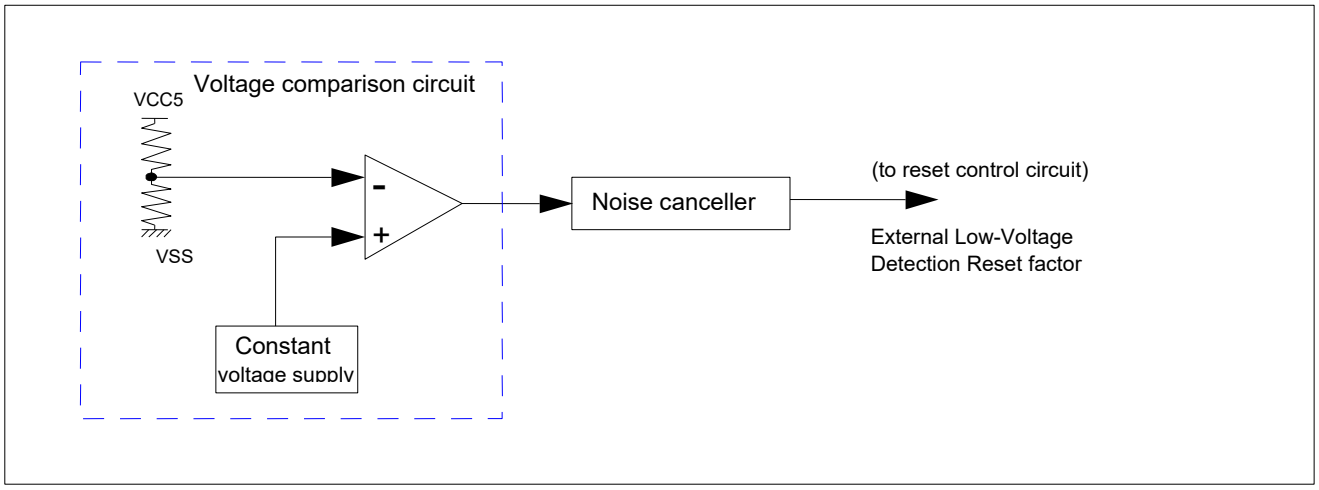
This section explains the features of the low-voltage detection (external low-voltage detection).

- External Low-voltage Detection Circuit
  - Method: Generates settings initialization reset if a voltage lower than LVDV ± 0.2 V is detected.
    (Rising LVDV: 2.3 V (fixed), falling LVDV: 3.7 to 4.3 V (variable in units of 0.2 V))
  - Number of units: One
  - Operation: Switches operation/stop by user setting.
    During writes to the internal RAM, the low-voltage reset occurs after the write has finished.
  - Voltage comparison circuit: Compares the external voltage to the detection voltage, and outputs "L" if low-voltage is detected.

29.3 Configuration

This section explains the configuration of the low-voltage detection (external low-voltage detection).

Figure 29-2. Configuration Diagram



29.4 Registers

This section explains the registers of the low-voltage detection (external low-voltage detection).

Table 29-1. 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

29.4.1 External Low-voltage Detection Rise Detection Register: LVD5R (Low-voltage Detect external 5 V Rise register)

The bit configuration of the external low-voltage detection rise detection register (LVD5R) is explained.

This register is used as the microcontroller unit external voltage rise detection flag.

LVD5R: Address 0584H (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 5 V Rise Flag): External voltage rise detection flag

This is an external voltage rise detection flag.

LVD5R_F	External Voltage 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".

It is cleared when external reset is input.

29.4.2 External Low-voltage Detection Fall Detection Register: LVD5F (Low-voltage Detect external 5 V Fall register)

The bit configuration of the external low-voltage detection fall detection register (LVD5F) is explained.

This register is used to clear the low-voltage detection reset flag and set low-voltage detection, etc.

LVD5F: Address 0585H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LVD5F_PD	Reserved	LVD5F_SEL[1:0]		LVD5F_OE	Reserved		LVD5F_F
Initial value	0	0	1	0	0	0	0	1
Attribute	R/W	R0,WX	R/W	R/W	R/W	R0,WX	R0,WX	R(RM1), W

[bit7] LVD5F_PD (Low-Voltage Detect external 5 V Fall Power Down): External voltage fall power down setting

This bit is used to set whether to detect a fall in external voltage or not.

LVD5F_PD	External Voltage Fall Power Down Setting
0	Disable (Performs detection)
1	Enable (Stops detection)

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 set it before 100 μs, some detection flag setting will be occur.

[bit6] Reserved

[bit5, bit4] LVD5F_SEL[1:0] (Low-Voltage Detect 5 V Fall Select): External fall detection voltage setting

These bits are the selection signal for a detection level of external voltage fall detection.

LVD5F_SEL[1:0]	External Voltage Fall Detection Voltage Setting
00	3.7 V ± 0.2 V
01	3.9 V ± 0.2 V
10	4.1 V ± 0.2 V
11	4.3 V ± 0.2 V

LVD5F_SEL[1:0] bits can be rewritten only when LVD5F_OE = "1".

[bit3] LVD5F_OE (Low-Voltage Detect external 5 V 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 Voltage Fall Detection Output Enable Setting
0	Enable
1	Stop

This bit is initialized by only power-on reset.

[bit2, bit1] Reserved

[bit0] LVD5F_F (Low-Voltage Detect external 5 V Fall Flag): External voltage fall detection flag

This 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 external reset is input.

29.5 Operation

This section explains the low-voltage detection (external low-voltage detection).

29.5.1 External Low-voltage Detection

29.5.1 External Low-voltage Detection

This section explains the external low-voltage detection.

The external low-voltage detection monitors the external voltage and generates a initialization reset if the external voltage drops below the configured value. The contents 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 is executed without delaying for the oscillation stabilization wait time, and then the program is restarted from the address specified by the reset vector.

29.6 Notes

This section explains notes of the low-voltage detection (external low-voltage detection).

When Using the Low-voltage Detection Reset Circuit

- Program Operation
  - ☐ The low-voltage detection reset circuit operates according to settings, except for the external low-voltage detection rise detection which 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 stop mode is cleared.
- Hysteresis for Detection/Release
  - ☐ The detection/release voltage becomes set value ±0.125 V because detection/release may have the hysteresis of 0.125 V. Set value shows the detecting voltage for the fall detection voltage. For example, when 4.1 V±0.2 V is set, the release voltage is 4.225 V±0.2 V. Set value shows the release voltage for the rise detection voltage. For example, when 2.5 V±0.2 V is set, the detection voltage is 2.375 V±0.2 V.

30. Wild Register

This chapter explains the wild register.

30.1 Overview
30.2 Features
30.3 Configuration
30.4 Registers
30.5 Operation
30.6 Usage Example

30.1 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.

30.2 Features

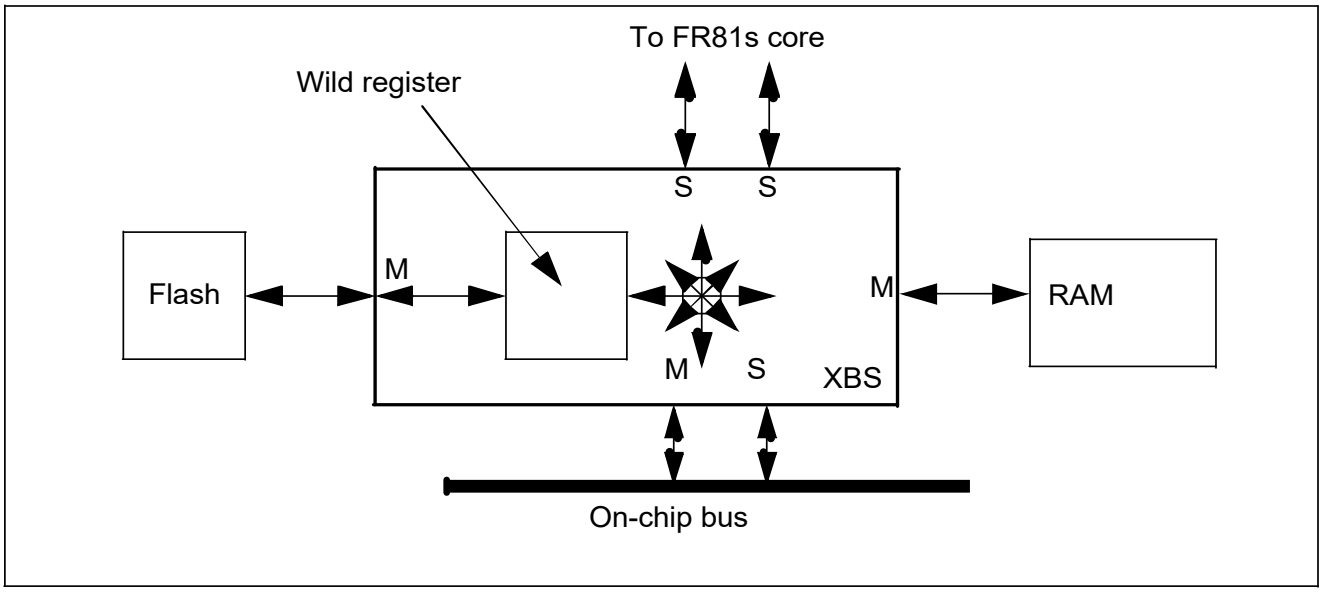
This section explains features of the wild register.

- 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

30.3 Configuration

This section explains the configuration of the wild register.

Figure 30-1. Configuration Diagram



Note:

When the access wait to the Flash memory is set to one cycle, this function cannot be used.

30.4 Registers

This section explains registers of the wild register.

Table 30-1. 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

Address	Registers				Register Function
	+0	+1	+2	+3	
0x08D0	WRAR10				Wild register address register 10
0x08D4	WRDR10				Wild register data register 10
0x08D8	WRAR11				Wild register address register 11
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

30.4.1 Wild Register Data Enable Register: WREN

The bit configuration of the wild register data enable register is shown.

These bits set whether the wild register function is enabled or disabled on each channel.

WREN: Address085A$_H$ (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 ch.n
1	Enables the wild register function of ch.n

30.4.2 Wild Register Address Register 00 to 15: WRAR00 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.

30.4.3 Wild Register Data Register 00 to 15: WRDR00 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 0884H 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.

30.5 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.

The setting addresses need to be set so that they do not overlap each other. When addresses overlap, the reading value is undefined.

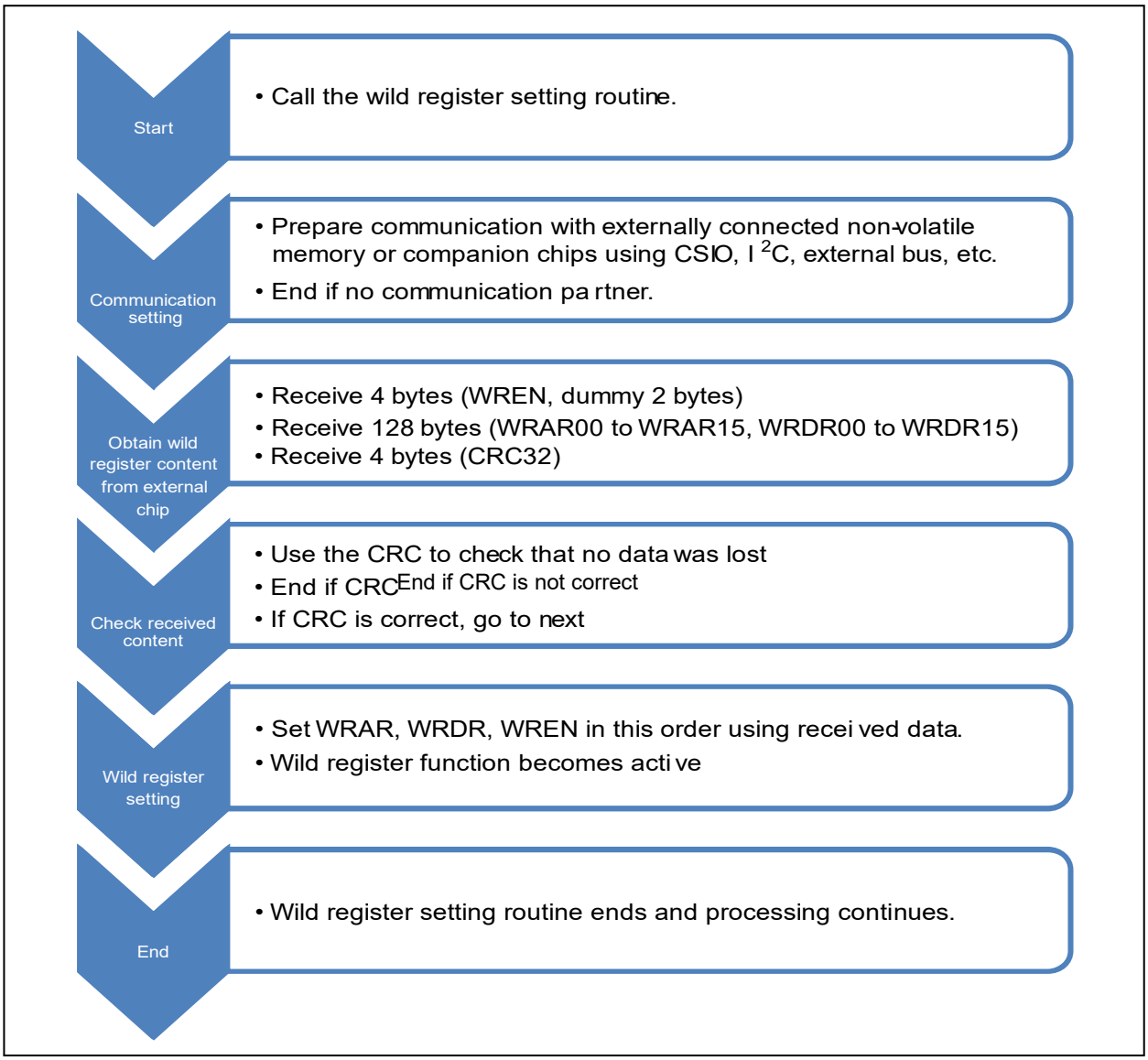
The data's byte line is the big endian.
The target area to replace is the Flash area only.

30.6 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 30-2. Usage Example



31. Clock Supervisor

This chapter explains overview, features, and register, etc of the clock supervisor.

31.1 Overview
31.2 Configuration
31.3 Register
31.4 Operation

31.1 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.

The clock supervisor for the sub clock is independent with the clock supervisor for the main. The clock supervisor can be enabled, and disabled separately.

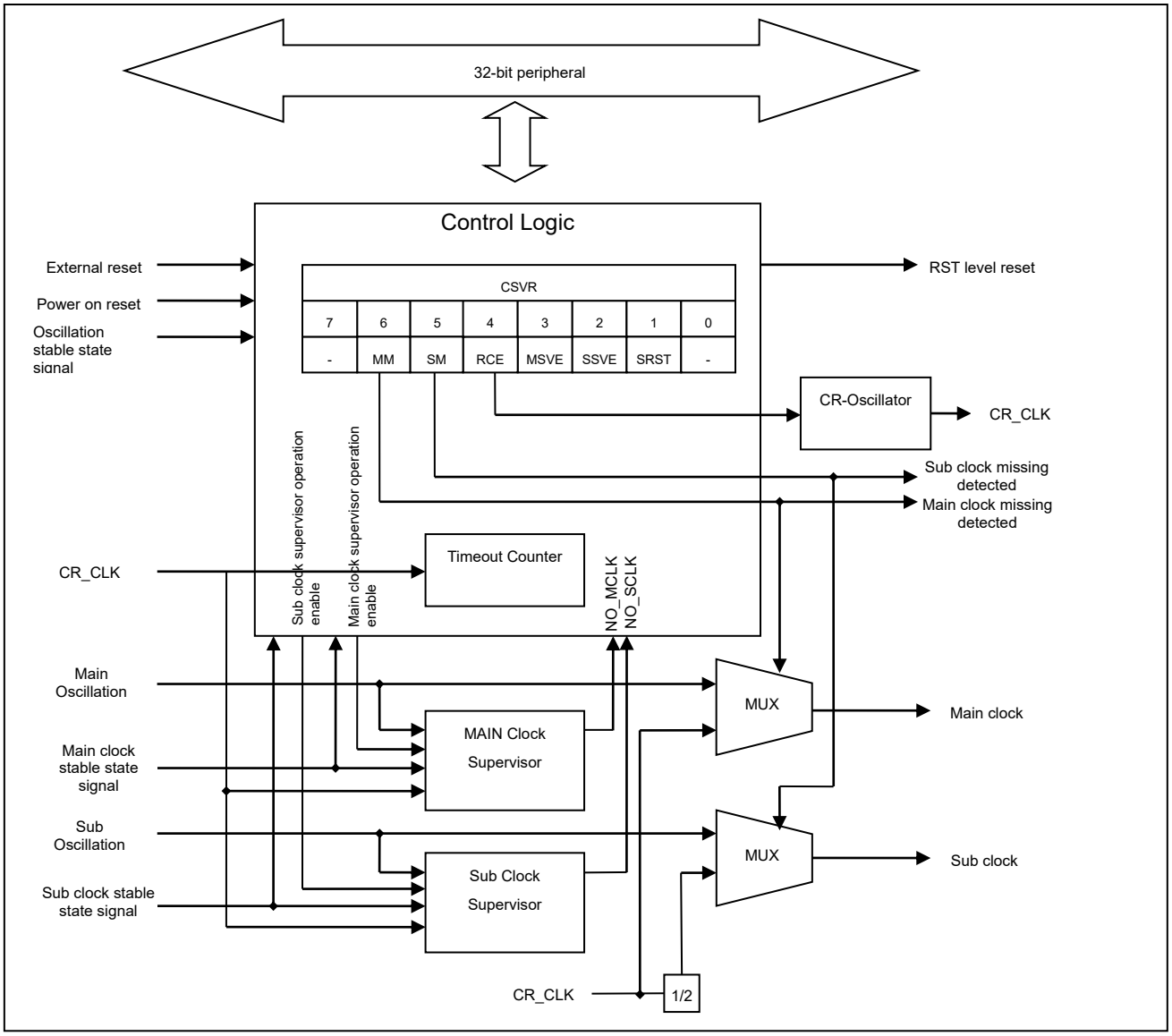
31.2 Configuration

This section shows the configuration of the clock supervisor.

The blocks that make up the clock supervisor are shown below.

- Clock supervisor
- Timeout counter
- Control logic
- CR oscillator

Figure 31-1. Block Diagram of Clock Supervisor (Detailed)



External reset: On assert of RSTX pin (including simultaneous assert with NMIX)

Note:

- The sub clock supervisor can be used for dual clock products.

31.3 Register

This section explains a register of the clock supervisor.

Table 31-1. Register Map

Address	Register				Register Function
	+0	+1	+2	+3	
0x056C	Reserved	CSVCR	Reserved	Reserved	Clock supervisor control register

Clock Supervisor Control Register: CSVCR

This register sets 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	SM	RCE	MSVE	SSVE	SRST	Reserved
Initial value	0	0	0	1	1/0	1	0	0
Attribute	R0/W0	R,W	R,W	R/W	R/W	R/W	R0/W0	R0/W0

Note: Initial value of bit3 depends on the part number.

[bit7] Reserved

"0" should be written to this bit.

[bit6] MM (Main clock Missing): Main clock stop

When this bit is "1", it indicates that any 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 have no effect on 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] SM (Sub clock Missing): Sub clock stop

When this bit is "1", it indicates that any problem is found in the sub oscillation clock.
When this bit is "0", there are no problems in the sub clock.
When the sub 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 have no effect on this bit.

SM	Read	Write
0	Sub oscillation clock stop undetected	When the sub clock is restored oscillating, this bit can be cleared
1	Sub oscillation clock stop detected	No effect

[bit4] RCE (CR-oscillator Enable)

The oscillation of the CR oscillator is permitted at the standby mode when this bit is set to "1". The thing to set this bit to "0" is prohibited while main clock supervisor or the sub-clock supervisor has been still permitted.

First of all, it is necessary to confirm the MM bit and the SM bit are "0" after prohibiting the supervisor. Afterwards, sets the RCE bit to "0".
Please do not set the RCE bit to "0" when either of the MM bit or the SM bit is "1". This bit is cleared to "1" by turning on the power supply or external reset. Other types of resets have no effect on 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)

When this bit is set to "1", the main clock supervisor is enabled.
This bit is only initialized to "1" when the power is turned on.
Other types of resets have no effect on this bit.

MSVE	Description
0	Main clock supervisor disabled (Initial Value of the products whose initial state of the clock supervisor is OFF)
1	Main clock supervisor enabled (Initial Value of the products whose initial state of the clock supervisor is ON)

Note:

Initial state of the clock supervisor depends on the part number. Therefore, initial value of this bit depends on the part number. Refer "1.3 Product Line-up" in "Chapter: Overview" for the detail of the part number.

[bit2] SSVE (Sub Clock Supervisor Enable)

When this bit is set to "1", the sub clock supervisor is enabled.
This bit is only initialized to "1" when the power is turned on.
Other types of resets have no effect on this bit.

SSVE	Description
0	Sub clock supervisor disabled
1	Sub clock supervisor enabled (Initial value)

[bit1] SRST (Sub Clock Mode Reset)

"0" should be written to this bit.

SRST	Description
0	No reset occurs when user changes main clock to sub clock while sub clock missing (Initial value).
1	Reset occurs when user changes main clock to sub clock while sub clock missing.

[bit0] Reserved

"0" should be written to this bit.

31.4 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 working with the main clock. When the period of 30 µs to 40 µs and the clock is not input, it is judged that it stops. Because the bit that shows the thing that the main clock stops remains in the register, the thing that the problem occurs with software can be judged.

After the clock replaces the CR oscillator, it is reset at once when sub clock stops while CPU is working with sub clock. When the period of 310 µs to 320 µs and the clock is not input, it is judged that it stops. Because the bit that shows the thing that a sub clock stops remains in the register, the thing that the problem occurs with software can be judged.

When sub clock stops while CPU is working with the main clock, reset is not generated at once. It operates with the CR clock when changing to the sub clock mode.

The main clock supervisor stops automatically when the main clock is stopped intending it. When sub clock is stopped intending it, the sub clock supervisor stops automatically.

The CR oscillator stops automatically when the standby mode changes when the CR oscillation at 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 when the main clock is replaced with the CR oscillator and works after detecting the main clock stop.

The following explains the operational mode of the clock supervisor.

31.4.1 Initial State

This section explains the initial state of the clock supervisor.

When initial setting, the oscillation of the CR oscillator, main clock supervisor function, and sub clock supervisor function have been 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) at the standby mode, it stops. When the standby mode is made clear, the oscillation is automatically restarted.

Main Clock Supervisor

Main clock supervisor is enabled after the main oscillation stabilization wait time has elapsed for the products whose initial state of the clock supervisor is ON.

Main clock supervisor is disabled initially for the products whose initial state of the clock supervisor is OFF. Main clock supervisor is enabled if it is enabled by the software.

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.

Notes:

For the products whose initial state of the clock supervisor is ON, 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 time 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.

Sub Clock Supervisor

After the timeout period measured by internal CR oscillator passes, it is enabled.

Behavior when a sub-clock stops when the sub-clock supervisor has been permitted changes according to whether MCU operates with the main clock or it operates with a sub-clock.

- For the Main Clock Mode

  When a sub clock stops while operating in the main clock mode, a sub clock replaces two dividing frequency of the CR oscillation clock. Afterwards, reset keeps being not generated and operating in the main clock mode though the SM bit of the CSVCR register is set to "1".

  Under such a condition, clock changes to the sub-clock mode that operates with the CR oscillation clock when changing to the sub-clock mode.

- For the Sub Clock Mode

  When a sub clock stops while operating in the sub-clock mode, two dividing frequency of the CR oscillation clock replaces a sub clock. Afterwards, the SM bit of the CSVCR register is set to "1", and reset of the RST level is generated.

31.4.2 Stopping CR Oscillator and the Clock Supervisor Function

This section explains stopping CR oscillator and the clock supervisor function.

CR Oscillator

The CR oscillator can be stopped only at the standby mode. Please change to the standby mode after setting oscillation permission bit (CSVCR:RCE) at the standby mode to "0".

The thing to stop the CR oscillator when there is a problem in the main clock or a sub-clock is prohibited. It can be confirmed whether or not the problem exists in the clock by the MM bit and the SM bit of the CSVCR register.

Note:

The operation clock stops, too, when the CR oscillation is stopped because the operation clock has already replaced 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".

Sub Clock Supervisor

The SSVE bit of the CSVCR register is set to "0".

31.4.3 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".

The thing to permit the main clock supervisor function with the CR oscillator has stopped is prohibited.

Notes:

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 time 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.

Sub Clock Supervisor

To permit the sub clock supervisor function again, the SSVE bit of the CSVCR register is set to "1". The thing to permit the sub clock supervisor function with the CR oscillator has stopped is prohibited.

31.4.4 Sub Clock Mode

This section explains the sub clock mode of the clock supervisor.

The main clock supervisor function stops automatically when the device changes to the sub clock mode with the main clock supervisor function has been permitted.

The main clock supervisor enable bit (CSVCR:MSVE) does not become "0".

After the oscillation stabilization wait time of the main clock passes, the main clock supervisor function is permitted again when the device changes from the sub-clock mode to the main clock mode.

Notes:

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 time 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.

31.4.5 Stop Mode

This section explains stop mode of the clock supervisor.

CR Oscillator

The oscillation stops when oscillation permission bit (CSVCR:RCE) at the stop mode is set to "0" by changing to the stop mode.

After the stop mode is made clear, it is permitted automatically again.

Main Clock Supervisor

When the main clock supervisor function is enabled, it automatically stops when stop mode is entered.

The main clock supervisor enable bit (CSVCR:MSVE) does not change to "0".

After stop mode is released, the supervisor is automatically re-enabled after waiting for the main oscillation stabilization wait time.

Notes:

- 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 time 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.
- When the main clock supervisor function is disabled, if stop mode is entered, the supervisor remains disabled even after recovering from stop mode.

Sub Clock Supervisor

When the sub clock supervisor function is enabled, it automatically stops when stop mode is entered.

The sub clock supervisor enable bit (CSVCR:SSVE) does not change to "0".

After stop mode is released, the supervisor is automatically re-enabled after waiting for the main oscillation stabilization wait time.

Note:

When the sub clock supervisor function is disabled, if stop mode is entered, the supervisor remains disabled even after recovering from stop mode.

31.4.6 Watch Mode

This section explains watch mode.

Main Clock Supervisor

The main clock supervisor function is not influenced from the transition to the watch mode.

When the main clock stops, the change dividing and reset are issued to the CR oscillation clock when the main clock is connected with the permission of the main clock supervisor function and RTC.

The watch mode is made clear, and RTC is initialized.

The RTC clock stops only because it does not detect it even if the main clock stops when the main clock is connected with the prohibition of the main clock supervisor function and RTC.

Sub Clock Supervisor

The sub clock supervisor function is not influenced from the transition to the watch mode.

When a sub clock is connected with the permission of the sub clock supervisor function and RTC, reset is not issued though it cuts in the CR oscillation clock when a sub clock stops.

The RTC clock stops only because it does not detect it even if the sub clock stops when the sub clock is connected with the prohibition of the sub clock supervisor function and RTC.

31.4.7 Checking the Reset Factor Using the Clock Supervisor

Checking the reset factor using the clock supervisor is explained.

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 "7.4.1 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 either 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. Also, read the RSTRR register (see "7.4.1 Reset Source Register: RSTRR (Reset Result Register)" in "Chapter: Reset") and confirm the reset factor.

The reset factor can be checked as follows.

MM	SM	Reset factor
1	0	Main clock supervisor reset
0	1	Sub clock supervisor reset
1	1	Main clock supervisor reset or Sub clock supervisor reset

Note:

Because the MM bit and SM bit are 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 "7.4.1 Reset Source Register: RSTRR (Reset Result Register)" in "Chapter: Reset").

31.4.8 Return from CR Clock

Return from the CR clock is explained.

Main Clock Supervisor

The main clock stops when the MPU detects that the MM bit has been set after recovering from a reset, and it can be judged that these has been a change in 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 when the main clock works 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+SM
btsth #0b0110,@r1 ;; Check: Is one of them 1?
bne clear_CSV_loop
```

Note:

Set "0" to PMUCTLR.SHDE to return to the main clock.

Sub Clock Supervisor

A sub clock stops when the MPU detects that the SM bit has been set and it can be judged that there has been a change in the CR oscillation clock. At this time, it is possible to return to the sub clock by writing "0" in the SM bit if it can be confirmed that the sub clock is restored.

When a sub clock is not restored, writing "0" in the SM bit does not have any influence. The SM bit keeps maintaining "1".

The SM bit is cleared when a sub clock works when "0" is written in the SM bit, and the clock returns to a sub clock via a synchronous stage.

It can perform polling on the SM bit until a sub-clock is restored. (The same method as main clock supervisor can be used.)

Note:

Set "0" to PMUCTLR.SHDE to return to the sub clock.

32. Sound Generator

This chapter explains the sound generator.

32.1 Overview

32.2 Features

32.3 Configuration

32.4 Registers

32.5 Operation

32.1 Overview

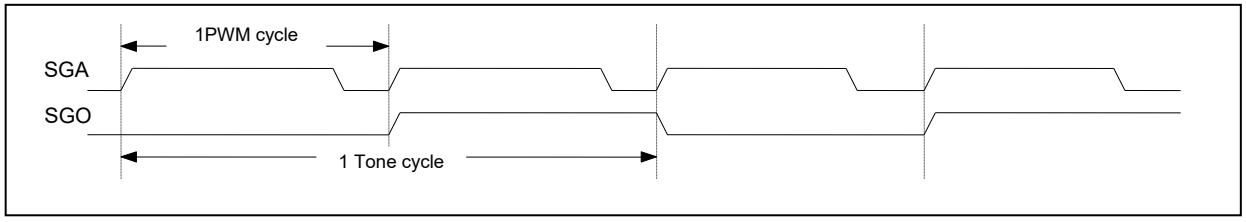
This section explains the overview of the sound generator.

This series includes a sound generator with 5 channels. The sound generator generates and outputs tone pulse signals (or mixed signals of tone pulse signals and PWM pulse signals) and PWM pulse signals according to the setting from the CPU. The frequency of tone pulse signals to output, sound volume (amplitude of PWM pulse), and sound length can be specified.

The sound generator consists of registers and counters below:

- DMA transfer update enable register
- Sound control register
- Amplitude data register
- Frequency data register
- Cycle register
- Tone outputs number register
- Increment decrement data register
- PWM cycles number data register
- DMA transfer indirect register
- PWM pulse generator
- Frequency counter
- Decrement counter
- Tone pulse counter

Figure 32-1. External Pin Output



32.2 Features

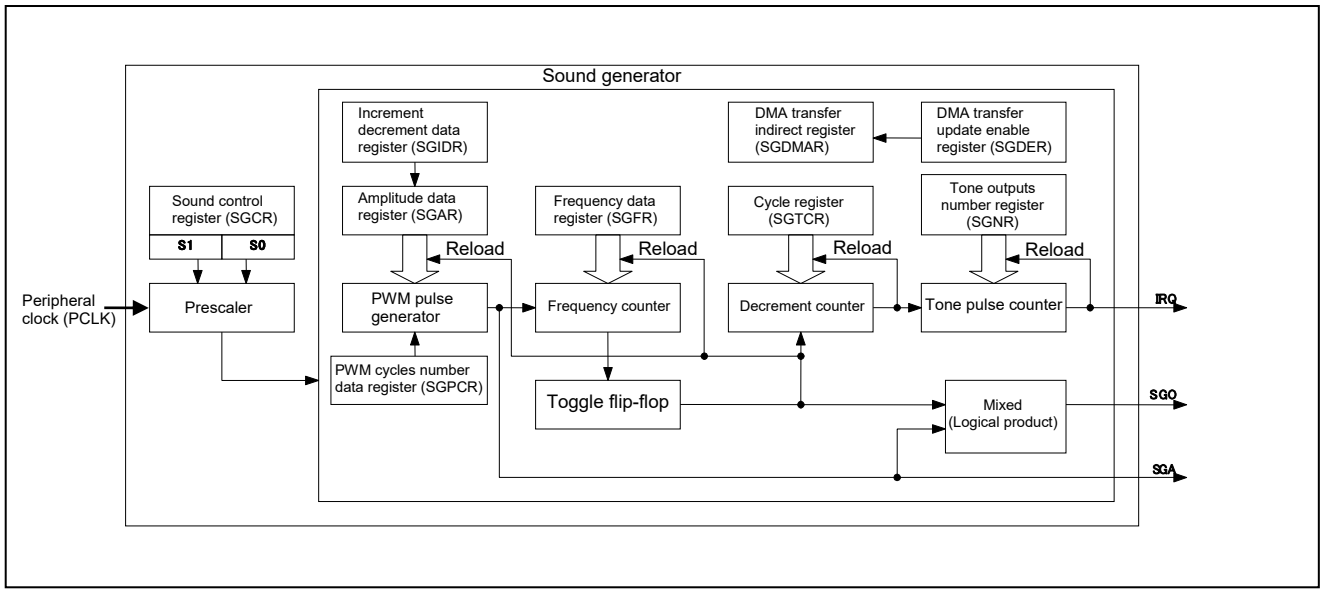
This section explains features of the sound generator.

No	Item	Function
1	Operating clock	Peripheral clock (PCLK) 16 MHz to 40 MHz
2	Clock input	For clock input for the sound generator, the peripheral clock (PCLK) is used with divided frequency. Peripheral clock (PCLK) 1/2 × peripheral clock (PCLK) 1/4 × peripheral clock (PCLK) 1/8 × peripheral clock (PCLK)
3	Waveform	Square wave for sound (sound output from SGO pin)
4	Sound volume	Any volume can be set (amplitude output from SGA pin)
5	Frequency	Sound signal frequency can be set to any value (frequency setting and PWM cycles number setting)
6	Sound length	Any value can be set.
7	Interrupt	When the specified sound length is finished to output, an interrupt request can be generated. (Tone pulse counter overflow) In the DMA mode (SGCR:DMA="1"), when "1" is written to the start bit (SGCR:ST), an interrupt can be generated.

32.3 Configuration

This section explains the configuration of the sound generator.

Figure 32-2. Block Diagram



32.4 Registers

This section explains registers of the sound generator.

Table of Base Addresses (Base_addr) and External Pins

Table 32-1. Table of Base Addresses and External Pins

Channel	Base_addr	External Pin	
		SGO	SGA
0	0x1040	SGO0	SGA0
1	0x1060	SGO1	SGA1
2	0x1080	SGO2	SGA2
3	0x10A0	SGO3	SGA3
4	0x10C0	SGO4_0/ SGO4_1	SGA4_0/ SGA4_1

Registers Map

Table 32-2. Registers Map

Address	Registers				Register function
	+0	+1	+2	+3	
0x1040	Reserved	SGDER0	SGCR0		DMA transfer update register 0 Sound control register 0
0x1044	SGAR0		SGFR0	SGNR0	Amplitude data register 0 Frequency data register 0 Tone outputs number register 0
0x1048	SGTCR0	SGIDR0	SGPCR0		Cycle register 0 Increment decrement amount data register 0 PWM cycles number register 0
0x104C	SGDMAR0				DMA transfer indirect register 0
0x1060	Reserved	SGDER1	SGCR1		DMA transfer update register 1 Sound control register 1
0x1064	SGAR1		SGFR1	SGNR1	Amplitude data register 1 Frequency data register 1 Tone outputs number register 1

Address	Registers				Register function
	+0	+1	+2	+3	
0x1068	SGTCR1	SGIDR1	SGPCR1		Cycle register 1 Increment decrement amount data register 1 PWM cycles number register 1
0x106C	SGDMAR1				DMA transfer indirect register 1
0x1080	Reserved	SGDER2	SGCR2		DMA transfer update register 2 Sound control register 2
0x1084	SGAR2		SGFR2	SGNR2	Amplitude data register 2 Frequency data register 2 Tone outputs number register 2
0x1088	SGTCR2	SGIDR2	SGPCR2		Cycle register 2 Increment decrement amount data register 2 PWM cycles number register 2
0x108C	SGDMAR2				DMA transfer indirect register 2
0x10A0	Reserved	SGDER3	SGCR3		DMA transfer update register 3 Sound control register 3
0x10A4	SGAR3		SGFR3	SGNR3	Amplitude data register 3 Frequency data register 3 Tone outputs number register 3
0x10A8	SGTCR3	SGIDR3	SGPCR3		Cycle register 3 Increment decrement amount data register 3 PWM cycles number register 3
0x10AC	SGDMAR3				DMA transfer indirect register 3
0x10C0	Reserved	SGDER4	SGCR4		DMA transfer update register 4 Sound control register 4
0x10C4	SGAR4		SGFR4	SGNR4	Amplitude data register 4 Frequency data register 4 Tone outputs number register 4
0x10C8	SGTCR4	SGIDR4	SGPCR4		Cycle register 4 Increment decrement amount data register 4 PWM cycles number register 4
0x10CC	SGDMAR4				DMA transfer indirect register 4

32.4.1 DMA Transfer Update Enable Register: SGDER (SG DMA Enable Register)

This section explains the bit configuration for the DMA transfer update enable register (SGDER).

The DMA transfer update enable register (SGDER) is to set registers (SGAR, SGFR, SGNR, SGTCR, SGIDR, SGPCR) in bytes to be updated on DMA transfer. The sound generator determines the register to be updated on DMA transfer according to the configuration of this register. In addition, the transfer count of DMA transfers, number of transfer bytes, and DMA transfer indirect register (SGDMAR) enabled bytes location are determined with the configuraion of this register.

SGDER: Address Base_addr + 01 H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ARE1	ARE0	FRE	NRE	TCRE	IDRE	PCRE1	PCRE0
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

Bit name		Function
bit7	ARE1: Amplitude data (upper byte) update enable bit	On DMA transfer, update of the amplitude data (upper byte) of the amplitude data register (SGAR) is enabled via the DMA transfer indirect register. When this bit is set to "0": Amplitude data (upper byte) is not updated. When this bit is set to "1": Amplitude data (upper byte) is updated.
bit6	ARE0: Amplitude data (lower byte) update enable bit	On DMA transfer, update of the amplitude data (lower byte) of the amplitude data register (SGAR) is enabled via the DMA transfer indirect register. When this bit is set to "0": Amplitude data (lower byte) is not updated. When this bit is set to "1": Amplitude data (lower byte) is updated.
bit5	FRE: Frequency data update enable bit	On DMA transfer, update of the frequency data of the frequency data register (SGFR) is enabled via the DMA transfer indirect register. When this bit is set to "0": Frequency data is not updated. When this bit is set to "1": Frequency data is updated.
bit4	NRE: Tone output number update enable bit	On DMA transfer, update of the tone outputs number of the tone outputs number register (SGNR) is enabled via the DMA transfer indirect register. When this bit is set to "0": Tone output number is not updated. When this bit is set to "1": Tone output number is updated.
bit3	TCRE: Cycle update enable bit	On DMA transfer, update of the cycle of the cycle register (SGTCR) is enabled via the DMA transfer indirect register. When this bit is set to "0": Cycle is not updated. When this bit is set to "1": Cycle is updated.
bit2	IDRE: Increment decrement amount data update enable bit	On DMA transfer, update of the increment decrement amount data of the increment decrement data register (SGIDR) is enabled via the DMA transfer indirect register. When this bit is set to "0": Increment decrement amount data is not updated. When this bit is set to "1": Increment decrement amount data is updated.

Bit name		Function
bit1	PCRE1: PWM cycles number data (upper byte) update enable bit	On DMA transfer, update of the PWM cycles number data (upper byte) of the PWM cycles number data register (SGPCR) is enabled via the DMA transfer indirect register. When this bit is set to "0": PWM cycles number data (upper byte) is not updated. When this bit is set to "1": PWM cycles number data (upper byte) is updated.
bit0	PCRE0: PWM cycles number data (lower byte) update enable bit	On DMA transfer, update of the PWM cycles number data (lower byte) of the PWM cycles number data register (SGPCR) is enabled via the DMA transfer indirect register. When this bit is set to "0": PWM cycles number data (lower byte) is not updated. When this bit is set to "1": PWM cycles number data (lower byte) is updated.

32.4.2 Sound Control Register: SGCR (SG Control Register)

This section explains the bit configuration for the sound control register (SGCR).

The sound control register (SGCR) controls interrupts and the operating state of the sound generator.

SGCR: Address Base_addr + 02_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	SRST	DMA	GID	GEN	Reserved	BUSY	Reserved
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W0	R0,W	R/W	R/W	R/W	R0,W0	R/W	R0,W0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	S1	S0	TONE	Reserved		INTE	INT	ST
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R(RM1),W	R,W

Bit name		Function
bit15	Reserved	This bit is reserved. This bit must always be written to "0".
bit14	SRST: Software reset bit	This bit is a software reset bit. When "1" is set to this bit, hardware issues a software reset. On read, "0" is always read, therefore, setting of this bit to "0" is invalid.
bit13	DMA: DMA transfer start interrupt set enable bit	This bit is a DMA transfer start interrupt set enable bit. When this bit is set to "0": When "1" is written to a start bit (SGCR:ST), an interrupt bit (SGCR:INT) is not set. When this bit is set to "1": When "1" is written to a start bit (SGCR:ST), an interrupt bit (SGCR:INT) is set. **Note:** Do not change the setting of this bit during the sound generator is being operated (SGCR:ST = 1).
bit12	GID: Increment decrement setting bit	With the cycle register (SGTCR), increment decrement amount data register (SGIDR), and automatic increment decrement enable bit (SGCR:GEN), this bit is designed for automatic increment decrement of sound. The value stored in the amplitude data register is incremented or decremented by the value of the increment decrement amount data register on each counting by the decrement counter of the tone pulses number from the toggle flip-flop specified by the cycle register (increment decrement setting reflection timing) according to the setting of the increment decrement setting bit. When this bit is set to "0": The value stored in the amplitude data register is decremented. When this bit is set to "1": The value stored in the amplitude data register is incremented. **Note:** This bit is enabled only when the automatic increment decrement enable bit is enabled (SGCR:GEN = 1). If the setting is changed during operation, the register value of the increment decrement setting reflection timing is enabled.

Bit name		Function
bit11	GEN: Automatic increment decrement enable bit	With the cycle register (SGTCR), increment decrement amount data register (SGIDR), and increment decrement setting bit (SGCR:GID), this bit is designed for automatic increment decrement of sound. When this bit is set to "0": Automatic increment decrement of sound is disabled. When this bit is set to "1": Automatic increment decrement of sound is enabled. The value stored in the amplitude data register is incremented or decremented by the value of the increment decrement amount data register on each counting by the decrement counter of the tone pulses number from the toggle flip-flop specified by the cycle register (increment decrement setting reflection timing) according to the setting of the increment decrement setting bit. **Note:** When the automatic increment decrement enable setting is changed to disable setting, the amplitude data register keeps the current value. When the automatic increment decrement disable setting is changed to enable setting, the current value of the amplitude data register is incremented or decremented from the increment decrement setting reflection timing.
bit10	Reserved	This bit is reserved. On read, "0" is read, and on write, write "0".
bit9	BUSY: Busy bit	This bit indicates whether the sound generator is being operated or not. Condition for "1": When the SGCR:ST bit is set to "1", this bit is set to "1". Condition for "0": When the SGCR:ST bit is set to "0" and operation is completed on 1 tone cycle completion, this bit is cleared to "0". **Note:** This bit becomes "0" with a software reset (SGCR:SRST = 1 write).
bit8	Reserved	This bit is reserved. On read, "0" is read, and on write, write "0".
bit7, bit6	S1, S0: Operating clock selection bits	These bits select a clock input signal for the sound generator. {S1,S0} Clock input 00 1/1 Input clock 01 1/2 Input clock 10 1/4 Input clock 11 1/8 Input clock
bit5	TONE: Tone output bit	This bit sets a SGO signal. When this bit is set to "0": The SGO signal is a mixed signal (logical multiply) of a tone pulse and a PWM pulse. When this bit is set to "1": The SGO signal is a simple square waveform (tone pulse) signal from the toggle flip-flop.
bit4	Reserved	This bit is reserved. Written value is ignored.
bit3	Reserved	This bit is reserved. Written value is ignored.
bit2	INTE: Interrupt enable bit	Interrupt signals of the sound generator are enabled. When this bit is set to "0": The interrupt by the SGCR:INT bit is disabled. When this bit is set to "1": The interrupt by the SGCR:INT bit is enabled. With SGCR:INT =1, an interrupt signal is output.

Bit name		Function
bit1	INT: Interrupt bit	When the tone pulse counter counts the tone pulses number specified to the tone outputs number register and cycle register, this bit is set to "1". In addition, in the DMA mode (SGCR:DMA = 1) when the amplitude data register (SGAR), frequency data register (SGFR), tone outputs number register (SGNR), cycle register (SGTCR), and increment decrement amount data register (SGIDR) are not written but the start bit (SGCR:ST) is written to "1" also, this bit becomes "1", and this can be used as a DMA transfer start request. When this bit is written to "0", it is cleared to "0", however, even if it is written to "1", it is disabled and the previous value is held. Condition for "1": Tone pulse count number ≥ (Cycle register value + 1) × (Tone outputs number register value + 1) Condition for "0": When this bit is written to "0", it is cleared to "0". **Note:** On a read access by a read-modify-write instruction, "1" is always read. This bit becomes "0" with a software reset (SGCR:SRST = 1 write).
bit0	ST: Start bit	This bit is for operation start for the sound generator. When this bit is set to "1": The operation of the sound generator is started. When this bit is set to "0": The operation of the sound generator is stopped after completion of the current tone cycle. **Note:** While this bit is "1", the sound generator is being operated. When this bit is set to "0", it is cleared to "0", and the operation of the sound generator is stopped after the current tone cycle is completed. Whether the sound generator is stopped completely or not is indicated by the SGCR:BUSY bit. This bit becomes "0" with a software reset (SGCR:SRST = 1 write).

32.4.3 Amplitude Data Register: SGAR (SG Amplitude Register)

This section explains the bit configuration for the amplitude data register (SGAR).

The amplitude data register (SGAR) stores reload values for the PWM pulse generator. The register value inidicates the amplitude of sound. The register value is reloaded to the PWM pulse generator for each completion of a tone cycle.

SGAR: Address Base_addr + 04_H (Access: Byte, Half-word, 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

[bit15 to bit0] D[15:0] (Data): Amplitude data bits

These bits store reload values for the PWM pulse generator. Software sets the reload value for the PWM pulse generator. When the increment decrement enable bit is enabled (SGCR:GEN = 1), hardware increments or decrements the value stored in the amplitude data register by the value of the increment decrement amount data register (SGIDR) on each counting by the decrement counter of the tone pulses number from the toggle flip-flop specified by the cycle register (SGTCR) according to the setting of the increment decrement setting bit (SGCR:GID). When the increment decrement setting bit is set with decrement (SGCR:GID = 0) and the amplitude data register value is "0x0000", no more decrement is performed. When the increment decrement setting bit is set with increment decrement (SGCR:GID = 1) and the amplitude data register value is "0xFFFF", no more increment is performed. However, the operation of the sound generator is continued until the start bit (SGCR:ST) is cleared.

32.4.4 Frequency Data Register: SGFR (SG Frequency Register)

This section explains the bit configuration for the frequency data register (SGFR).

The frequency data register (SGFR) stores the reload value for the frequency counter. The stored value indicates the frequency of sound (or the tone signal from the toggle flip-flop). The register value is reloaded to the counter for each transition of the toggle signal.

SGFR: Address Base_addr + 06_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D[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] D[7:0] (Data): Frequency data bits

These bits store reload values for the frequency counter. Software sets the reload value for the frequency counter. Hardware uses the bits for the frequency of sound (or the tone signal from the toggle flip-flop). The register value is reloaded to the counter for each transition of the toggle signal.

Note:

Note that when the register value is changed during operation, 50% of the duty cycle of the sound (or the tone signal from the toggle flip-flop) might be changed depending on the change timing.

32.4.5 Tone Outputs Number Register: SGNR (SG tone Number Register)

This section explains the bit configuration for the tone outputs number register (SGNR).

The tone outputs number register (SGNR) stores the reload value for the tone pulse counter. The tone pulse counter accumulates the tone pulses number (or times of amplitude operations of the sound), and when the number reaches the reload value, the interrupt bit (SGCR:INT) is set. This aims to reduce the frequency of interrupts.

SGNR: Address Base_addr + 07_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D[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] D[7:0] (Data): Tone outputs number bits

These bits store reload values for the tone pulse counter. Software sets the reload value for the tone pulse counter. Hardware accumulates the tone pulses number (or times of amplitude operations of the sound) with the tone pulse counter, and when the number reaches the reload value, the interrupt bit (SGCR:INT) is set.

When "0x00" is set to the tone count register, the tone pulse counter sets the SGCR:INT bit for each the carry-out signal from the decrement counter. Accumulated tone pulses number is obtained by the formula below. In addition, the cycle regiser stores the reload value of the decrement counter.

(Cycle register value + 1) × (Tone outputs number register value + 1)

When both the tone output register and the cycle register are set to "0x00", the interrupt bit (SGCR:INT) is set for each tone cycle.

32.4.6 Cycle Register: SGTCR (SG Tone Cycle Register)

This section explains the bit configuration for the cycle register (SGTCR).

The cycle register (SGTCR) stores the reload value for the decrement counter. This is designed to increment or decrement automatically the value stored in the amplitude data register (SGAR).

When the automatic increment decrement enable bit (SGCR:GEN) is set to enable, the value stored in the amplitude data register (SGAR) is inremented or decremented by the value of the increment decrement amount data register (SGIDR) on each counting by the decrement counter of the tone pulses number from the toggle flip-flop specified by the cycle register according to the setting of the increment decrement setting bit (SGCR:GID).

This behavior allows less CPU intervention to the sound automatic increment decrement performance.

Note that the pulses number specified by this register is "register value + 1". When it is set to "0x00", the automatic increment decrement behavior is performed for each tone cycle.

SGTCR: Address Base_addr + 08_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D[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] D[7:0] (Data): Cycle bits

These bits store reload values for the decrement counter. Software sets reload values for the decrement counter. Hardware increments or decrements the value stored in the amplitude data register (SGAR) by the value of the increment decrement amount data register (SGIDR) on each counting by the decrement counter of the tone pulses number from the toggle flip-flop specified by the cycle register according to the setting of the increment decrement setting bit (SGCR:GID).

Note:

Note that the pulses number specified by this register is "register value + 1".

32.4.7 Increment Decrement Data Register: SGIDR (SG Increment Decrement Register)

This section explains the bit configuration for the increment decrement data register (SGIDR).

The increment decrement data register (SGIDR) stores an increment and decrement amount for the amplitude data register (SGAR). It increments or decrements the value of the amplitude data register (SGAR) on each counting by the decrement counter of the tone pulses number from the toggle flip-flop specified by the cycle register according to the setting of the increment decrement setting bit (SGCR:GID).

SGIDR: Address Base_addr + 09_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D[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] D[7:0] (Data): Increment Decrement Data bits

Theses bits store increment and decrement amount for the amplitude data register (SGAR). Software sets increment and decrement amount for the amplitude data register (SGAR). Hardware increments or decrements the value stored in the amplitude data register (SGAR) on each counting by the decrement counter of the tone pulses number from the toggle flip-flop specified by the cycle register according to the setting of the increment decrement setting bit (SGCR:GID). When the increment decrement setting bit is set with decrement (SGCR:GID = "0") and the amplitude data register (SGAR) value is "0x0000", no more decrement is performed. When the increment decrement setting is increment (SGCR:GID = 1) and the amplitude data register (SGAR) value is "0xFFFF", no more increment is performed. In addition, when the increment decrement amount data register value is "0x00", increment or decrement is not performed.

32.4.8 PWM Cycles Number Data Register: SGPCR (SG PWM Cycle Register)

This section explains the bit configuration for the PWM cycles number data register (SGPCR).

The PWM cycles number data register (SGPCR) stores the cycles number of 1PWM cycle.

SGPCR: Address Base_addr + $0A_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	D[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
	D[7:0]							
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 bit0] D[15:0] (Data): PWM cycles number data bits

These bits store the cycles number of 1PWM cycle. The reference clock cycle is the input clock (prescaler).

Note:

Note that the clock cycles number specified by this register is "register value + 1".

32.4.9 DMA Transfer Indirect Register: SGDMAR (SG DMA Register)

This section explains the bit configuration for the DMA transfer indirect register (SGDMAR).

The DMA transfer indirect register (SGDMAR) is used for DMA transfer for the amplitude data register (SGAR), frequency data register (SGFR), tone outputs number register (SGNR), cycle register (SGTCR), increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR). The read value is always "0". To read the value written to this register, read the relevant register of the PWM cycles number data register (SGPCR) from the amplitude data register (SGAR).

This register must be accessed with 1/2/4 bytes according to the setting of the DMA transfer update enable register (SGDER). The access location is fixed to bit31 to bit24 for 1-byte access and bit31 to bit16 for 2-byte access.

When the operation enable (SGCR:ST = 1) is set while the DMA transfer start interrupt set is enabled (SGCR:DMA = 1), if sound is output only the number of times specified by the cycle register (SGTCR) and tone outputs number register (SGNR), the sound generator sets the interrupt bit (SGCR:INT) and asserts the interrupt signal (PIRQ).

When receiving an interrupt from the sound generator, the DMA controller performs the DMA transfer for this register. In addition, for the DMA controller, fix this register to the transfer destination address.

The sound generator writes the data written to this register to the amplitude data register (SGAR), frequency data register (SGFR), tone outputs number register, cycle register (SGTCR), increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR) according to the setting of the DMA transfer update enable register (SGDER).

SGDMAR: Address Base_addr + $0C_H$ (Access: Byte, Half-word, Word)

	bit31	bit30	-	-	-	bit2	bit1	bit0
	D[31:0]							
Initial value	0	0	-	-	-	0	0	0
Attribute	R0/W	R0/W	-	-	-	R0/W	R0/W	R0/W

[bit31 to bit0] D[31:0] (Data): DMA transfer data bits

These are registers to be used for DMA transfer for the amplitude data register (SGAR), frequency data register (SGFR), tone outputs number register (SGNR), cycle register (SGTCR), increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR).
The DMA transfer size and transfer number of times must be set according to the setting of the DMA transfer update enable register (SGDER). Bit31 to bit24 are fixed for 1-byte access and bit31 to bit16 are fixed for 2-byte access.
In addition, the number of transfers is either once or twice, and the data transferred to this register in a single transfer must be all or a part of the "amplitude data, frequency data, and tone outputs number" or "cycle, increment decrement amount data, and PWM cycles number data". It is inhibited to transfer the data, which is in the address space from the amplitude data register (SGAR) to the PWM cycles number data register (SGPCR) and whose address exceeds 4-byte boundary, in a single transfer. (Example: The frequency data and increment decrement amount data cannot be transferred in a single transfer. They will be transferred in two transfers.)

Note:

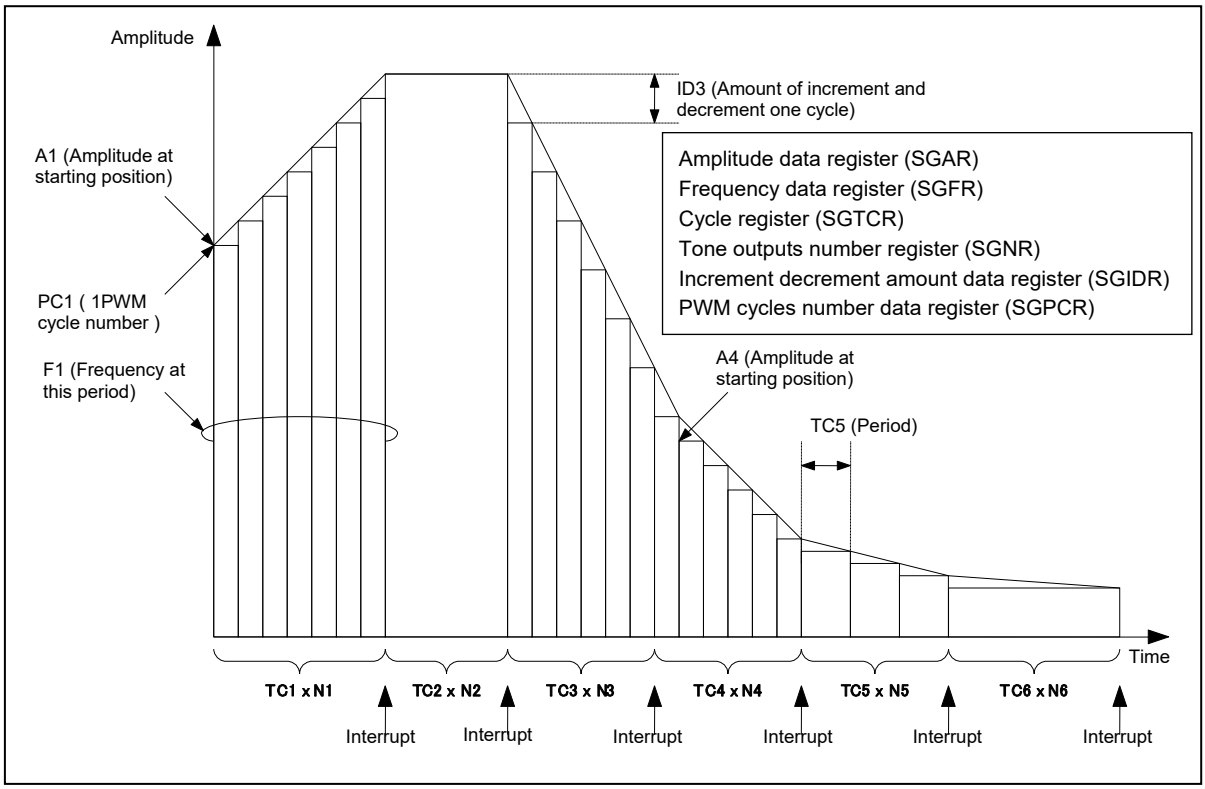
If the number of DMA transfers is more than the transfers count set in the DMA transfer update enable register (SGDER), the value will be updated.

32.5 Operation

This section explains the operation of the sound generator.

This section explains the operations of the sound generator. The sound generator operation concept diagram is shown below.

Figure 32-3. Operation Concept Diagram for Sound Generator



An amplitude of the output start position is set to the amplitude data register (SGAR), a frequency of the tone pulse signal is set to the frequency data register (SGFR), an output number of the tone pulse signal for one cycle is set to the cycle register (SGTCR), a frequency that generates an interrupt is set to the tone outputs number register (SGNR), the increment or decrement amount for one cycle is set to the increment decrement amount data register (SGIDR), 1PWM cycles number is set to the PWM cycles number data register (SGPCR), and other sound generator control information is set to the sound control register (SGCR).

The sound generator outputs the tone pulse signal and amplitude data with these settings.

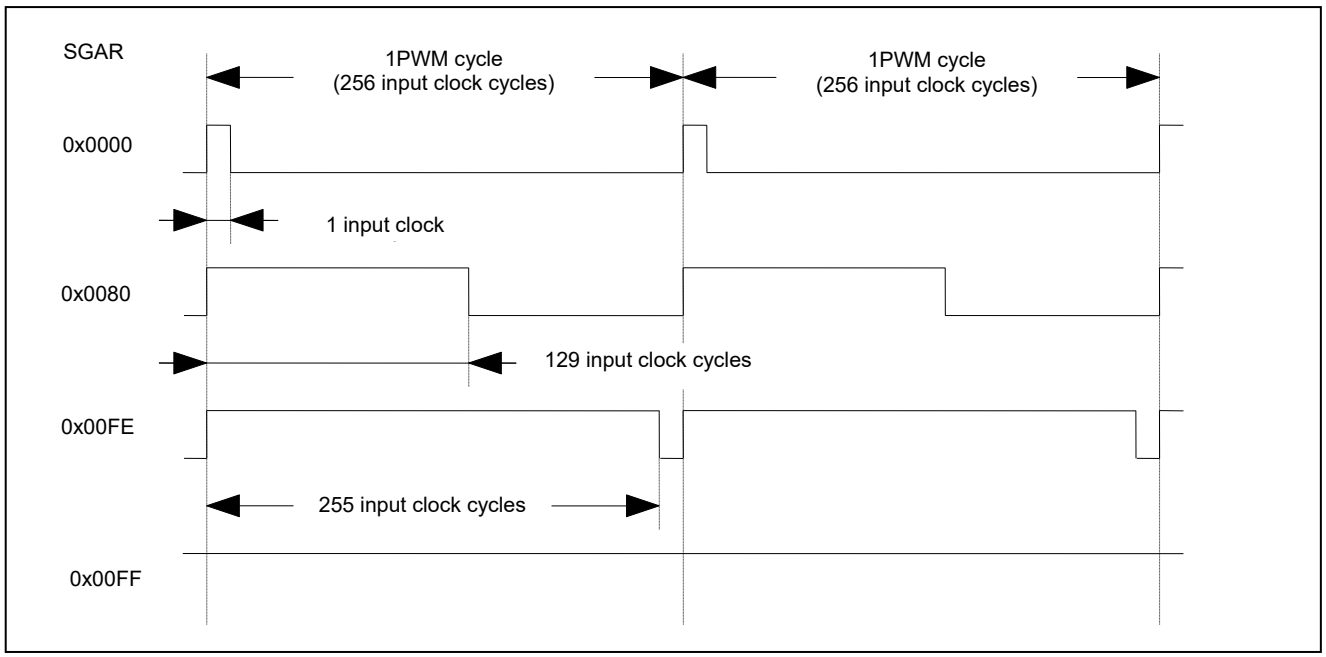
In the operation concept diagram above, the sound generator outputs six types of signals. Various register values such as the amplitude data register are set for output start and each interrupt occurrence.

32.5.1 Relation of Amplitude Data Register (SGAR) and PWM Pulse

The relation of the amplitude data register (SGAR) and PWM pulse is explained.

The relation of the amplitude data register (SGAR) and PWM pulse is explained. The relation diagram for the amplitude data register (SGAR) and PWM pulse is shown below.

Figure 32-4. Relation Diagram for Amplitude Data Register (SGAR) and PWM Pulse



The amplitude data is output by the SGA pin as PWM (Pulse Width Modulation) pulses.

One PWM cycle is 256 input clock cycles (SGPCR = "0x00FF"), which can be set with the PWM cycles number data register (SGPCR).

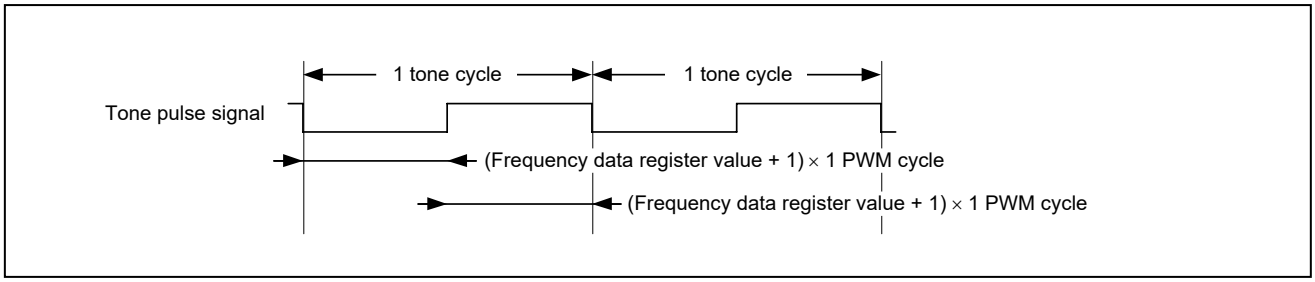
The value "register value + 1" of the amplitude data register (SGAR) is the input clock cycles number for SGA pin = "H" during one PWM cycle. In addition, when the amplitude data register (SGAR) ≥ PWM cycles number data register (SGPCR), the SGA pin is always "H".

32.5.2 Relation of Frequency Data Register (SGFR) and Tone Pulse Signals

The relation of the frequency data register (SGFR) and tone pulse signals is explained.

The relation of the frequency data register (SGFR) and tone pulse signals is explained. The relation diagram for the frequency data register (SGFR) and tone pulse signals is shown below.

Figure 32-5. Relation Diagram for Frequency Data Register (SGFR) and Tone Pulse Signals



Tone pulse signals repeat "L" and "H" with the cycle of "(Frequency data register value + 1) × 1 PWM cycle". They are generated by the toggle flip-flop.

When the tone output bit (TONE) of the sound control register (SGCR) is "0", tone pulse signals are mixed with PWN pulses (logical multiply), and output from the SGO pin. In addition, when the tone output bit (TONE) of the sound control register (SGCR) is "1", tone pulse signals are output from the SGO pin without mixing.

Note:

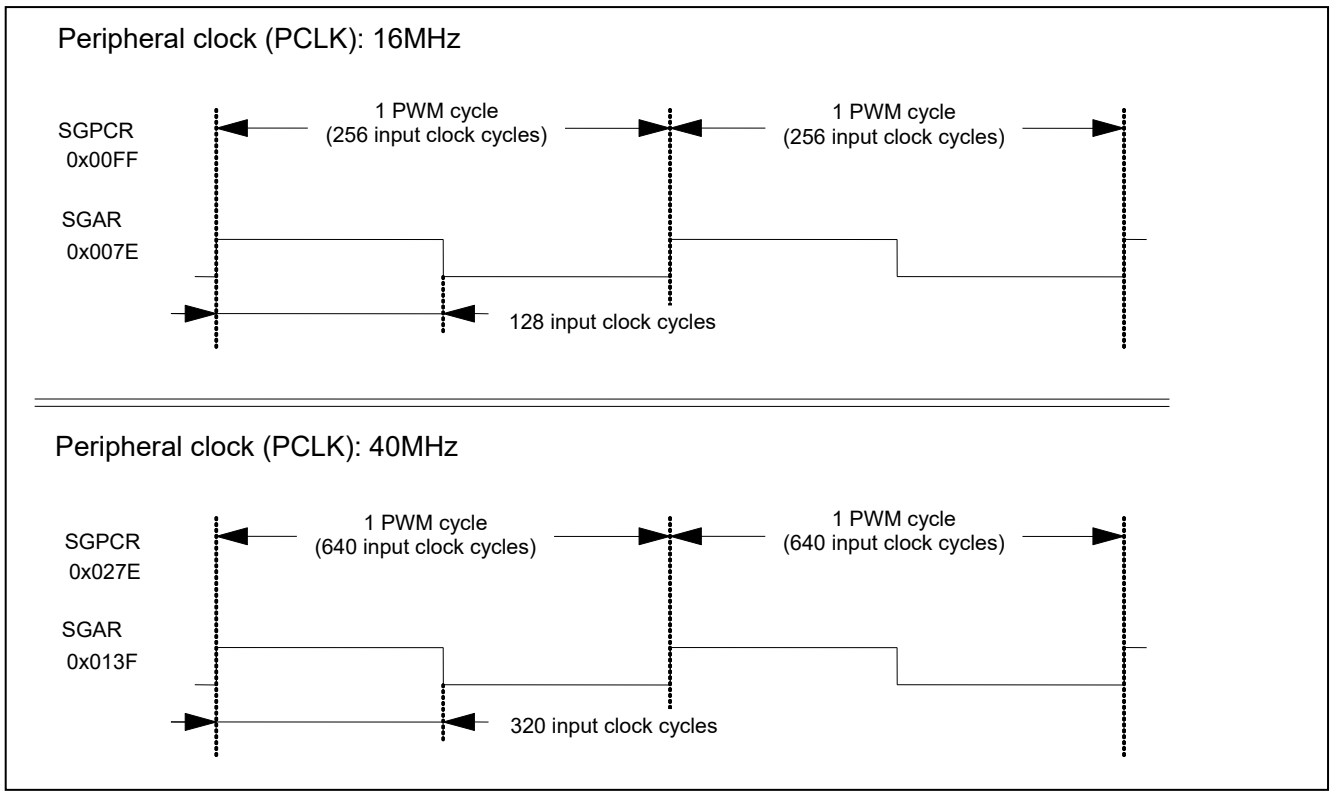
Note that when the register value is changed during operation of the sound generator, 50% of the duty cycle might be changed depending on the change timing.

32.5.3 Relation of PWM Cycles Number Data Register (SGPCR) and PWM Cycle

The relation of the PWM cycles number data register (SGPCR) and the PWM cycle is explained.

The relation of the PWM cycles number data register (SGPCR) and the PWM cycle is explained. The relation diagram for the PWM cycles number data register (SGPCR) and the PWM cycle is shown below.

Figure 32-6. Relation Diagram for PWM Cycles Number Data Register (SGPCR) and PWM cycle



The PWM cycle can be set with the PWM cycles number data register (SGPCR).

"Register value + 1" of the PWM cycles number data register (SGPCR) is the input clock cycles number for one PWM cycle. The input clock is the division of the frequency of the peripheral clock (PCLK).

The PWM cycle is a reference cycle for the tone pulse signals (or mixed signals of tone pulse signals with the PWM pulse signals) and PWM pulse signals.

You can make the sound output for a peripheral clock (PCLK) of 16 MHz equal to the sound output for a peripheral clock of 40 MHz by changing the PWM cycles number data register (SGPCR) value and the amplitude data register (SGAR) value. This can be realized by making the ratio of the register values above 1:2.5 since the ratio of 16 MHz and 40 MHz for the peripheral clock (PCLK) is 1:2.5.

32.5.4 Relation of DMA Transfer Update Enable Register (SGDER) and DMA Transfers Count/DMA Transfer Size/Transfer Byte Location

The relation of the DMA transfer update enable register (SGDER) and the DMA transfers count/DMA transfer size/Transfer byte location is explained.

32.5.4.1 DMA Transfers Count

The DMA transfers count is explained below.

Whether the DMA transfer is n byte × once or n byte × twice depends on the setting of the DMA transfer update enable register (SGDER).

When all of SGDER:ARE1, SGER:ARE0, SGDER:FRE, and SGDER:NRE are "0", or, all of SGDER:TCRE, SGDER:IDRE, SGDER:PCRE1, and SGDER:PCRE0 are "0", the DMA transfers count is once. For other than that, the DMA transfers count is twice.

32.5.4.2 DMA Transfer Size

The DMA transfer size is explained below.

Whether the DMA transfer size is 1 byte, 2 bytes, or 4 bytes depends on the setting of the DMA transfer update enable register (SGDER). In addition, the DMA transfer size is larger one of the setting value of SGDER:ARE1, SGER:ARE0, SGDER:FRE, and SGDER:NRE and the setting value of SGDER:TCRE, SGDER:IDRE, SGDER:PCRE1, and SGDER:PCRE0. Furthermore, transfer for 3 bytes or more is assumed as 4 bytes.

32.5.4.3 Transfer Byte Location for DMA Transfer Indirect Register

The transfer byte location for the DMA transfer indirect register is explained below.

The transfer byte location for the DMA transfer indirect register (SGDMAR) is decided by the DMA transfer update enable register (SGDER) setting and DMA transfer size.

When the DMA transfer size for one transfer to the DMA transfer indirect register (SGDMAR) is less than 4 bytes, the transfer byte location is left-aligned.(Also when the DMA transfer size for one transfer is 3 bytes, the location is left-aligned.)

The relation of the DMA transfer update enable register (SGDER) and the transfer byte location of the amplitude data (SGAR[15:0]), frequency data (SGFR[7:0]), and tone outputs number (SGNR[7:0]) for the DMA transfer indirect register (SGDMAR) is shown below. When the transfer size #1 for the DMA transfer indirect register (SGDMAR) calculated from SGDER:ARE1, SGDER:ARE0, SGDER:FRE, and SGDER:NRE of the DMA transfer update enable register (SGDER) is 2 bytes or less, the transfer byte location is left-aligned.

Table 32-3. DMA Transfer Update Enable Register (SGDER) and Transfer Byte Location for SGDMAR #1

	SGDER setting				Transfer size #1 [1]	Transfer byte location for SGDMAR			
	ARE1	ARE0	FRE	NRE		SGAR [15:8]	SGAR [7:0]	SGFR [7:0]	SGNR [7:0]
1	0	0	0	0	0	-	-	-	-
2	1	0	0	0	1	SGDMAR [31:24]	-	-	-
3	0	1	0	0	1	-	SGDMAR [31:24]	-	-
4	1	1	0	0	2	SGDMAR [31:24]	SGDMAR [23:16]	-	-
5	0	0	1	0	1	-	-	SGDMAR [31:24]	-
6	1	0	1	0	2	SGDMAR [31:24]	-	SGDMAR [23:16]	-
7	0	1	1	0	2	-	SGDMAR [31:24]	SGDMAR [23:16]	-
8	1	1	1	0	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	-
9	0	0	0	1	1	-	-	-	SGDMAR [31:24]
10	1	0	0	1	2	SGDMAR [31:24]	-	-	SGDMAR [23:16]
11	0	1	0	1	2	-	SGDMAR [31:24]	-	SGDMAR [23:16]

	SGDER setting				Transfer size #1 [1]	Transfer byte location for SGDMAR			
	ARE1	ARE0	FRE	NRE		SGAR [15:8]	SGAR [7:0]	SGFR [7:0]	SGNR [7:0]
12	1	1	0	1	4	SGDMAR [31:24]	SGDMAR [23:16]	-	SGDMAR [15:8]
13	0	0	1	1	2	-	-	SGDMAR [31:24]	SGDMAR [23:16]
14	1	0	1	1	4	SGDMAR [31:24]	-	SGDMAR [23:16]	SGDMAR [15:8]
15	0	1	1	1	4	-	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]
16	1	1	1	1	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	SGDMAR [7:0]

[1]: Transfer size calculated from {SGDER:ARE1, SGER:ARE0, SGDER:FRE, SGDER:NRE}

X, - : Do not care

The relation of the DMA transfer update enable register (SGDER) and the transfer byte location of the cycle (SGTCR[7:0]), increment decrement amount data (SGFR[7:0]), and PWM cycles number data (SGPCR[15:0]) for the DMA transfer indirect register (SGDMAR) is shown below. When the transfer size #2 for the DMA transfer indirect register (SGDMAR) calculated from SGDER:TCRE, SGDER:IDRE, SGDER:PCRE1, and SGDER:PCRE0 of the DMA transfer update enable register (SGDER) is 2 bytes or less, the transfer byte location is left-aligned.

Table 32-4. DMA Transfer Update Enable Register (SGDER) and Transfer Byte Location for SGDMAR #2

No.	SGDER setting				Transfer size #2 [1]	Transfer byte location for SGDMAR			
	TCRE	IDRE	PCRE1	PCRE0		SGTCR [7:8]	SGIDR [7:0]	SGPCR [15:0]	SGPCR [7:0]
1	0	0	0	0	0	-	-	-	-
2	1	0	0	0	1	SGDMAR [31:24]	-	-	-
3	0	1	0	0	1	-	SGDMAR [31:24]	-	-
4	1	1	0	0	2	SGDMAR [31:24]	SGDMAR [23:16]	-	-
5	0	0	1	0	1	-	-	SGDMAR [31:24]	-

No.	SGDER setting				Transfer size #2 [1]	Transfer byte location for SGDMAR			
	TCRE	IDRE	PCRE1	PCRE0		SGTCR [7:8]	SGIDR [7:0]	SGPCR [15:0]	SGPCR [7:0]
6	1	0	1	0	2	SGDMAR [31:24]	-	SGDMAR [23:16]	-
7	0	1	1	0	2	-	SGDMAR [31:24]	SGDMAR [23:16]	-
8	1	1	1	0	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	-
9	0	0	0	1	1	-	-	-	SGDMAR [31:24]
10	1	0	0	1	2	SGDMAR [31:24]	-	-	SGDMAR [23:16]
11	0	1	0	1	2	-	SGDMAR [31:24]	-	SGDMAR [23:16]
12	1	1	0	1	4	SGDMAR [31:24]	SGDMAR [23:16]	-	SGDMAR [15:8]
13	0	0	1	1	2	-	-	SGDMAR [31:24]	SGDMAR [23:16]
14	1	0	1	1	4	SGDMAR [31:24]	-	SGDMAR [23:16]	SGDMAR [15:8]
15	0	1	1	1	4	-	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]
16	1	1	1	1	4	SGDMAR [31:24]	SGDMAR [23:16]	SGDMAR [15:8]	SGDMAR [7:0]

[1]: Transfer size calculated from {SGDER:TCRE, SGDER:IDRE, SGDER:PCRE1, SGDER:PCRE0}

X: Do not care

32.5.4.4 DMA Transfer Image

The DMA transfer image is shown below.

DMA transfer update enable register (SGDER) setting:

{SGDER:ARE1, SGER:ARE0, SGDER:FRE, SGDER:NRE} = 1001
{SGDER:TCRE, SGDER:IDRE, SGDER:PCRE1, SGDER:PCRE0} = 0100

An example with the setting above is described below:

DMA transfers count: Twice

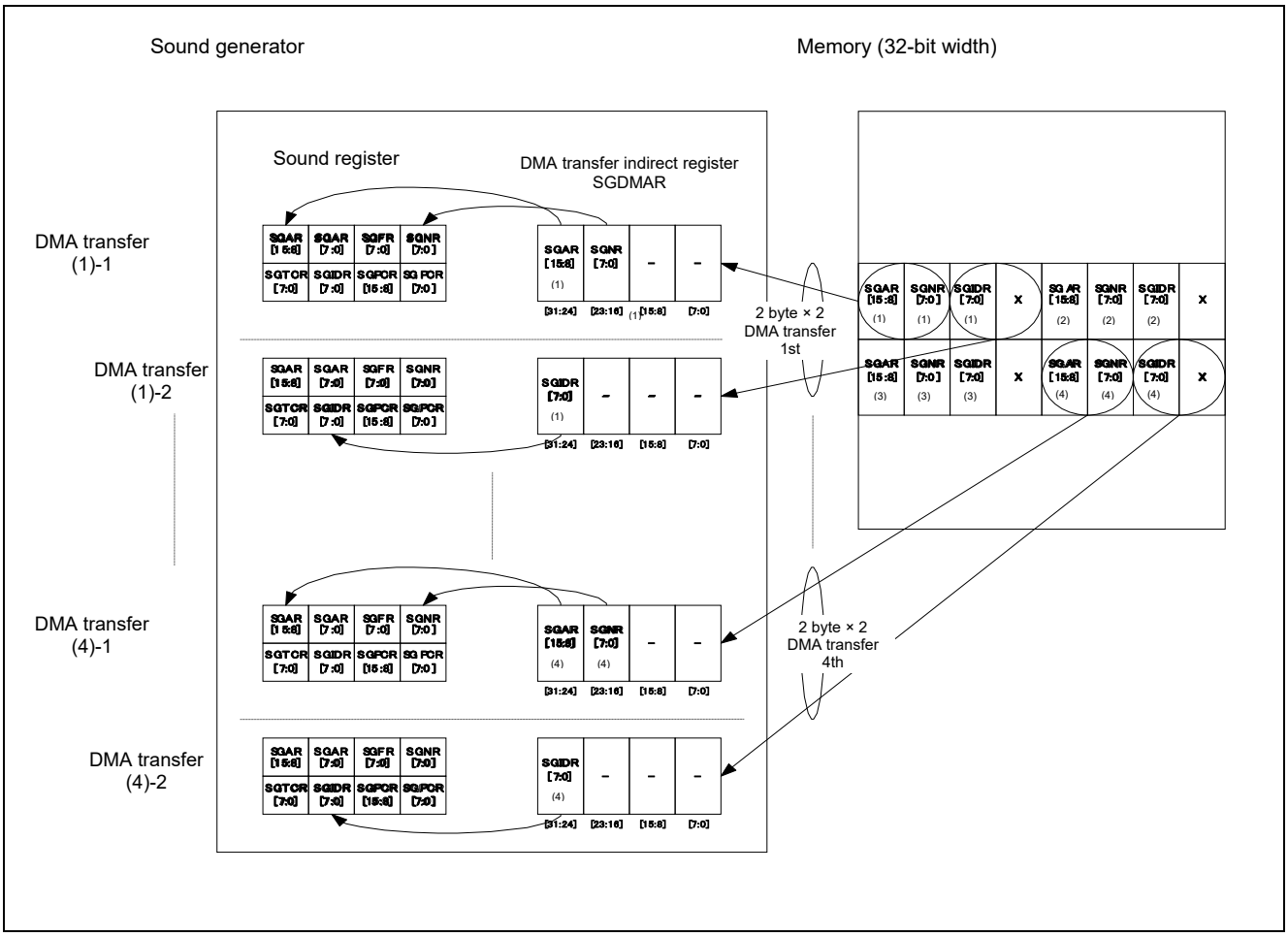
DMA transfer size: 2 bytes

Transfer byte location for DMA transfer indirect register:

First
- SGDMAR[31:24] ← Amplitude data (upper byte) / SGAR[15:8]
- SGDMAR[23:16] ← Tone outputs number / SGNR[7:0]
- SGDMAR[15:0] ← Don't care

Second
- SGDMAR[31:24] ← Increment decrement amount data / SGIDR[7:0]
- SGDMAR[23:0] ← Don't care

Figure 32-7. DMA Transfer Image

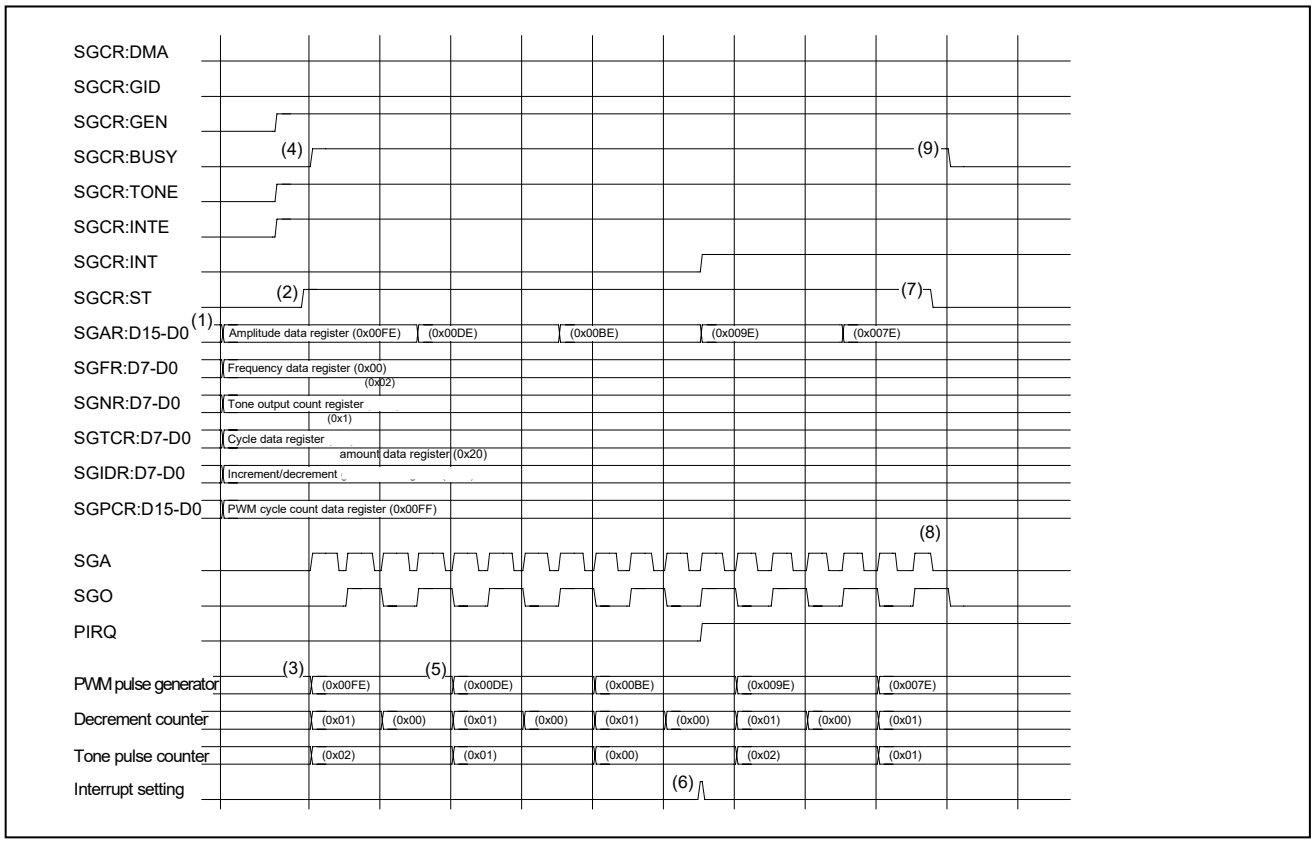


32.5.5 Operation of Sound Generator

The operation of the sound generator is shown below.

This section explains the operations of the sound generator. The sound generator operation is shown below.

Figure 32-8. Operation of Sound Generator



1. With software, write the reload value to the amplitude data register (SGAR), frequency data register (SGFR), tone outputs number register (SGNR), and cycle register (SGTCR), the amplitude increment or decrement amount to the increment decrement amount data register (SGIDR), and the 1PWM cycles number to the PWM cycles number data register (SGPCR). In addition, set other sound generator control information to the sound control register (SGCR). Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
2. Set the start bit (SGCR:ST) to "1".
3. By setting of the start bit (SGCR:ST) to "1", the amplitude data register (SGAR) value is loaded to the PWM generator, the frequency data register (SGFR) value is loaded to the frequency counter, the tone outputs number register (SGNR) value is loaded to the tone pulse counter, and the cycle register (SGTCR) value is loaded to the decrement counter.
4. The operation flag (SGCR:BUSY) is turned to "1".
5. By counting of the tone pulses number until the reload value by the decrement counter, the amplitude data register (SGAR) value increments or decrements according to the automatic increment decrement enable bit (SGCR:GEN) and the increment decrement setting bit (SGCR:GID).
6. When the tone pulses number specified with the tone outputs number register (SGNR) and the cycle register (SGTCR) is counted by the tone pulse counter (the timing of the tone pulse counter = "0x00", the decrement counter = "0x00", and SGO = "L" → "H"), an interrupt set request is generated, the interrupt bit (SGCR:INT) is set, and an interrupt (PIRQ) is generated.
7. Set the start bit (SGCR:ST) to "0". The operation is continued until the busy bit (SGCR:BUSY) is turned to "0".
8. The operation of the sound generator is stopped after completion of the current tone cycle.

9. The operation flag (SGCR:BUSY) is turned to "0".

DMA Transfer Start Interrupt Set Enable Bit

The assert condition for the first interrupt after the start instruction by the CPU differs according to the setting of the DMA transfer start interrupt set enable bit.

- Normal mode: When the sound generator outputs the tone pulses number set in the cycle register (SGTCR)
- DMA mode: Immediately after the start instruction (DMA transfer request)

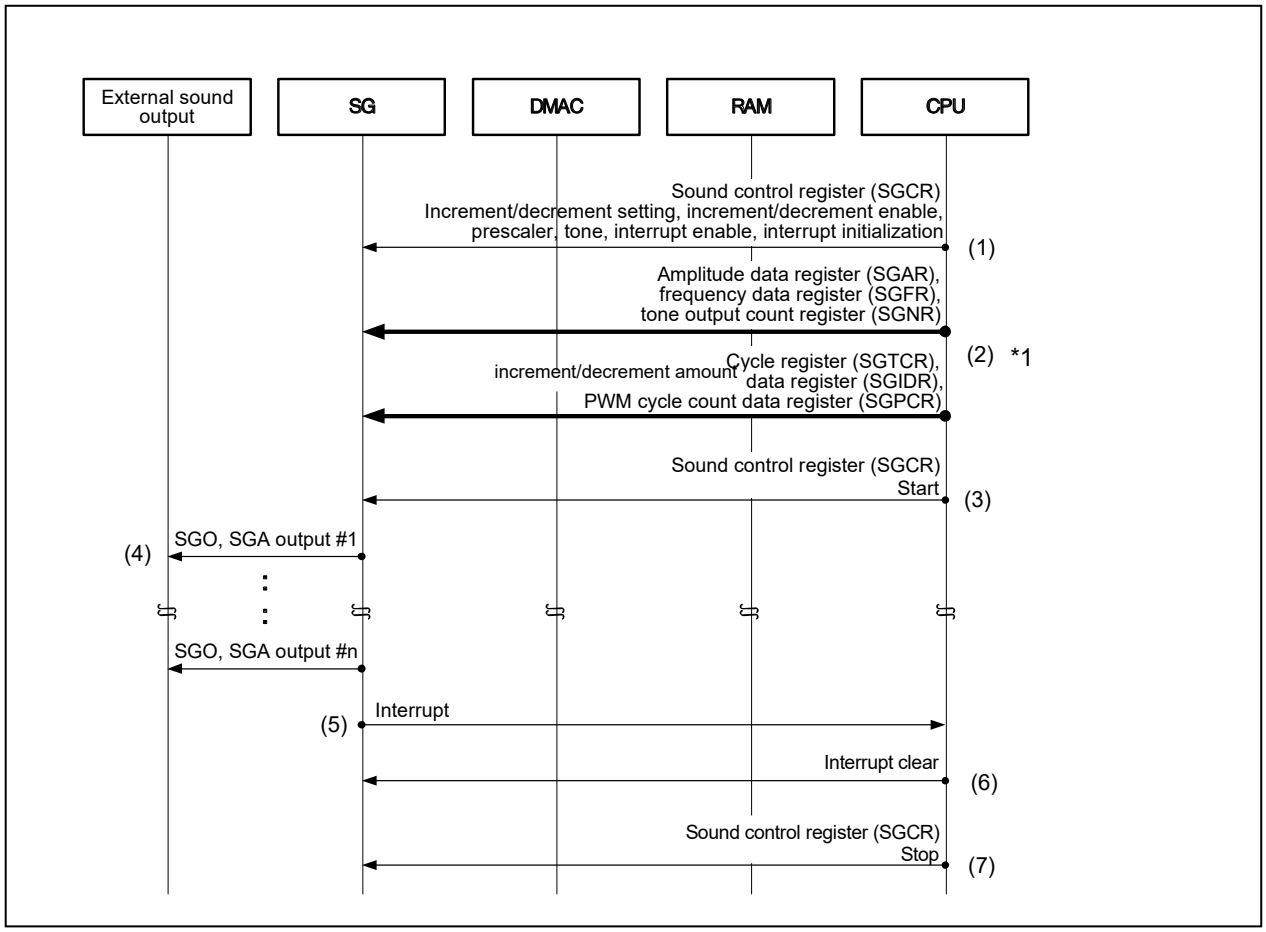
DMA Transfer

The setting of SGAR, SGFR, SGNR, SGTCR, SGIDR, and SGPCR is performed through the DMA transfer indirect register (SGDMAR).

Sound Generator Single Operation by CPU

The flow of the sound generator single operation by the CPU is shown below.

Figure 32-9. Sound Generator Single Operation by CPU



1. With software, set the sound generator control information into the sound control register (SGCR) of the sound generator. Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
2. With software, set "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)."(*1: In addition, setting to all registers is not mandatory.)
3. Set the start bit (SGCR:ST) to "1".
4. The sound generator SGO and SGA output is started.
5. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
6. The CPU clears the interrupt.
7. The CPU sends the stop instruction (SGCR:ST = 0) to the sound generator.

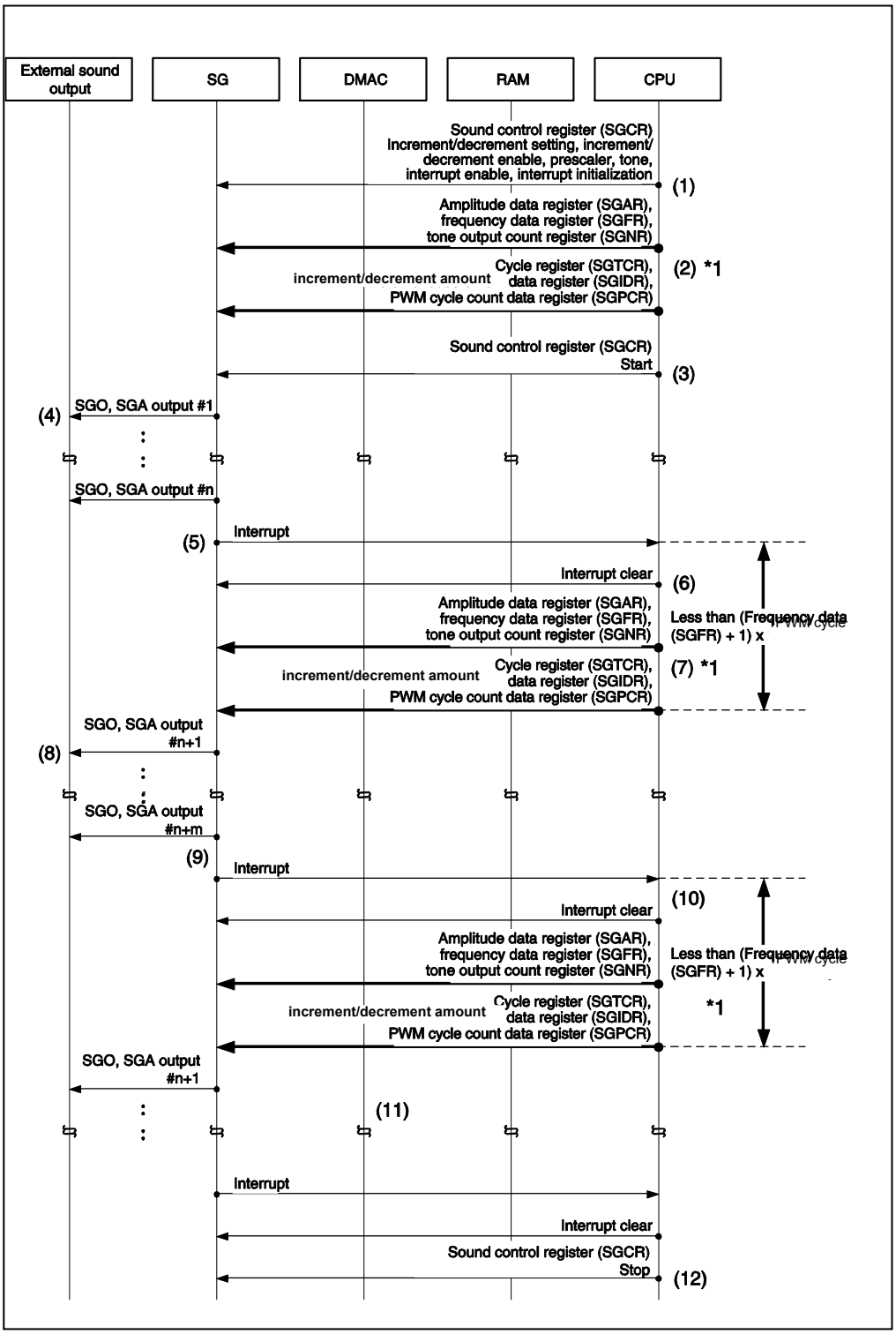
32.5.6 Sound Generator Continuous Operation by CPU

The sound generator continuous operation by CPU is shown below.

The flow of the sound generator continuous operation by the CPU is shown below.

(7) and later steps differ from the flow of the sound generator single operation by CPU.

Figure 32-10. Sound Generator Continuous Operation by CPU



1. With software, set the sound generator control information into the sound control register (SGCR) of the sound generator. Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
2. With software, set "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)."(*1: In addition, setting to all registers is not mandatory.)
3. Set the start bit (SGCR:ST) to "1".
4. The sound generator SGO and SGA output is started.
5. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
6. The CPU clears the interrupt.
7. With software, set "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)."(*1: In addition, setting to all registers is not mandatory.)
8. Perform the SGO and SGA output of the sound generator with the setting values.
9. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
10. The CPU clears the interrupt.
11. From here, (7) through (11) is repeated.
12. The CPU sends the stop instruction (SGCR:ST = 0) to the sound generator.

Note:

- Until the sound generator notifies of the interrupt and necessary settings are performed like Step (5) to (7), operations must be completed within (frequency data (SGFR) + 1) × 1PWM cycle.
- When the increment decrement setting is changed, the increment decrement setting bit (SGCR: GID) and automatic increment decrement enable bit (SGCR:GEN) of the sound control register (SGCR) must be changed within the cycle above.

32.5.7 Sound Generator Operation Coordinated with DMA

The sound generator operation coordinated with DMA is shown below.

The flow of the sound generator operation coordinated with DMA is shown.

DMAC performs setting of the data register for sound. The first interrupt assert differs from the flow of the sound generator operation by the CPU. In addition, the data register for sound is transferred through the DMA transfer indirect register (SGDMAR).

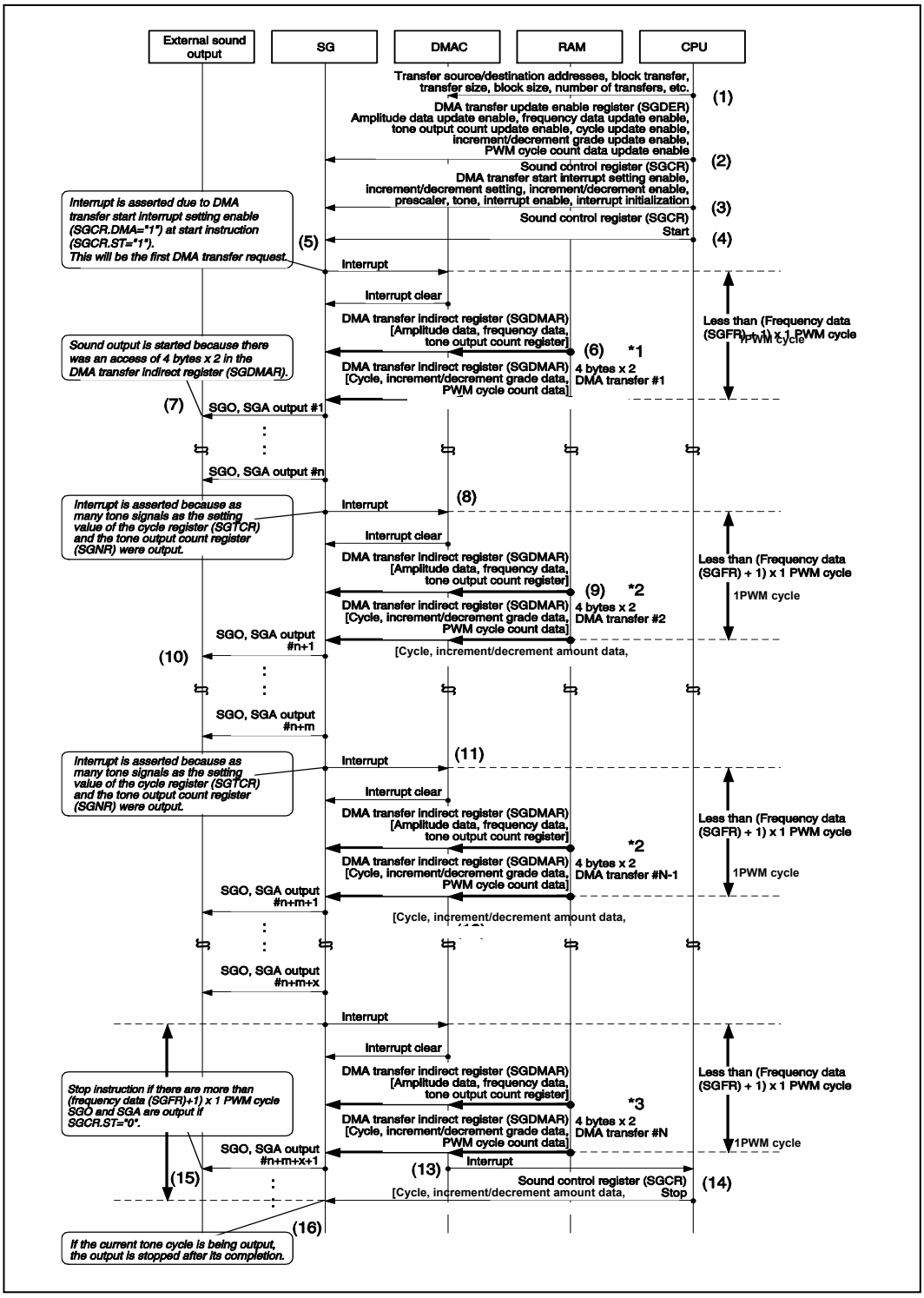
Note:

To make an interrupt signal a DMA transfer request, the interrupt enable (SGCR:INTE = 1) must be set with software.

32.5.8 When DMA Transfer of 4 Bytes × 2 is Performed N Times

N times transfer of the data (4 bytes x2) is shown.

Figure 32-11. Sound Generator Operation Coordinated with DMA (For DMA Transfer of 4 Bytes × 2, N Times)



1. By software, set the configuration of DMAC required for DMA transfer. In addition, for the DMA transfer, perform a block transfer of 4 bytes × 2 N times. Transfer data of "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)" through the DMA transfer indirect register (SGDMAR). In addition, the setting of the transfer destination address for DMAC is fixed (address of the DMA transfer indirect register).
2. By software, set up "DMA transfer update enable register (SGDER)" for the registers that should be transferred and updated. Here, set to update all registers of "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)."
3. With software, set the sound generator control information into the sound control register (SGCR) of the sound generator. In addition, set "DMA transfer start interrupt enable bit" to enable. Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
4. Set the start bit (SGCR:ST) to "1".
5. Due to the start instruction (SGCR:ST = 1) and DMA transfer start interrupt set enable setting (SGCR:DMA = 1), the interrupt bit (SGCR:INT) is set and the interrupt (PIRQ) is generated. This is used as a DMA transfer request.
6. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), with the first transfer, the values of the amplitude data register (SGAR), frequency data register (SGFR), and tone outputs number register (SGNR) are transferred, and with the second transfer, the values of the cycle register (SGTCR), increment decrement amount data register (SGDR), and PWM cycles number data register (SGPCR) are transferred.(*1: The block transfer of 4 bytes × 2 to the DMA transfer indirect register is mandatory.)
7. Since the block transfer of 4 bytes × 2 was performed for the DMA transfer indirect register (SGDMAR), the SGO and SGA output of the sound generator is started.
8. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
9. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the DMA transfer is performed.(*2: For the second DMA transfer and later also, the block transfer of 4 bytes × 2 to the DMA transfer indirect register is mandatory.)
10. Sound is output with the data transferred with DMA.
11. When the DMAC completes DMA transfer for the number of times specified, it notifies the CPU of an interrupt.
12. Hereafter, the same operation is continued.
13. Since the DMAC completes DMA transfer for specified number of times (transfer of 4 bytes × 2 N times), an interrupt is generated for the CPU.
14. The CPU sends the stop instruction (SGCR:ST = 0) to the sound generator.
15. When the stop instruction (SGCR:ST = 0) is sent within (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data are not output.(*3: The Nth transfer data is written to the register for sound, but sound is not output. The Nth transfer is to generate an interrupt for the CPU from the DMAC.)
16. When the stop instruction (SGCR:ST = 0) is sent after (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data stop output after the current tone cycle is completed.(*3: The Nth transfer data is output with sound.)

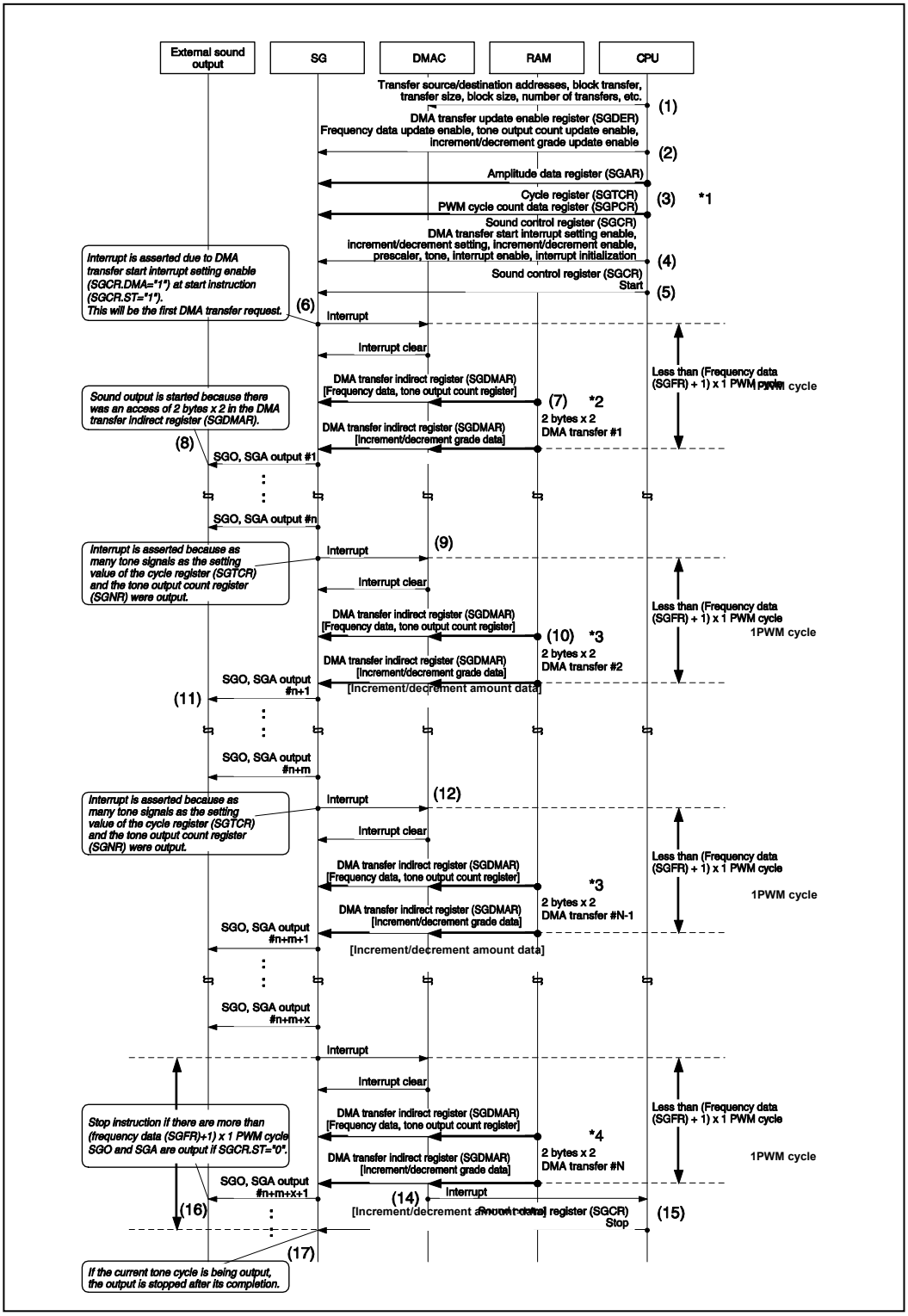
Note:

Until the sound generator notifies of the interrupt and the DMA transfer is completed like Step (5) to (6), operations must be completed within (frequency data (SGFR) + 1) × 1PWM cycle.

32.5.8.1 When DMA Transfer of 2 Bytes × 2 Is Performed N Times

N times transfer of the data (2 bytes x2) is shown.

Figure 32-12. Sound Generator Operation Coordinated with DMA (For DMA Transfer of 2 Bytes × 2, N Times)



1. By software, set the configuration of DMAC required for DMA transfer. In addition, for the DMA transfer, perform a block transfer of 2 bytes × 2 N times. Through the DMA transfer indirect register (SGDMAR), transfer data of "Frequency data register (SGFR) and Tone outputs number register (SGNR)" and "Increment decrement amount data register (SGIDR)". In addition, the setting of the transfer destination address for DMAC is fixed (address of the DMA transfer indirect register).
2. By software, set up "DMA transfer update enable register (SGDER)" for the registers that should be transferred and updated. Here, set to update "Frequency data register (SGFR) and Tone outputs number register (SGNR)" and "Increment decrement amount data register (SGIDR)".
3. With software, set "Amplitude data register (SGAR)" and "Cycle register (SGTCR) and PWM cycles number data register (SGPCR)" of the sound generator. (*1: Set registers that are not updated with the DMA transfer.)
4. With software, set the sound generator control information into the sound control register (SGCR) of the sound generator. In addition, set "DMA transfer start interrupt enable bit" to enable. Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
5. Set the start bit (SGCR.ST) to "1".
6. Due to the start instruction (SGCR:ST = 1) and DMA transfer start interrupt set enable setting (SGCR:DMA = 1), the interrupt bit (SGCR:INT) is set and the interrupt (PIRQ) is generated. This is used as a DMA transfer request.
7. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), with the first transfer, the values of the frequency data register (SGFR) and tone outputs number register (SGNR) are transferred, and with the second transfer, the value of the increment decrement amount data register (SGDR) is transferred.(*2: The block transfer of 2 bytes × 2 to the DMA transfer indirect register is mandatory.)
8. Since the block transfer of 2 bytes × 2 was performed for the DMA transfer indirect register (SGDMAR), the SGO and SGA output of the sound generator is started.
9. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
10. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the DMA transfer is performed. (*3: For the second DMA transfer and later also, the block transfer of 2 bytes × 2 to the DMA transfer indirect register is mandatory.)
11. Sound is output with the data transferred with DMA.
12. When the DMAC completes DMA transfer for the number of times specified, it notifies the CPU of an interrupt.
13. Hereafter, the same operation is continued.
14. Since the DMAC completes DMA transfer for specified number of times (transfer of 2 bytes × 2 N times), an interrupt is generated for the CPU.
15. The CPU sends the stop instruction (SGCR:ST = 0) to the sound generator.
16. When the stop instruction (SGCR:ST = 0) is sent within (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data are not output. (*4: The Nth transfer data is written to the register for sound, but sound is not output. The Nth transfer is to generate an interrupt for the CPU from the DMAC.)
17. When the stop instruction (SGCR:ST = 0) is sent after (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data stop output after the current tone cycle is completed. (*4: The Nth transfer data is output with sound.)

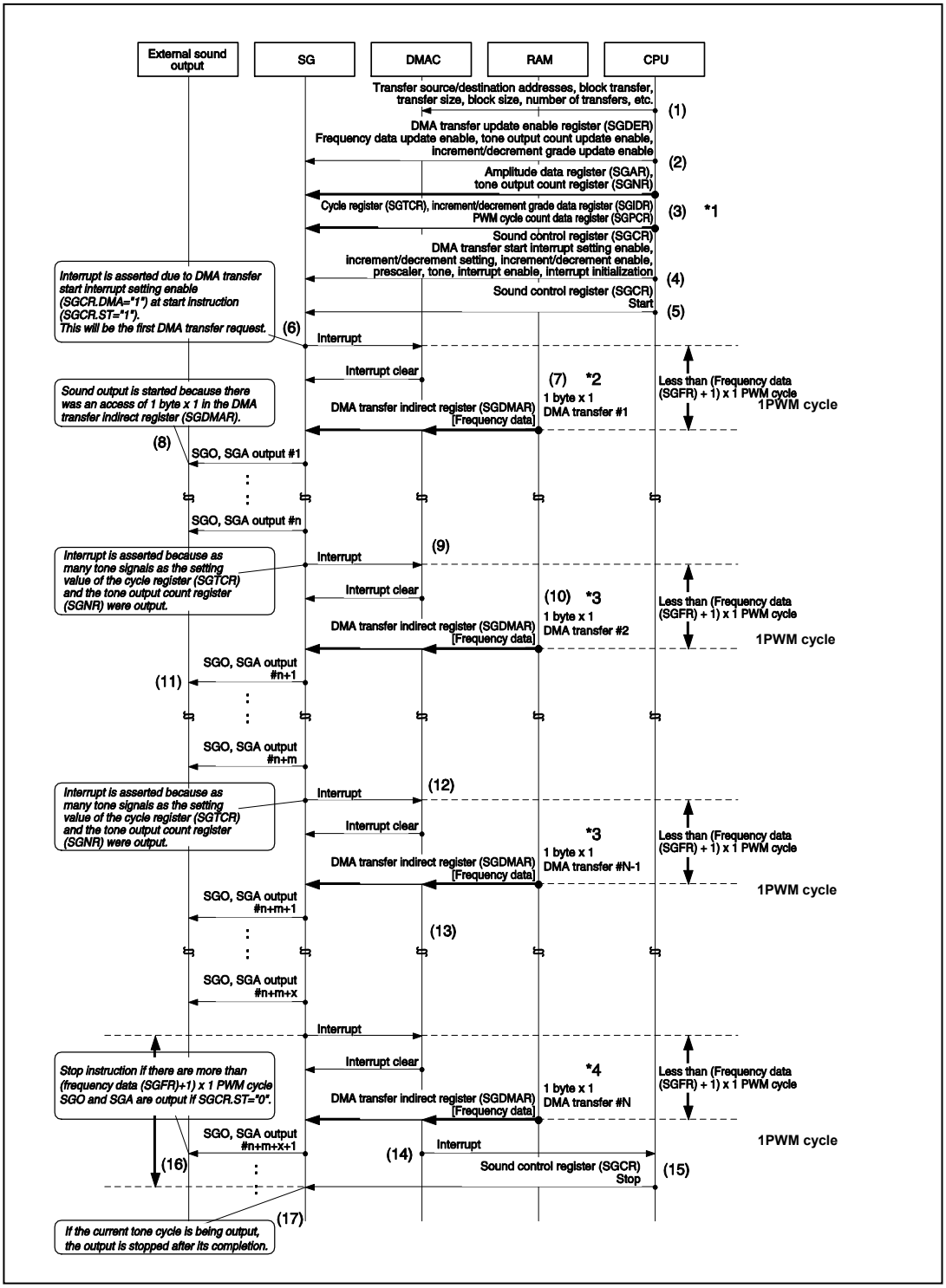
Note:

Until the sound generator notifies of the interrupt and the DMA transfer is completed like Step (6) to (7), operations must be completed within (frequency data (SGFR) + 1) × 1PWM cycle.

32.5.8.2 When DMA Transfer of 1 Byte × 1 is Performed N Times

N times transfer of the data (1 byte x1) is shown.

Figure 32-13. Sound Generator Operation Coordinated with DMA (For DMA Transfer of 1 Byte × 1, N Times)



1. By software, set the configuration of DMAC required for DMA transfer. In addition, for the DMA transfer, perform a block transfer of 1 byte × 1 N times. Through the DMA transfer indirect register (SGDMAR), transfer data of "Frequency data register (SGFR)." In addition, the setting of the transfer destination address for DMAC is fixed (address of the DMA transfer indirect register).
2. By software, set up "DMA transfer update enable register(SGDER)" for the registers that should be transferred and updated to the DMA transfer update enable register (SGDER) of the sound generator. Here, set to update "Frequency data register (SGFR)."
3. With software, set "Amplitude data register (SGAR) and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)" of the sound generator.(*1: Set registers that are not updated with the DMA transfer.)
4. With software, set the sound generator control information into the sound control register (SGCR) of the sound generator. In addition, set "DMA transfer start interrupt enable bit" to enable. Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
5. Set the start bit (SGCR:ST) to "1".
6. Due to the start instruction (SGCR:ST = 1) and DMA transfer start interrupt set enable setting (SGCR:DMA = 1), the interrupt bit (SGCR:INT) is set and the interrupt (PIRQ) is generated. This is used as a DMA transfer request.
7. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the value of the frequency data register (SGFR) is transferred. (*2: The block transfer of 1 byte × 1 to the DMA transfer indirect register is mandatory.)
8. Since the block transfer of 1 byte × 1 was performed for the DMA transfer indirect register (SGDMAR), the SGO and SGA output of the sound generator is started.
9. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
10. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the DMA transfer is performed. (*3: For the second DMA transfer and later also, the block transfer of 1 byte × 1 to the DMA transfer indirect register is mandatory.)
11. Sound is output with the data transferred with DMA.
12. When the DMAC completes DMA transfer for the number of times specified, it notifies the CPU of an interrupt.
13. Hereafter, the same operation is continued.
14. Since the DMAC completes DMA transfer for specified number of times (transfer of 1 byte × 1 N times), an interrupt is generated for the CPU.
15. The CPU sends the stop instruction (SGCR:ST = 0) to the sound generator.
16. When the stop instruction (SGCR:ST = 0) is sent within (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data are not output. (*4: The Nth transfer data is written to the register for sound, but sound is not output. The Nth transfer is to generate an interrupt for the CPU from the DMAC.)
17. When the stop instruction (SGCR:ST = 0) is sent after (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data stop output after the current tone cycle is completed.(*4: The Nth transfer data is output with sound.)

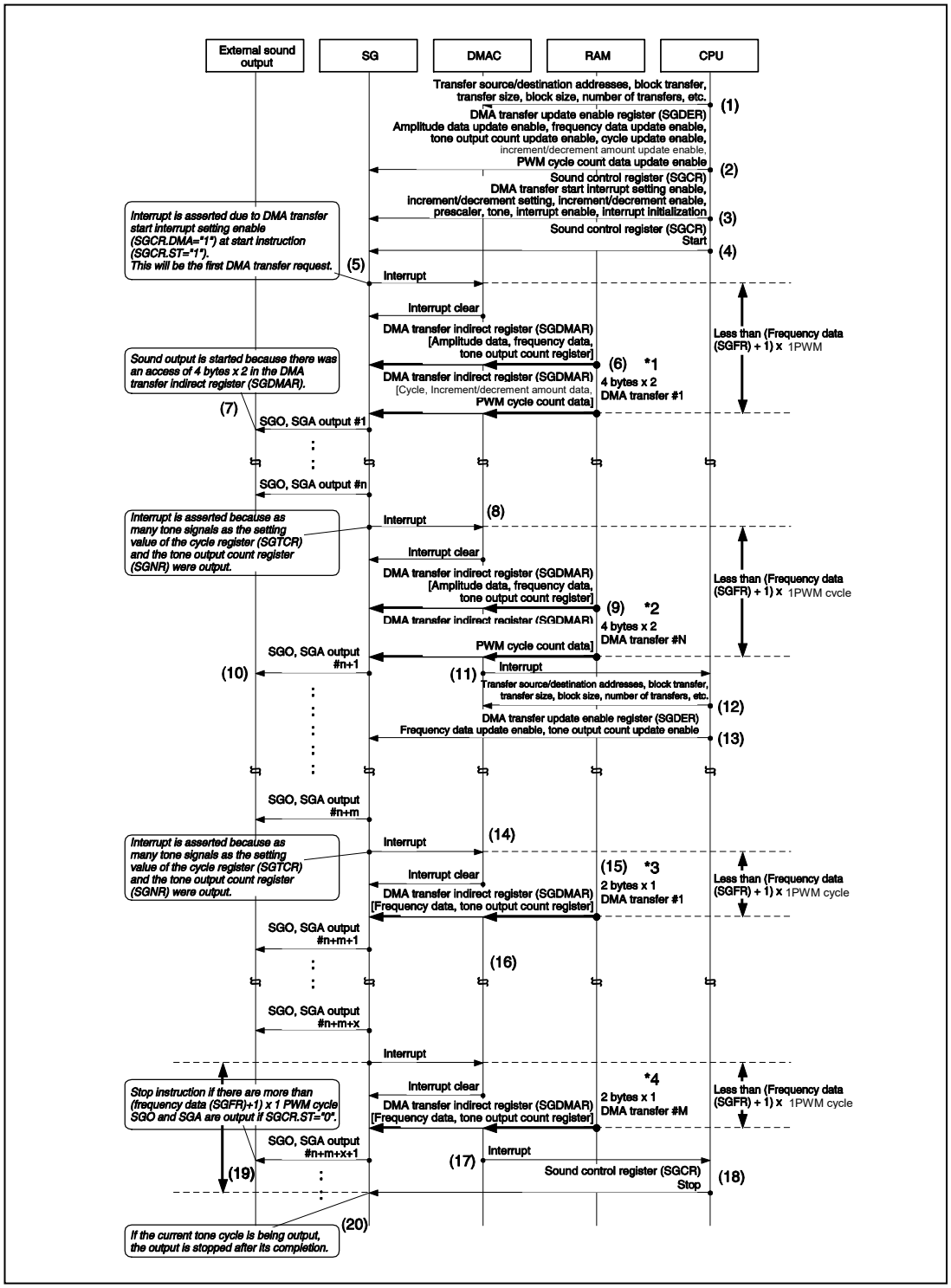
Note:

Until the sound generator notifies of the interrupt and the DMA transfer is completed like Step (6) to (7), operations must be completed within (frequency data (SGFR) + 1) × 1PWM cycle.

32.5.8.3 For DMA Transfer of 4 Bytes × 2 N Times and DMA Transfer of 2 Bytes × 1 M Times (Transfer Bytes Number Change During Sound Output)

N times transfer of the data (4 bytes x2) and M times transfer of the data (2 bytes x1) are shown.

Figure 32-14. Sound Generator Operation Coordinated with DMA (Transfer Bytes Number Change During Sound Output)



1. By software, set the configuration of DMAC required for DMA transfer. In addition, for the DMA transfer, perform a block transfer of 4 bytes × 2 N times. Transfer data of "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)" through the DMA transfer indirect register (SGDMAR). In addition, the setting of the transfer destination address for DMAC is fixed (address of the DMA transfer indirect register).
2. By software, set up "DMA transfer update enable register (SGDER)" for the registers that should be transferred and updated to the DMA transfer update enable register (SGDER) of the sound generator. Here, set to update all registers of "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)."
3. With software, set the sound generator control information into the sound control register (SGCR) of the sound generator. In addition, set "DMA transfer start interrupt enable bit" to enable. Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
4. Set the start bit (SGCR:ST) to "1".
5. Due to the start instruction (SGCR:ST = 1) and DMA transfer start interrupt set enable setting (SGCR:DMA = 1), the interrupt bit (SGCR:INT) is set and the interrupt (PIRQ) is generated. This is used as a DMA transfer request.
6. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), with the first transfer, the values of the amplitude data register (SGAR), frequency data register (SGFR), and tone outputs number register (SGNR) are transferred, and with the second transfer, the values of the cycle register (SGTCR), increment decrement amount data register (SGDR), and PWM cycles number data register (SGPCR) are transferred.(*1: The block transfer of 4 bytes × 2 to the DMA transfer indirect register is mandatory.)
7. Since the block transfer of 4 bytes × 2 was performed for the DMA transfer indirect register (SGDMAR), the SGO and SGA output of the sound generator is started.
8. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
9. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the DMA transfer is performed. (*2: For the second DMA transfer and later also, the block transfer of 4 bytes × 2 to the DMA transfer indirect register is mandatory.)
10. Sound is output with the data transferred with DMA.
11. When the DMAC completes DMA transfer for the number of times specified (transfer of 4 bytes × 2, N times), it notifies the CPU of an interrupt.
12. Set up DMAC for translation by software. In addition, for the DMA transfer, perform a block transfer of 2 bytes × 1 M times. Through the DMA transfer indirect register (SGDMAR), transfer data of "Frequency data register (SGFR) and Tone outputs number register (SGNR)."
13. By software, set up "DMA transfer update enable register (SGDER)" for the registers that should be transferred and updated to the DMA transfer update enable register (SGDER) of the sound generator. Here, set to update "Frequency data register (SGFR) and Tone outputs number register (SGNR)."
14. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
15. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the DMA transfer is performed. (*3: The block transfer of 2 bytes × 1 to the DMA transfer indirect register is mandatory.)
16. Hereafter, the same operation is continued.
17. Since the DMAC completes DMA transfer for specified number of times (transfer of 2 bytes × 1, M times), an interrupt is generated for the CPU.

18. The CPU sends the stop instruction (SGCR:ST = 0) to the sound generator.
19. When the stop instruction (SGCR:ST = 0) is sent within (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data are not output. (*4: The Nth transfer data is written to the register for sound, but sound is not output. The Nth transfer is to generate an interrupt for the CPU from the DMAC.)
20. When the stop instruction (SGCR:ST = 0) is sent after (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data stop output after the current tone cycle is completed. (*4: The Nth transfer data is output with sound.)

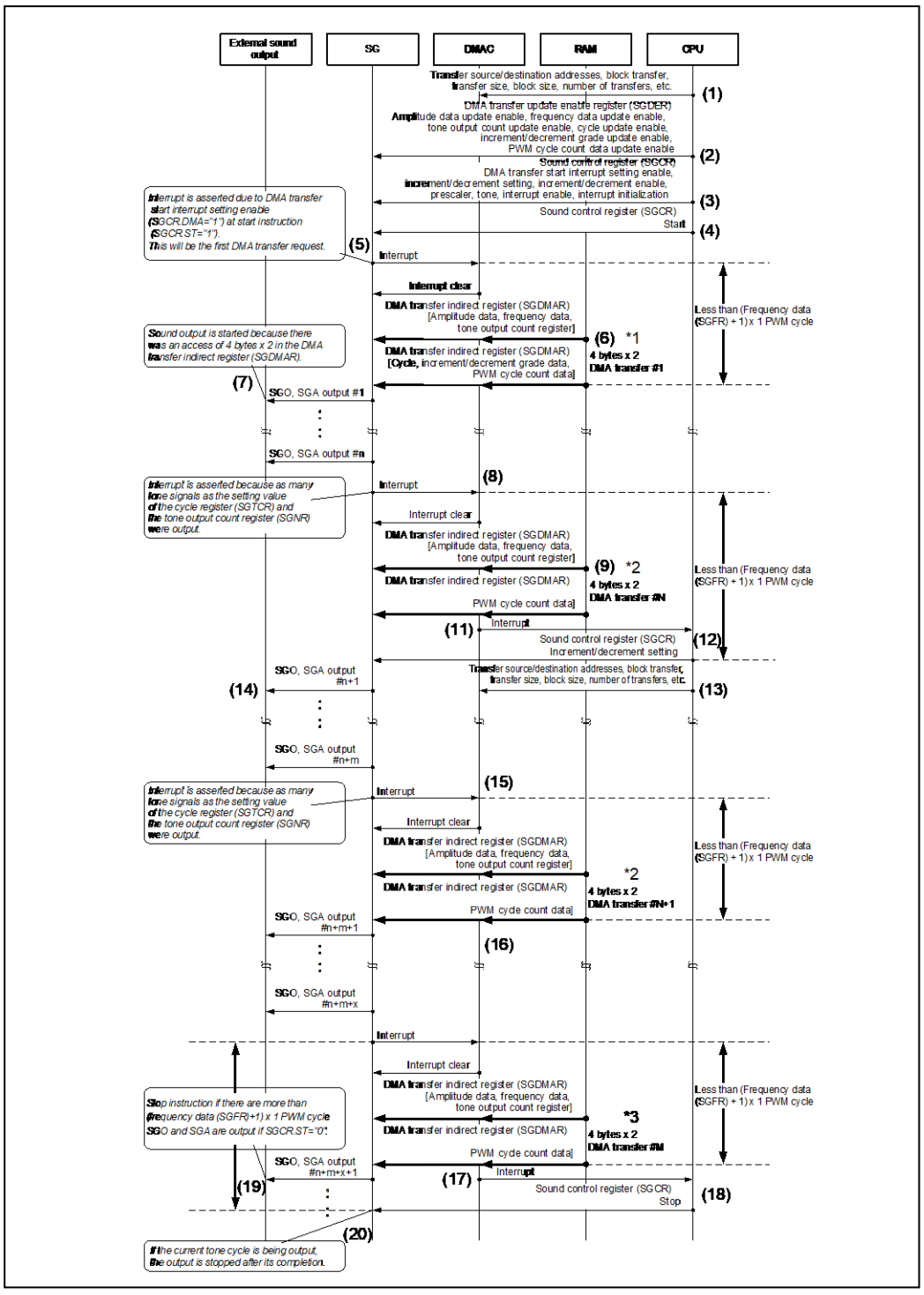
Note:

Until the sound generator notifies of the interrupt and the DMA transfer is completed like Step (5) to (6), operations must be completed within (frequency data (SGFR) + 1) × 1PWM cycle.

32.5.8.4 For DMA Transfer of 4 Bytes × 2 N Times and DMA Transfer of 4 Bytes × 2 M Times (Transfer Bytes Number and Increment Decrement Setting Change During Sound Output)

N times transfer of the data (4 bytes x2) and M times transfer of the data (4 bytes x2) are shown.

Figure 32-15. Sound Generator Operation Coordinated with DMA (Transfer Bytes Number and Increment Decrement Setting Change During Sound Output)



1. By software, set the configuration of DMAC required for DMA transfer. In addition, for the DMA transfer, perform a block transfer of 4 bytes × 2 N times. Transfer data of "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)" through the DMA transfer indirect register (SGDMAR). In addition, the setting of the transfer destination address for DMAC is fixed (address of the DMA transfer indirect register).
2. By software, set up "DMA transfer update enable register (SGDER)" for the registers that should be transferred and updated to the DMA transfer update enable register (SGDER) of the sound generator. Here, set to update all registers of "Amplitude data register (SGAR), Frequency data register (SGFR), and Tone outputs number register (SGNR)" and "Cycle register (SGTCR), Increment decrement amount data register (SGIDR), and PWM cycles number data register (SGPCR)."
3. With software, set the sound generator control information into the sound control register (SGCR) of the sound generator. In addition, set "DMA transfer start interrupt enable bit" to enable. Initialize the interrupt bit (SGCR:INT), and set the interrupt enable bit (SGCR:INTE).
4. Set the start bit (SGCR:ST) to "1".
5. Due to the start instruction (SGCR:ST = 1) and DMA transfer start interrupt set enable setting (SGCR:DMA = 1), the interrupt bit (SGCR:INT) is set and the interrupt (PIRQ) is generated. This is used as a DMA transfer request.
6. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), with the first transfer, the values of the amplitude data register (SGAR), frequency data register (SGFR), and tone outputs number register (SGNR) are transferred, and with the second transfer, the values of the cycle register (SGTCR), increment decrement amount data register (SGDR), and PWM cycles number data register (SGPCR) are transferred.(*1: The block transfer of 4 bytes × 2 to the DMA transfer indirect register is mandatory.)
7. Since the block transfer of 4 bytes × 2 was performed for the DMA transfer indirect register (SGDMAR), the SGO and SGA output of the sound generator is started.
8. When the sound generator outputs the tone pulses number set in the cycle register (SGTCR) and tone outputs number register (SGNR), an interrupt is generated.
9. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the DMA transfer is performed. (*2: For the second DMA transfer and later also, the block transfer of 4 bytes × 2 to the DMA transfer indirect register is mandatory.)
10. Sound is output with the data transferred with DMA.
11. When the DMAC completes DMA transfer for the number of times specified (transfer of 4 bytes × 2, N times), it notifies the CPU of an interrupt.
12. With software, change the increment decrement setting of the sound control register (SGCR).
13. By software, set the configuration of DMAC required for DMA transfer. Here, set the transfer of 4 bytes × 2 M times.
14. Sound is output with the data transferred with DMA.
15. The DMAC clears the interrupt, and through the DMA transfer indirect register (SGDMAR), the DMA transfer is performed. (*2: The block transfer of 4 bytes × 2 to the DMA transfer indirect register is mandatory.)
16. Hereafter, the same operation is continued.
17. Since the DMAC completes DMA transfer for specified number of times (transfer of 4 bytes × 2, M times), an interrupt is generated for the CPU.
18. The CPU sends the stop instruction (SGCR:ST = 0) to the sound generator.
19. When the stop instruction (SGCR:ST = 0) is sent within (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data are not output. (*3: The Nth transfer data is written to the register for sound, but sound is not output. The Nth transfer is to generate an interrupt for the CPU from the DMAC.)
20. When the stop instruction (SGCR:ST = 0) is sent after (frequency data (SGFR) + 1) × 1PWM cycle, SGO and SGA by the Nth transfer data stop output after the current tone cycle is completed. (*3: The Nth transfer data is output with sound.)

Note:

- Until the sound generator notifies of the interrupt and the DMA transfer is completed like Step (5) to (6), operations must be completed within (frequency data (SGFR) + 1) × 1PWM cycle.

Note:

- Until the sound generator notifies of the interrupt and the software changes the increment decrement setting like Step (8) to (12), operations must be completed within (frequency data (SGFR) + 1) × 1PWM cycle.

Note:

- The increment decrement setting is enabled from the sound output in Step (14). With the use of the Nth transfer data of 4 bytes × 2, the automatic increment decrement is performed.

33. Stepping Motor Controller

This chapter explains the stepping motor controller.

33.1 Overview

33.2 Features

33.3 Configuration

33.4 Registers

33.5 Operation

33.6 Setting

33.7 Q&A

33.8 Sample Programs

33.9 Notes

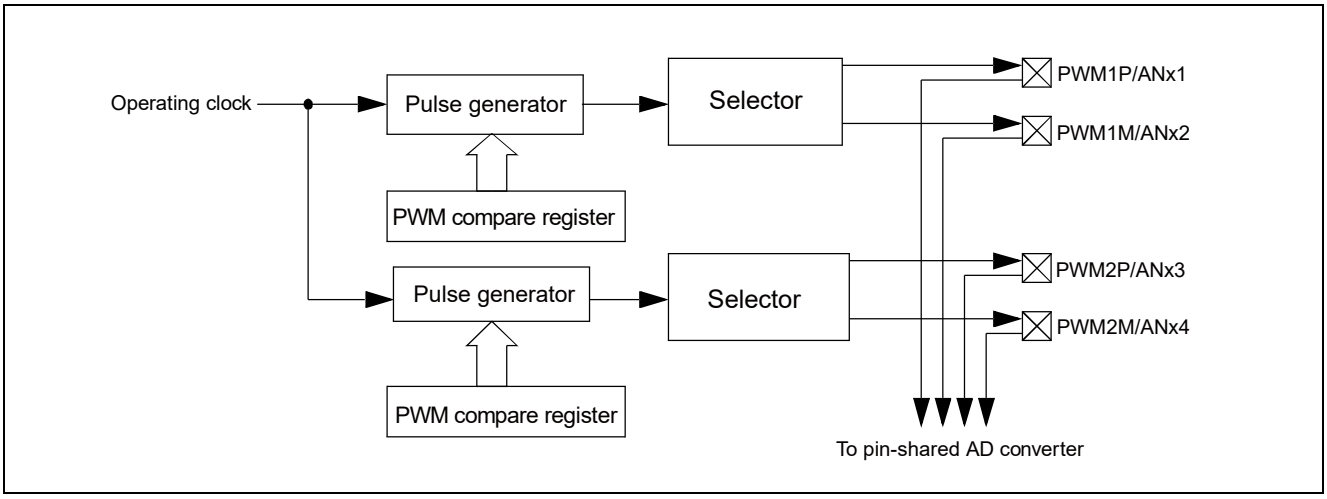
33.1 Overview

This section explains the overview of the stepping motor controller.

This stepping motor controller consists of 2 PWM pulse generators and 4 motor drivers.

The 4 set of motor drivers have high output power driving capacity and 2 set of motor coils are able to connect directly to the 4 terminals. This controller is designed to control the motor revolution with the combination of the PWM pulse generator and the selector logic. The synchronization mechanism ensures the synchronization operation between 2 set of PWMs.

Figure 33-1. Block Diagram (1 Channel, Overview)



33.2 Features

This section explains features of the stepping motor controller.

PWM operation mode:

- The PWM operation mode can be selected from 8/10−bit operations.

Number of set: 6

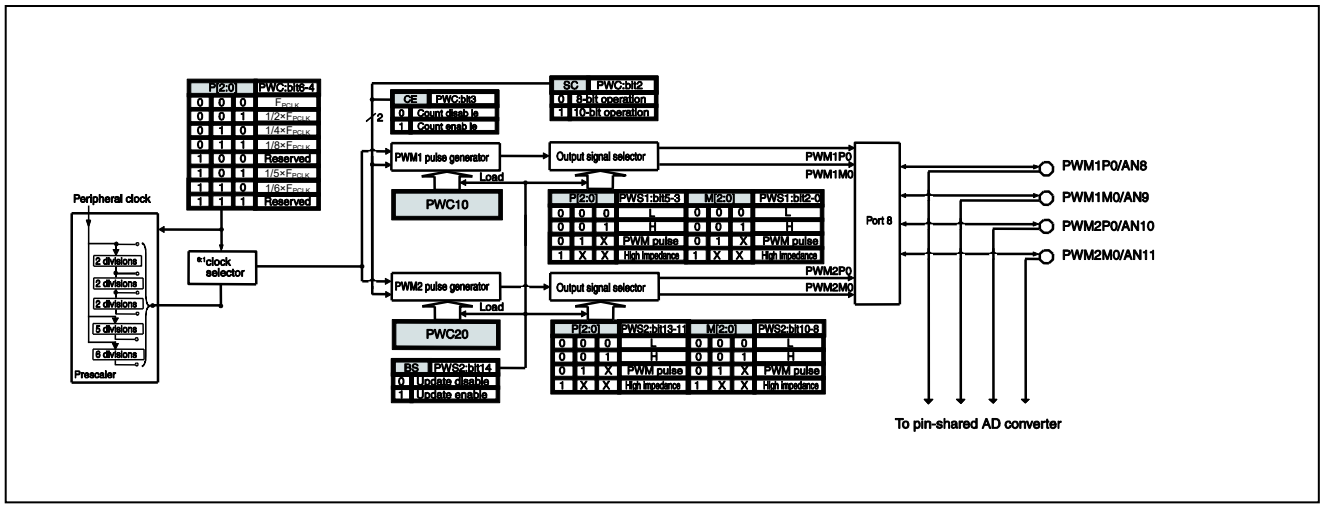
PWM operating clock:

- Peripheral clock/1, /2, /4, /5, /6, /8

33.3 Configuration

This section explains the configuration of the stepping motor controller.

Figure 33-2. Block Diagram (1 Channel, Detailed)



33.4 Registers

This section explains registers of the stepping motor controller.

List of Base_addresses (Base_addr) and External Pins

Channel number	Base_addr	External pin name			
		PWM1P	PWM1M	PWM2P	PWM2M
0	0x200	PWM1P0	PWM1M0	PWM2P0	PWM2M0
1	0x208	PWM1P1	PWM1M1	PWM2P1	PWM2M1
2	0x210	PWM1P2	PWM1M2	PWM2P2	PWM2M2
3	0x218	PWM1P3	PWM1M3	PWM2P3	PWM2M3
4	0x220	PWM1P4	PWM1M4	PWM2P4	PWM2M4
5	0x228	PWM1P5	PWM1M5	PWM2P5	PWM2M5

Registers Map

Figure 33-3. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0200	PWC20		PWC10		PWM2 compare register 0 PWM1 compare register 0
0x204	Reserved	PWC0	PWS20	PWS10	PWM control register 0 PWM2 selection register 0 PWM1 selection register 0
0x0208	PWC21		PWC11		PWM2 compare register 1 PWM1 compare register 1
0x020C	Reserved	PWC1	PWS21	PWS11	PWM control register 1 PWM2 selection register 1 PWM1 selection register 1
0x0210	PWC22		PWC12		PWM2 compare register 2 PWM1 compare register 2
0x0214	Reserved	PWC2	PWS22	PWS12	PWM control register 2 PWM2 selection register 2 PWM1 selection register 2
0x0218	PWC23		PWC13		PWM2 compare register 3 PWM1 compare register 3
0x021C	Reserved	PWC3	PWS23	PWS13	PWM control register 3 PWM2 selection register 3 PWM1 selection register 3
0x0220	PWC24		PWC14		PWM2 compare register 4 PWM1 compare register 4
0x0224	Reserved	PWC4	PWS24	PWS14	PWM control register 4 PWM2 selection register 4 PWM1 selection register 4
0x0228	PWC25		PWC15		PWM2 compare register 5 PWM1 compare register 5
0x022C	Reserved	PWC5	PWS25	PWS15	PWM control register 5 PWM2 selection register 5 PWM1 selection register 5

33.4.1 PWM Control Register: PWC

The bit configuration of the PWM control register (PWC) is shown below.

The PWC is used to set activation/stop for the stepping motor controller.

PWC: Address Base_addr + 05_H (Access: Byte)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	P2	P1	P0	CE	SC	-	Reserved
Initial value	0	0	0	0	0	0	-	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R1,WX	R/W0

[bit7] Reserved

[bit6 to bit4] P2 to P0 : Operating clock selection bits

P2 to P0 bits select a clock input signal for the PWM pulse generator.

P2	P1	P0	Clock input	PWM cycle (F_{PCLK} = 16MHz)	
				SC=0	SC=1
0	0	0	F_{PCLK}	16.0 μs	64.0 μs
0	0	1	1/2 × F_{PCLK}	32.0 μs	128.0 μs
0	1	0	1/4 × F_{PCLK}	64.0 μs	256.0 μs
0	1	1	1/8 × F_{PCLK}	128.0 μs	512.0 μs
1	0	0	Reserved	-	-
1	0	1	1/5 × F_{PCLK}	80.0 μs	320.0 μs
1	1	0	1/6 × F_{PCLK}	96.0 μs	384.0 μs
1	1	1	Reserved	-	-

F_{PCLK}: Peripheral clock (PCLK)

[bit3] CE: Count enable bit

The CE bit enables the operation of the PWM pulse generator. When the CE bit is set to "1", the PWM pulse generator will start operating. The PWM2 pulse generator will start operating 1 machine cycle after the PWM1 pulse generator's start to reduce switching noise from the output drivers.

When the CE bit is cleared to "0" while the PWM pulse generator is operating, the PWM pulse generator will be initialized and stop operating.

Note:

If you set "1" to the CE bit, the operating clock selection must be completed.

[bit2] SC: 8/10 bit switching bit

When the SC bit is set to "1", the PWM will operate in 10-bit mode. When the SC bit is set to "0", the PWM will operate in 8-bit mode.

[bit1] - : Undefined bit

The read value is always "1". This does not affect the writing operation.

[bit0] Reserved

"0" should be written to this bit.

33.4.2 PWM1&2 Compare Register: PWC1/PWC2

The bit configuration of PWM1&2 compare registers (PWC1/PWC2) is shown below.

The 2 sets of 8 (10) bit compare registers for PWM1&2 are used to determine the width of PWM pulse. Memorized value "00_H"("000_H") means that the PWM duty is 0%, and "FF_H"("$3FF_H$") means 99.6% (99.9%).

The PWM1&2 compare registers can be accessed at given timing but modified value will be reflected to the pulse width at the end of current PWM cycle after the BS bit of the PWM2 selection register is set to "1".

When the SC bit of the PWM control register is set to "0" and the PWM is operating in 8-bit mode, the values of D9 and D8 will be unknown.

PWM1&2 compare registers must be accessed in half-word or word.

PWC1: Address Base_addr + 02_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-	-	-	-	-	-	D9	D8
Initial value	-	-	-	-	-	-	X	X
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
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

PWC2: Address Base_addr + 00H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-	-	-	-	-	-	D9	D8
Initial value	-	-	-	-	-	-	X	X
Attribute	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R1,WX	R/W	R/W
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
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 to bit10] -: Undefined bits

The read value is always "1". This does not affect the writing operation.

[bit9 to bit0] D9 to D0: Compare data

Set a pulse width for the PWM to these bits.

33.4.3 PWM1 Selection Register: PWS1

The bit configuration of the PWM1 selection register (PWS1) is shown below.

The PWM1 selection register is used to determine the output of the stepping motor controller's external pins from "0", "1", PWM pulse or high impedance.

PWS1: Address Base_addr + 07_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	P2	P1	P0	M2	M1	M0
Initial value	-	-	0	0	0	0	0	0
Attribute	R1,WX	R1,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit7, bit6] - : Undefined bits

The read value is always "1". This does not affect the writing operation.

[bit5 to bit3] P2 to P0: Output selection bits

P2, P1 and P0 bits are used to select output signals for the PWM1P0 to PWM1P5.

[bit2 to bit0] M2 to M0: Output selection bits

M2, M1 and M0 bits are used to select output signals for the PWM1M0 to PWM1M5.

Following table shows the relation between the output level and selected bits:

P2	P1	P0	PWM1Pn	M2	M1	M0	PWM1Mn
0	0	0	L	0	0	0	L
0	0	1	H	0	0	1	H
0	1	X	PWM pulse	0	1	X	PWM pulse
1	X	X	High impedance	1	X	X	High impedance

n = 0 to 5

33.4.4 PWM2 Selection Register: PWS2

The bit configuration of the PWM2 selection register (PWS2) is shown below.

The PWM2 selection register is used to determine the output of the stepping motor controller's external pins from "0", "1", PWM pulse or high impedance.

PWS2: Address Base_addr + 06_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	BS	P2	P1	P0	M2	M1	M0
Initial value	-	0	0	0	0	0	0	0
Attribute	R1,WX	R,W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7] - : Undefined bits

The read value is always "1". This does not affect the writing operation.

[bit6] BS: Rewrite bit

The BS bit is used to synchronize the setting for PWM outputs. The changes to the 2 set of compare registers and 2 set of selection registers will not be reflected to the output signals before the BS bit is set.

When the BS bit is set to "1", the PWM pulse generator and selector will load the contents of the registers at the end of PWM cycle. The BS bit will automatically be cleared to "0" at the beginning of next cycle. If the BS bit is set to"1"with software at the same time of this automatic clearing, the BS bit will be set to "1" (no change) and automatic clearing will be released.

If the BS bit is set to"0"with software at the same time of this automatic clearing, the BS bit will be cleared to "0" and the PWM pulse generator and selector will not load the contents of the registers at the end of PWM cycle.

Note:

If a read-modify-write instruction is executed to a bit other than BS with BS = "1", "1" will be read from the BS and "1" will be written to the BS bit once again. If the BS bit is automatically cleared at the beginning of PWM cycle between read and write, "1" will be written to the BS bit once again after cleared. Therefore, if "1" is not set to the BS bit by the end of next PWM cycle, the PWM pulse generator and selector will load the contents of the registers.

[bit5 to bit3] P2 to P0: Output selection bits

P2, P1 and P0 bits are used to select output signals for the PWM2P0 to PWM2P5.

[bit2 to bit0] M2 to M0: Output selection bits

M2, M1 and M0 bits are used to select output signals for the PWM2M0 to PWM2M5.

Following table shows the relation between the output level and selected bits:

P2	P1	P0	PWM2Pn	M2	M1	M0	PWM2Mn
0	0	0	L	0	0	0	L
0	0	1	H	0	0	1	H
0	1	X	PWM pulse	0	1	X	PWM pulse
1	X	X	High impedance	1	X	X	High impedance

n = 0 to 5

33.5 Operation

This section explains operating of the stepping motor controller.

33.5.1 PWM Operation

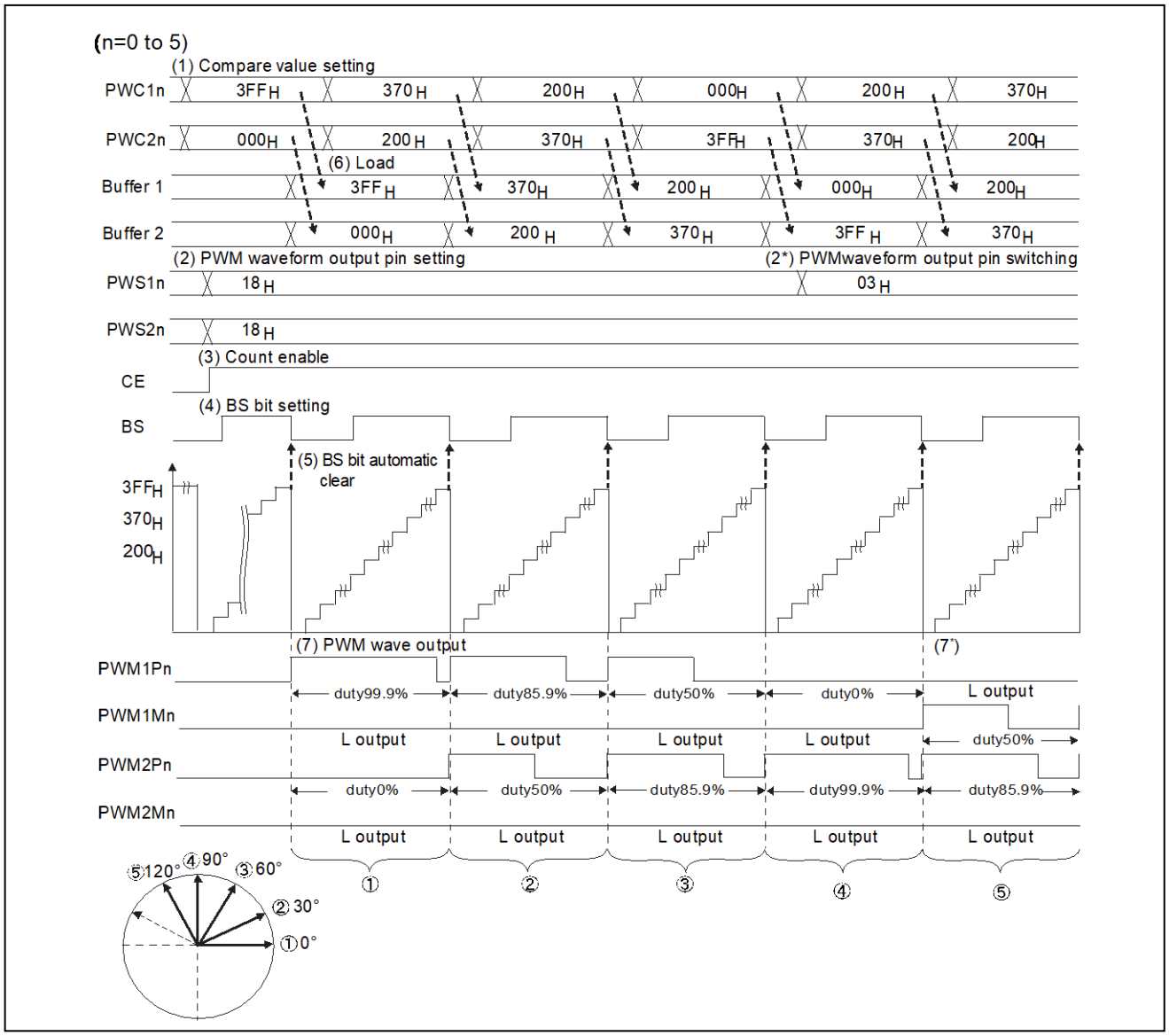
33.5.2 PWM Compare Register Loading with the BS Bit

33.5.3 Selection of Motor Drive Signals

33.5.1 PWM Operation

The PWM operation is explained.

Figure 33-4. PWM Operation

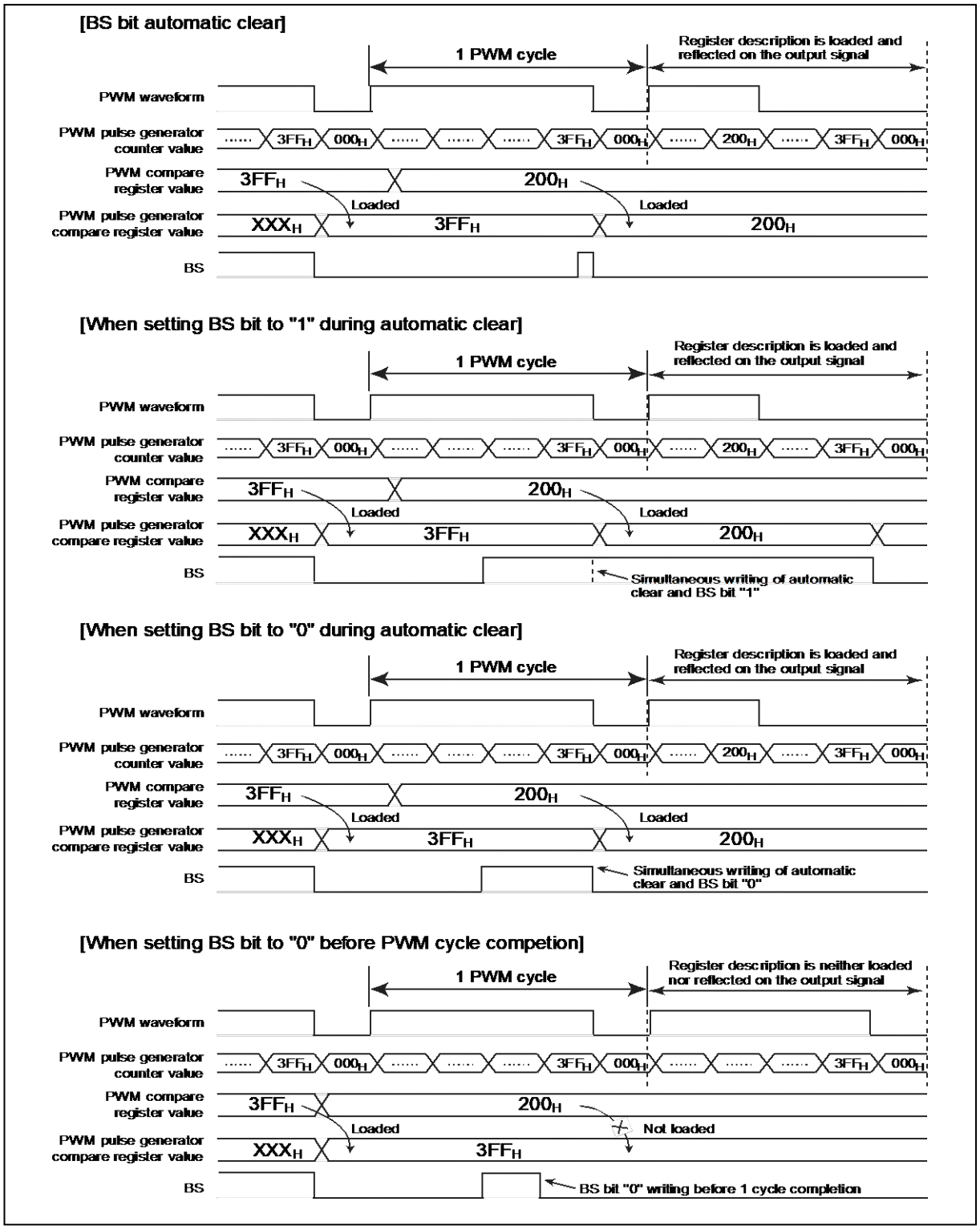


1. Set compare value
2. PWM waveform output pin setting/(2*) PWM waveform output pin switching
3. Count enable
4. The BS bit setting
5. The BS bit automatic clearing
6. Load compare value
7. PWM wave output/ PWM wave output switching by (7*) and (2*)
8. Repeat step (1) to (7).

33.5.2 PWM Compare Register Loading with the BS Bit

PWM compare register loading with the BS bit is explained.

Figure 33-5. Loading PWM Compare Register Values



1. The BS bit automatic clearing: Loading will be performed to be reflected to the output signals.
2. Automatic clearing at the same time of setting "1" to the BS bit: Loading will be performed to be reflected to the output signals.
3. Automatic clearing at the same time of setting "0" to the BS bit: Loading will be performed to be reflected to the output signals.
4. "0" is set to the BS bit before the end of PWM cycle: Loading will not be performed and no reflection
(See "Notes" for "[bit6] BS: Rewrite bit".)

33.5.3 Selection of Motor Drive Signals

This section explains selection of motor drive signals.

The motor drive signals which output to the pins related to the stepping motor controller can be selected for each pin from 4 types with the setting of the PWM selection registers.

Table 33-1 shows the motor drive signal selection and the settings for the PWM selection register 1 and 2.

Writing "1" to the BS bit of the PWM selection register 2 after these settings are made, the setting values will be valid at the end of current PWM cycle. This BS bit will automatically be cleared at the beginning of next cycle. If writing to the BS bit and BS bit clearing are occurred at the same time of the beginning of next cycle, writing to the BS bit is prioritized and the BS bit clearing will be cancelled.

Table 33-1. Motor Drive Signal Selection and Settings for PWM Selection Register

P2, P1, P0 Bits	PWM1P0 to PWM1P5 Outputs PWM2P0 to PWM2P5 Outputs	M2, M1, M0 Bits	PWM1M0 to PWM1M5 Outputs PWM2M0 to PWM2M5 Outputs
000_B	L	000_B	L
001_B	H	001_B	H
$01X_B$	PWM pulse	$01X_B$	PWM pulse
$1XX_B$	High impedance	$1XX_B$	High impedance

33.6 Setting

This section explains setting of the stepping motor controller.

Table 33-2. Settings Required for PWM Operation

Setting	Setting Register	Setting Method
PWM operation start	PWM control (PWC0 to PWC5)	See 33.7.2
PWM operating clock setting		See 33.7.4
8/10-bit mode switching		See 33.7.1
Compare value (Duty value) setting	PWM1&2 compare (PWC10 to PWC15/PWC20 to PWC25)	
Selection of motor drive signals	PWS1&2 selection (PWS10 to PWS15/PWS20 to PWS25)	See 33.7.5
PWM pin output setting	Set the pins as peripheral output. See "Chapter: I/O Ports".	

Table 33-3. Settings Required to Stop PWM

Setting	Setting Register	Setting Method
PWM operation stop	PWM control (PWC0 to PWC5)	See 33.7.2

Table 33-4. Settings Required for Changing PWM Output

Setting	Setting Register	Setting Method
Compare value (Duty value) setting	PWM1&2 compare (PWC10 to PWC15/PWC20 to PWC25)	See 33.7.1
Selection of motor drive signals	PWS1&2 selection (PWS10 to PWS15/PWS20 to PWS25)	See 33.7.5
Rewrite bit(BS bit) setting	PWM control (PWC0 to PWC5)	See 33.7.3

33.7 Q&A

This section explains Q&A of the stepping motor controller.

33.7.1 How to Set Cycle and Duty
33.7.2 How to Enable/Stop PWM Operation
33.7.3 How to Reflect the Duty Change
33.7.4 Type and Selection of Operating Clock
33.7.5 How to Change the Motor Drive Signals
33.7.6 How to Assign a Pin as a PWM Output Pin
33.7.7 How to Assign a Pin as an A/D Converter Analog Input Pin

33.7.1 How to Set Cycle and Duty

How to set the cycle and duty is explained.

Cycle value setting and duty value setting

- Set the cycle value (operating clock selection, 8/10-bit operation selection) in the PWM control register PWC0 to PWC5.
- Set the duty value in the PWM1&2 compare register (PWC10 to PWC15, PWC20 to PWC25).

Calculation formulas:

☐ Cycle:
8-bit operation (PWC0 to PWC5:SC = 0) : (1/operating clock) × 256
10-bit operation (PWC0 to PWC5:SC = 1) : (1/operating clock) × 1024

Note:

Specify the operating clock with PWC0 to PWC5:P[2:0]. (F_{PCLK}, 1/2 × F_{PCLK}, 1/4 × F_{PCLK}, 1/8 × F_{PCLK}, 1/5 × F_{PCLK}, 1/6 × F_{PCLK} (F_{PCLK}: peripheral clock))

☐ Duty:
8-bit operation (PWC0 to PWC5:SC = 0) : PWC1&2 compare register value = duty × (256/100)
10-bit operation (PWC0 to PWC5:SC = 1) : PWC1&2 compare register value = duty × (1024/100)

Available setting range

☐ Cycle:
16µs, 32µs, 64µs, 80µs, 96µs, 128µs, 256µs, 320µs, 384µs, 512µs (F_{PCLK} = 16MHz)

☐ PWC1&2 compare register value:
8-bit operation (PWC0 to PWC5:SC = 0) : 0 to 99.6% (0 to FF_H)
10-bit operation (PWC0 to PWC5:SC = 1) : 0 to 99.9% (0 to $3FF_H$)

33.7.2 How to Enable/Stop PWM Operation

How to enable/stop the PWM operation is shown below.

PWM operation enable

Use the count enable bit (PWC0 to PWC5:CE).

Control	Count enable Bit (CE)
How to stop PWM operation	Set to "0"
How to enable PWM operation	Set to "1"

If you enable the counting, the operating clock selection must be completed.

Note:

See "[bit3] CE: Count enable bit" in "33.4.1 PWM Control Register: PWC".

33.7.3 How to Reflect the Duty Change

How to reflect the duty change is shown below.

Duty change

Writing "1" to the BS bit of the PWM1&2 selection registers, the pulse width will be updated at the end of current PWM cycle.

Control	Rewrite Bit (BS)
How to change the duty	Set to "1"

Note:

See Figure 33-5 for details on the load timing of the PWM1&2 compare registers.

33.7.4 Type and Selection of Operating Clock

The type and selection of the operating clock are shown below.

Operating clock selection

Use the operating clock selection bits (PWC0 to PWC5:P[2:0]).

See "33.4.1 PWM Control Register: PWC" for setting the operating clock selection bits (PWC0 to PWC5: P[2:0]).

33.7.5 How to Change the Motor Drive Signals

How to change the motor drive signals is shown below.

Motor drive signal change

The motor drive signals can be selected from L, H, PWM pulse, or high impedance with the output selection bits (PWS10 to PWS15/PWS20 to PWS25).

See "[bit2 to bit0] M2 to M0: Output selection bits" in "33.4.3 PWM1 Selection Register: PWS1" and "[bit2 to bit0] M2 to M0: Output selection bits" in "33.4.4 PWM2 Selection Register: PWS2" for details on the output selection bit setting.

33.7.6 How to Assign a Pin as a PWM Output Pin

How to assign a pin as a PWM output pin is shown below.

Set the pins as peripheral output. For details, see "Chapter: I/O Ports".

33.7.7 How to Assign a Pin as an A/D Converter Analog Input Pin

How to assign a pin as an A/D converter analog input pin is shown below.

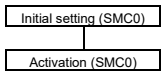
Set the pins as an A/D converter input. For details, see "Chapter: I/O Ports".

33.8 Sample Programs

This section explains sample programs.

Setting Procedure Example 1

PWM pulse output from PWM1P0 and PWM2P0



<Initial setting (SMC0)>

- Port

SMC0 output setting for ports	See "Chapter : I/O Ports".

- SMC0 control Register name.Bit name

PWM control register setting	PWC0
Sampling clock selection>>	.S2
Operating clock selection>>	.P[2:0]
Count setting>>	.CE
8/10 bit switch>>	.SC

- Duty setting Register name.Bit name

PWC1 compare register setting	PWC10
PWC2 compare register setting	PWC20

- Output pin setting Register name.Bit name

PWS1 selection register setting	PWS10
PWM1P0 pin output selection>>	.P[2:0]
PWM1M0 pin output selection>>	.M[2:0]
PWS2 selection register setting	PWS20
PWM2P0 pin output selection>>	.P[2:0]
PWM2M0 pin output selection>>	.M[2:0]

<Activation (SMC0)>

- SMC0 activation Register name.Bit name

Count enable	PWC0.CE

- Duty change Register name.Bit name

PWC1 compare register setting	PWC10
PWC2 compare register setting	PWC20

- BS bit set Register name.Bit name

PWS2 selection register setting	PWS20.BS

Program Example 1

```
void SMC0_sample_1(void)
{
        SMC0_initial();
        SMC0_start();
}
void SMC0_initial(void)
{
        IPORT_SETTING_SMC0_OUT()  /* Set the SMC0 pins as peripheral output. */

        IO_PWC0.byte = 0x36;      /* Setting value = 0011_0110 */
                                  /* bit7 = 0 S2 sampling clock setting */
                                  /* bit6 to 4 = 011 P[2:0] Operating clock setting */
                                  /* bit3 = 0 CE count disable */
                                  /* bit2 = 1 SC 10bit operation */
                                  /* bit1 = 1 Undefined bit */
                                  /* bit0 = 0 Reserved bit */

        IO_PWC10.hword = 0x03ff;  /* PWM10 Duty setting */
        IO_PWC20.hword = 0x0000;  /* PWM20 Duty setting */

        IO_PWS10.byte = 0x1f;     /* Setting value = 0001_1111 */
                                  /* bit7 to 6 = 00 Undefined bits */
                                  /* bit5 to 3 = 011 P[2:0] PWM1P0 = PWM output  */
                                  /* bit2 to 0 = 111 M[2:0] PWM1M0 = Hi-Z output */
        IO_PWS20.byte = 0x58;     /* Setting value = 0101_1000 */
                                  /* bit7 = 0 Undefined bit */
                                  /* bit6 = 1 BS rewrite setting */
                                  /* bit5 to 3 = 011 P[2:0] PWM2P0 = PWM output */
                                  /* bit2 to 0 = 000 M[2:0] PWM2M0 = L output */
}

Void SMC0_start(void)
{
        IO_PWC0.bit.CE = 1;       /* bit3= 1 CE count enable */
        ……
        ……                        /* BS bit automatic clearing waiting */

        IO_PWC10.hword = 0x0370;  /* PWM10 Duty change */
        IO_PWC20.hword = 0x0200;  /* PWM20 Duty change */

        IO_PWS20.byte = 0x58;     /* Setting value = 0101_1000 */
        ……
}
```

33.9 Notes

This section explains notes of the stepping motor controller.

Notes for PWM Setting Value Change

- The PWM compare registers 1/2 (PWC10 to PWC15, PWC20 to PWC25) and the PWM selection registers 1/2 (PWS10 to PWS15, PWS20 to PWS25) can be always accessed. However, to change the width of the "H" level of the PWM or PWM output, "1" must be written to the BS bit of the PWM2 selection register after (or at the same time) the setting values are written to these registers.
- After "1" is set to the BS bit, new setting values will become valid at the end of the current PWM cycle and the BS bit will automatically be cleared.
- In addition, if writing "1" to the BS bit and the reset of the BS bit at the end of the PWM cycle are occurred at the same time, writing to the BS bit is prioritized and the BS bit reset will be cancelled.

34. Regulator Control

This chapter explains the overview, features and configurations of the regulator control.

34.1 Overview
34.2 Features
34.3 Configuration
34.4 Register
34.5 Operation

34.1 Overview

This section explains the overview of the regulator control.

The operation of the regulator that generates the internal voltage is automatically changed according to the device state transition.

It is changed automatically to following three regulator modes.

- Main mode (at normal operation)
- Sub mode (at sub run)
- Standby mode(at STOP mode and Watch mode)

34.2 Features

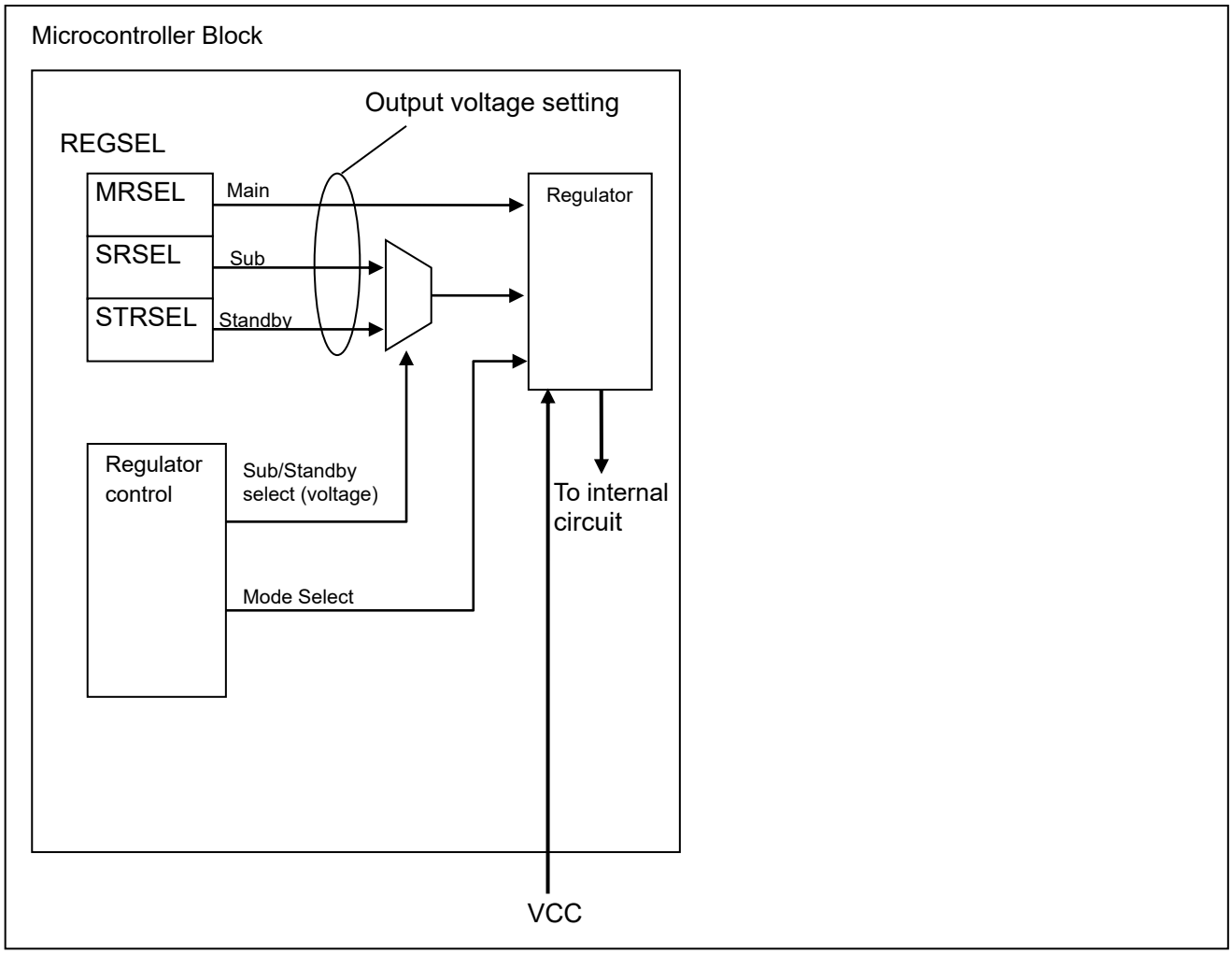
This section explains features of the regulator control.

- The regulator mode is automatically changed according to the device state transition.

34.3 Configuration

This section explains the configuration of the regulator control.

Figure 34-1. Regulator Control Block Diagram



Note:

The difference between the sub mode and the standby mode is only the output voltage settings.

34.4 Register

This section explains a register of the regulator control.

Table 34-1. Register Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0580	REGSEL	Reserved	Reserved	Reserved	Regulator Output Voltage Selection Register

34.4.1 Regulator Output Voltage Select Register: REGSEL

The bit configuration of the regulator output voltage selection register is shown below.

It is a register that selects the output voltage level of each regulator mode (main/sub/standby).

REGSEL: Address 0580$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MRSEL[1:0]		SRSEL[1:0]		STRSEL[2:0]			Reserved
Initial value	0	1	1	0	0	1	1	0
Attribute	R/W0	R/W1	R/W1	R/W0	R/W1	R/W1	R/W0	R0,WX

[bit7, bit6] MRSEL[1:0] (Main Regulator voltage Select)

These bits set the output voltage level of main regulator.

MRSEL[1:0]	Main Regulator Output Voltage
00	Reserved
01	1.2 ± 0.1 V
10	Reserved
11	Reserved

[bit5, bit4] SRSEL[1:0] (Sub Regulator voltage Select)

These bits set the output voltage level of sub regulator.

SRSEL[1:0]	Sub Regulator Output Voltage
00	Reserved
01	Reserved
10	1.2 ± 0.1 V
11	Reserved

[bit3 to bit1] STRSEL[2:0] (Standby Regulator Voltage Select)

These bits set the output voltage level of standby regulator.

STRSEL[2:0]	Standby Regulator Output Voltage
000	Reserved
001	Reserved
010	Reserved
011	0.9 ± 0.1V
100	Reserved
101	Reserved
110	1.2 ± 0.1V
111	Reserved

Note:

Please use 1.2 V as the set value (STRSEL[2:0]=110).

[bit0] Reserved

34.5 Operation

This section explains the operation of the regulator control.

Before entering standby mode, set STRSEL[2:0] to 110. After a reset this value (STRSEL[2:0] = 110) note that it has not been set.

35. LCD Controller

This chapter explains the LCD controller.

35.1 Overview

35.2 Features

35.3 Configuration

35.4 Registers

35.5 Operation

35.6 Setting

35.7 Q&A

35.8 Sample Program

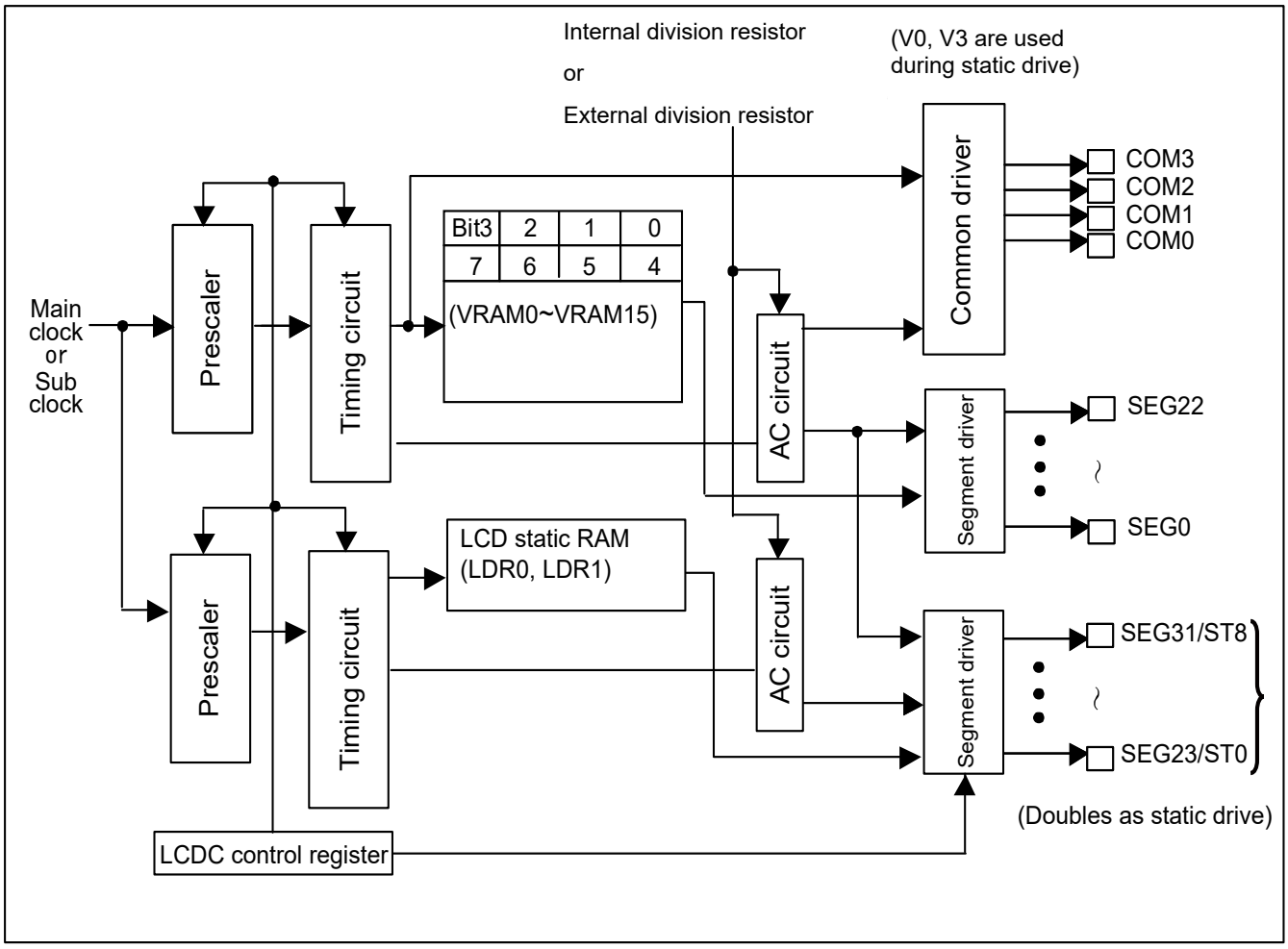
35.9 Notes

35.1 Overview

This section explains the overview of the LCD controller.

The LCD enables the selection of duty from 1/2, 1/3, and 1/4, and up to 128-element display.

Furthermore, it enables up to 8-element display as static LCD output.



35.2 Features

This section explains the features of the LCD controller.

Duty drive

- Number: 1 (4-common × 32-segment)
- Display: Up to 128-element (on 1/4 duty)
- Duty: Selection from three types (1/2, 1/3, 1/4)
- Bias: Selection from 1/2 and 1/3

Combination of Bias, Duty, and Common Output

Bias	1/2 Duty Output Mode	1/3 Duty Output Mode	1/4 Duty Output Mode
1/2 bias	○	×	×
1/3 bias	×	○	○

○ : Recommended mode
× : Prohibited

- Frame cycle: Selectable from four types (For a clock, a main clock or sub clock can be used.)
- Driver: Built-in (Internal division resistor), or an external division resistor can be connected to V0 to V3 pins.
- Data memory: Built-in 16-byte data memory for display
- Nondisplay selection: Enabled
  - ☐ Pin: COM0 to COM3, SEG0 to SEG31, V0, V1, V2, V3 pins are used for general-purpose ports also and switching is enabled.
    SEG23 to SEG31 pins are used as static drive pins (ST0 to ST8) also and switching is enabled.
    VCC can be selected instead of V3.
- During duty driving, up to 4 common outputs (COM0 to COM3) and 32 segment outputs (SEG0 to SEG31) can be used.
  - ☐ Others: The external division resistor also can block a current when LCD is stopped.

Static drive

- Number: 1 (1-common × 8-segment)
- Display: Up to 8 element
- Frame cycle: Selectable from four types (For a clock, a main clock or sub clock can be used.)
- Pin: ST0 to ST8 pins are used also as general-purpose ports and duty drive pins (SEG23 to SEG31) and switching is enabled.
- During static driving, up to 1 static common output and 8 static segment outputs (ST0 to ST8) can be used.

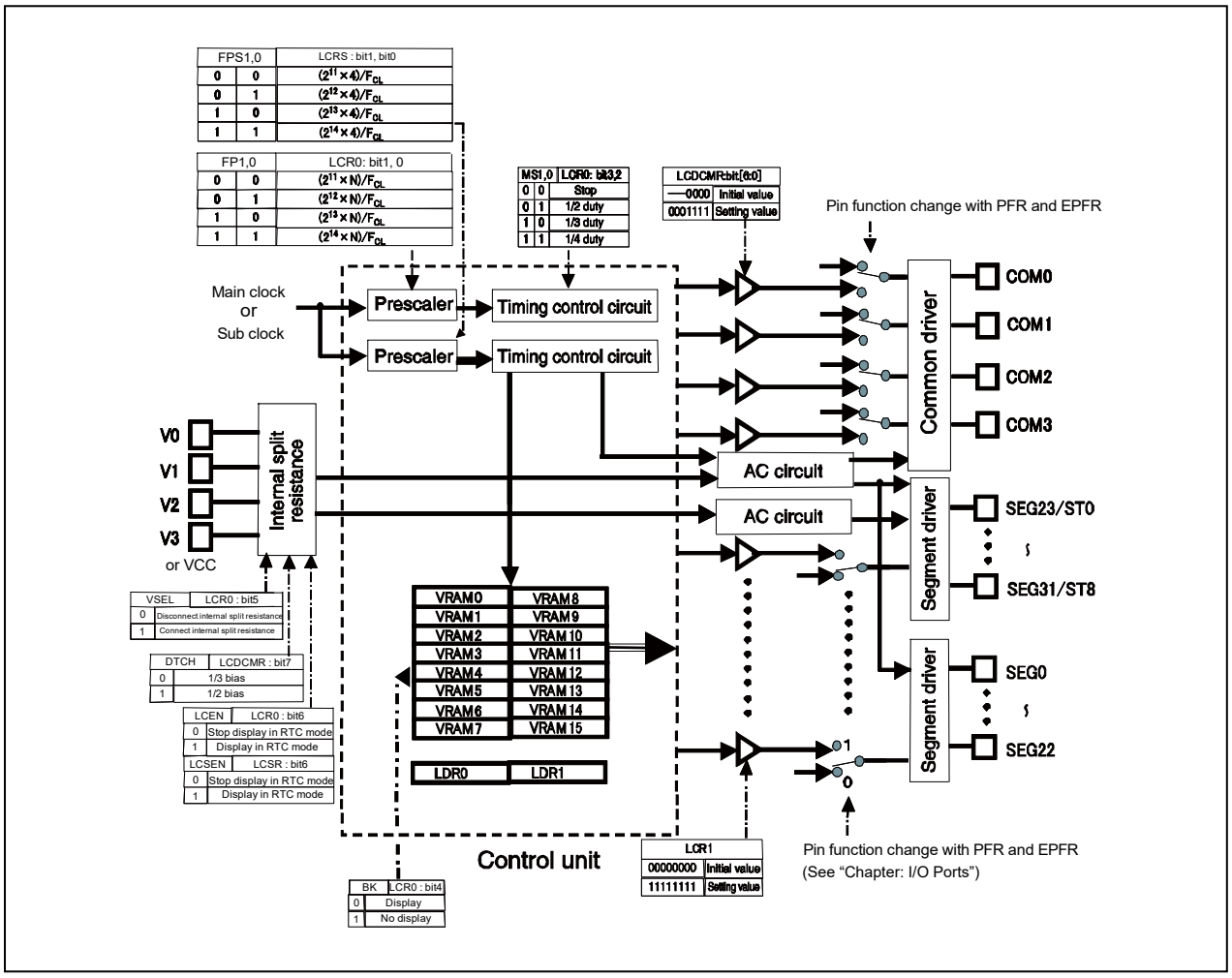
35.3 Configuration

This section shows the configuration of the LCD controller.

Configuration Diagram of LCD Controller

Figure 35-1 shows the configuration diagram of the LCD controller.

Figure 35-1. Configuration Diagram



35.4 Registers

This section explains the registers of the LCD controller.

Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x05A8	LCDCMR	LCRS	LCR0	LCR1	Common pin switching register LCDC static control register LCD control register 0 LCD control register 1
0x05AC	VRAM0	VRAM1	VRAM2	VRAM3	Data memory for display
0x05B0	VRAM4	VRAM5	VRAM6	VRAM7	
0x05B4	VRAM8	VRAM9	VRAM10	VRAM11	
0x05B8	VRAM12	VRAM13	VRAM14	VRAM15	
0x05BC	LDR		Reserved		Static LCD display data register

35.4.1 LCD Control Register 0: LCR0

The bit configuration of the LCD control register 0 is shown below.

This register selects a frame cycle with its clock to be used, selects display mode, enables the LCD controller to operate in the condition for selecting display/non-display mode, and controls LCD drive power.

LCR0: Address 05AA$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	CSS	LCEN	VSEL	BK	MS1	MS0	FP1	FP0
Initial value	0	0	0	1	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit7] CSS: Clock selection bit

Select a clock to be used for this module.

CSS	Operation
0	Main clock
1	Sub clock

For the single clock products, set "0".

[bit6] LCEN: Watch mode operation enable

LCEN	Operation
0	LCD display stop in the watch mode
1	LCD display in the watch mode

[bit5] VSEL: LCD drive power control

VSEL	Operation
0	Internal division resistor disconnection
1	Internal division resistor connection

When an external division resistor is to be connected, the LCD drive power control bit (VSEL) must be set to "0".

[bit4] BK: Blanking selection

BK	Operation
0	LCD display
1	LCD nondisplay

[bit3, bit2] MS1, MS0: Display mode selection

MS1	MS0	Display Mode
0	0	LCD operation stop
0	1	1/2 duty output mode (Time division number: N=2, COM0, COM1)
1	0	1/3 duty output mode (Time division number: N=3, COM0 to COM2)
1	1	1/4 duty output mode (Time division number: N=4, COM0 to COM3)

When the display mode selection bits (MS[1:0]) are set to "00", the LCD controller's operation is stopped, and the common pin/segment pin output "L" level.

[bit1, bit0] FP1, FP0: Frame cycle

FP1	FP0	Frame Cycle
0	0	For CSS=0 : $(2^{11} \times N)/F_{CL}$ For CSS=1 : $(2^{3} \times N)/F_{CL}$
0	1	For CSS=0 : $(2^{12} \times N)/F_{CL}$ For CSS=1 : $(2^{4} \times N)/F_{CL}$
1	0	For CSS=0 : $(2^{13} \times N)/F_{CL}$ For CSS=1 : $(2^{5} \times N)/F_{CL}$
1	1	For CSS=0 : $(2^{14} \times N)/F_{CL}$ For CSS=1 : $(2^{6} \times N)/F_{CL}$

F_{CL}: Main clock (LCR0:CSS=0) or sub clock (LCR0:CSS=1)

N: Time division number (depends on the setting of the display mode selection bits MS1, MS0)

Select the setting for FP1 and FP0 that is the optimum condition for the frame frequency of the LCD panel to be used.

35.4.2 Data Memory for Display: VRAM

The data memory for display is shown below.

Memory (VRAM) area for setting of data for display

VRAM0 (SEG0, SEG1): Address 05AC$_H$ (Access: Byte, Half-word, Word)

VRAM1 (SEG2, SEG3): Address 05AD$_H$ (Access: Byte, Half-word, Word)

VRAM2 (SEG4, SEG5): Address 05AE$_H$ (Access: Byte, Half-word, Word)

VRAM3 (SEG6, SEG7): Address 05AF$_H$ (Access: Byte, Half-word, Word)

VRAM4 (SEG8, SEG9): Address 05B0$_H$ (Access: Byte, Half-word, Word)

VRAM5 (SEG10, SEG11): Address 05B1$_H$ (Access: Byte, Half-word, Word)

VRAM6 (SEG12, SEG13): Address 05B2$_H$ (Access: Byte, Half-word, Word)

VRAM7 (SEG14, SEG15): Address 05B3$_H$ (Access: Byte, Half-word, Word)

VRAM8 (SEG16, SEG17): Address 05B4$_H$ (Access: Byte, Half-word, Word)

VRAM9 (SEG18, SEG19): Address 05B5$_H$ (Access: Byte, Half-word, Word)

VRAM10 (SEG20, SEG21): Address 05B6$_H$ (Access: Byte, Half-word, Word)

VRAM11 (SEG22, SEG23): Address 05B7$_H$ (Access: Byte, Half-word, Word)

VRAM12 (SEG24, SEG25): Address 05B8$_H$ (Access: Byte, Half-word, Word)

VRAM13 (SEG26, SEG27): Address 05B9$_H$ (Access: Byte, Half-word, Word)

VRAM14 (SEG28, SEG29): Address 05BA$_H$ (Access: Byte, Half-word, Word)

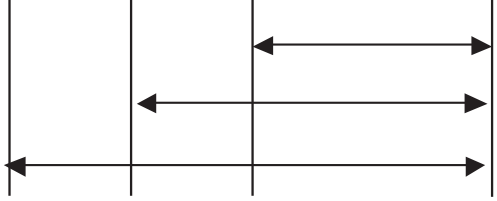
VRAM15 (SEG30, SEG31): Address 05BB$_H$ (Access: Byte, Half-word, Word)

	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
Atrribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The RAM for display can be read /written with any timing regardless of the LCD controller/driver operation.

Correspondence between VRAM and common pins/segment pins

VRAM0	bit3	bit2	bit1	bit0	SEG0
	bit7	bit6	bit5	bit4	SEG1
VRAM1	bit3	bit2	bit1	bit0	SEG2
	bit7	bit6	bit5	bit4	SEG3
VRAM2	bit3	bit2	bit1	bit0	SEG4
	bit7	bit6	bit5	bit4	SEG5
VRAM3	bit3	bit2	bit1	bit0	SEG6
	bit7	bit6	bit5	bit4	SEG7
VRAM4	bit3	bit2	bit1	bit0	SEG8
	bit7	bit6	bit5	bit4	SEG9
VRAM5	bit3	bit2	bit1	bit0	SEG10
	bit7	bit6	bit5	bit4	SEG11
VRAM6	bit3	bit2	bit1	bit0	SEG12
	bit7	bit6	bit5	bit4	SEG13
VRAM7	bit3	bit2	bit1	bit0	SEG14
	bit7	bit6	bit5	bit4	SEG15
VRAM8	bit3	bit2	bit1	bit0	SEG16
	bit7	bit6	bit5	bit4	SEG17
VRAM9	bit3	bit2	bit1	bit0	SEG18
	bit7	bit6	bit5	bit4	SEG19
VRAM10	bit3	bit2	bit1	bit0	SEG20
	bit7	bit6	bit5	bit4	SEG21
VRAM11	bit3	bit2	bit1	bit0	SEG22
	bit7	bit6	bit5	bit4	SEG23
VRAM12	bit3	bit2	bit1	bit0	SEG24
	bit7	bit6	bit5	bit4	SEG25
VRAM13	bit3	bit2	bit1	bit0	SEG26
	bit7	bit6	bit5	bit4	SEG27
VRAM14	bit3	bit2	bit1	bit0	SEG28
	bit7	bit6	bit5	bit4	SEG29
VRAM15	bit3	bit2	bit1	bit0	SEG30
	bit7	bit6	bit5	bit4	SEG31
	COM3	COM2	COM1	COM0	



RAM area and a common pin used in for 1/2 duty output mode

RAM area and a common pin used in for 1/3 duty output mode

RAM area and a common pin used in for 1/4 duty output mode

35.4.3 LCDC Control Register 1: LCR1

The bit configuration of the LCDC control register 1 is shown below.

LCR1: Address $05AB_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1

[bit7 to bit0] Reserved

When the LCD is used, set "11111111" always.

35.4.4 Common Pin Switching Register: LCDCMR

The bit configuration of the common pin switching register is shown below.

LCDCMR: Address $05A8_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DTCH	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W0	R/W0	R/W0	R/W1	R/W1	R/W1	R/W1

[bit7] DTCH: Bias selection

DTCH	Operation
0	1/3 bias
1	1/2 bias

[bit6 to bit4] Reserved

When the LCD is used, set "000" always.

[bit3 to bit0] Reserved

When the LCD is used, set "1111" always.

35.4.5 LCDC Static Control Register: LCRS

The bit configuration of the LCDC static control register is shown below.

LCRS: Address 05A9$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SCSS	LCSEN	LCS3	LCS2	LCS1	LCS0	FPS1	FPS0
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] SCSS: Frame cycle generation clock for static drive selection bit

SCSS	Operation
0	Main clock
1	Sub clock

For the single clock products, set "0".

[bit6] LCSEN: Watch mode operation enable

LCSEN	Operation
0	LCD display stop in the watch mode
1	LCD display in the watch mode

[bit5 to bit2] LCS3 to LCS0: Static drive selection

LCS3	LCS2	LCS1	LCS0	Static Drive Selection Port
				Common/Segment Output
0	0	0	0	OFF
0	0	0	1	ST0, ST1
0	0	1	0	ST0 to ST2
0	0	1	1	ST0 to ST3
0	1	0	0	ST0 to ST4
0	1	0	1	ST0 to ST5

LCS3	LCS2	LCS1	LCS0	Static Drive Selection Port
				Common/Segment Output
0	1	1	0	ST0 to ST6
0	1	1	1	ST0 to ST7
1	X	X	X	ST0 to ST8

Note:

For the use method for the static drive selection, refer to "35.5.5 Static Drive Output Waveform".

[bit1, bit0] FPS1, FPS0: Frame cycle

FPS1	FPS0	Frame Cycle
0	0	For SCSS=0 : $(2^{11}\times4)/F_{CL}$ For SCSS=1 : $(2^{3}\times4)/F_{CL}$
0	1	For SCSS=0 : $(2^{12}\times4)/F_{CL}$ For SCSS=1 : $(2^{4}\times4)/F_{CL}$
1	0	For SCSS=0 : $(2^{13}\times4)/F_{CL}$ For SCSS=1 : $(2^{5}\times4)/F_{CL}$
1	1	For SCSS=0 : $(2^{14}\times4)/F_{CL}$ For SCSS=1 : $(2^{6}\times4)/F_{CL}$

F_{CL}: Main clock (LCRS:SCSS=0) or sub clock (LCRS:SCSS=1)

Select the setting for FPS1 and FPS0 that is the optimum condition for the frame frequency of the LCD panel to be used.

The frame cycle during static drive is equal to the cycle of the 1/4 duty output mode.

The static drive is enabled when PFR setting and any setting other than LCS[3:0] = "0000" are executed.

The duty drive is enabled when PFR and LCS[3:0] = "0000" are set.

35.4.6 Static LCD Display Data Register: LDR

The bit configuration of the static LCD display data register is shown below.

LDR0: Address 05BC$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-		-	-	-	-	-	ST8
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 bits

The read value is always "0". Writing has no effect on operation.

[bit0] ST8: Static output data

This bit is static output data for ST8.

LDR1: Address 05BDH (Access: Byte, Half-word, Word)

	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

[bit7 to bit0] ST7 to ST0: Static output data

These bits are static output data for ST0 to ST7.

Static Output Data Correspondence Table

LDR	Operation
ST8	ST8 static output data
ST7	ST7 static output data
ST6	ST6 static output data
ST5	ST5 static output data
ST4	ST4 static output data
ST3	ST3 static output data
ST2	ST2 static output data
ST1	ST1 static output data
ST0	ST0 static output data

This register is for setting of data output to the LCD static drive port (set by the LCRS register). The LCD static drive is executed by inverting output of the specified data periodically.

Common output pins are not specified especially. Assign common output to one of segments. For example, if ST0 to ST8 are assigned static drive pins, ST8 is a common pin and set to LDR0:ST8 = "0". Furthermore, by setting LDR1:ST[7:0] = "11111111", the LCD selected with ST0 to ST7 is displayed.

35.5 Operation

The following sections explain the operation of the LCD controller.

35.5.1 Operation of LCD Controller/Driver (LCDC)
35.5.2 1/2 Duty Output Waveform
35.5.3 1/3 Duty Output Waveform
35.5.4 1/4 Duty Output Waveform
35.5.5 Static Drive Output Waveform

35.5.1 Operation of LCD Controller/Driver (LCDC)

The operation of LCD controller/driver (LCDC) is shown below.

1. Data memory for display

   Duty drive

   Set a value to the data memory for display (VRAM) in advance.

   Static drive

   Set a value to the data memory for display (LDR0, LDR1) in advance.

2. Write necessary settings to each register.

3. Output pin

   Duty drive

   When a clock for frame cycle generation oscillates, waveforms that drive the LCD are output to the common/segment output pins (COM0 to COM3, SEG0 to SEG31).

   The content of the VRAM is read automatically in synchronization with the timing of the common signal and output from the segment output pin.

   (When the bit is set to "1", a selection waveform is output from the segment output pin.

   When the bit is set to "0", a non-selection waveform is output from the segment output pin.)

   Non-selection level waveforms are output from the COM2 and COM3 pins in 1/2 duty display mode and from the COM3 pin in 1/3 duty display mode.

   Static drive

   When a clock for frame cycle generation oscillates, waveforms that drive the LCD are output to the common/segment output pins (ST0 to ST8).

4. Output waveform

   Duty drive

   Output waveforms drive in 2 frames alternating waveform according to the duty setting.

   Static drive

   Output waveforms drive in 1 frame alternating waveform.

5. Operation in the watch mode

   Duty drive

   For operation enable (LCEN = "1"), the LCD is displayed.

   Static drive

   For operation enable (LCSEN = "1"), the LCD is displayed.

6. Blanking function

   Duty drive

   Light of the LCD can be turned off with selection of non-display (BK = "1") of blanking.

   However, non-selection waveforms are output.

7. LCD stop state

Duty drive

When display operation of the LCD is stopped (MS[1:0] = "00"), both the common/segment output pins become "L" level.

Static drive

When display operation of the LCD is stopped (LCS[3:0] = "0000"), both the common/segment output pins become "L" level.

When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1:ST[7:0] = "00000000" without change of the static drive selection port (LCS[3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

35.5.2 1/2 Duty Output Waveform

1/2 duty output waveform is shown below.

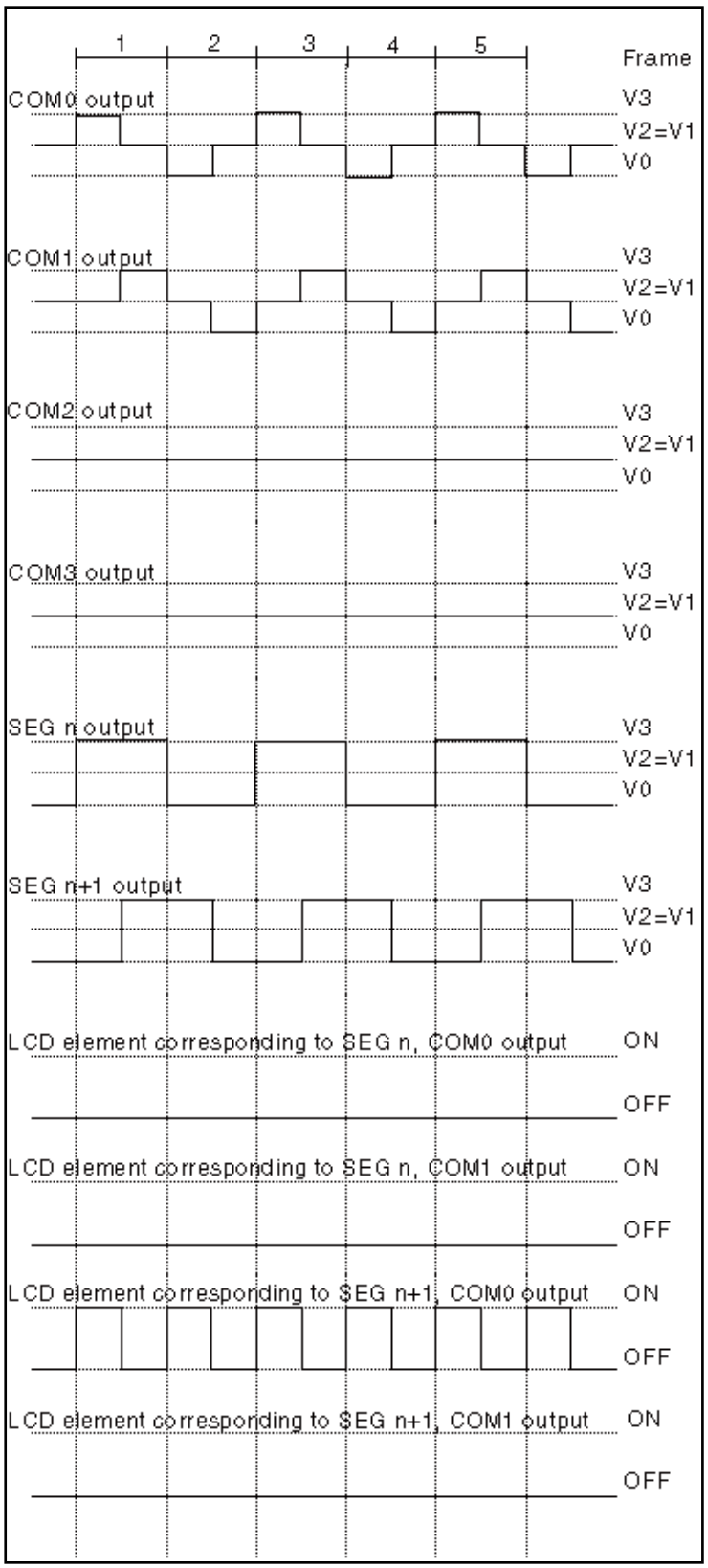
Only COM0 output and COM1 output are used for LCD display. COM2 output and COM3 output are not used.

1/2 bias output waveform case

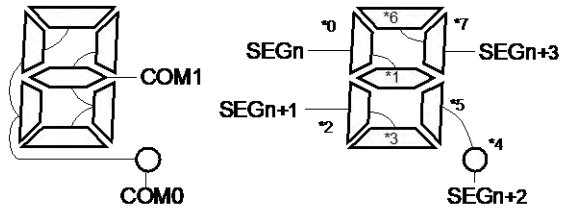
Liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

Example of Content of Data Memory for Display

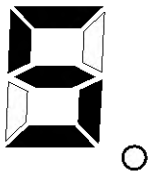
Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n Output	-	-	0	0
SEG n+1 Output	-	-	0	1


1
2
3
4
5
Frame
COM0 output
COM1 output
COM2 output
COM3 output
SEG n output
SEG n+1 output
V3
V2=V1
V0
LCD element corresponding to SEG n, COM0 output
LCD element corresponding to SEG n, COM1 output
LCD element corresponding to SEG n+1, COM0 output
LCD element corresponding to SEG n+1, COM1 output
ON
OFF

- LCD panel connection case and display data case (1/2 duty drive method)



Example for displaying 5:

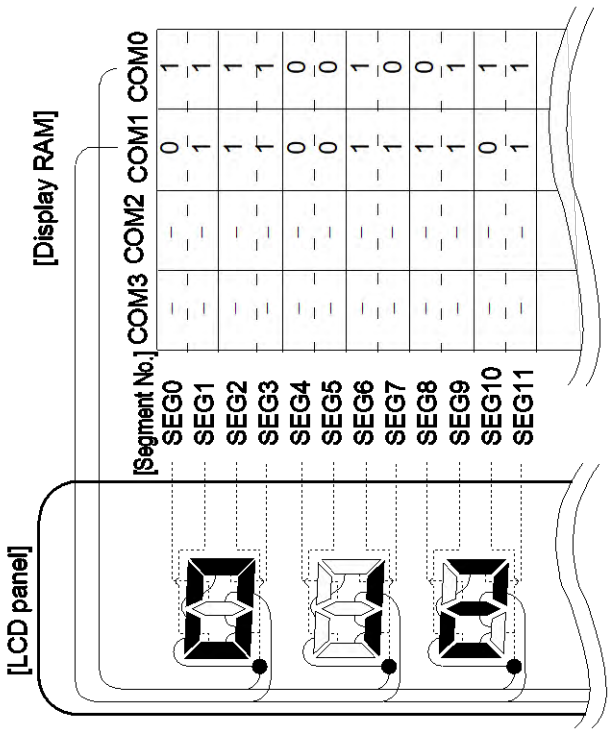


Address	COM3	COM2	COM1	COM0	
nH	bit3	bit2	bit1*1	bit0*0	SEGn
	bit7	bit6	bit5*3	bit4*2	SEGn+1
n+1H	bit3	bit2	bit1*5	bit0*4	SEGn+2
	bit7	bit6	bit5*7	bit4*6	SEGn+3

COM3	COM2	COM1	COM0	
–	–	1	1	SEG0
–	–	1	0	SEG1
–	–	1	0	SEG2
–	–	0	1	SEG3

*0 to *7 : Indicates correspondence with display RAM
bit2, 3, 6, 7 are unused

0 : OFF
1 : ON



LCD display	Example of data correspondence from 0 to 9							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	–	–	1	1	–	–	0	1
1	–	–	1	1	–	–	1	1
2	–	–	0	0	–	–	0	0
	–	–	1	0	–	–	1	1
3	–	–	1	1	–	–	1	0
	–		1	1			0	1
4	–	–	1	0	–	–	1	0
	–	–	1	1	–	–	1	1
5	–	–	0	0	–	–	1	1
	–	–	1	0	–	–	1	1
6	–	–	1	0	–	–	1	1
	–	–	0	1	–	–	1	1
7	–	–	1	1	–	–	1	1
	–	–	0	1	–	–	1	1
8	–	–	0	0	–	–	0	1
	–	–	1	1	–	–	1	1
9	–	–	1	1	–	–	1	1
	–	–	1	1	–	–	1	1
8.	–	–	1	0	–	–	1	1
	–	–	1	1	–	–	1	1

35.5.3 1/3 Duty Output Waveform

1/3 duty output waveform is shown below.

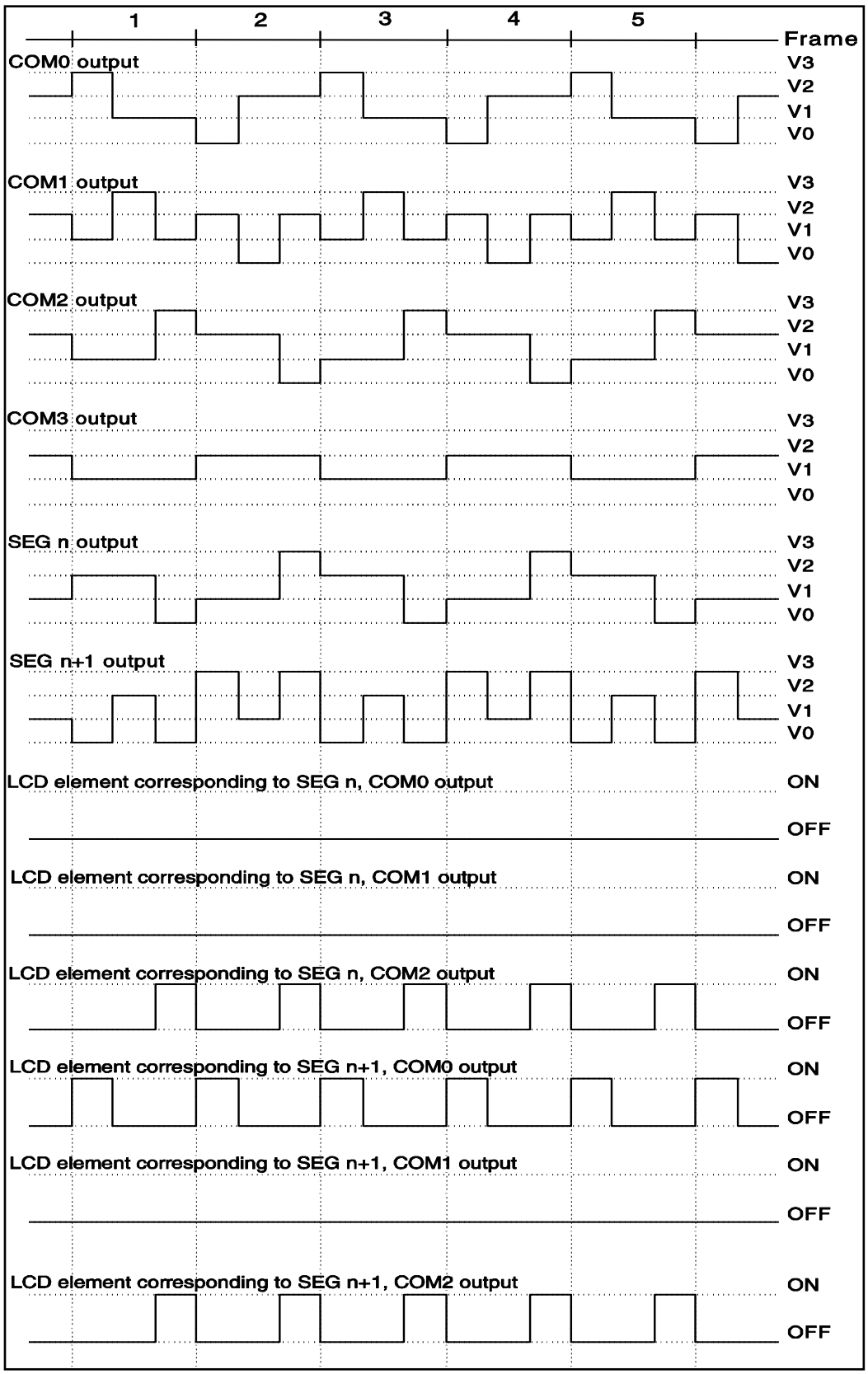
For 1/3 duty output mode, COM0 output, COM1 output, and COM2 output are used for LCD display. COM3 output is not used.

1/3 bias output waveform case

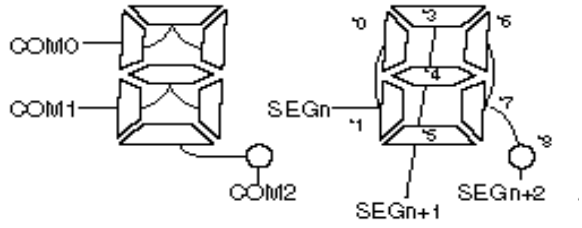
Liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

Example of Content of Data Memory for Display

Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n Output	-	1	0	0
SEG n+1 Output	-	1	0	1


1
2
3
4
5
Frame
COM0 output
COM1 output
COM2 output
COM3 output
SEG n output
SEG n+1 output
V3
V2
V1
V0
LCD element corresponding to SEG n, COM0 output
LCD element corresponding to SEG n, COM1 output
LCD element corresponding to SEG n, COM2 output
LCD element corresponding to SEG n+1, COM0 output
LCD element corresponding to SEG n+1, COM1 output
LCD element corresponding to SEG n+1, COM2 output
ON
OFF

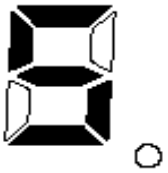
- LCD panel connection case and display data case (1/3 duty drive method)



Address	COM3	COM2	COM1	COM0	
nH	bit3	bit2*2	bit1*1	bit0*0	SEGn
	bit7	bit6*5	bit5*4	bit4*3	SEGn+1
n+1H	bit3	bit2*8	bit1*7	bit0*6	SEGn+2

*0 to *8 : Indicates correspondence with display RAM bit3, bit7 and *2 are unused

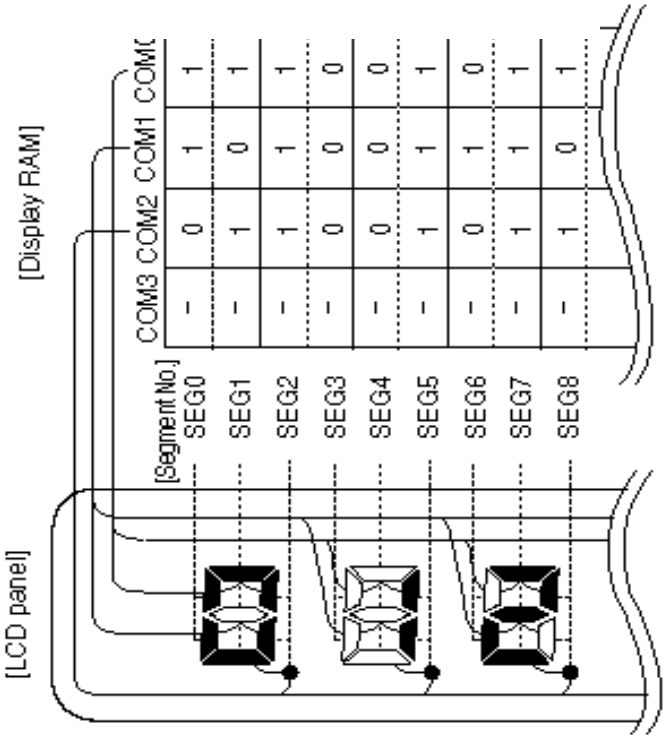
Example for displaying 5:



COM3	COM2	COM1	COM0	
–	0	0	1	SEG0
–	1	1	1	SEG1
–	0	1	0	SEG2
–	0	0	1	SEG3
–	1	1	1	SEG4
–	0	1	0	SEG5

From bit0 (SEG0–SEG2), From bit4 (SEG3–SEG5)

0 : OFF
1 : ON



LCD display	Example of display data from 0 to 9							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	–	1	0	1	–	0	1	1
	–	0	1	1	–	1	1	1
	–	1	1	1	–	1	0	1
1	–	0	0	0	–	0	0	0
	–	0	0	0	–	1	1	1
	–	1	1	1	–	0	0	0
2	–	1	1	1	–	0	1	0
	–	0	1	0	–	1	0	1
	–	1	0	1	–	1	1	1
3	–	1	1	1	–	0	0	0
	–	0	0	0	–	1	1	1
	–	1	1	1	–	1	1	1
4	–	0	1	0	–	0	0	1
	–	0	0	1	–	1	1	1
	–	1	1	1	–	0	1	0
5	–	1	1	1	–	0	0	1
	–	0	0	1	–	1	1	0
	–	1	1	0	–	1	1	1
6	–	1	1	1	–	0	1	1
	–	0	1	1	–	1	1	0
	–	1	1	0	–	1	1	1
7	–	0	0	1	–	0	0	1
	–	0	0	1	–	1	1	1
	–	1	1	1	–	0	0	1
8	–	1	1	1	–	0	1	1
	–	0	1	1	–	1	1	1
	–	1	1	1	–	1	1	1
9	–	1	1	1	–	0	0	1
	–	0	0	1	–	1	1	1
	–	1	1	1	–	1	1	1

: Data when beginning with bit4
: Data when beginning with bit0

As 1/3 duty displays two digits with three bytes, data array provides data from the first byte bit0 and data from the second byte bit4.

35.5.4 1/4 Duty Output Waveform

1/4 duty output waveform is shown below.

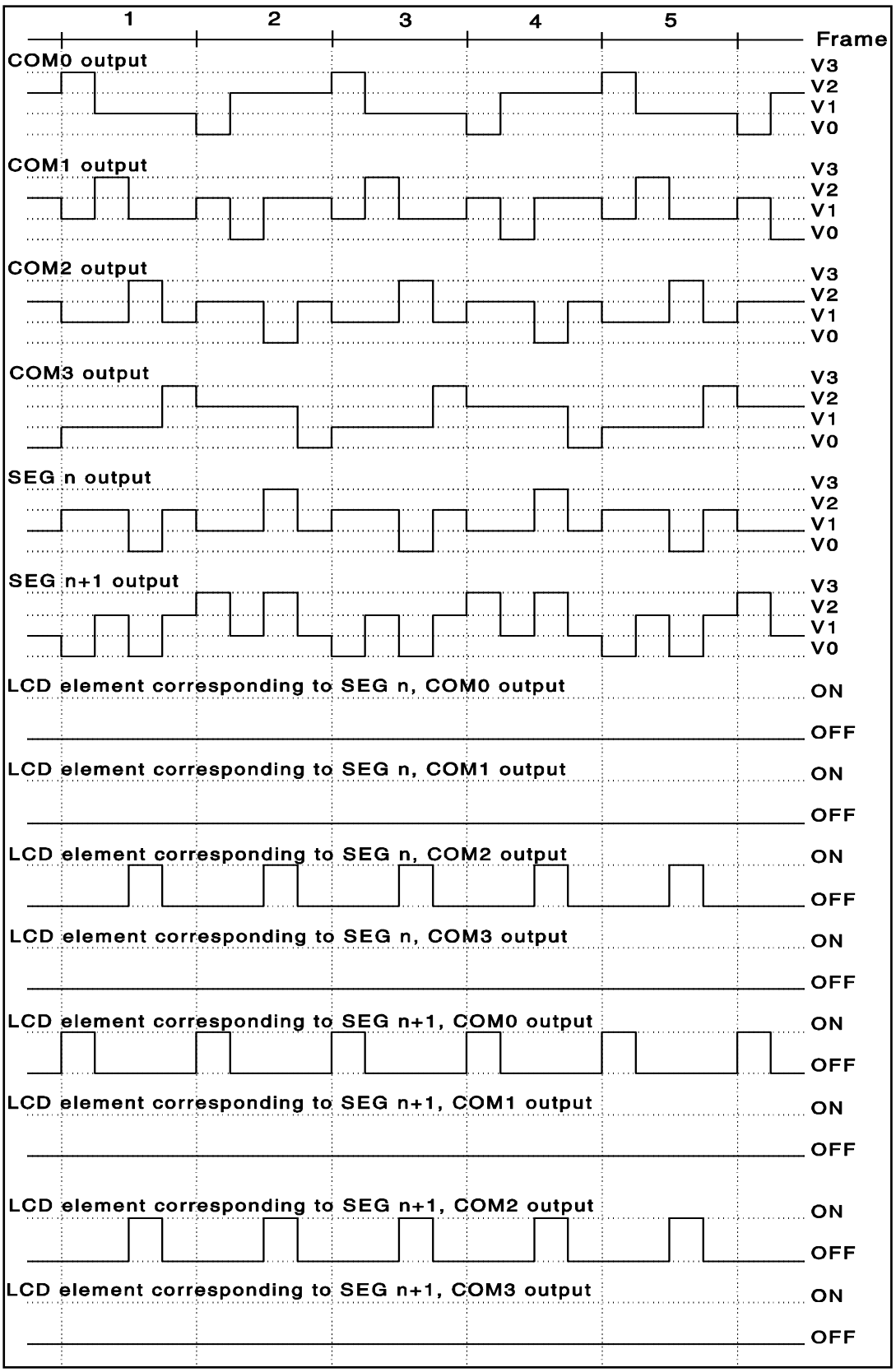
For 1/4 duty output mode, all of COM0 output, COM1 output, COM2 output, and COM3 output are used for LCD display.

1/4 bias output waveform case

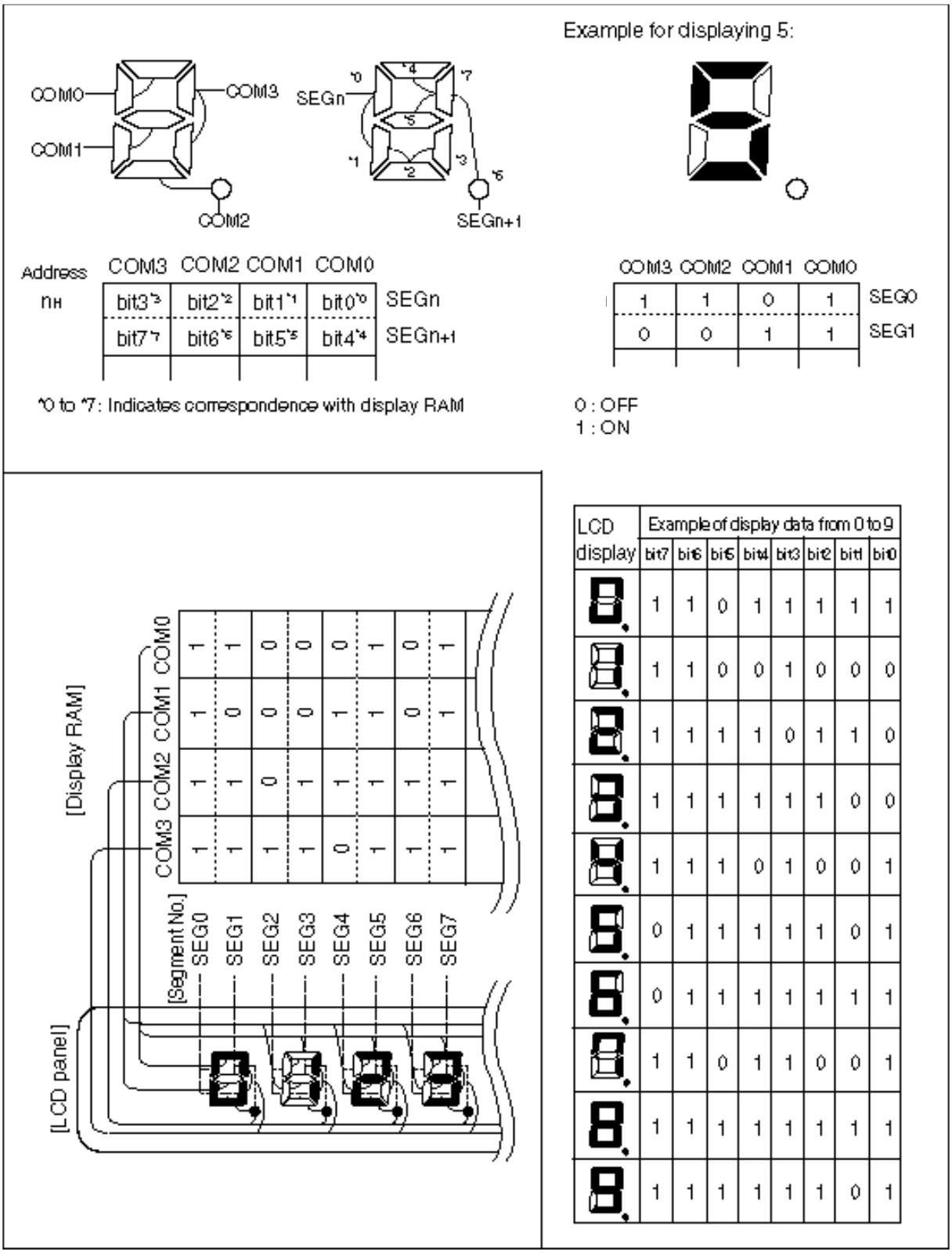
For LCD, liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

Example of Content of Data Memory for Display

Segment	Content of Data Memory for Display			
	COM3 Output	COM2 Output	COM1 Output	COM0 Output
SEG n Output	0	1	0	0
SEG n+1 Output	0	1	0	1


1
2
3
4
5
Frame
COM0 output
COM1 output
COM2 output
COM3 output
SEG n output
SEG n+1 output
V3
V2
V1
V0
LCD element corresponding to SEG n, COM0 output
LCD element corresponding to SEG n, COM1 output
LCD element corresponding to SEG n, COM2 output
LCD element corresponding to SEG n, COM3 output
LCD element corresponding to SEG n+1, COM0 output
LCD element corresponding to SEG n+1, COM1 output
LCD element corresponding to SEG n+1, COM2 output
LCD element corresponding to SEG n+1, COM3 output
ON
OFF

- LCD panel connection case and display data case (1/4 duty drive method)



35.5.5 Static Drive Output Waveform

Static drive output waveform is shown below.

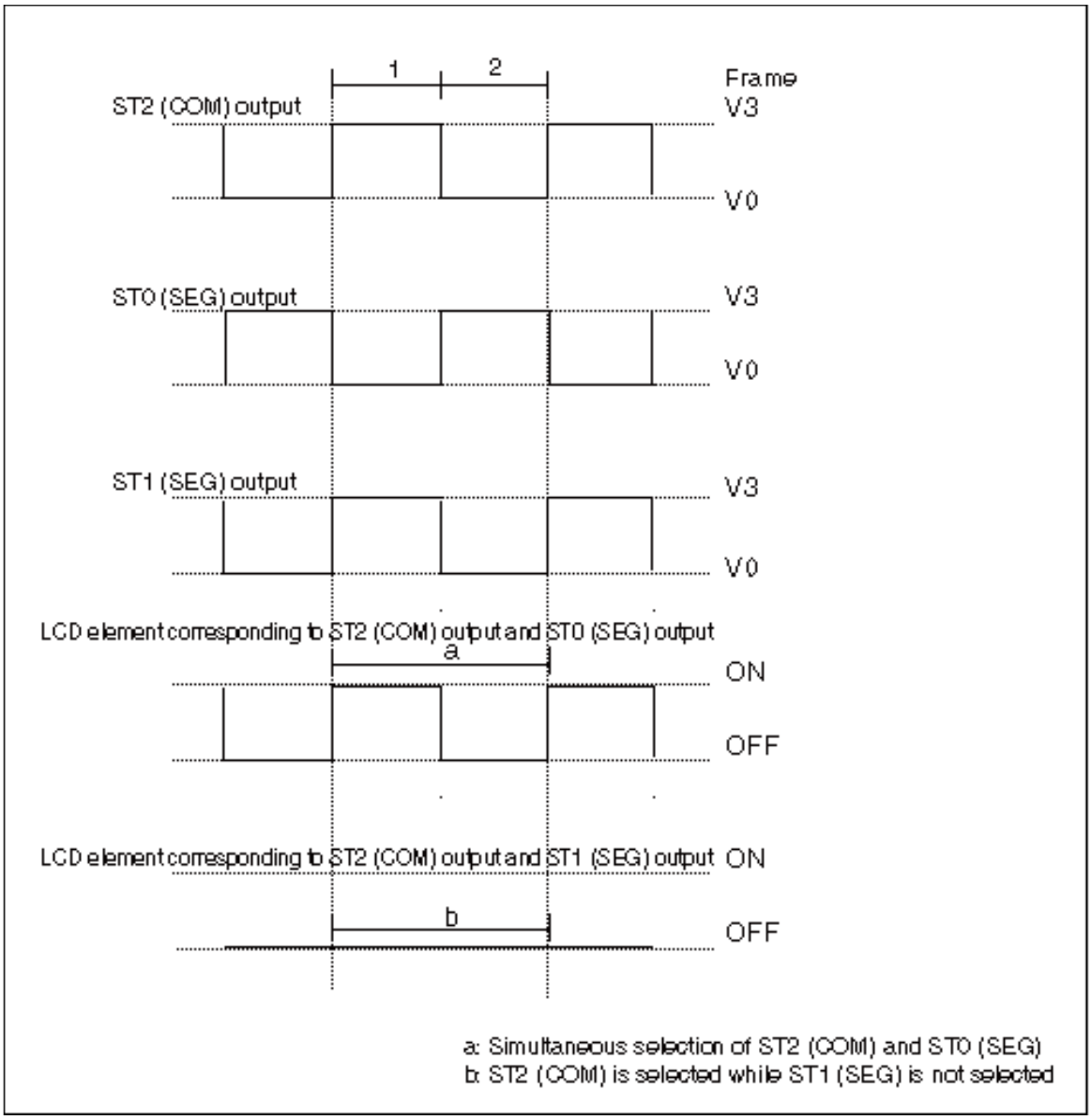
For the static drive output mode, the common/segment output pins (ST0 to ST8) are used for LCD display.

Static drive output waveform case

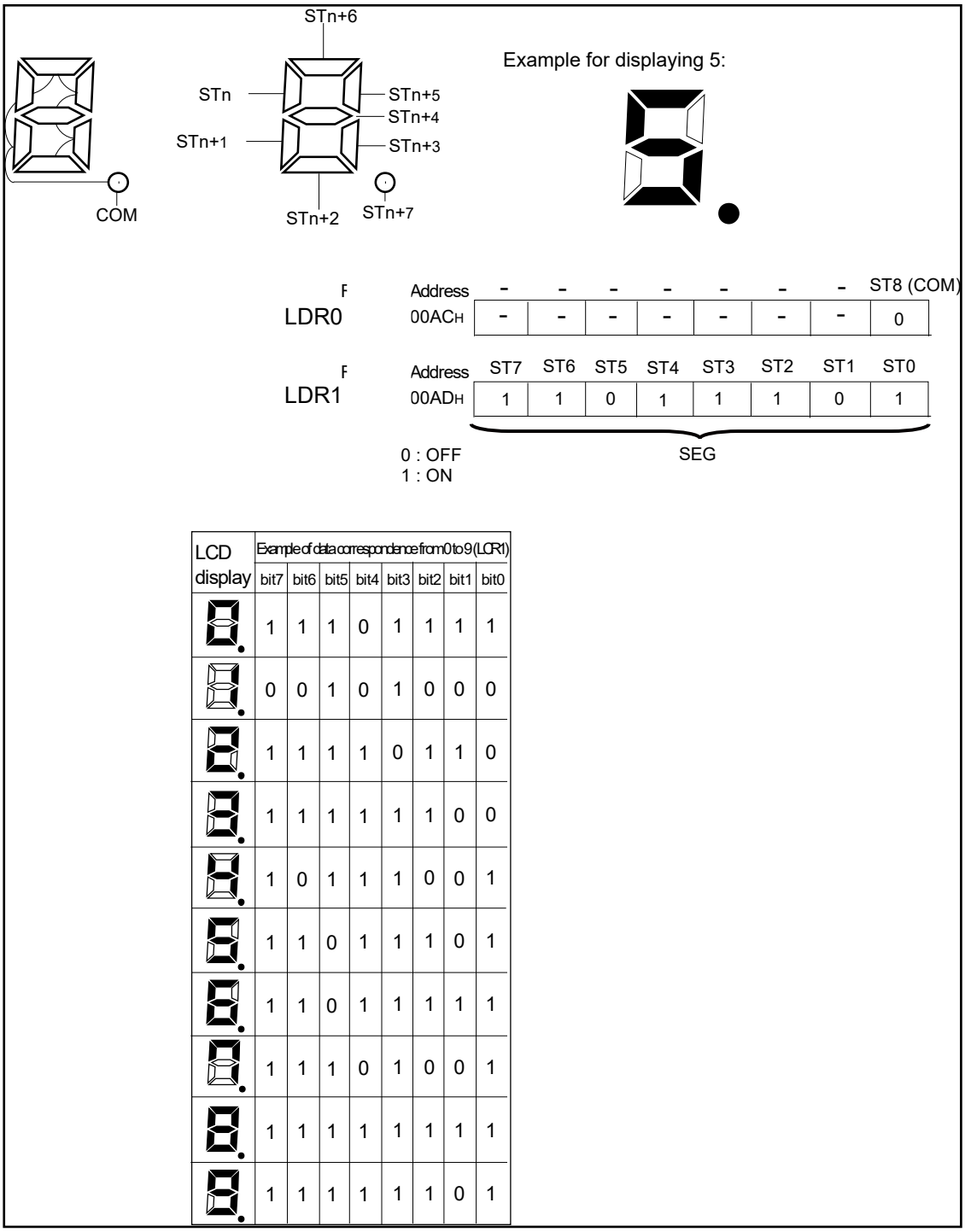
For LCD, liquid crystal elements that have maximum difference in potential between the common output and segment output light up.

Example of Content of Data Memory for Display (Example of static SEG output to the output pins ST0 to ST1 and static COM output to ST2)

Content of Data Memory for Display (LDR0, LDR1)		
ST2 output LDR0[2]	**ST1 output LDR0[1]**	**ST0 output LDR0[0]**
0	0	1


1
2
Frame
ST2 (COM) output
V3
V0
ST0 (SEG) output
V3
V0
ST1 (SEG) output
V3
V0
LCD element corresponding to ST2 (COM) output and ST0 (SEG) output
a
ON
OFF
LCD element corresponding to ST2 (COM) output and ST1 (SEG) output
ON
b
OFF
a: Simultaneous selection of ST2 (COM) and ST0 (SEG)
b: ST2 (COM) is selected while ST1 (SEG) is not selected

- LCD panel connection case and display data case (Static drive method)



LCD display	Example of data correspondence from 0 to 9 (LCR1)							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	1	1	1	0	1	1	1	1
	0	0	1	0	1	0	0	0
	1	1	1	1	0	1	1	0
	1	1	1	1	1	1	0	0
	1	0	1	1	1	0	0	1
	1	1	0	1	1	1	0	1
	1	1	0	1	1	1	1	1
	1	1	1	0	1	0	0	1
	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	0	1

35.6 Setting

This section shows the setting of the LCD controller.

Duty drive

Setting Necessary for Use of LCD

Setting	Set Register	Set Method
Preliminary setting	LCD control register 1 (LCR1) Common pin switching register (LCDCMR)	See 35.4.3 35.4.4
Division resistor setting	LCD control register 0 (LCR0)	See 35.7.9 35.7.11
Port setting	Port function register (PFR)	See 35.7.1
Display data setting	Data memory for display (VRAM)	See 35.7.2
Frame cycle setting	LCD control register 0 (LCR0)	See 35.7.3
Duty selection (activation)		See 35.7.5
Display selection		See 35.7.7

Setting Necessary for Cancellation of LCD Display

Setting	Set Register	Set Method
Non-display selection	LCD control register 0 (LCR0)	See 35.7.7

Setting Necessary for LCD Operation Stop

Setting	Set Register	Set Method
LCD operation stop	LCD control register 0 (LCR0)	See 35.7.6

Setting Necessary for LCD Display during Watch Mode

Setting	Set Register	Set Method
Display selection during watch mode	LCD control register 0 (LCR0)	See 35.7.8
Transition to watch mode	See chapter of "Power Consumption Control".	-

Static drive

Setting Necessary for Use of LCD

Setting	Set Register	Set Method
Preliminary setting	LCD control register 1 (LCR1) Common pin switching register (LCDCMR)	See 35.4.3 35.4.4
Port setting	Set the pin to peripheral output. See chapter of "I/O Ports".	
	LCD control register (LCRS)	See 35.4.5
Display data setting	Data memory for display (LDR0,LDR1)	See 35.4.6
Frame cycle setting	LCD control register (LCRS)	See 35.7.3

Setting Necessary for LCD Operation Stop

Setting	Set Register	Set Method
LCD operation stop	LCD control register (LCRS)	See 35.7.6

Setting Necessary for LCD Display during Watch Mode

Setting	Set Register	Set Method
Display selection during watch mode	LCD control register (LCRS)	See 35.7.8
Transition to watch mode	See chapter of "Power Consumption Control".	-

35.7 Q&A

This section shows Q&A of the LCD controller.

35.7.1 Set Pins to COM Output Pins or SEG Output Pins
35.7.2 Set VRAM
35.7.3 Set the Frame Cycle
35.7.4 Set the Bias
35.7.5 Set the Duty
35.7.6 Control the LCD Operation Start/Stop
35.7.7 Execute/Cancel the Display
35.7.8 Display During the Watch Mode
35.7.9 Select Either Internal or External for the Division Resistor
35.7.10 Select Pin of V3 Voltage
35.7.11 Select Either Internal or External for the Division Resistor
35.7.12 Adjust the Brightness When the Internal Division Resistor is Used
35.7.13 Block the Current with the External Division Resistor When the LCD Stops
35.7.14 Display/Non-display the LCD with Static Drive (ST0 to ST8)

35.7.1 Set Pins to COM Output Pins or SEG Output Pins

This section shows how to set pins to COM output pins or SEG output pins.

COM output and SEG output setting.

With software setting, ports can be switched to COM, SEG output.

Set pins for COM output and SEG output to peripheral output. See chapter of "I/O PORTS" for pin setting.

Duty drive

Pin	Pin Setting Method	PFR Register/Bit Number
V0	Set pins to LCDC V0/V1/V2/V3 (LCDC reference power input). See chapter of "I/O Ports".	See "Chapter: I/O Ports".
V1		
V2		
V3		
COM0	Set pins to peripheral output. See chapter of "I/O Ports".	
COM1		
COM2		
COM3		
SEG0		
SEG1		
SEG2		
SEG3		
SEG4		
SEG5		
SEG6		
SEG7		
SEG8		
SEG9		
SEG10		
SEG11		
SEG12		
SEG13		
SEG14		
SEG15		
SEG16		
SEG17		

Pin	Pin Setting Method	PFR Register/Bit Number
SEG18	Set pins to peripheral output. See chapter of "I/O Ports".	See "Chapter: I/O Ports".
SEG19		
SEG20		
SEG21		
SEG22		
SEG23/ST0		
SEG24/ST1		
SEG25/ST2		
SEG26/ST3		
SEG27/ST4		
SEG28/ST5		
SEG29/ST6		
SEG30/ST7		
SEG31/ST8		

Static drive

COM output and SEG output setting.

With software setting, ports can be switched to COM, SEG output.

Set pins for static drive to peripheral output. See chapter of "I/O Ports" for pin setting.

Pin	Pin Setting Method	PFR Register/Bit Number
V0	Set pins to LCDC V0/V3 (LCDC reference power input). See chapter of "I/O Ports".	See "Chapter: I/O Ports".
V3		
SEG23/ST0	Set pins to peripheral output. See chapter of "I/O Ports".	
SEG24/ST1		
SEG25/ST2		
SEG26/ST3		
SEG27/ST4		
SEG28/ST5		
SEG29/ST6		
SEG30/ST7		
SEG31/ST8		

35.7.2 Set VRAM

This section shows how to set VRAM.

Duty drive

The matrix of pins and bit locations of VRAM (n) is shown below. (n = 0 to 15)

1/2 Duty

Pin	COM1	COM0
SEG 2n	bit1	bit0
SEG 2n+1	bit5	bit4

1/3 Duty

Pin	COM2	COM1	COM0
SEG 2n	bit2	bit1	bit0
SEG 2n+1	bit6	bit5	bit4

1/4 Duty

Pin	COM3	COM2	COM1	COM0
SEG 2n	bit3	bit2	bit1	bit0
SEG 2n+1	bit7	bit6	bit5	bit4

(Non-selection waveforms are output from irrelevant pins.)

Example: 1/4 duty

When the bit6 of VRAMn is set to "1", the selection waveform is output from the SEGn+1 of COM2.

For bits with "0"setting, non-selection waveforms are output to pins.

35.7.3 Set the Frame Cycle

This section shows how to set the frame cycle.

Duty drive

The frame cycle can be set with the frame cycle bits (LCR0:FP[1:0]).The following settings are available.

Frame Cycle (When a Main Clock is Selected)	Selection Value
	Frame Cycle Bits (FP[1:0])
$(2^{11} \times N)$/ Main clock frequency	Set "00".
$(2^{12} \times N)$/ Main clock frequency	Set "01".
$(2^{13} \times N)$/ Main clock frequency	Set "10".
$(2^{14} \times N)$/ Main clock frequency	Set "11".

N (Time division number) = Value of MS[1:0] + "1"

Frame Ccycle (When a Sub Clock is Selected)	Selection Value
	Frame Cycle Bits (FP[1:0])
$(2^{3} \times N)$/ Sub clock frequency	Set "00".
$(2^{4} \times N)$/ Sub clock frequency	Set "01".
$(2^{5} \times N)$/ Sub clock frequency	Set "10".
$(2^{6} \times N)$/ Sub clock frequency	Set "11".

N (Time division number) = Value of MS[1:0] + "1"

Static drive

The frame cycle can be set with the frame cycle bits (LCRS:FPS[1:0]).The following settings are available.

Frame Cycle	Selection Value
	Frame Cycle Bits (FPS[1:0])
$(2^{11} \times 4)$/Main clock (F_{CL})	Set "00".
$(2^{12} \times 4)$/ Main clock (F_{CL})	Set "01".
$(2^{13} \times 4)$/ Main clock (F_{CL})	Set "10".
$(2^{14} \times 4)$/ Main clock (F_{CL})	Set "11".

35.7.4 Set the Bias

This section shows how to set the bias.

Duty drive

Set the bias selection bit (LCDCMR:DTCH).

Bias	Bias Selection Bit (DTCH)
To set 1/3 bias	Set "0".
To set 1/2 bias	Set "1".

35.7.5 Set the Duty

This section shows how to set the duty.

Duty drive

Set the display mode selection bits (LCR0:MS[1:0]).

Control Detail	Display Mode Selection Bits (MS[1:0])	N (Time Division Number)
LCD operation stop (Pin output "L")	Set "00".	-
To set 1/2 duty output mode	Set "01".	2
To set 1/3 duty output mode	Set "10".	3
To set 1/4 duty output mode	Set "11".	4

The display mode selection bits are also the control bit for operation start/stop.

35.7.6 Control the LCD Operation Start/Stop

This section shows how to control the LCD operation start/stop.

Duty drive

Operation start/stop can be controlled with the display mode selection bits (LCR0:MS[1:0]).

See "35.7.5 Set the Duty".

Static drive

Operation start/stop can be controlled with the display mode selection bits (LCRS:LCS[3:0]).

35.7.7 Execute/Cancel the Display

This section shows how to execute/cancel the LCD display.

Duty drive

The two methods below are available.

- Setting of the Blanking Selection Bit (LCR0:BK)

Control Detail	Blanking Selection Bit (BK)
To execute LCD display	Set "0".
To cancel LCD display (A non-selection waveform is output to a segment pin.)	Set "1".

- Cancellation of Display with Operation Stop by the Display Mode Selection Bits (LCR0:MS[1:0]).

Control Detail	Display Mode Selection Bits (MS[1:0])
LCD operation stop ("L" output from the common pin and segment pin.)	Set "00".

35.7.8 Display During the Watch Mode

This section shows how to display during the watch mode.

Duty drive

Set the watch mode operation enable bit (LCR0:LCEN).

Control Detail	Watch Mode Operation Enable Bit (LCEN)
For no LCD display in the watch mode	Set "0".
For LCD display in the watch mode	Set "1".

Static drive

Set the watch mode operation enable bit (LCRS:LCSEN).

Control detail	Watch mode operation enable bit (LCSEN)
For no LCD display in the watch mode	Set "0".
For LCD display in the watch mode	Set "1".

35.7.9 Select Either Internal or External for the Division Resistor

This section shows how to select either internal or external for the division resistor.

Duty drive

Set the LCD drive power control bit (LCR0:VSEL).

Control Detail	LCD Drive Power Control Bit (VSEL)
For use of the external division resistor (The internal division resistor is disconnected.)	Set "0".
For use of the internal division resistor (The internal division resistor is connected.)	Set "1".

35.7.10 Select Pin of V_3 Voltage

This section shows how to select the pin of V_3 voltage.

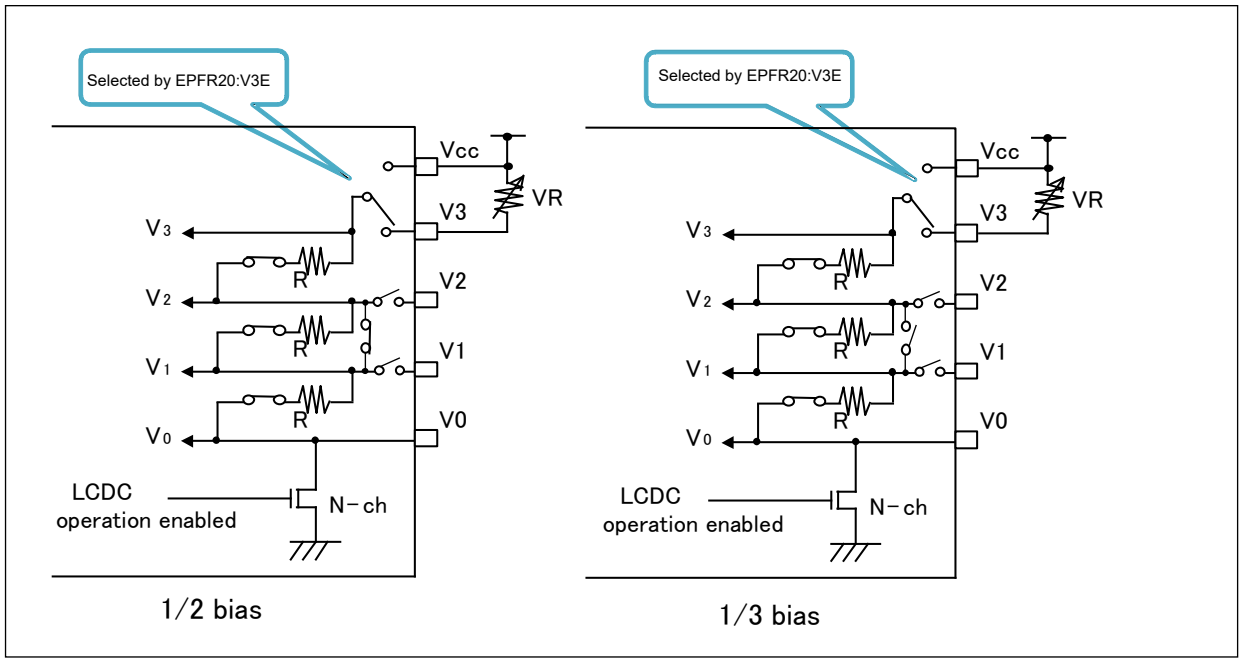
Set the EPFR20:V3E bit. See chapter of "I/O PORTS" for details.

35.7.11 Select Either Internal or External for the Division Resistor

This section shows how to select either internal or external for the division resistor.

- When the Internal Division Resistor is Selected

  Even when the internal division resistor is used if the V3 pin is used as V_3 voltage, the external resistor must be connected between Vcc and V3.



- When the External Division Resistor is Selected

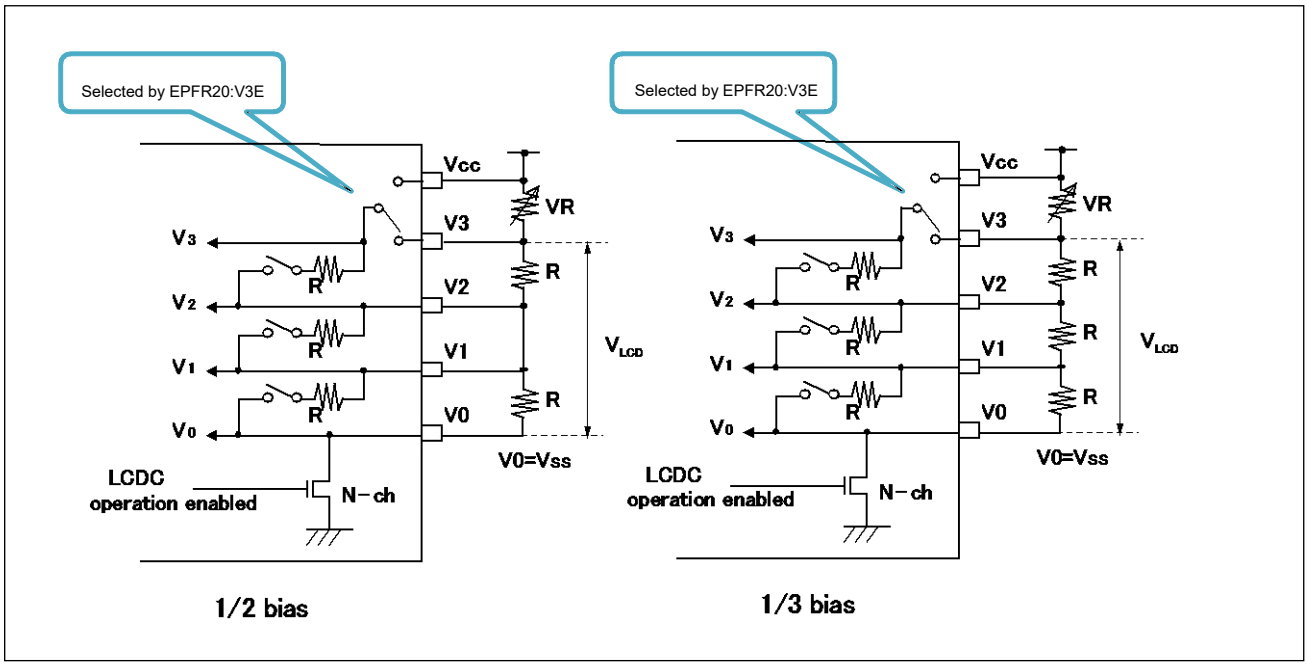
  Voltage for the LCD drive is set with the external division resistor connected to the power pins for LCD drive (V0 to V3).

  Setting of LCD Drive Voltage

	V3	V2	V1	V0
1/2 bias	VLCD	1/2VLCD	1/2VLCD	VSS
1/3 bias	VLCD	2/3VLCD	1/3VLCD	VSS

V_0 to V_3: Voltage of V0 to V3 pins

V_{LCD}: Operation voltage of the LCD



To avoid the effect of the internal division resistor, the LCD drive power control bit (LCR0:VSEL) must be set to "0" to disconnect the internal division resistor.

35.7.12 Adjust the Brightness When the Internal Division Resistor is Used

This section shows how to adjust the brightness when the internal division resistor is used.

If desired brightness cannot be obtained with the use of the internal division resistor, adjust the voltage V3 by putting a variable resister (VR) between the outer terminals Vcc and V3.

35.7.13 Block the Current with the External Division Resistor When the LCD Stops

This section shows how to block the current with the external division resistor when the LCD stops.

The V0 pin is connected to the Vss (GND) via a transistor internally. Therefore, when the external division resistor is used, by connecting the Vss side of the external division resistor to the V0 pin, the current which flows on the LCD controller which is stopped can be blocked. Block the current with the display mode selection bits (MS[1:0] = "00").

35.7.14 Display/Non-display the LCD with Static Drive (ST0 to ST8)

This section shows how to display/non-display the LCD with static drive (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched to non-display after reset, set LDR0:ST8 = "0" and LDR1:ST[7:0] = "00000000", and set the static drive selection port (LCS[3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

When the LCD with static drive (ST0 to ST8) is switched from non-display to display, set LDR0:ST8 and LDR1:ST[7:0] without change of (LCS[3:0]).

When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1:ST[7:0] = "00000000" without change of the static drive selection port (LCS[3:0]).

The same potential pulses are output from the static drive pins (ST0 to ST8).

35.8 Sample Program

This section shows sample program.

Setting procedure 1

With the 1/2 duty drive method, make the LCD display a four-digit number, "0 1 2 3".

Initial setting (LCDC)

<Initial setting>

1. - Port

	Register name
COM, SEG output setting for the port	See Chapter of "I/O Ports".

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM07

3. - Setting of control register

	Register name
Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

<Others>

(Note)

Clock-related setting and setting of _set_il (numerical value) in advance are required. See Chapter of "Clock" and Chapter of "Interrupt Control (Interrupt Controller)" for details.

Program 1

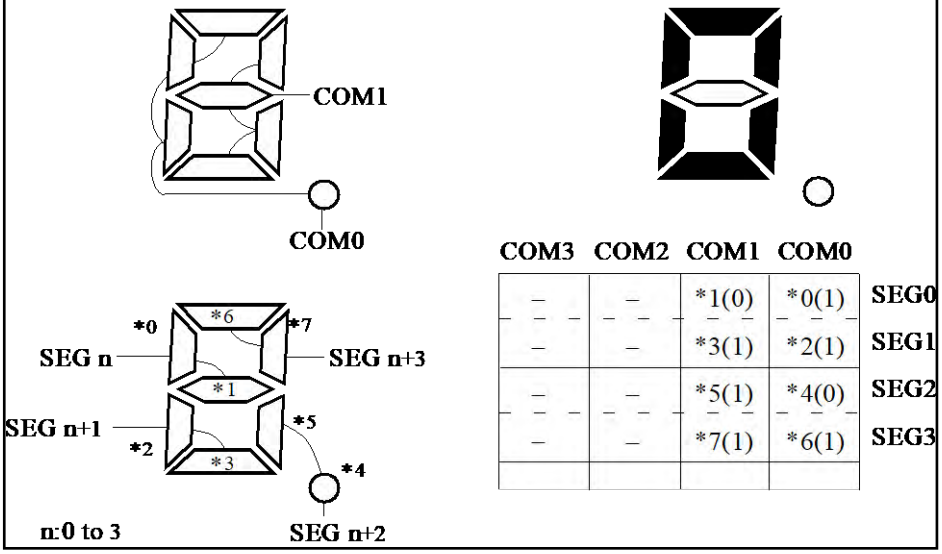
```
void LCD_sample_1(void)

{
        LCDC_initial();
}
void LCDC_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output.  */

        IO_VRAM00 = 0x31;
        IO_VRAM01 = 0x32;
        IO_VRAM02 = 0x00;
        IO_VRAM03 = 0x22;
        IO_VRAM04 = 0x32;
        IO_VRAM05 = 0x30;
        IO_VRAM06 = 0x22;
        IO_VRAM07 = 0x32;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x8F;         /* 1/2 bias */
        IO_LCR0.byte = 0x04;            /* Setting value =0000_0100 */
                                        /* bit7 = 0 Clock selection  Main clock */
                                        /* bit6 = 0 LCEN  Display stopped with watch mode  */
                                        /* bit5 = 0 VSEL Internal division resistor disconnected */
                                        /* bit4 = 0 BK Blanking selection bit */
                                        /* bit3-2 = 01 MS[1:0]       1/2 duty mode */
                                        /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 0 and its value (in parentheses)



COM3	COM2	COM1	COM0	
–	–	*1(0)	*0(1)	SEG0
–	–	*3(1)	*2(1)	SEG1
–	–	*5(1)	*4(0)	SEG2
–	–	*7(1)	*6(1)	SEG3

Setting procedure 2

With the1/3 duty drive method, make the LCD display a four-digit number , "0 1 2 3 ".

Initial setting (LCDC)

<Initial setting>

1. - Port

	Register name
COM, SEG output setting for the port	See Chapter of " I/O Ports".

2. - Setting of VRAM

	Register name
Setting of VRAM	VRAM00-VRAM05

3. - Setting of control register

	Register name
Fixed value	LCR1
Bias setting	LCDCMR
Setting of control register	LCR0
	. LCEN
	. VSEL
	. BK
	. MS[1:0]
	. FP[1:0]

<Others>
(Note)

Clock-related setting and setting of _set_il (numerical value) in advance are required. See Chapter of "Clock" and Chapter of "Interrupt Control (Interrupt Controller)" for details.

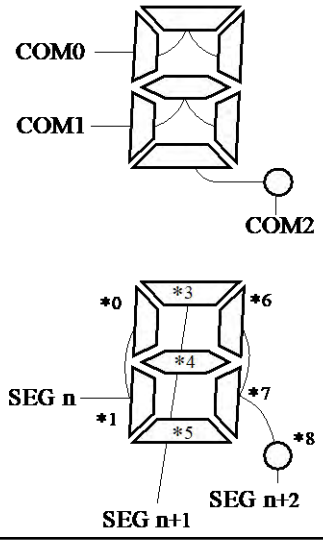
Program 2

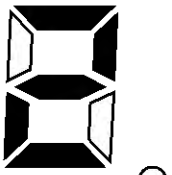
```
void LCD_sample_2(void)
{
        LCDC_initial();
}
void LCDC_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output.  */

        IO_VRAM00 = 0x53;
        IO_VRAM01 = 0x03;
        IO_VRAM02 = 0x30;
        IO_VRAM03 = 0x72;
        IO_VRAM04 = 0x01;
        IO_VRAM05 = 0x37;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x0F;         /* 1/3 bias */
        IO_LCR0.byte = 0x08;            /* Setting value =0000_1000 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCEN Display stopped with watch mode */
                                        /* bit5 = 0 VSEL Internal division resistor disconnected */
                                        /* bit4 = 0 BK Blanking selection bit */
                                        /* bit3-2 = 10 MS[1:0] 1/3 duty  mode */
                                        /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 2 and its value (in parentheses)





COM3	COM2	COM1	COM0		
–	(-)	*1(1)	*0(0)	SEG0	First digit
–	*5(1)	*4(1)	*3(1)	SEG1	
–	*8(0)	*7(0)	*6(1)	SEG2	
–	–	1	0	SEG3	Second digit
–	1	1	1	SEG4	
–	0	0	1	SEG5	

Setting procedure 3

With the 1/4 duty drive method, make the LCD display a four-digit number, "0 1 2 3 ".

Initial setting (LCD)

<Initial setting>

1. - PORT

Register name
See Chapter of "I/O Ports".

2. - Setting of VRAM

Register name
VRAM00-VRAM03

3. - Setting of control register

registername . bit name
LCR1
LCDCMR
LCR0
. LCEN
. VSEL
. BK
. MS[1:0]
. FP[1:0]

<Others>
(Note)

Clock-related setting and setting of _set_il (numerical value) in advance are required. See Chapter of "Clock" and Chapter of "Interrupt Control (Interrupt Controller)" for details.

Program 3

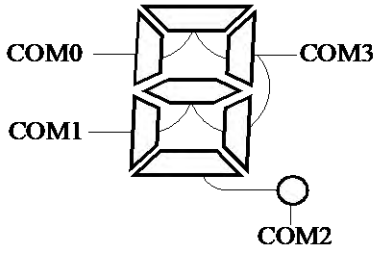
```
void LCD_sample_3(void)
{
        LCD_initial();
}

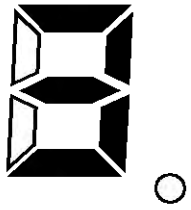
void lcdc_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output. */

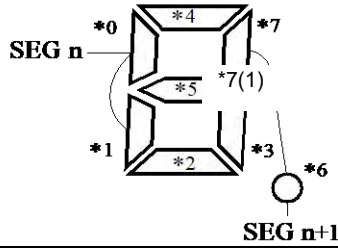
        IO_VRAM00 = 0x9F;
        IO_VRAM01 = 0x88;
        IO_VRAM02 = 0xB6;
        IO_VRAM03 = 0xBC;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x0F;         /* 1/3 bias */
        IO_LCR0.byte = 0x0C;            /* Setting value =0000_1100 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCEN  Display stopped with watch mode */
                                        /* bit5 = 0 VSEL Internal division resistor disconnected */
                                        /* bit4 = 0 BK Blanking selection bit */
                                        /* bit3-2 = 11 MS[1:0] 1/4 duty mode */
                                        /* bit1-0 = 00 FP[1:0] */
}
```

Example for displaying 3 and its value (in parentheses)







COM3	COM2	COM1	COM0	
*0(0)				
	*2(1)	*1(0)	*0(1)	SEG0
*7(0)	*6(0)	*5(1)	*4(1)	SEG1

Setting procedure 4

With the static drive method, make the LCD display an one-digit number.

Initial setting (LCD)

<Initial setting>

1. - PORT

COM, SEG output setting for the port	Register name See Chapter of "I/O Ports".

2. - Data setting

Data setting	Register name LDR0 LDR1

3. - Setting of control register

Fixed value	Register name . bit name LCR1
Bias setting	LCDCMR
Setting of control register	LCRS
	. LCSEN
	. LCS[3:0]
	. FPS[1:0]

Program 4

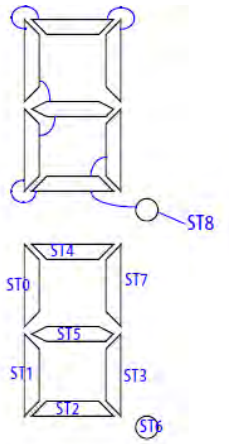
```
void LCD_sample_4(void)
{
        LCD_initial();
}

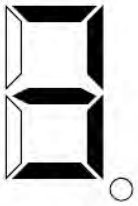
void lcdc_initial(void)
{
        PORT_SETTING_LCDC_OUT();        /* Set the LCD controller pin to */
                                        /* peripheral output.  */

        IO_LDR0.byte = 0x00;
        IO_LDR1.byte = 0xBC;

        IO_LCR1.byte = 0xFF;            /* Set to FF. */
        IO_LCDCMR. byte = 0x0F;         /* 1/3 bias */
        IO_LCRS.byte = 0x20;            /* Setting value =0010_0000 */
                                        /* bit7 = 0 Clock selection Main clock */
                                        /* bit6 = 0 LCSEN  Non-display with watch mode  */
                                        /* bit5-2 = 1000  LCS[3:0]  ST0 to ST8 */
                                        /* bit1-0 = 00 FPS[1:0]  Frame cycle setting */
}
```

Example) Values to display 3





LDR0/ LDR1 register

ST 8	ST 7	ST 6	ST 5	
0	1	0	1	
ST 4	**ST 3**	**ST 2**	**ST 1**	**ST 0**
1	1	1	0	0

35.9 Notes

This section shows notes.

- Since resistance value of the external division resisters depends on the LCD to be used, connect the resistors with a suitable value.
- Non-selection level waveforms are output from the COM2 and COM3 pins in 1/2 duty display mode and from the COM3 pin in 1/3 duty display mode.
- If the settings of the LCD drive power control (VSEL), duty selection (MS[1:0]), frame cycle selection (FP[1:0]), etc. are not proper, the LCD does not display correctly.
- When neither the LCD nor general-purpose ports are used, connect a pull-up or pull-down resistor to V3 to V0 pins.
- The static drive is enabled when any setting other than LCS[3:0] = "0000" is set. The duty drive is enabled when LCS[3:0] = "0000" is set.
- Setting of display/non-display of the LCD with static drive (ST0 to ST8)
  - ☐ When the LCD with static drive (ST0 to ST8) is switched to non-display after reset, set LDR0:ST8 = "0" and LDR1:ST[7:0] = "00000000", and set the static drive selection port (LCS[3:0]). The same potential pulses are output from the static drive pins (ST0 to ST8).
  - ☐ When the LCD with static drive (ST0 to ST8) is switched from non-display to display, set LDR0:ST8 and LDR1:ST[7:0] without change of the static drive selection port (LCS[3:0]).
    When the LCD with static drive (ST0 to ST8) is switched from display to non-display, set LDR0:ST8 = "0" and LDR1:ST[7:0] = "00000000" without change of the static drive selection port (LCS[3:0]).

  The same potential pulses are output from the static drive pins (ST0 to ST8).
- When the LCD is used (static drive, duty drive), set the following.
  - ☐ LCR1:ST[7:0] = "11111111"
  - ☐ LCDCMR[3:0] = "1111"

  Operation can be continued at the watch mode(w. power shutdown) as similar as at the watch mode(w/o. power shutdown). See "35.7.8 Display During the Watch Mode" for the operation setting at the watch mode. Moreover, the LCD controller is not initialized in reset by the Wake up from the watch mode (power shutdown). The operation continues.

36. External Bus Interface

This chapter explains the external bus interface.

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 external bus interface.

This chapter will explain each of the functions of the external bus interface.

36.2 Features

This section explains the features of the external bus interface.

- Address up to 22 bits long (4MB space) can be output.(The address space can be extended to 8MB by treating the lowermost bit as fixed and extending the upper bit by 1 bit, depending on setting the ACR0 to ACR3:ADTY bit.)
- Supports split address/data bus
- Able to connect to asynchronous memory
- Supports multiplexed address/data bus
- Four independent chip select areas (called CS areas below) can be configured, and chip select output corresponding to each area can be performed
- The size of each CS area can be selected from 16 options in the range of 64 KB to 2 GB
- Each CS area can be set to an arbitrary position within the external bus area
- The following functions can be set independently for each CS area
  - ☐ Enabled or disabled
  - ☐ Data bus width (8-bit or 16-bit)
  - ☐ Write prohibited (read-only) setting
  - ☐ Byte order
    CS0 area: Big endian
    Not CS0 area: Supports big and little endian
  - ☐ Address shift output mode
- Bus type selectable for each CS area
- Split address/data bus
- Multiplexed address/data bus
- Type 0 (byte write strobe signal output)
- The following timings are configurable for each CS area
  - ☐ Common to read/write access
    Address --> CS signal setup cycle count
  - ☐ Address strobe signal output cycle count
  - ☐ Extend read/write bus cycle by external ready input
  - ☐ Read access
  - ☐ Read access automatic wait
  - ☐ CS signal → Read strobe signal setup cycle count
  - ☐ Read strobe signal → CS signal hold cycle count
  - ☐ Read access → Insert idle cycle between write accesses
  - ☐ Write access
  - ☐ Write access automatic wait
  - ☐ CS signal → Write strobe signal setup cycle count
  - ☐ Write strobe signal → CS signal hold cycle count
- Insert write recovery cycles
  - ☐ Multiplexed address/data bus
  - ☐ Address output cycle count

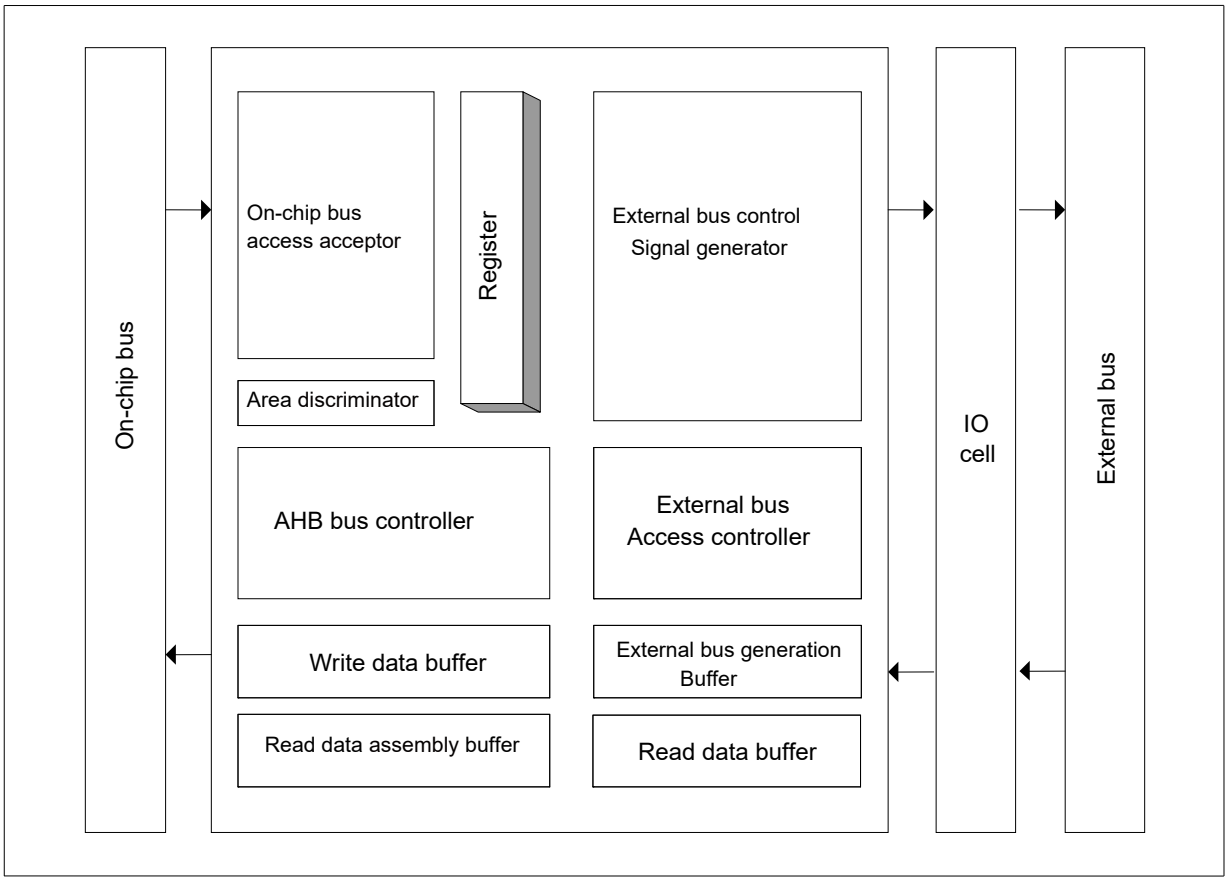
36.3 Configuration

This section shows the configuration of the external bus interface.

Block Diagram in External Bus Interface

Figure 36-1 shows the block diagram in the external bus interface.

Figure 36-1. Block Diagram of External Bus Interface



36.4 Registers

This section explains the registers of the external bus interface.

Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x0600	ASR0				CS0 area register
0x0604	ASR1				CS1 area register
0x0608	ASR2				CS2 area register
0x060C	ASR3				CS3 area register
0x0640	ACR0				CS0 bus setting register
0x0644	ACR1				CS1 bus setting register
0x0648	ACR2				CS2 bus setting register
0x064C	ACR3				CS3 bus setting register
0x0680	AWR0				CS0 wait register
0x0684	AWR1				CS1 wait register
0x0688	AWR2				CS2 wait register
0x068C	AWR3				CS3 wait register
0x06C0	Reserved (DMAR0)				ch.0 external DMA transfer register (This function is not supported by this series.)
0x06C4	Reserved (DMAR1)				ch.1 external DMA transfer register (This function is not supported by this series.)
0x06C8	Reserved (DMAR2)				ch.2 external DMA transfer register (This function is not supported by this series.)
0x06CC	Reserved (DMAR3)				ch.3 external DMA transfer register (This function is not supported by this series.)

36.4.1 CS Area Setting Registers: ASR0 to ASR3 (Area Setting Register 0-3)

The bit configurations of the CS area setting registers are shown below.

These registers configure the CS areas CS0 to CS3. Each CS area has a single ASR register. Set the CS areas such that they do not overlap. See "36.5.10 CS Setting Flow" for the setting procedure for these registers.

ASR0: Address 0600H (Access: Word)

ASR1: Address 0604H (Access: Word)

ASR2: Address 0608H (Access: Word)

ASR3: Address 060CH (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	SADR[31:24]							
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

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	SADR[23:16]							
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	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
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
	ASZ[3:0]				Reserved	WREN	LEDN	CSEN
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Attribute	R/W	R/W	R/W	R/W	R0,W0	R/W	R/W[2]	R/W

[1]: [Initial value] ASR0 0000_0000_0000_0000_0000_0000_1111_0001B
Not ASR0 XXXX_XXXX_XXXX_XXXX_0000_0000_XXXX_0XX0B

[2]: The attribute is "R0,W0" only ASR0 register.

[bit31 to bit16] SADR[31:16] (Start Address): CS Area Start Address

SADR specifies the start address of the CS area. The initial value for ASR0 is "0000_0000_0000_0000", and for not ASR0 is undefined. The start address sets the upper 16 bits of the 32-bit address. The CS area is the area starting from the address specified in these registers with a range as specified by ASZ[3:0]. The CS area boundary is determined according to the setting of bits 7 to 4:ASZ[3:0] of these registers. For example, when the CS area is configured as 1MB using ASZ[3:0]=0100, bit[19:16] of SADR are ignored and only SADR[31:20] has meaning.

Note:

The address range that can be allocated to the CS area depends on the model. See "Appendix".

[bit15 to bit8] Reserved

Always write "0" to these bits.

[bit7 to bit4] ASZ[3:0] (Area SiZe): CS Area Size

These bits configure the size of the CS area as follows. These bits also specify the bit position within SADR that is actually compared to the address.

ASZ[3:0]	CS Area Size	SADR Bits that are Actually Compared to the Address
0000	64KB	SADR[31:16]
0001	128KB	SADR[31:17]
0010	256KB	SADR[31:18]
0011	512KB	SADR[31:19]
0100	1MB	SADR[31:20]
0101	2MB	SADR[31:21]
0110	4MB	SADR[31:22]
0111	8MB	SADR[31:23]
1000	16MB	SADR[31:24]
1001	32MB	SADR[31:25]
1010	64MB	SADR[31:26]
1011	128MB	SADR[31:27]
1100	256MB	SADR[31:28]
1101	512MB	SADR[31:29]
1110	1GB	SADR[31:30]
1111	2GB (Initial value)	SADR[31]

[bit3] Reserved

Always write "0" to this bit.

[bit2] WREN (Write Enable)

This bit sets whether writes to the CS area are enabled or disabled.

WREN	Writes Enabled or Disabled
0	Writes disabled
1	Writes enabled

The initial value for ASR0 is "0", and for not ASR is undefined.
If a write to a write-disabled area is generated from the internal bus, that access is ignored and the external access is not performed. For an area to be written such as data area, set WREN to "1".

[bit1] LEDN (Little Endian)

LEDN sets the byte order of the CS area.
ASR0 does not have this bit, and reading this bit always returns "0".

LEDN	Endian
0	Big endian
1	Little endian

Initial value other than ASR0 are undefined.

[bit0] CSEN (Chip Select Enable): CS Area Enable

This bit sets whether the CS area is enabled or disabled. Operation starts according to the settings of the ASR register, ACR register, and AWR register by setting CSEN to "1".

CSEN	CS area Enabled or Disabled
0	Disabled
1	Enabled

The initial value for ASR0 is "1", and for not ASR is "0".

36.4.2 CS Bus Setting Registers: ACR0 to ACR3 (Area Configuration Register 0-3)

The bit configurations of the CS bus setting registers are shown below.

These registers set the bus of the CS area. Each CS area has a single ACR register. See "5.10 CS Setting Sequence" for the setting procedure for these registers.

ACR0: Address 0640_H (Access: Word)

ACR1: Address 0644_H (Access: Word)

ACR2: Address 0648_H (Access: Word)

ACR3: Address $064C_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Attribute	R0,W0	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	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
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							
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
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
	DBW[1:0]		Reserved		ADTY	BSTY	Reserved	Reserved
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	
Attribute	R/W	R/W	R0,W0	R0,W0	R/W	R/W	RX,W0	RX,W0

[1]: [Initial value] ACR0 $0000_0000_0000_0000_0000_0000_0100_0000_B$
Not ACR0 $0000_0000_0000_0000_0000_0000_XX00_XX0X_B$

[bit 31 to bit8] Reserved

Always write "0" to these bits.

[bit7, bit6] DBW[1:0] (Data Bus Width): Data Bus Width

These bits set the data bus width.

DBW[1:0]	Data Bus Width	Positions of Bits Used (D31 to D16)
00	8-bit	D[31:24]
01	16-bit	D[31:16]
10	Reserved (32-bit)	-
11	Reserved (32-bit)	-

In this series, 32-bit data bus width is not supported.
The initial value for ACR0 is "00". The initial value other than ACR0 is undefined.

[bit5, bit4] Reserved

Always write "0" to these bits.

[bit3] ADTY (Address output Type): Address Type

This bit sets the address output type.
The initial value for ACR0 is "00", and for not ASR is undefined.

ADTY	Description
0	Normal output
1	During 16-bit addressing, addresses are shifted by 1 bit and output. See "36.5.6 Address Information" for details.

[bit2] BSTY (Bus Type)

This bit sets the bus type.
The initial value for ACR0 is "00", and for not ASR is undefined.

BSTY	Description
0	Split address/data bus
1	Multiplexed address/data bus

[bit1, bit0] Reserved

Always write "0" to these bits.

36.4.3 CS Wait Registers: AWR0 to AWR3 (Area Wait Register 0-3)

The bit configurations of the CS wait registers are shown below.

These registers configure each type of wait for the CS areas CS0 to CS3. Each CS area has a single AWR register. See "36.5.10 CS Setting Flow" for the setting procedure for these registers.

AWR0: Address 0680$_H$ (Access: Word)

AWR1: Address 0684$_H$ (Access: Word)

AWR2: Address 0688$_H$ (Access: Word)

AWR3: Address 068C$_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved				RWT[3:0]			
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	WWT[3:0]				RIDL[1:0]		WRCV[1:0]	
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	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CSRD[1:0]		RDCS[1:0]		CSWR[1:0]		WRCS[1:0]	
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

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADCY[1:0]		ACS[1:0]		ASCY	Reserved	RDYE	Reserved
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Attribute	R/W	R/W	R/W	R/W	R/W	R0,W0	R/W	R0,W0

[1]: [Initial value] AWR0 0000_1111_0000_0000_1111_0000_0000_0000$_B$
Not AWR0 0000_XXXX_XXXX_XXXX_XXXX_XXXX_XXXX_X0X0$_B$

[bit31 to bit28] Reserved

Always write "0" to these bits.

[bit27 to bit24] RWT[3:0] (Read Access Auto Wait)

RWT[3:0] sets the number of auto wait cycles when fetching data during the read access cycle.

RWT[3:0]	Read Access Wait
0000	0 cycle
0001	1 cycle
0010	2 cycles
0011	3 cycles
:	:
1110	14 cycles
1111	15 cycles (AWR0 Initial value)

[bit23 to bit20] WWT[3:0] (Write Access Auto Wait)

WWT[3:0] sets the number of auto wait cycles during the write access cycle.

WWT[3:0]	Write Access Wait
0000	0 cycle (AWR0 Initial value)
0001	1 cycle
0010	2 cycles
0011	3 cycles
:	:
1110	14 cycles
1111	15 cycles

[bit19, bit18] RIDL[1:0] (Read Access Idle Cycle)

RIDL[1:0] is configured in order to prevent conflicts on the data bus between the read data from a device with a long output off time and the data of the subsequent access. If an access meeting any of the following conditions occurs in sequence after a read access, the idle cycles specified in RIDL are inserted after the read access.

- Write access
- Access to another CS area
- Access to a CS area configured with address/data multiplexed bus type

For the case of sequential read accesses to the same CS area configured with split bus type (ACR:BSTY=0), idle cycles are not inserted by RIDL. During idle cycles, all CS signals are negated and the data pins are put in the high-impedance state.

RIDL[1:0]	Read Access Idle Cycle
00	0 cycle (AWR0 Initial value)
01	1 cycle
10	2 cycles
11	3 cycles

[bit17, bit16] WRCV[1:0] (Write Recovery Cycle)

WRCV[1:0] is the write recovery cycle setting and is configured to control access to devices that have a limit on the interval between a write access and the next access. During write recovery cycles, all of the chip select signals are negated and write strobe signals WRnX (n=0, 1) is also held negated. Furthermore, new accesses are not started within this period. When the write recovery cycle is set to 1 cycle or higher, the write recovery cycle is always inserted after the write access.

WRCV[1:0]	Write Recovery Cycle
00	0 cycle (AWR0 Initial value)
01	1 cycle
10	2 cycles
11	3 cycles

[bit15, bit14] CSRD[1:0] (CSnX to RDX Setup Cycle)

CSRD[1:0] configures the read access CSnX to RDX setup cycles which set the period until RDX is asserted after CSnX is asserted.
In order to correctly establish the protocol when address/data multiplex bus is configured (ACR:BSTY=1), set the AWR parameters to satisfy the following conditions.

ACS + CSRD ≥ 1 and ACS + CSWR ≥ 1

CSRD[1:0]	CSnX → RDX Setup Extension Cycle
00	0 cycle
01	1 cycle
10	2cycles
11	3 cycles (AWR0 Initial value)

[bit13, bit12] RDCS[1:0] (RDX to CSnX Hold Cycle)

RDCS[1:0] configures the read access RDX to CSnX hold cycles which set the period until CSnX is negated after RDX is negated.

RDCS[1:0]	RDX → CSnX Hold Extension Cycle
00	0 cycle
01	1 cycle
10	2 cycles
11	3 cycles (AWR0 Initial value)

[bit11, bit10] CSWR[1:0] (CSnX to WRnX Setup Cycle)

CSWR[1:0] configures the write access CSnX to WRnX setup cycles which set the period until WRnX is asserted after CSnX is asserted.
In order to correctly establish the protocol when address/data multiplex bus is configured (ACR.BSTY=1), set the AWR parameters to satisfy the following conditions.

ACS + CSRD ≥ 1 and ACS + CSWR ≥ 1

CSWR[1:0]	CSnX → WRnX Setup Extension Cycle
00	0 cycle (AWR0 Initial value)
01	1 cycle
10	2 cycles
11	3 cycles

[bit9, bit8] WRCS[1:0] (WRnX to CSnX Hold Cycle)

WRCS[1:0] configures the write access WRnX to CSnX hold cycles which set the period until CSnX is negated after WRnX is negated.

WRCS[1:0]	WRnX → CSnX Hold Extension Cycle
00	0 cycle (AWR0 Initial value)
01	1 cycle
10	2 cycles
11	3 cycles

[bit7, bit6] ADCY[1:0] (Address Cycle): Address Output Extension Cycle Count

ADCY[1:0] sets the number of extension cycles for outputting addresses to the data bus during access to CS areas configured with address/data multiplexed bus type. The settings of these bits are only valid when the bus type is set to address/data multiplexed.

In order to correctly establish the protocol when ADCY is set to 1 or higher, set the AWR parameters to satisfy the following conditions.

ADCY + 1 ≤ ACS + CSRD and ADCY + 1 ≤ ACS + CSWR

ADCY[1:0]	Number of Address Output Extension Cycles During Address/Data Multiplexing
00	0 cycle (AWR0 Initial value)
01	1 cycle
10	2 cycles
11	3 cycles

[bit5, bit4] ACS[1:0] (A00 to A21 to CSnX delay cycle): A00 to A21 to CSnX Delay Cycle Count

ACS[1:0] sets the number of delay cycles from outputting A00 to A21 and ASX to outputting CSnX. This is used when the address for CSnX assert needs to be setup for a fixed time, or when CSnX edges are required when accessing the same chip select area in sequence.

ACS[1:0]	A00 to A21 → CSnX Delay Cycle Count
00	0 cycle (AWR0 Initial value)
01	1 cycle
10	2 cycles
11	3 cycles

[bit3] ASCY (ASX Cycle): ASX Output Extension Cycle Count

ASCY sets the number of cycles to extend ASX output. ASX outputs a minimum of 1 cycle.

ASCY	ASX Output Extension Delay Cycle Count
0	0 cycle (AWR0 Initial value)
1	1 cycle

[bit2] Reserved

Always write "0" to this bit.

[bit1] RDYE (RDY Enable)

RDYE sets whether the wait insertion function by external RDY pin is enabled or disabled.

RDYE	RDY Pin Enable
0	Wait insertion by RDY pin disabled (AWR0 initial value)
1	Wait insertion by RDY pin enabled

[bit0] Reserved

Always write "0" to this bit.

36.4.4 External DMA Transfer Registers: DMAR0-3 (DMA transfer Register 0-3)

The bit configuration of the external DMA transfer registers is shown below.

These registers set the external pins for DMA transfers. This function is not supported by this series.

DMAR0: Address $06C0_H$ (Access: Word)

DMAR1: Address $06C4_H$ (Access: Word)

DMAR2: Address $06C8_H$ (Access: Word)

DMAR3: Address $06CC_H$ (Access: Word)

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
	Reserved							
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
Attribute	R0,W0	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	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
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							
Initial value	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
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				REQL	ACKMD	ACKL	EOPL
Initial value	[1]	[1]	[1]	[1]	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R/W0	R/W0	R/W0	R/W0

[1]: [Initial value] $0000_0000_0000_0000_0000_0000_0000_0000_B$

[bit31 to bit4] Reserved

Always write "0" to these bits.

[bit3] REQL

When writing, always write "0" to this bit.

[bit2] ACKMD

When writing, always write "0" to this bit.

[bit1] ACKL

When writing, always write "0" to this bit.

[bit0] EOPL

When writing, always write "0" to this bit.

36.5 Operation

This section explains the operation of the external bus interface.

36.5.1 External Pin Table
36.5.2 External Bus Signal Protocol
36.5.3 Address Alignment
36.5.4 Split Access
36.5.5 Data Alignment
36.5.6 Address Information
36.5.7 Idle Cycle Insertion Function
36.5.8 External Bus Output Signal Timing Settings
36.5.9 RDY Pin Access Cycle Extension Function
36.5.10 CS Setting Flow
36.5.11 Example of Connecting to Asynchronous Memory
36.5.12 Example of Connection to Little Endian Device

36.5.1 External Pin Table

This section shows the external pin table.

This section shows the external pins for the external bus interface of this series.

Table 36-1. External pin table

External Pin of this Series	Pin Number of this Series	Description
SYSCLK	35	System clock output
ASX	5	Address strobe output
CS0X, CS1X, CS2X, CS3X	6, 7, 38, 39	Chip selected output
RDX	8	Read strobe output
WR0X, WR1X	9, 10	Write strobe output
RDY	40	Bus ready input
D16_0 to D31_0/ D16_1 to D31_1	131 to 138, 139 to 143, 2 to 4/ 139 to 143, 2 to 4, 131 to 138	Data input/output and address output (during address multiplexing)
A00 to A21	11 to 19, 22 to 34	Address output

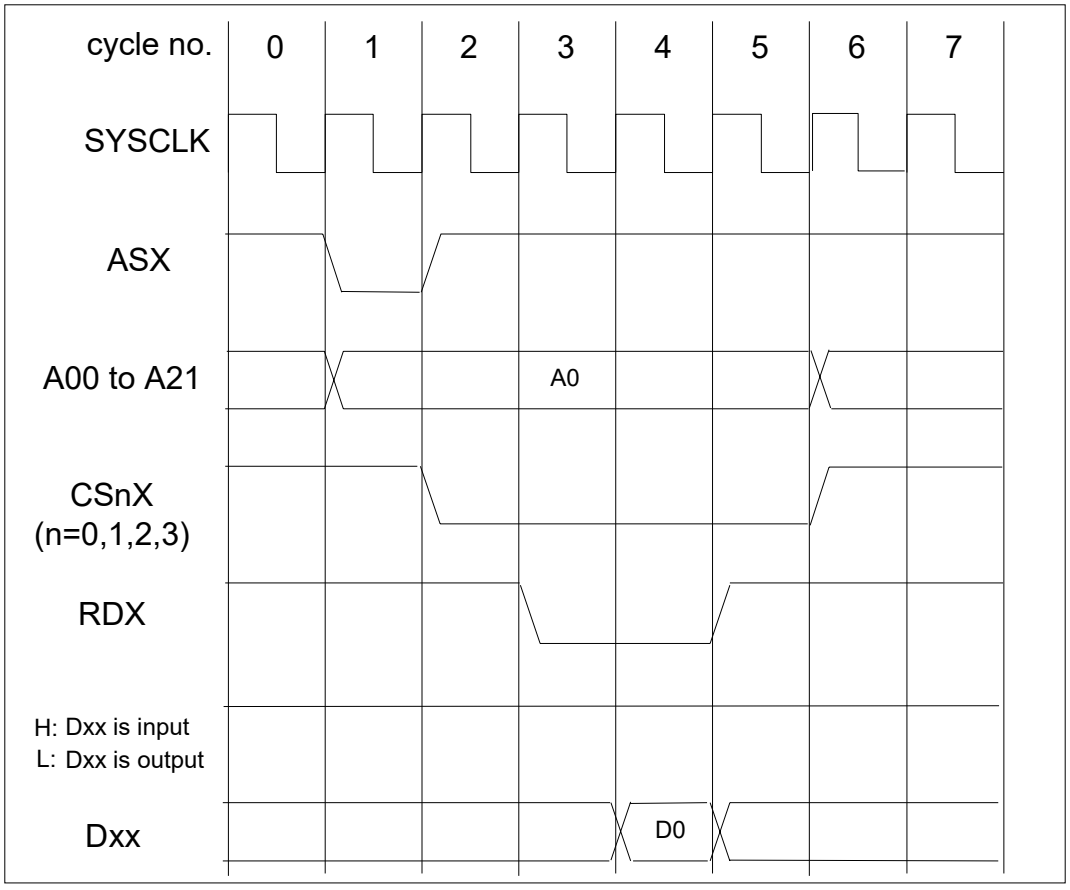
36.5.2 External Bus Signal Protocol

This section shows the external bus signal protocol.

- Address/Data Split Bus Read Protocol

  This section shows the protocol for read access using an address/data split bus.

Figure 36-2. Address/data split bus (Read operation example)



- Operating Example Description

  cycle1: "L" is output to ASX for 1 cycle to indicate that access is starting from this cycle. A00 to A21 indicate the address information of the access destination for this cycle.

  cycle2: After the configured count has finished from the access starting, "L" is output to CSnX (n= 0 to 3) continually until the access is complete. Devices on the external bus need to perform processing for the access only within the period where CSnX="L".

  cycle3: "L" is output to RDX after the configured count from when CSnX="L" output is started. External bus devices are required to return read data to D16 to D31 within the strobe period indicated by RDX="L".

  cycle4: The output to RDX returns to "H" after the configured count finishes after output of RDX="L" begins. STU fetches data from D16 to D31 to the internal buffer at the rising edge for the last SYSCLK within the period RDX=L.

  cycle5: The output of CSnX returns to "H" after the configured count finishes from when RDX returns to "H", and the read access finishes. In this example, CSnX returns to "H" when this cycle ends and the read access finishes.

- Signal description

  External bus output signals are synchronized to the rising edge of SYSCLK.

  ASX

  Indicates the start of access. This also functions as the address strobe.
  An "L" pulse is output for a period of 1 or 2 cycles from when the access starts.

  A00 to A21

  Outputs the address information of the access destination.
  This is output from when the access starts and continues until the access finishes.

  CSnX (n=0 to 3)

  Indicates that the access destination address is within the corresponding CS area. External bus devices are required to process requests from the bus only when this signal is "L". After the configured count has finished from when the access started, "L" begins to be output, and this continues until the access finishes.

  RDX

  Indicates the period of the read strobe. After the configured count ends from when CSnX (n=0 to 3) is driven, this outputs "L" for read access. This returns to output "H" after the read auto wait count has ended. The external bus device is required to return valid data in D16 to D31 within the period where RDX="L". This module fetches the D16 to D31 data into the internal buffer on the rising edge of the final SYSCLK within the period where RDX="L".

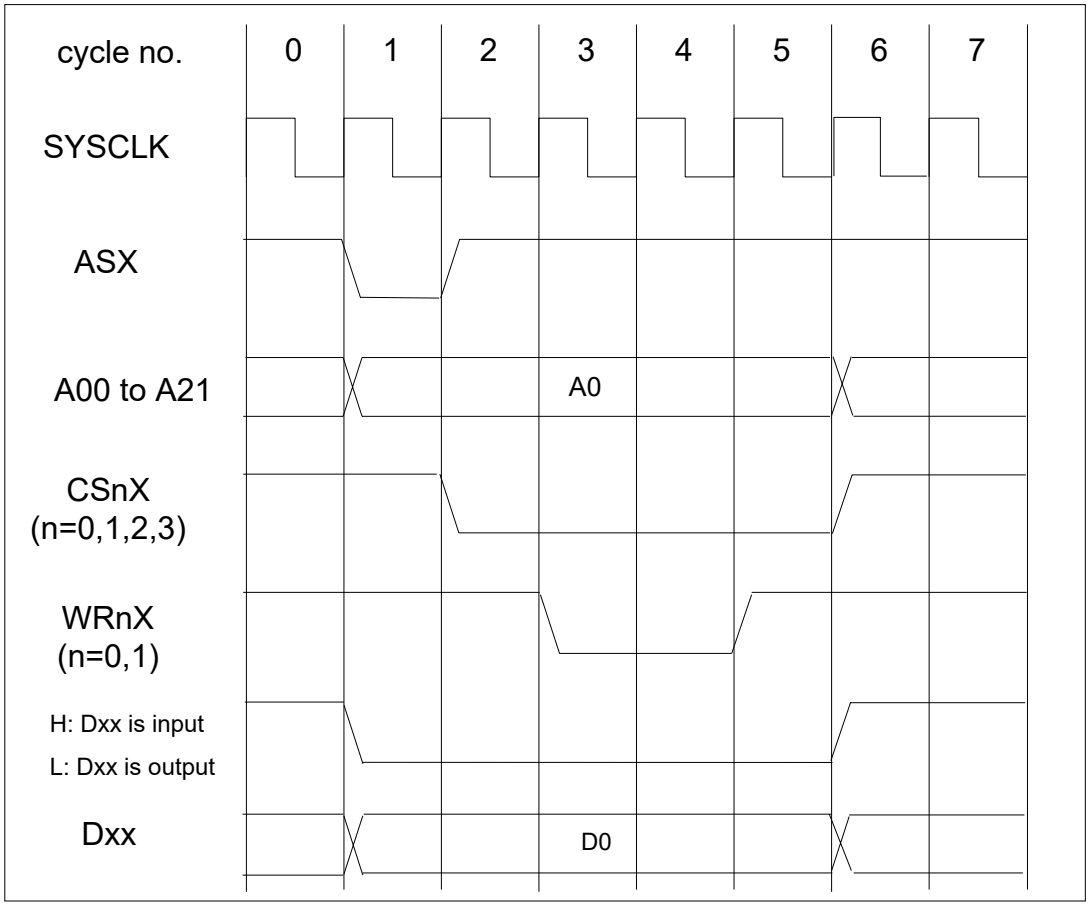
  D16 to D31

  The external bus device is required to return valid data in D16 to D31 within the period where RDX="L". This module fetches the D16 to D31 data into the internal buffer on the rising edge of the final SYSCLK within the period where RDX="L".

- Address/Data Split Bus Write Protocol

  This section shows the protocol for write access using an address/data split bus.

Figure 36-3. Address/Data Split Bus (Write Operation Example)



- Operating Example Description

  cycle1: "L" is output to ASX for 1 cycle to indicate that access is starting from this cycle. A00 to A21 indicate the address information of the access destination for this cycle.

  cycle2: After the configured count has finished from when the access was started, "L" is output to CSnX (n=0 to 3). CSnX continues to output "L" until the access is complete. Devices on the external bus need to execute processing for the access only within the period where CSnX="L".

  cycle3: After the configured count has finished after "L" starts being output to CSnX, "L" is output to WRnX (n=0, 1). External bus devices are required to fetch the value of D16 to D31 within the write strobe period where "L" is output to WRnX.

  cycle4: After the configured count has finished from when WRnX="L" starts being output, the output of WRnX returns to "H" and the write strobe period ends. In this example, the write strobe period is extended by 1 cycle. At the end of this cycle, the output of WRnX returns to "H" and the strobe period ends.

  cycle5: The output of CSnX returns to "H" after the configured count finishes from when WRnX returns to "H", and the write access finishes. In this example, CSnX returns to "H" when this cycle ends and the write access finishes.

- Signal Description

  External bus output signals are synchronized to the rising edge of SYSCLK.

  ASX

  Indicates the start of access. This also functions as the address strobe.
  An "L" pulse is output for a period of 1 or 2 cycles from when the access starts.

  A00 to A21

  Outputs the address information of the access destination.
  This is output from when the access starts and continues until the access finishes.

  CSnX (n=0 to 3)

  Indicates that the access destination address is within the corresponding CS area. External bus devices are required to process requests from the bus only when this signal is "L". After the configured count has finished from when the access started, "L" begins to be output, and this continues until the access finishes.

  WRnX (n=0, 1)

  Indicates the period of the write cycle strobe. After the configured count ends from when CSnX (n=0 to 3) is driven, this outputs "L" for write access. This returns to output "H" after the write auto wait count has ended. External bus devices are required to fetch the data of D16 to D31 within the period where WRnX (n=0, 1)="L".

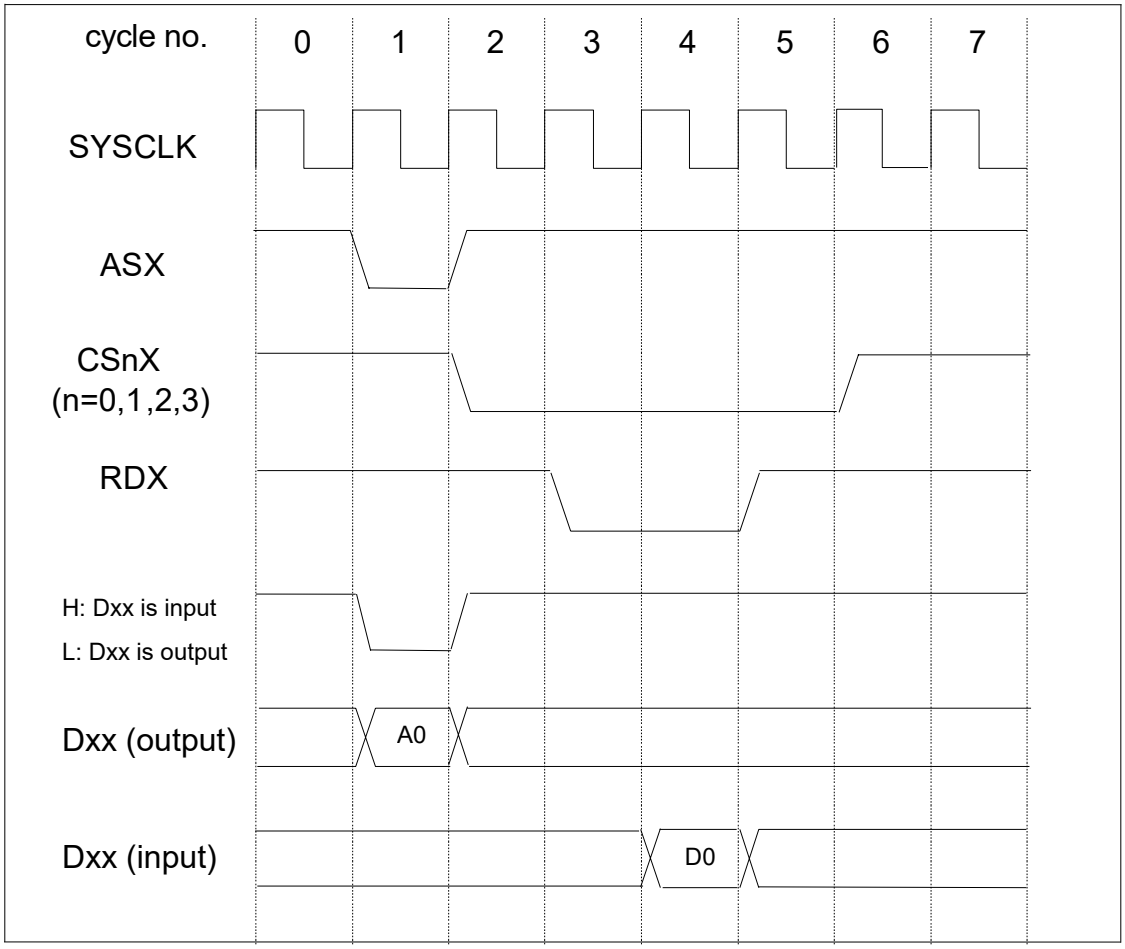
  D16 to D31

  Write data is output from when the access begins. The write data output continues until the access finishes. External bus devices are required to fetch the data of D16 to D31 within the period where WRnX="L".

- Address/Data Multiplexed Bus Read Protocol

  This section shows the protocol for read access using an address/data multiplexed bus.

Figure 36-4. Address/data multiplexed bus (Read operation example)



- Operating Example Description

  cycle1: "L" is output to ASX to indicate that access is starting from this cycle. Address information A0 is output to data bus D16 to D31. ASX functions as the strobe signal for this address information. This address information is output for the configured count cycles. After the configured count has finished, D16 to D31 are put into the input state.

  cycle2: After the configured count has finished from the access starting, "L" is output to CSnX (n= 0 to 3) continually until the access is complete. Devices on the external bus need to perform processing for the access only within the period where CSnX="L".

  cycle3: "L" is output to RDX after the configured count from when CSnX="L" output is started. External bus devices are required to return read data to D16 to D31 within the strobe period indicated by RDX="L".

  cycle4: The output to RDX returns to "H" after the configured count finishes after output of RDX="L" begins. The data on D16 to D31 is fetched into the internal buffer on the rising edge of the final SYSCLK within the period where RDX="L".

  cycle5: The output of CSnX (n=0 to 3) returns to "H" after the configured count finishes from when RDX returns to "H", and the read access finishes. In this example, CSnX returns to "H" when this cycle ends and the read access finishes.

- Signal Description

  External bus output signals are synchronized to the rising edge of SYSCLK.

  ASX

  Indicates the start of access. This also functions as the address strobe.

  An "L" pulse is output for a period of 1 or 2 cycles from when the access starts.

  CSnX (n=0 to 3)

  Indicates that the access destination address is within the corresponding CS area. External bus devices are required to process requests from the bus only when this signal is "L". After the configured count has finished from when the access started, "L" begins to be output, and this continues until the access finishes.

  RDX

  Indicates the period of the read strobe. After the configured count ends from when CSnX (n=0 to 3) is driven, this outputs "L" for read access. This returns to output "H" after the read auto wait count has ended. The external bus device is required to return valid data in D16 to D31 within the period where RDX="L". This module fetches the D16 to D31 data into the internal buffer on the rising edge of the final SYSCLK within the period where RDX="L".

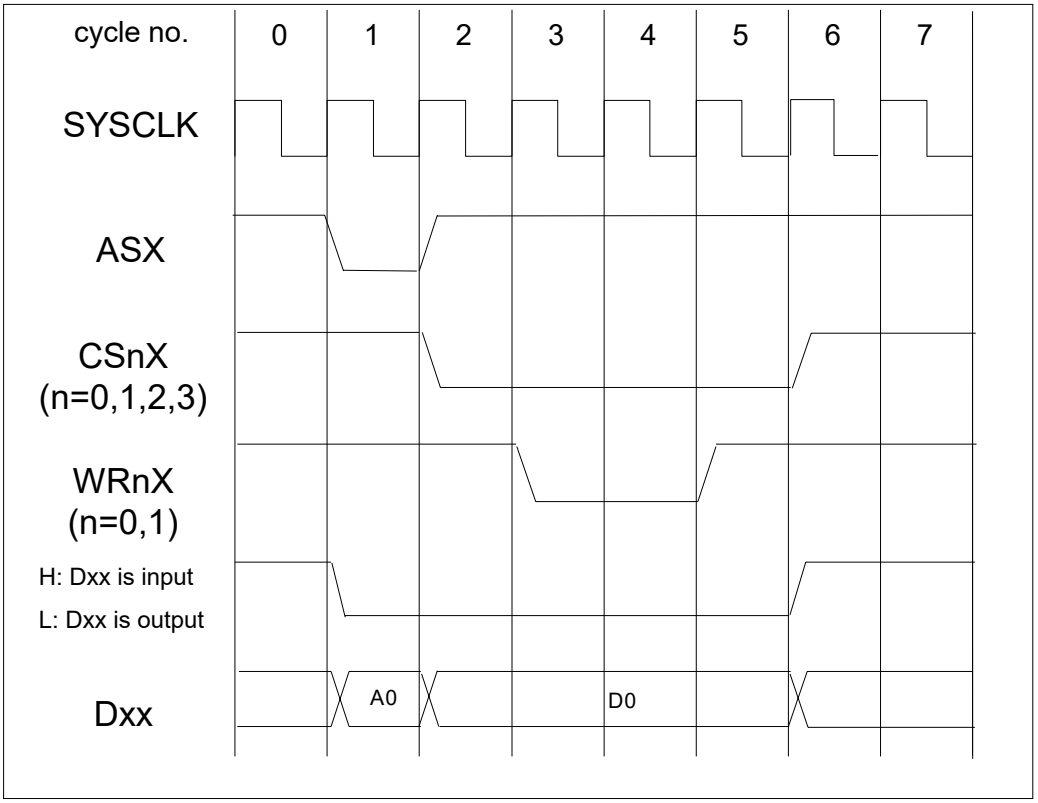
  D16 to D31

  Address information is output from when the access begins. After the configured count has finished, this enters the input state and accepts the read data from the external bus device. This module fetches the D16 to D31 data into the internal buffer on the rising edge of the final SYSCLK within the period where RDX="L".

- Address/Data Multiplexed Bus Write Protocol

  This section shows the protocol for write access using an address/data multiplexed bus.

Figure 36-5. Address/data multiplexed bus (Write operation example)



- Operating Example Description

  cycle1: The cycle where access begins. "L" is output to ASX to indicate the start of access. Address information is output to D16 to D31. ASX functions as the strobe signal for this address information. This address information is output for the configured count cycles.

  cycle2: After the configured count has finished from the access starting, "L" is output to CSnX (n= 0 to 3) continually until the access is complete. Devices on the external bus need to perform processing for the access only within the period where CSnX="L".

  cycle3: After the configured count has finished after "L" starts being output to CSnX, "L" is output to WRnX (n=0, 1). External bus devices are required to fetch the value of D16 to D31 within the write strobe period indicated by WRnX="L".

  cycle4: After the configured count has finished from when WRnX="L" starts being output, the output of WRnX returns to "H" and the write strobe period ends. In this example, the write strobe period is extended by 1 cycle. At the end of this cycle, the output of WRnX returns to "H" and the write strobe period ends.

  cycle5: The output of CSnX returns to "H" after the configured count finishes from when WRnX returns to "H", and the write access finishes. In this example, CSnX returns to "H" when this cycle ends and the write access finishes.

- Signal Description

  External bus output signals are synchronized to the rising edge of SYSCLK.

  ASX

  Indicates the start of access. This also functions as the address strobe.

  An "L" pulse is output for a period of 1 or 2 cycles from when the access starts.

  CSnX (n=0 to 3)

  Indicates that the access destination address is within the corresponding CS area. External bus devices are required to process requests from the bus only when this signal is "L". After the configured count has finished from when the access started, "L" begins to be output, and this continues until the access finishes.

  WRnX (n=0, 1)

  Indicates the period of the write strobe. After the configured count ends from when CSnX is driven, this outputs "L" for write access. This returns to output "H" after the write auto wait count has ended. External bus devices are required to fetch the data of D16 to D31 within the period where WRnX="L".

  D16 to D31

  Outputs the address information of the access destination from when the access starts. The write data begins to be output after the configured count ends, and continues until the access finishes. External bus devices are required to fetch the value of D16 to D31 within the write strobe period.

36.5.3 Address Alignment

This section shows the address alignment.

The external bus interface does not detect misalignment errors in the access destination address. As a result, word access and half-word access are performed as follows.

- Word Access

  Regardless of whether the lowermost 2 bits of the address specified by the program are "00", "01", "10", or "11", the lowermost 2 bits of the output address are "00".

- Half-word Access

  If the lowermost 2 bits of the address specified by the program are "00" or "01, the lowermost 2 bits of the output address are "00", and if the lowermost 2 bits are "10" or "11", then the lowermost 2 bits of the output address are "10".

36.5.4 Split Access

This section shows the split access.

If the access size is larger than the bus width, this is executed by splitting a single access.

- Number of Split Accesses

Bus Width	Access Size		
	Byte	Half-word	Word
8-bit	1 time	2 times	4 times
16-bit	1 time	1 time	2 times

36.5.5 Data Alignment

This section shows the data alignment.

Each CS area supports both big endian and little endian. However, CS0 only supports big endian. The data bus width can be selected between 8-bit and 16-bit for each CS area.

The following shows the data alignment for the external access size and the corresponding control signals for each endian setting and data bus width setting.

- Big Endian - 16 Bits

Access		Split Access	Output Pins				
Size	Address Lowermost 2 Bits		A01, A00	D31 to D24	D23 to D16	WR0X	WR1X
Byte	00	-	00	bit7 to bit0		○	
	01	-	01		bit7 to bit0		○
	10	-	10	bit7 to bit0		○	
	11	-	11		bit7 to bit0		○
Half-word	0n	-	00	bit15 to bit8	bit7 to bit0	○	○
	1n	-	10	bit15 to bit8	bit7 to bit0	○	○
Word	nn	First split access	00	bit31 to bit24	bit23 to bit16	○	○
		Second split access	10	bit15 to bit8	bit7 to bit0	○	○

- Big Endian - 8 Bits

Access		Split Access	Output Pins				
Size	Address Lowermost 2 Bits		A01, A00	D31 to D24	D23 to D16	WR0X	WR1X
Byte	00	-	00	bit7 to bit0	-	○	-
	01	-	01	bit7 to bit0	-	○	-
	10	-	10	bit7 to bit0	-	○	-
	11	-	11	bit7 to bit0	-	○	-
Half-word	0n	First split access	00	bit15 to bit8	-	○	-
		Second split access	01	bit7 to bit0	-	○	-
	1n	First split access	10	bit15 to bit8	-	○	-
		Second split access	11	bit7 to bit0	-	○	-
Word	nn	First split access	00	bit31 to bit24	-	○	-
		Second split access	01	bit23 to bit15	-	○	-
		Third split access	10	bit15 to bit8	-	○	-
		Fourth split access	11	bit7 to bit0	-	○	-

- Little Endian - 16 Bits

Access		Split Access	Output Pins				
Size	Address Lowermost 2 Bits		A01, A00	D31 to D24	D23 to D16	WR0X	WR1X
Byte	00	-	00	bit7 to bit0	-	○	-
	01	-	01	-	bit7 to bit0	-	○
	10	-	10	bit7 to bit0	-	○	-
	11	-	11	-	bit7 to bit0	-	○
Half-word	0n	-	00	bit7 to bit0	bit15 to bit8	○	○
	1n	-	10	bit7 to bit0	bit15 to bit8	○	○
Word	nn	First split access	00	bit7 to bit0	bit15 to bit8	○	○
		Second split access	10	bit23 to bit16	bit31 to bit24	○	○

- Little Endian - 8 Bits

Access		Split Access	Output Pins				
Size	Address Lowermost 2 Bits		A01, A00	D31 to D24	D23 to D16	WR0X	WR1X
Byte	00	-	00	bit7 to bit0	-	○	-
	01	-	01	bit7 to bit0	-	○	-
	10	-	10	bit7 to bit0	-	○	-
	11	-	11	bit7 to bit0	-	○	-
Half-word	0n	First split access	00	bit7 to bit0	-	○	-
		Second split access	01	bit15 to bit8	-	○	-
	1n	First split access	10	bit7 to bit0	-	○	-
		Second split access	11	bit15 to bit8	-	○	-
Word	nn	First split access	00	bit7 to bit0	-	○	-
		Second split access	01	bit15 to bit8	-	○	-
		Third split access	10	bit23 to bit16	-	○	-
		Fourth split access	11	bit31 to bit24	-	○	-

36.5.6 Address Information

This section shows the address information.

- Address Information and Output Pins
  - ☐ Split address/data bus

    22-bit address information is output to A00 to A21.
  - ☐ Multiplexed address/data bus

    In the address/data multiplexed bus, the address information is output to data bus pins D16 to D31 during the address output cycle. The address bit width that can be output is determined by the data bus width setting. Even while address/data multiplexed bus is selected, the address is output to address pins A00 to A21. The missing parts of address information output to pins D16 to D31 can be supplemented by using address pins A00 to A21.
- Address Type

  The output of address information can be selected from normal type that outputs as normal and the shift type that outputs using bit shift. This is set using ACR:ADTY.
  - ☐ ADTY=0

    The normal output mode. The address information is output directly to the pins without bit shifting.
  - ☐ ADTY=1

    Address shift output mode. The address bus information is output to the pins after bit shifting.

  The relationship between the address type (ACR:ADTY), bus type (ACR:BSTY), bus width, output address information, and address output pins is as follows.
- Output Address and Output Pins

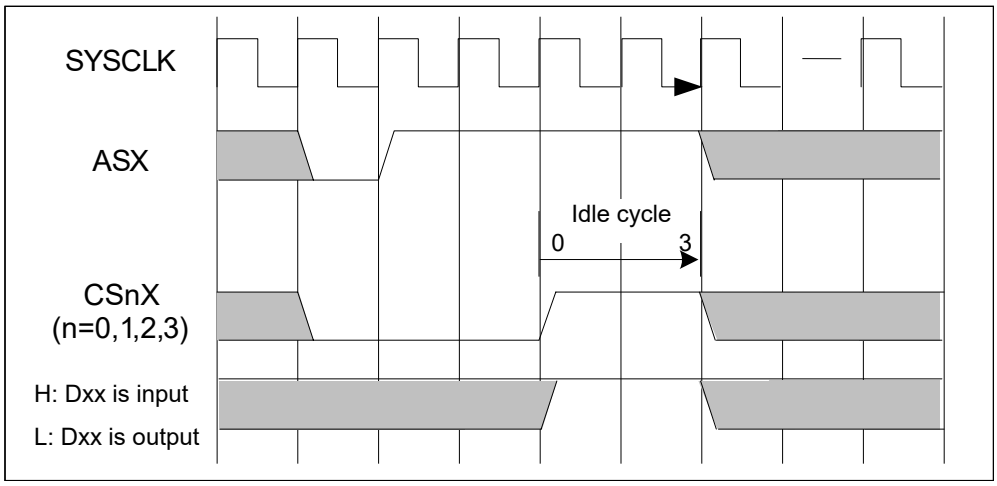
ACR Register		Bus Width [bit]	A21 to A00	Output Pins D31 to D16 During Address Output Cycle	
ADTY	BSTY			D31 to D24	D23 to D16
0	0	8	Address[21:0]	-	-
		16			
0	1	8	Address[21:0]	Address[7:0]	-
		16	Address[21:0]	Address[15:8]	Address[7:0]
1	0	8	Address[21:0]	-	-
		16	Address[22:1]		
1	1	8	Address[21:0]	Address[7:0]	-
		16	Address[22:1]	Address[16:9]	Address[8:1]

36.5.7 Idle Cycle Insertion Function

This section shows the idle cycle insertion function.

Idle cycles can be inserted between accesses. The next access does not start during the idle cycle even if there is a request, but starts after the idle cycle count finishes.

Figure 36-6. Idle Cycle Inserted



- Read Access Idle Cycles

  If an access meeting any of the following conditions occurs in sequence after a read access, idle cycles are inserted after the read access. This is configured using AWR:RIDL[1:0].

  - ☐ Write access
  - ☐ Access to another CS area
  - ☐ Access to a CS area configured with address/data multiplexed bus type

  Note:

  The only time when idle cycles are not inserted by RIDL is when sequential read accesses are performed on the same CS area configured for split bus type.

- Write Recovery Cycles

  Idle cycles are inserted after a write access ends. This is configured using AWR:WRCV[1:0].

36.5.8 External Bus Output Signal Timing Settings

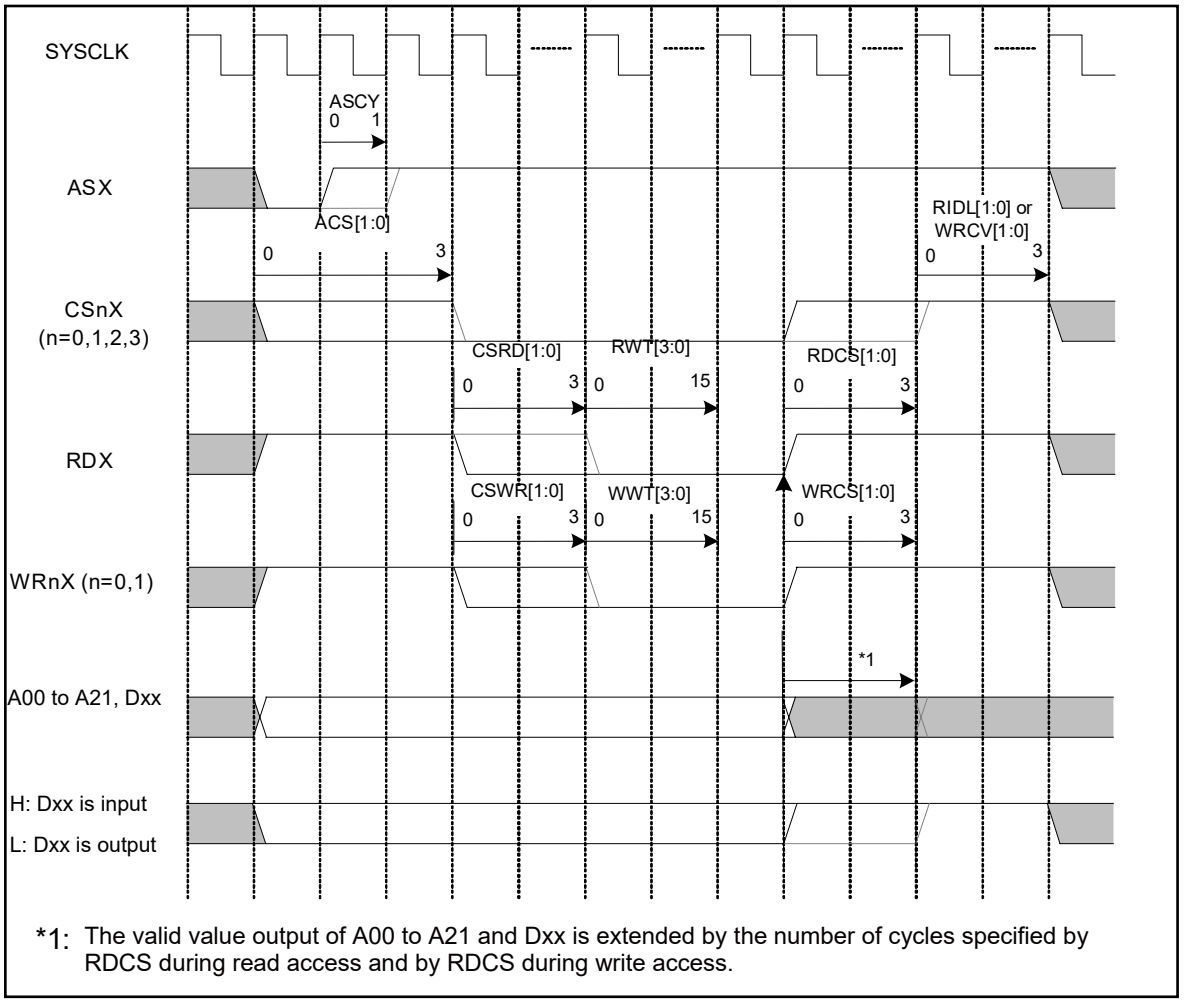
This section shows the external bus output signal timing settings.

The external bus signal output timing is determined by the following parameters. The timing parameters are determined by the values set in the registers.

- Address/Data Split Bus Timing Parameters

  This section shows the timing parameters that can be configured in the address/data split bus.

Figure 36-7. Address/Data Split Bus Timing Parameters



*1: The valid value output of A00 to A21 and Dxx is extended by the number of cycles specified by RDCS during read access and by RDCS during write access.

- Address/Data Split Bus Timing Parameters

Parameter Name	Function Name	Description
ASCY(ASX Cycle)	ASX output extension cycle count	"L" is output to ASX for (ASCY+1) cycles from when the access starts.
ACS[1:0] (A00 to A21 to CSnX delay cycle)	A00 to A21 → CSnX delay cycle count	Output of "L" to CSnX (n=0 to 3) starts after the ACS count has finished from ASX output.
CSRD[1:0] (CSnX to RDX setup cycle)	CSnX → RDX setup cycle	During read access, output of "L" to RDX begins after the CSRD count finishes after "L" output to CSnX begins.
RWT[3:0] (Read access auto Wait)	Read access auto wait	During read access, the RDX output returns to "H" after (RWT+1) count from when output of "L" to RDX begins.
RDCS[1:0] (RDX to CSnX hold cycle)	RDX → CSnX hold cycle	During read access, the output of CSnX returns to "H" after RDCS count from the cycle where the output of RDX returns to "H".
CSWR[1:0] (CSnX to WRnX setup cycle)	CSnX → WRnX setup cycle	During write access, output of "L" to WRnX begins after the CSWR count finishes after "L" output to CSnX begins.
WWT[3:0] (Write access auto Wait)	Write access auto wait	During write access, the output to WRnX (n=0, 1) returns to "H" after (WWT+1) count finishes.
WRCS[1:0] (WRnX to CSnX hold cycle)	WRnX → CSnX hold cycle	During write access, the output of CSnX returns to "H" after WRCS count from the cycle where the output of WRnX returns to "H".
RIDL[1:0] (Read access Idle cycle)	Read access idle cycle	After a read access has finished, the next access is able to start after RIDL count has finished.
WRCV[1:0] (Write Recovery cycle)	Write recovery cycle	After a write access has finished, the next access is able to start after WRCV count has finished.

The number of access cycles is determined from the following formula.

Number of read access cycles = Address & data output (1) + ACS (0 to 3) + CSRD (0 to 3) + RWT (0 to 15) +RDCS (0 to 3)

Minimum: 1 cycle; Maximum: 25 cycles

Number of write access cycles = Address & data output (1) + ACS (0 to 3) + CSWR (0 to 3) + WWT (0 to 15) + WRCS (0 to 3)

Minimum: 1 cycle; Maximum: 25 cycles

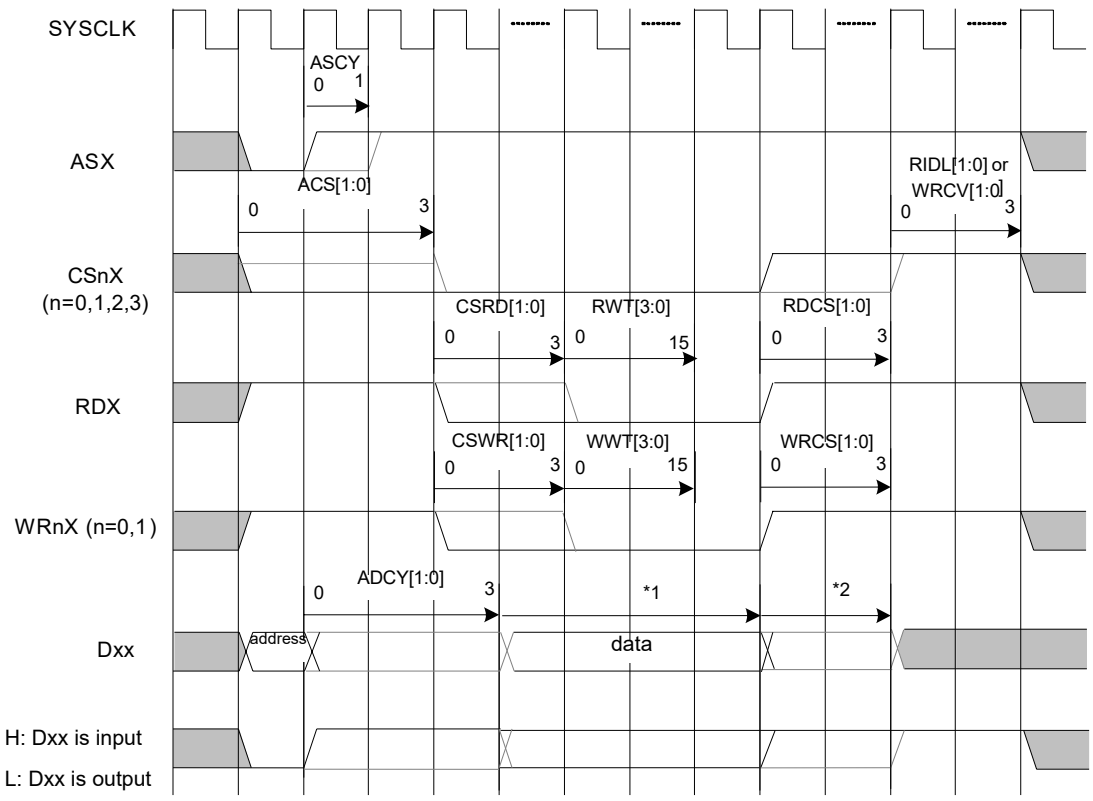
The following conditions need to be met in order to correctly establish the protocol.

ASCY ≤ ACS + CSRD + RWT + RDCS and ASCY ≤ ACS + CSWR + WWT + WRCS

- Address/Data Multiplexed Bus Timing Parameters

  This section shows the timing parameters that can be configured in the address/data multiplexed bus.

Figure 36-8. Address/Data Multiplexed Bus Timing Parameters



*1 : During write access, the write data of Dxx is output immediately after the address output cycle ends.
*2 : The valid value output of Dxx is extended by the number of cycles specified by WRCS during write access.

- Address/Data Multiplexed Bus Timing Parameters

Parameter Name	Function Name	Description
ASCY(ASX Cycle)	ASX output extension cycle count	"L" is output to ASX for (ASCY+1) cycles from when the access starts.
ACS[1:0] (A00 to A21 to CSnX delay cycle)	A00 to A21 → CSnX delay cycle count	Output of "L" to CSnX (n=0 to 3) starts after the ACS count has finished from ASX output.
ADCY[1:0](Address Cycle)	Address output extension cycle count	If ADCY ≥ ASCY D16 to D31 output (ADCY+1) cycle address information from when access starts. During writes, write data is output after the count finishes until the access finishes. If ADCY < ASCY The count value is changed from (ADCY+1) to (ASCY+1). There are no other differences. The ADCY count operates independently of the other counters. Furthermore, it is not affected by the start conditions of other counters. As a result, there are no limits on setting the counter value in order for the overall protocol to function correctly. See the prohibited setting conditions outside of the table.
CSRD[1:0] (CSnX to RDX setup cycle)	CSnX → RDX setup cycle	During read access, output of "L" to RDX begins after the CSRD count finishes after "L" output to CSnX begins.
RWT[3:0] (Read access auto Wait)	Read access auto wait	During read access, the RDX output returns to "H" after (RWT+1) count from when output of "L" to RDX begins.
RDCS[1:0] (RDX to CSnX hold cycle)	RDX → CSnX hold cycle	During read access, the output of CSnX returns to "H" after RDCS count from the cycle where the output of RDX returns to "H".
CSWR[1:0] (CSnX to WRnX setup cycle)	CSnX → WRnX setup cycle	During write access, output of "L" to WRnX (n=0, 1) begins after the CSWR count finishes after "L" output to CSnX begins.
WWT[3:0] (Write access auto Wait)	Write access auto wait	During write access, the output to WRnX returns to "H" after (WWT+1) count finishes.
WRCS[1:0] (WRnX to CSnX hold cycle)	WRnX → CSnX hold cycle	During write access, the output of CSnX returns to "H" after WRCS count from the cycle where the output of WRnX returns to "H".
RIDL[1:0] (Read access Idle cycle)	Read access idle cycle	After a read access has finished, the next access is able to start after RIDL count has finished.
WRCV[1:0] (Write Recovery cycle)	Write recovery cycle	After a write access has finished, the next access is able to start after WRCV count has finished.

The number of access cycles is determined from the following formula.

Number of read access cycles = Address output(1) + ACS(0 to 3) + CSRD(0 to3) + Data output(1) + RWT(0 to 15) + RDCS(0 to 3)

Minimum: 2 cycles; Maximum: 26 cycles

Number of write access cycles = Address output (1) + ACS(0 to 3) + CSWR(0 to 3) + Data output(1) + WWT(0 to 15) + WRCS(0 to 3)

Minimum: 2 cycles; Maximum: 26 cycles

The following four conditions need to be met in order to correctly establish the protocol.

ADCY + 1 ≦ ACS + CSRD

ADCY + 1 ≦ ACS + CSWR

ASCY + 1 ≦ ACS + CSRD

ASCY + 1 ≦ ACS + CSWR

36.5.9 RDY Pin Access Cycle Extension Function

This section shows the RDY pin access cycle extension function.

The read and write strobe cycles can be extended even after the auto wait cycles have finished by inputting "0" to the RDY pin.

This function can be enabled by the RDY pin for access to the corresponding area when setting AWR: RDYE to "1".

Use this function by setting the auto wait cycles of the corresponding area to 2 or more.

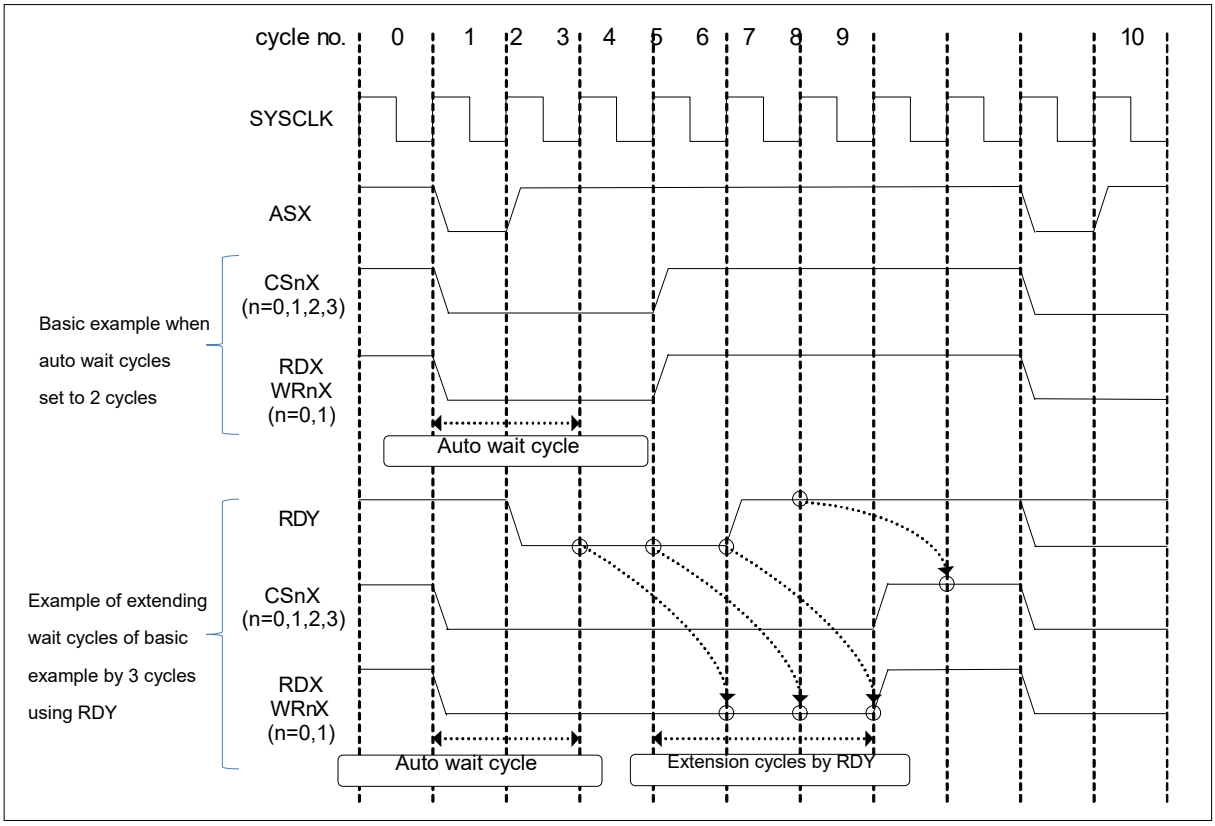
After the auto wait cycle has finished, the read and write strobe cycles are extended while "0" is input to RDY. If "1" is input to RDY after this, the read or write strobe cycle finishes in the next cycle.

RDY Signal Input Specifications

The input RDY signal adheres to the following specifications.

- Input RDY=1 except when extending the auto wait cycles.
- Begin inputting RDY=0 after checking that access to an area covered by auto wait cycle extension has started with ASX="L" and CSnX="L".
- Start inputting RDY=0 before the auto wait cycle ends. It is prohibited to input RDY=0 after the auto wait cycles have ended.
- Input RDY=1 after the required extension cycles have finished.

Figure 36-9. RDY Timing Example



36.5.10 CS Setting Flow

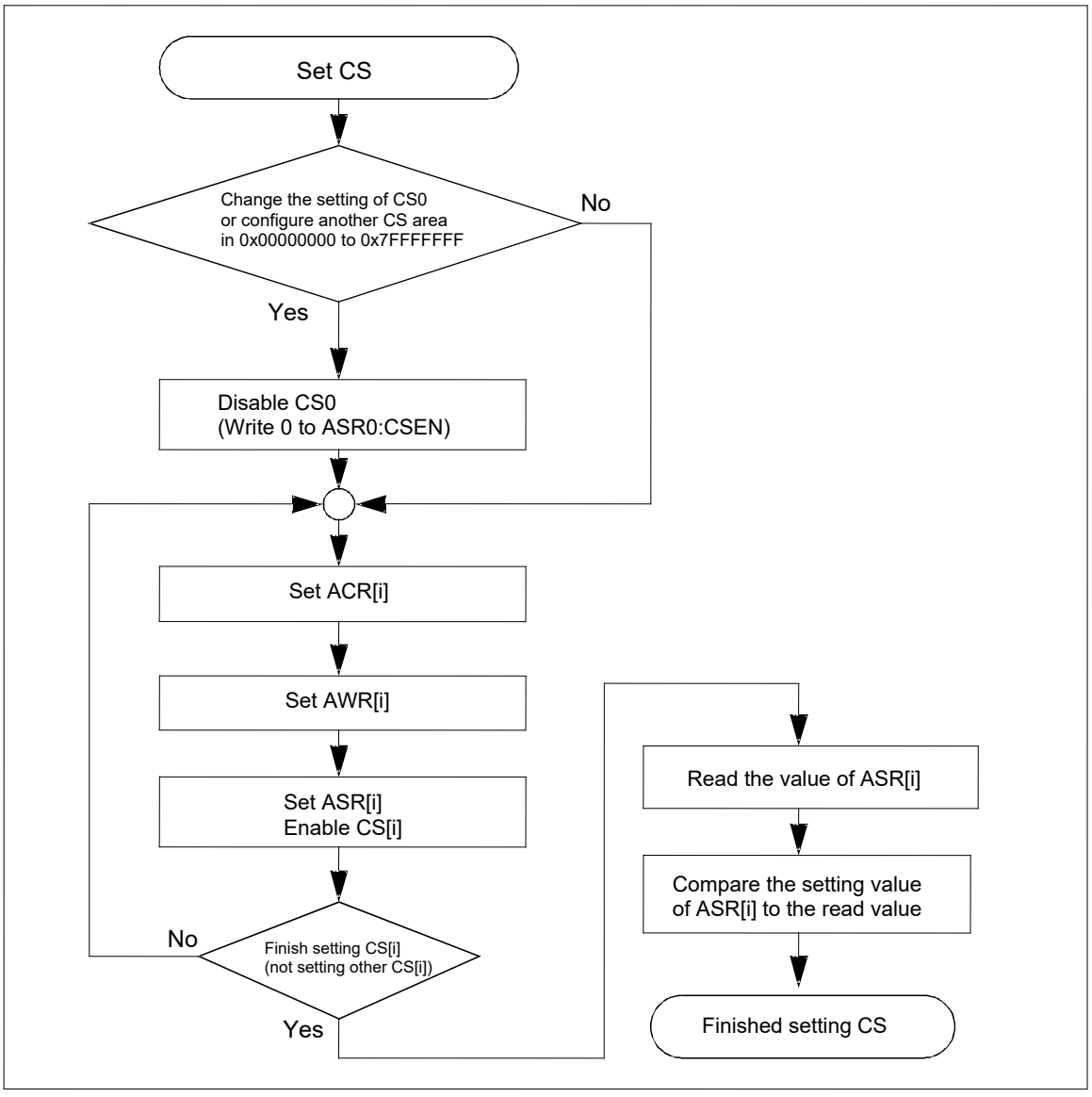
This section explains the CS setting method.

Notes:

- Perform the CS configuration during the initialization settings after reset, and do not change the settings thereafter.
- In models with built-in ROM, perform changes and settings of CS area in the initial settings program located in ROM.
- In models without built-in ROM, because instruction fetch after reset is performed in the CS0 area, in order to change the CS0 area first transfer the CS setting program to the built-in RAM and then branch to the program area in built-in RAM to configure the CS area.
- Operation is not guaranteed if the settings related to a CS area are changed while the CS area is being accessed.

The flow for configuring CS is shown below.

Figure 36-10. CS Setting Flow



- Disabling CS0

  In order to change CS0, CS0 first needs to be disabled. Write 0x0 to ASR0 as a word.

- Setting ACR

  The bus width, bus type, etc. of the CS area can be configured.

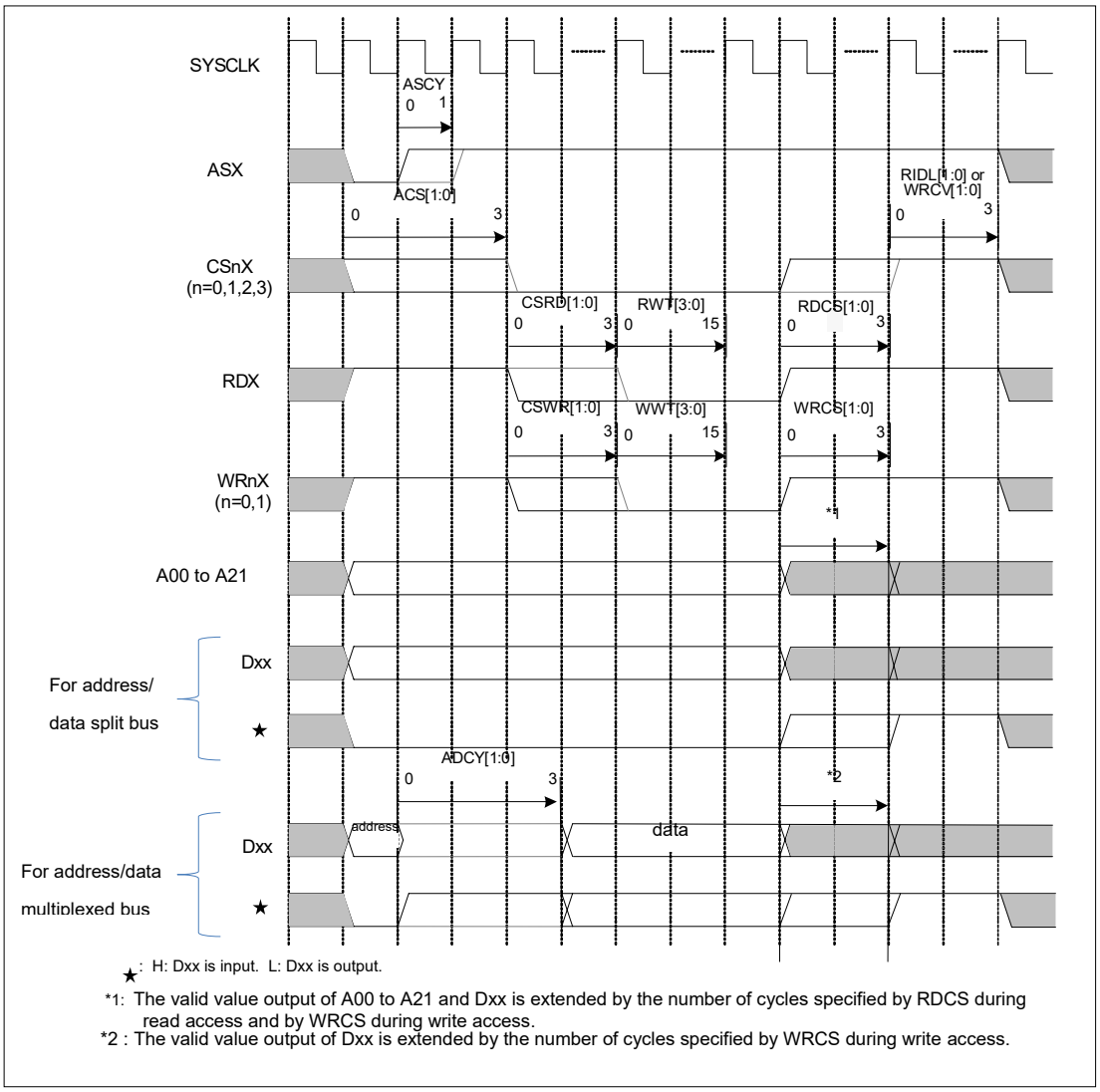
  1. The data bus width of the configured CS area can be selected from 8 bits and 16 bits.
  2. The address output type can be selected from normal output and shift output.
  3. The bus type can be selected from address/data split bus and address/data multiplexed bus.

  The above setting values are written to ACR as word units.

- Setting AWR

  The parameters that determine the output timing of the external bus signals and whether the RDY pin function is enabled or disabled can be configured. The setting values are written to AWR as words.

Figure 36-11. Parameters that can be Configured in AWR



★: H: Dxx is input. L: Dxx is output.

*1: The valid value output of A00 to A21 and Dxx is extended by the number of cycles specified by RDCS during read access and by WRCS during write access.

*2 : The valid value output of Dxx is extended by the number of cycles specified by WRCS during write access.

- List of Parameter

Parameter Nme	Description
RWT[3:0]	Sets the auto wait cycle count during the read access cycle. This is configured when you want to extend the read access cycle.
WWT[3:0]	Sets the auto wait cycle count during the write access cycle. This is configured when you want to extend the write access cycle.
RIDL[1:0]	Sets the idle cycle count after the read access. RIDL is configured in order to prevent conflicts on the data bus between the read data from a device with a long output off time and the data of the subsequent access.
WRCV[1:0]	Sets the write recovery cycle count. This is configured to control access to devices that have limits on the interval when performing an access after a write access.
CSRD[1:0]	Sets the number of cycles after CSnX (n=0 to 3) is asserted until RDX is asserted. This is configured if setup time is required for asserting CSnX when RDX is asserted during read access.
RDCS[1:0]	Sets the number of cycles after RDX is negated until CSnX (n=0 to 3) is negated. This is configured if hold time is required for the negation of CSnX after RDX is negated during read access.
CSWR[1:0]	Sets the number of cycles after CSnX is asserted until WRnX (n=0, 1) is asserted. This is configured if setup time is required for asserting CSnX when WRnX is asserted during write access.
WRCS[1:0]	Sets the number of cycles after WRnX is negated until CSnX is negated. This is configured if hold time is required for the negation of CSnX after WRnX is negated during write access.
ADCY[1:0]	Sets the number of cycles to extend address output to the data bus while address/data multiplexed bus is selected. Even if ADCY is set to "00", if ASCY is set to "1" then the address output cycle is extended by 1 cycle. Set this to"00" when the address/data split bus is selected.
ACS[1:0]	Sets the number of delay cycles from outputting A00 to A21 and ASX to outputting CSnX. This is used when the address for CSnX assert needs setup time, or when CSnX edges are required when accessing the same chip select area in sequence.
ASCY	Sets the number of ASX assert extensions cycles.
RDYE	Sets whether the wait insertion function by external RDY pin is enabled or disabled.

- Setting ASR

  The following settings are made using ASR.

  1. Configure the CS areas.
  2. Select whether writes are enabled or disabled.
  3. Select the byte ordering.
  4. Enable the CS.

  The above setting values are written to ASR as words.

  The CS area settings are explained below.

  1. Determine the size of the CS area and select the value of ASZ[3:0] from "36.4.1 CS Area Setting Registers: ASR0 to ASR3 (Area Setting Register 0-3)".
  2. Set the CS area start address. The starting address is configured by setting the upper bits of the address in SADR. However, the starting address has the boundaries determined in advance depending on the size of the area specified in the following table. Set the valid bits of SADR according to "36.4.1 CS Area Setting Registers: ASR0 to ASR3 (Area Setting Register 0-3)". Set invalid SADR bits to "0".

- The Size of the CS Area and the Aetting of ASZ and SADR

The size of the CS area	ASZ[3:0]	The valid SADR bit
64KB	0000	SADR[31:16]
128KB	0001	SADR[31:17]
256KB	0010	SADR[31:18]
512KB	0011	SADR[31:19]
1MB	0100	SADR[31:20]
2MB	0101	SADR[31:21]
4MB	0110	SADR[31:22]
8MB	0111	SADR[31:23]
16MB	1000	SADR[31:24]
32MB	1001	SADR[31:25]
64MB	1010	SADR[31:26]
128MB	1011	SADR[31:27]
256MB	1100	SADR[31:28]
512MB	1101	SADR[31:29]
1GB	1110	SADR[31:30]
2GB(initial value of ASR0)	1111	SADR[31]

Note:

Arrange each of the CS areas such that they do not overlap. Operation is not guaranteed if the CS areas are overlapping.

An example of the values set in SADR and ASZ and the actually allocated CS areas is shown below.

Setting example

CS0 settings

ASR0:ASZ[3:0]=0010
ASR0:SADR[31:16]=0x000C

→ 0x000C0000 to 0x000FFFFF becomes the CS0 area.

CS1 settings

ASR1:ASZ[3:0]=0000
ASR1:SADR[31:16]=0x0006

→ 0x00060000 to 0x0006FFFF becomes the CS1 area.

CS2 settings

To allocate the space from 0x00110000 to 1MB:

Set ASZ[3:0]=0100 to allocate a space of 1MByte. At that time, the SADR enable bit is [31:20]. SADR[19:16] is not the target of the comparison with the address. Therefore, the starting address of the CS2 area is 0x00100000 rather than 0x00110000.

ASR2:ASZ[3:0]=0100
ASR2:SADR[31:16]=0x0010

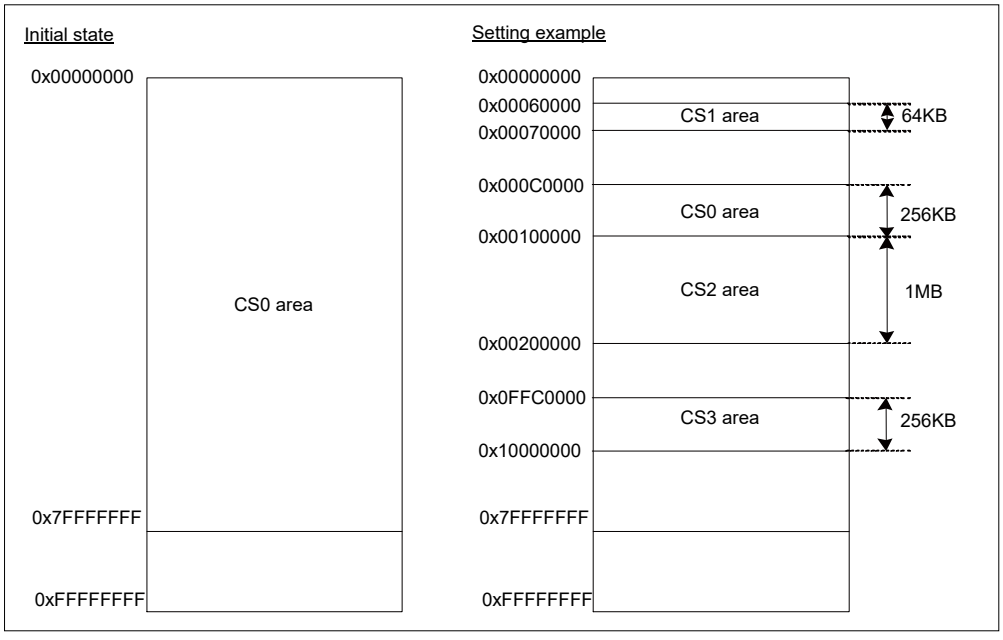
→ 0x00100000 to 0x001FFFFF becomes the CS2 area.

CS3 settings

ASR3:ASZ[3:0]=0010
ASR3:SADR[31:16]=0x0FFC

→ 0x0FFC0000 to 0x0FFFFFFF becomes the CS3 area.

Figure 36-12. Setting Example



- Reading and Comparing ASR

  After configuring the required ACR, AWR, and ASR settings for a CS, read the ASR which was configured last and compare to the set value in order to ensure that the CS settings will apply to subsequent accesses.

- CS Settings and Update Sample Program

  This section shows a CS configuration sample program that configures CS1.

Figure 36-13. CS1 Settings Sample Program

■ ACR1 Setting Example

Shows the setting values for the following table.

Data bus width	16bit
Address output type	Normal
Bus type	Address/data multiplexed bus
Write signal type	Write type 0

Bits other than the above setting bits are Reserved and are set to 0.

ACR1 setting value : 0x40

■ AWR1 Setting Example

Shows the setting values for the following table.

RWT	3 cycles
WWT	4 cycles
RIDL	2 cycles
WRCV	3 cycles
CSRD	1 cycle
RDCS	1 cycle
CSWR	2 cycles
WRCS	2 cycles
ADCY	Address/data multiplexed bus setting
ACS	0 cycle
ASCY	0 cycle
RDYE	Invalid

Bits other than the above setting bits are Reserved and are set to 0.

AWR1 setting value : 0x034b5a00

■ASR1 Setting Example

- CS1 area size : 64Kbyte
- CS1 area address: 0x0040_0000 to 0x0040_FFFF
- Write enable
- Big endain
- CS1 valid

ASR1 setting value : 0x00400005

■Program Example

```
_disable_CS0
ldi     #_ASR0, r0          //# _ASR0 is the ASR0 address value
ldi     0x0, r1
st      r1, @r0
_set_ACR1
ldi     #_ACR0, r0
ldi     #0x40, r1           // #_ACR1 is the ACR1 address value
st      r1, @r0             // Set ACR1 to 0x40
_set_AWR1
ldi     #_AWR1, r0          // #_AWR1 is the AWR1 address value
ldi     #0x034b5a00 r1
st      r1, @r0             // Set AWR1 to 0x034b5a00
_set_ASR1
ldi     #_ASR1, r0          // #_ASR1 is the ASR1 address value
ldi     #0x00400005 r1
st      r1, @r0             // Set ASR0 to 0x00400005
ld      @r0, r2
cmp     r1, r2              // Check the setting value of ASR1
```

36.5.11 Example of Connecting to Asynchronous Memory

This section shows an example of connecting to asynchronous memory.

This section shows an example of connecting external bus pins to asynchronous memory.

Figure 36-14. Example 1 of Connection to SRAM (8-bit SRAM×2)

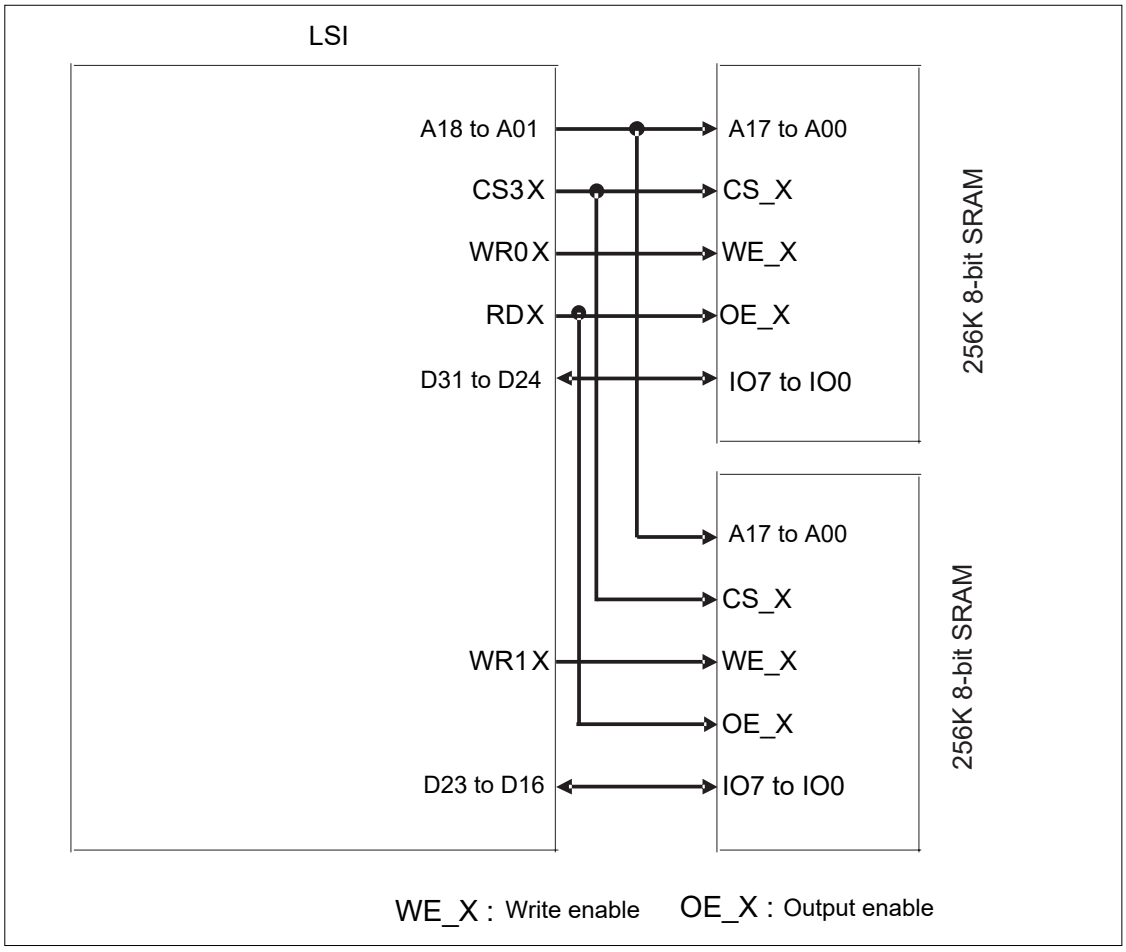
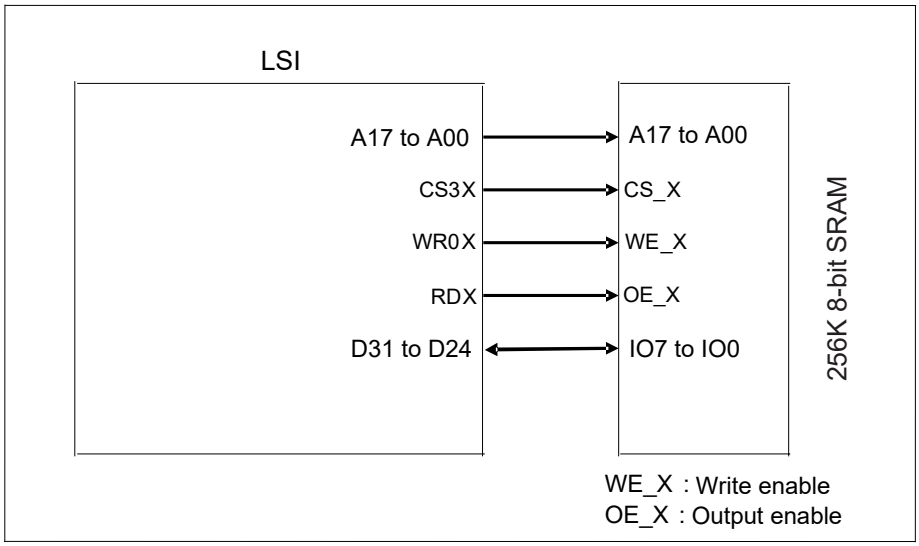


Figure 36-15. Example 2 of Connection to SRAM (8-bit SRAM×1)



36.5.12 Example of Connection to Little Endian Device

This section shows an example of connection to little endian device.

This section shows the method of connecting the data bus and byte enable signals to a little endian device.

Figure 36-16. 16-bit Bus Width

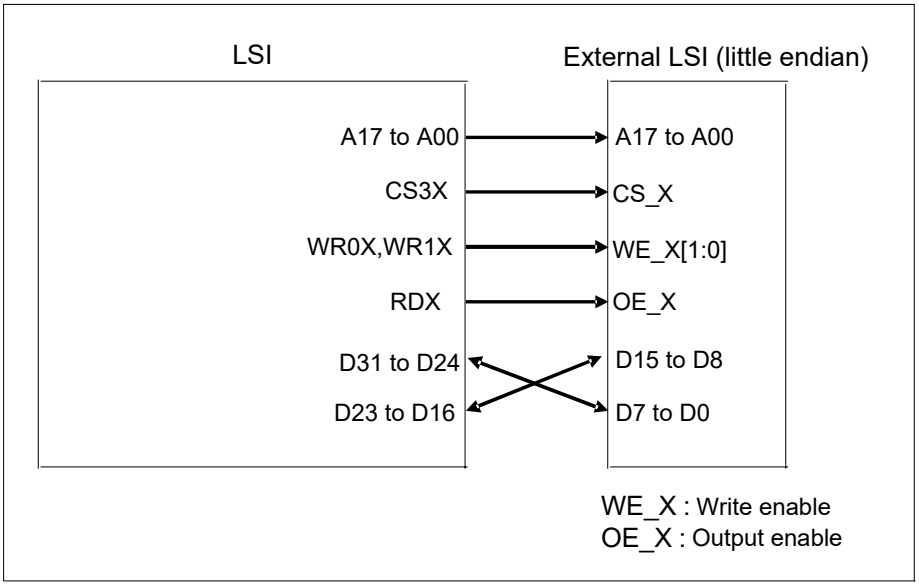
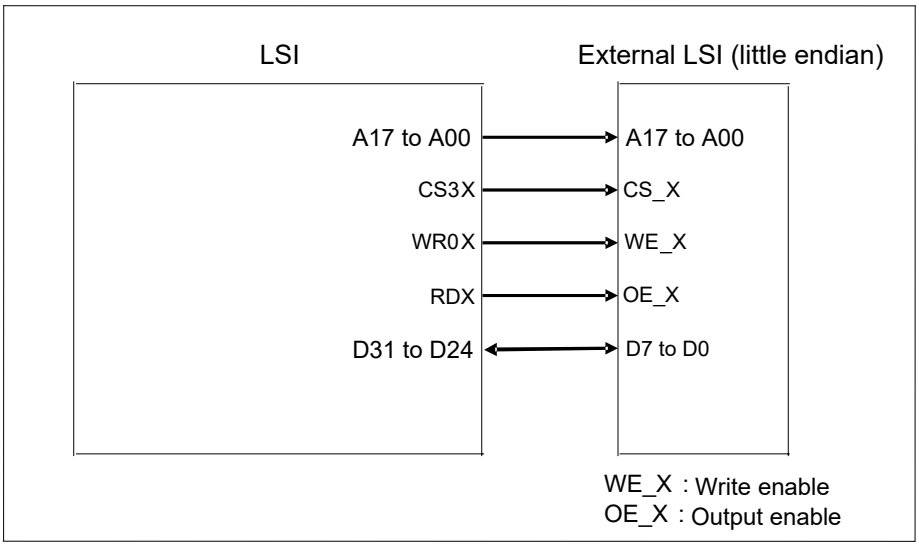


Figure 36-17. 8-bit Bus Width



37. Bus Performance Counters

This chapter explains the bus performance counters.

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 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.

37.2 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 × 3 channels (BPC-A, BPC-B, BPC-C)
- Overflow detection: None
- Counter value rewrite: Allowed

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

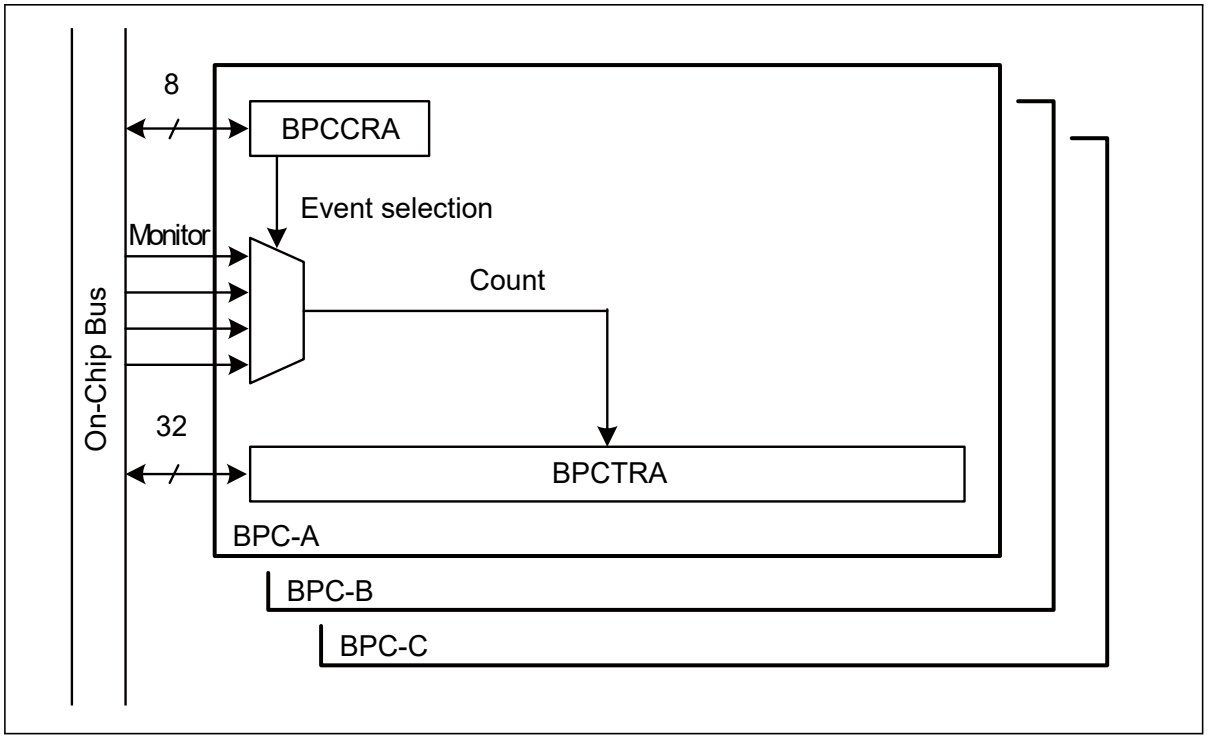
One of the following operations can be selected for counting in each channel

- Specific bus master (CPU, DMAC, other, or all)
- Specific target (ICH, MCH, other, or all)

37.3 Configuration

This section explains the configuration of the bus performance counters.

Figure 37-1. Block Diagram



37.4 Registers

This section explains the registers of the bus performance counters.

Table 37-1. Registers 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

37.4.1 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
0101 to 1111	Reserved

[bit1, bit0] SLV[1:0] (slave select): Slave selection

These bits select the slave for the events which are measured by BPC.

SLV[1:0]	Slave
00	All slaves (initial value)
01	MCH (registers, external bus)
10	ICH (peripherals)
11	Anything other than MCH/ICH

37.4.2 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

37.4.3 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

37.4.4 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 register 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.

37.4.5 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

37.4.6 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

37.5 Operation

This section explains the operations of the bus performance counters.

37.5.1 Setting
37.5.2 Starting and Stopping
37.5.3 Operation
37.5.4 Measurement and Result Processing

37.5.1 Setting

This section explains the setting of the bus performance counters.

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 37-2. List of BPC Settings

FUNC[1:0]	MST[3:0]	SLV[1:0]	Target Event
01	0000	00	Read access 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 access 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

FUNC[1:0]	MST[3:0]	SLV[1:0]	Target Event
10	0000	00	Write access 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 access from XBS
		01	MCH write from XBS
		10	ICH write from XBS
		11	Other than MCH/ICH write from XBS
	0100	00	Write access from DMAC
		01	MCH write from DMAC
		10	ICH write from DMAC
		11	Other than MCH/ICH write from DMAC
11	0000	00	Wait cycle 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

37.5.2 Starting and Stopping

This section explains the starting and stopping of the bus performance counters.

The target event count is started by setting the FUNC[1:0] field 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".

37.5.3 Operation

This section explains the operation of the bus performance counters.

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 after a reset occurs.

37.5.4 Measurement and Result Processing

This section explains the measurement and result processing of the bus performance counters.

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 program
- Measure a reference time base

These are explained below.

- Measuring between two points in a 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

- Measuring the 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 DMA 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 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.

38. CRC

This chapter explains the CRC.

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 CRC (Cyclic Redundancy Check).

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.

38.2 Features

This section explains features of the CRC (Cyclic Redundancy Check).

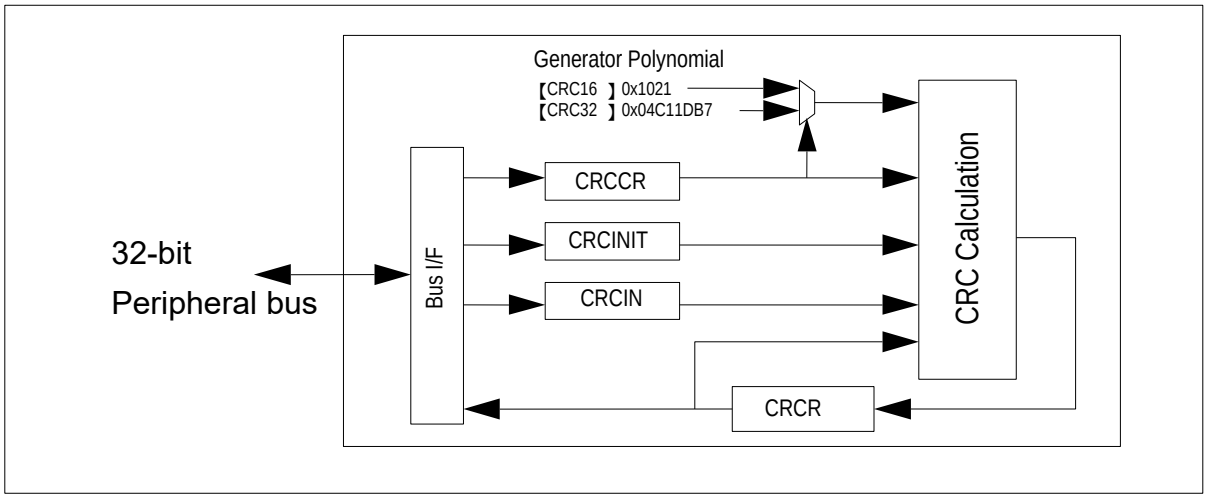
This module calculates CCITT CRC16 and IEEE-802.3 CRC32. This module cannot calculate CRC values based on other generator polynomials because the generator polynomials of this module are fixed for the values of CCITT CRC16 and IEEE-802.3 CRC32.

- CCITT CRC16 generator polynomials : 0x1021
- IEEE-802.3 CRC32 generator polynomials : 0x04C11DB7

38.3 Configuration

This section explains the configuration of the CRC (Cyclic Redundancy Check).

Figure 38-1. Block Diagram



38.4 Registers

This section explains registers of the CRC (Cyclic Redundancy Check).

Table 38-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x1130	Reserved			CRCCR	CRC control register
0x1134	CRCINIT				CRC initial value register
0x1138	CRCIN				CRC input data register
0x113C	CRCR				CRC register

38.4.1 CRC Control Register: CRCCR

The bit configuration of the CRC control register is shown below.

This register controls the CRC calculation.

CRCCR: Address 1133H (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

This bit must always be written to "0".

[bit6] FXOR (Final XOR): Final XOR Control bit

The result is calculated XOR of "XOR value" and itself. The XOR values are ALL"H" 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 (8 bits). When this bit is "0", MSB First is applied, and when this bit is "1", LSB First is applied. Four patterns of process order can be specified by combining the LTLEND bit setting.

[bit2] LTLEND (Litttle-Endian): Byte order setting bit

This bit sets byte orders. This bit 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.

38.4.2 CRC Initial Value Register: CRCINIT

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)

	bit31	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[31:0] (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.

38.4.3 CRC Input Data Register: CRCIN

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)

	bit31	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[31:0] (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 or 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.

38.4.4 CRC Register: CRCR

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)

	bit31	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[31:0] (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.

38.5 Operation

This section explains the operation of the CRC (Cyclic Redundancy Check).

38.5.1 CRC Definition
38.5.2 Reset Operation
38.5.3 Initialization
38.5.4 Byte and Bit Orders
38.5.5 CRC Calculation Sequence
38.5.6 Examples

38.5.1 CRC Definition

This section explains the CRC (Cyclic Redundancy Check) definition.

- 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)		

38.5.2 Reset Operation

This section explains the reset operation of the CRC (Cyclic Redundancy Check).

To reset, set 0xFFFF_FFFF to the CRC initial value register (CRCINIT) and CRC register (CRCR). Other registers are cleared to "0".

38.5.3 Initialization

This section explains the initialization of the CRC (Cyclic Redundancy Check).

In initialization by CRCCR:INIT, the value of the initial value register is loaded to the CRC register (CRCR).

38.5.4 Byte and Bit Orders

This section explains the byte and bit orders of the CRC (Cyclic Redundancy Check).

This section explains the byte and bit orders using examples. Inputs the following one 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	110101011	0000000
(first)			(last)

Notes:

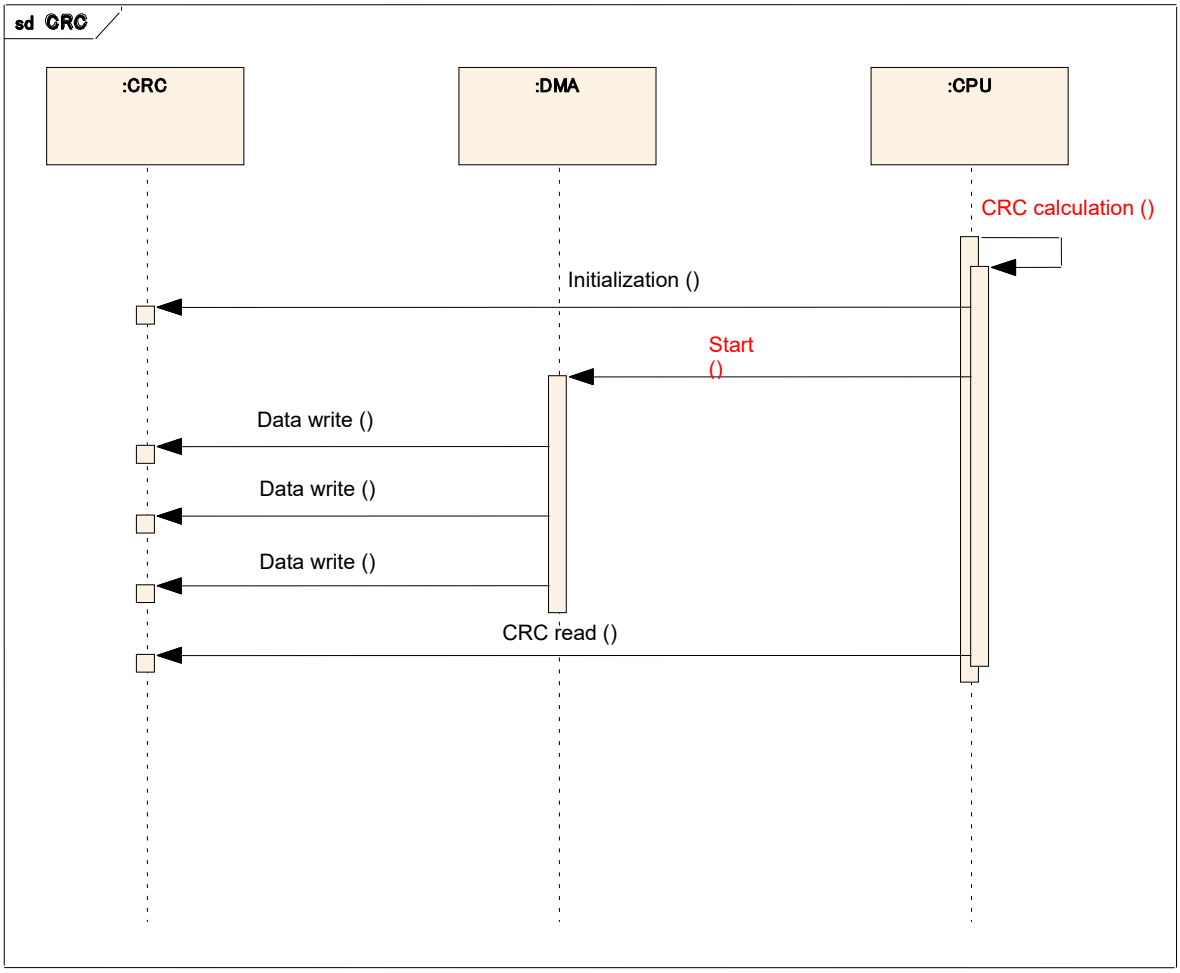
- When CRCCR:CRCLTE=1 is true, the byte order for the CRC result is changed in 32-bit width both for CRC16 and CRC32.
- Note that output position for CRC16 is bit31 to bit16.

38.5.5 CRC Calculation Sequence

This section explains the CRC calculation sequence of the CRC.

The sequence for the CRC calculation is shown below. In the following explanation, the CRC 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 ALL "H" is acceptable, the setting for the initial value register (CRCINIT) can be omitted.)

Figure 38-2. 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 CRC 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).

38.5.6 Examples

This section explains examples of the Cyclic Redundancy Check (CRC) operation.

38.5.6.1 Example 1 CRC16, Fixed Byte Input

38.5.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths

38.5.6.3 Example 3 CRC32, Byte Order, Big-endian

38.5.6.4 Example 4 CRC32, Byte Order, Little-endian

38.5.6.1 Example 1 CRC16, Fixed Byte Input

Example 1 CRC16 and fixed byte input are shown below.

Figure 38-3. 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);

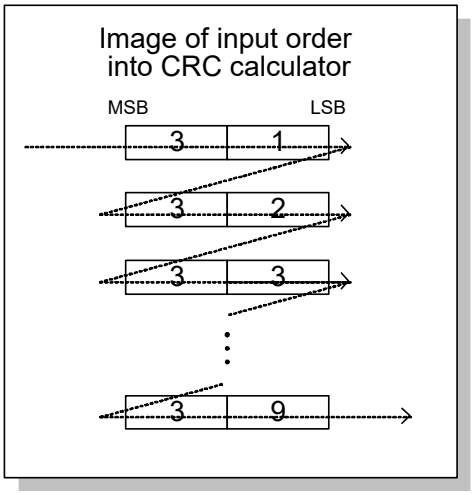
// check result
assert (data == 0x0000);
```

(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_READ (Half-word reading) is +2 in the example.

38.5.6.2 Example 2 CRC16, Mixture of Different Input Bit Widths

Example 2 CRC16 and Mixture of Different Input Bit Widths are shown below.

Figure 38-4. 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, 0x3556);
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);

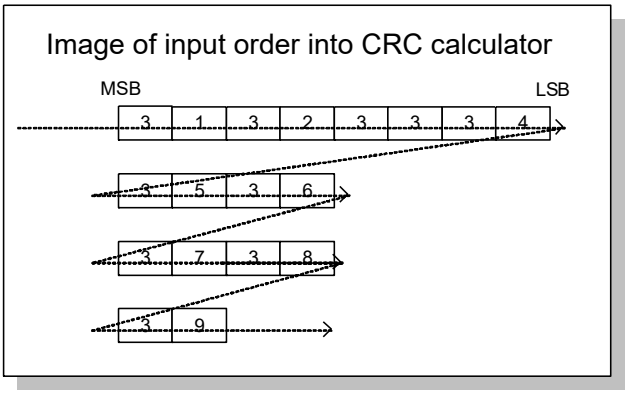
// check result
assert (data == 0x0000);
```

(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.

38.5.6.3 Example 3 CRC32, Byte Order, Big-endian

Example 3 CRC32, the byte order and big-endian are shown below.

Figure 38-5. 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);

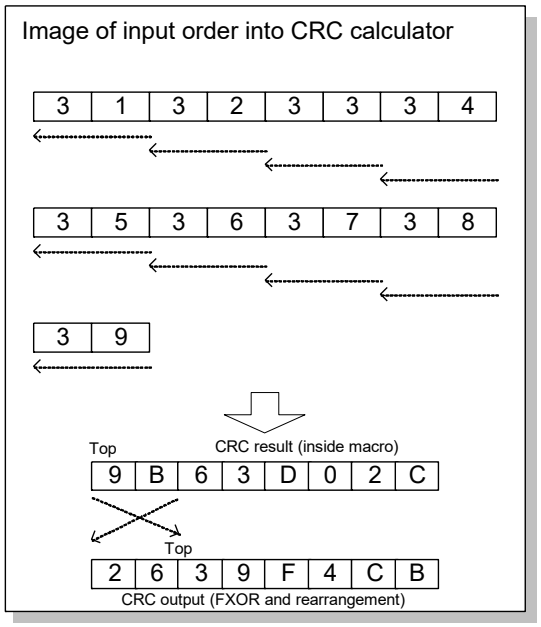
// check CRC result
assert (data == 0x2639F4CB);
 // <- big endian & LSB First
```

(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.

38.5.6.4 Example 4 CRC32, Byte Order, Little-endian

Example 4 CRC32, the byte order and Little-endian are shown below.

Figure 38-6. 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);

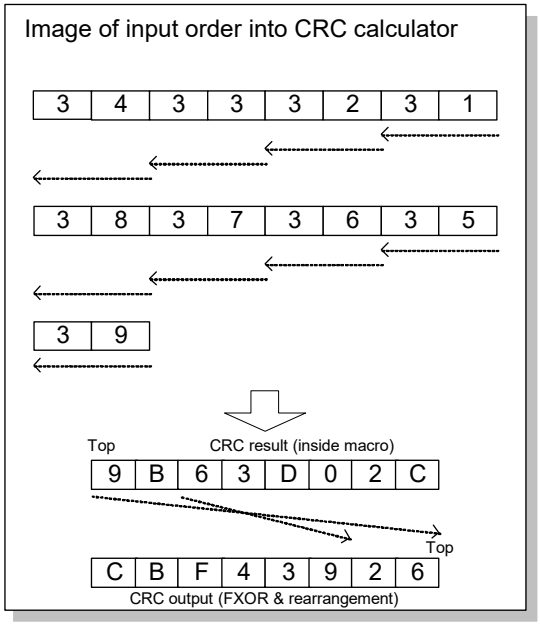
// check result
assert (data == 0xCBF43926);
 // <- little endian & LSB First
```

(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 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.

39. RAMECC

This chapter explains the RAMECC.

39.1 Overview

39.2 Features

39.3 Configuration

39.4 Registers

39.5 Operation

39.1 Overview

This section provides the overview of the RAMECC.

The RAMECC has a dual function in order to increase RAM’s tolerance to soft error while the CPU is writing to or reading from the RAM. One function is to correct a single bit error in 1-byte units, and the other is to generate and check the code to detect a double-bit error in 1-byte units as well.

39.2 Features

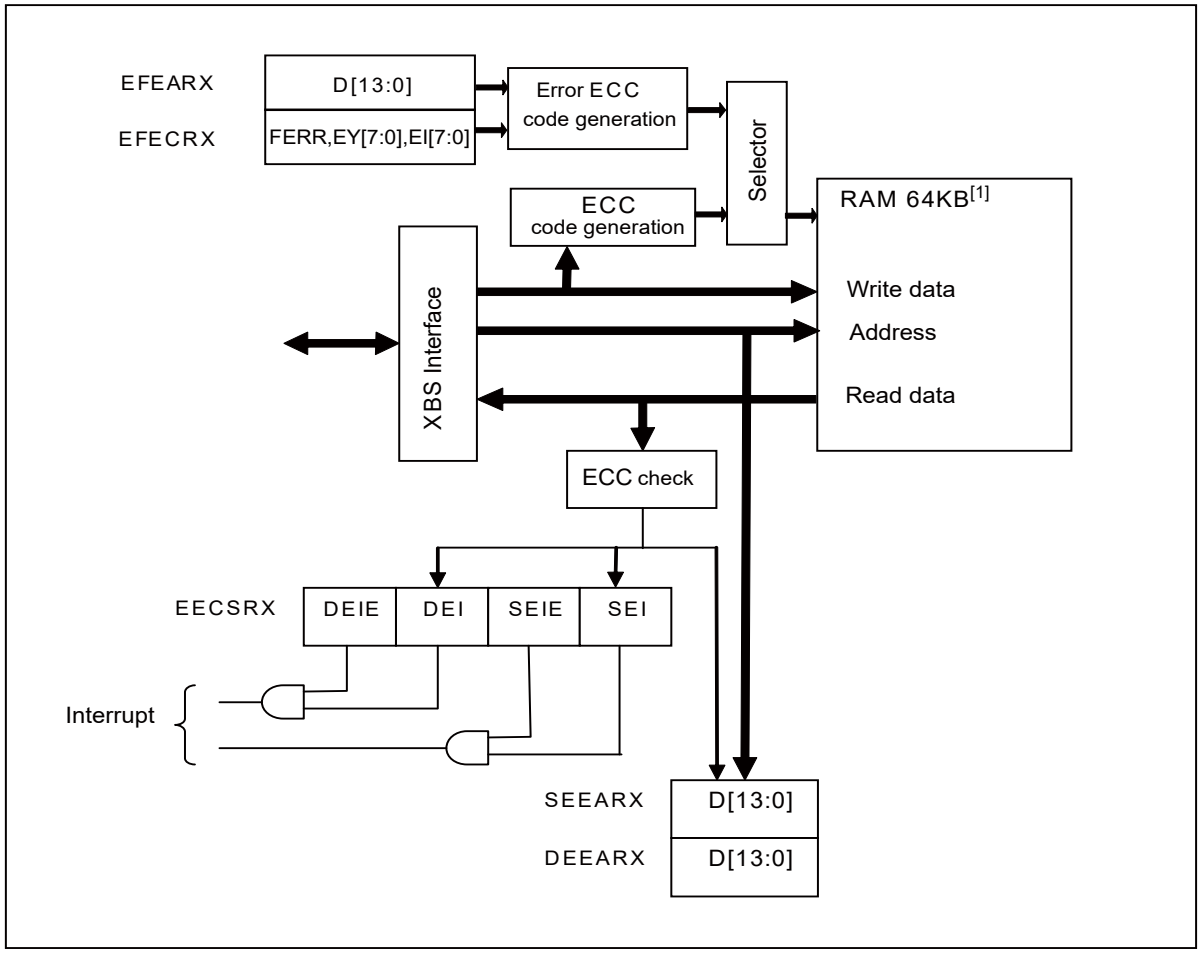
This section explains features of the RAMECC.

- Target RAM:
  XBS RAM: 40KB (CY91F575)
  64KB (CY91F577)
  Backup RAM: 8KB
- ECC: 5-bit ECC is added by byte. Single-bit error correction and double-bit error detection are enabled.
- Interrupt function: Single bit error is perceived and RAM single bit error interrupt signal is generated, and a double bit error is perceived and RAM double bit error interrupt signal is generated.
- Test function: A pseudo error occurs for the software debugging.

39.3 Configuration

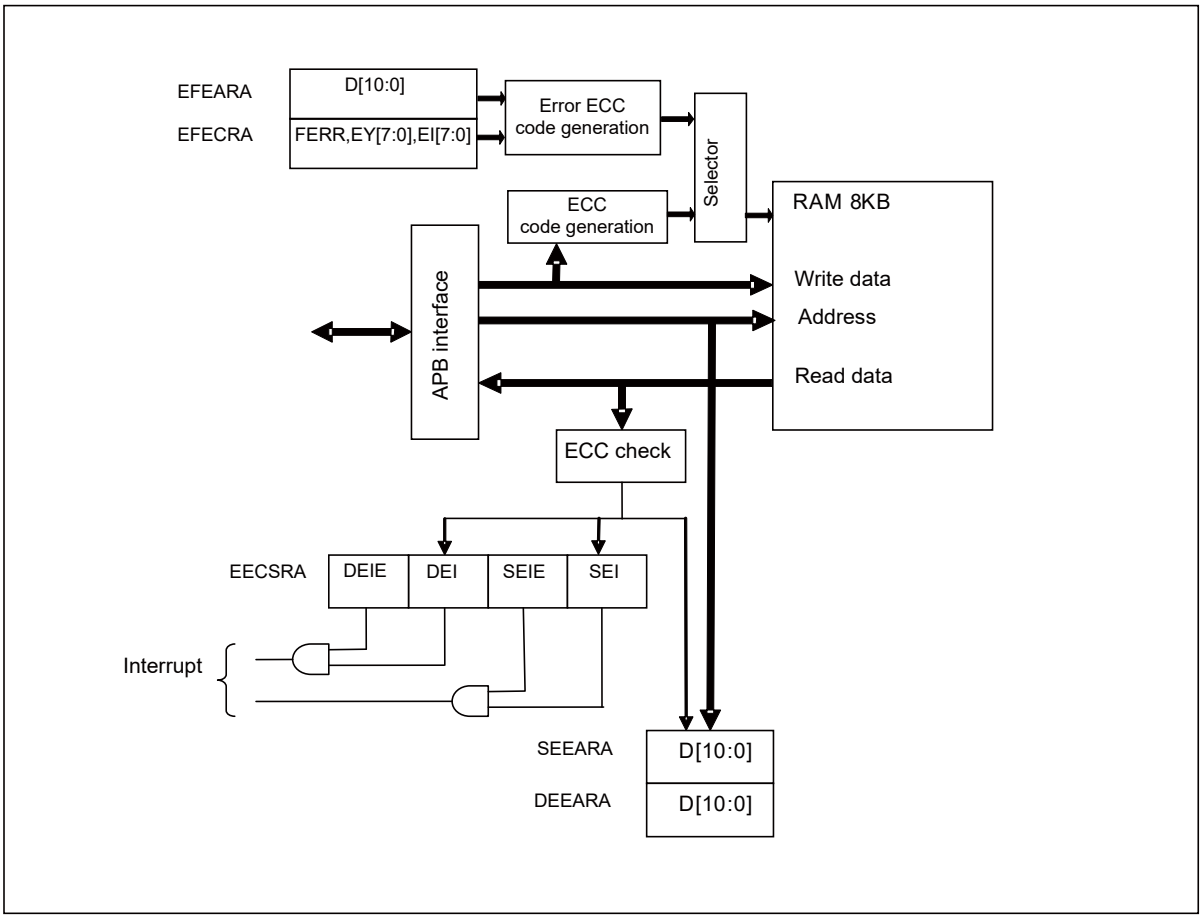
This section shows the configuration of the RAMECC.

Figure 39-1. Block Diagram (XBS RAM)



[1]: 40KB for CY91F575

Figure 39-2. Block Diagram (Backup RAM)



39.4 Registers

This section explains the registers of the RAMECC.

Table 39-1. 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 address register XBS RAM
0x2408	Reserved	EFECRX			ECC false error control register XBS RAM
0x3000	SEEARA		DEEARA		Single-bit ECC error address register BACKUP-RAM Double-bit ECC error address register BACKUP-RAM
0x3004	EECSRA	Reserved	EFEARA		ECC error control register BACKUP-RAM ECC false error address register BACKUP-RAM
0x3008	Reserved	EFECRA			ECC false error control register BACKUP-RAM

39.4.1 ECC Error Control Register XBS RAM: EECSRX (ECC Error Control and Status Register XBS RAM)

The bit configuration of the ECC error control register XBS RAM is explained.

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 such events.

EECSRX: Address 2404_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				DEIE	DEI	SEIE	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/W	R(RM1),W

[bit7 to bit4] Reserved

Always write "0". Reading these bits returns "0".

[bit3] DEIE: Double-bit error factor interrupt enable bit

DEIE	Description of Setting
0	Disables interrupts.
1	Enables interrupts.

[bit2] DEI: Double-bit error occurrence bit

DEI	Read	Write
0	Double-bit error has not occurred.	Clears this bit.
1	Double-bit error has occurred.	No effect.

[bit1] SEIE: Single-bit error factor interrupt enable bit

SEIE	Description of Setting
0	Disables interrupts.
1	Enables interrupts.

[bit0] SEI: Single-bit error occurrence bit

SEI	Read	Write
0	Single-bit error has not occurred.	Clears this bit.
1	Single-bit error has occurred.	No effect.

39.4.2 Single-bit ECC Error Address Register XBS RAM: SEEARX

The bit configuration of the single-bit ECC error address register XBS RAM is explained.

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		D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	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, bit14] Reserved

Reading these bits returns "0".

[bit13 to bit0] D13 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) = (0x00010000) + (Offset indicated by SEEARX +2b’00)

39.4.3 Double-bit ECC Error Address Register XBS RAM: DEEARX

The bit configuration of the double-bit ECC error address register XBS RAM is explained.

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 2402H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved		D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	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, bit14] Reserved.

Reading these bits returns "0".

[bit13 to bit0] D13 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) = (0x00010000)+(Offset indicated by DEEARX +2b’00)

39.4.4 ECC False Error Generation Address Register XBS RAM: EFEARX (ECC False Error Address Register XBS RAM)

The bit configuration of the ECC false error (a pseudo ECC error) generation address register (EFEARX) is explained.

The ECC false error (a pseudo ECC error) generation address register (EFEARX) specifies the address where a false error (a pseudo error) is generated.

EFEARX: Address 2406_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved		D13	D12	D11	D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	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,bit14] Reserved

Always write "0". Reading these bits returns "0".

[bit3 to bit0] D13 to D0: False error generation address setting bits

These bits set the address where false ECC error (a pseudo ECC error) is caused.

ECC error is caused because the write access to this address is generated at EFECRX:FERR ="1", and the written data contains the error according to the setting of EFECRX by intention.

39.4.5 ECC False Error Generation Control Register XBS RAM: EFECRX (ECC False Error Control Register XBS RAM)

The bit configuration of the ECC false error (a pseudo ECC error) generation control register (EFECRX) is explained.

The ECC false error (a pseudo ECC 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". Reading these bits returns "0".

[bit16] FERR: false error generation enable bit

FERR	Description of Setting
0	false error (a pseudo error) generation disable
1	false error (a pseudo error) enable

[bit15 to bit8] EY7 to EY0: false error generation byte setting bits

These bits specify byte position of the target that causes false ECC error (a pseudo ECC error).

EYn	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]

For example, when EY2 is filled with "1" and other EYn's are filled with "0", the target byte where a false error (a pseud 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

These bits specify bit position of the target that causes false ECC error (a pseudo ECC error).

EIn	Target Bit on Byte
EI0	bit0
EI1	bit1
EI2	bit2
EI3	bit3
EI4	bit4
EI5	bit5
EI6	bit6
EI7	bit7

For example, when both EY2 and EI4 are filled with "1", and others are filled with "0", the target bit where a false error (a pseudo 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 (a pseudo 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 (a pseudo error) is generated are RAM data[28] and RAM data[20]. As a result, a single bit error can be corrected in each byte.

39.4.6 ECC Error Control Register Backup-RAM: EECSRA (ECC Error Control and Status Register Backup-RAM)

The bit configuration of the ECC error control register BACKUP-RAM is explained.

During the ECC check of Backup 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 such events.

EECSRA: Address 3004_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved				DEIE	DEI	SEIE	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/W	R(RM1),W

[bit7 to bit4] Reserved

Always write "0". Reading these bits returns "0".

[bit3] DEIE: Double-bit error factor interrupt enable bit

DEIE	Description of Setting
0	Disables interrupts.
1	Enables interrupts.

[bit2] DEI: Double-bit error occurrence bit

DEI	Read	Write
0	Double-bit error has not occurred.	Clears this bit.
1	Double-bit error has occurred.	No effect.

[bit1] SEIE: Single-bit error factor interrupt enable bit

SEIE	Description of Setting
0	Disables interrupts.
1	Enables interrupts.

[bit0] SEI: Single-bit error occurrence bit

SEI	Read	Write
0	Single-bit error has not occurred.	Clears this bit.
1	Single-bit error has occurred.	No effect.

39.4.7 Single-bit ECC Error Address Register Backup-RAM: SEEARA

The bit configuration of the single-bit ECC error address register BACKUP-RAM is explained.

When the single-bit error correction is performed during the ECC check of Backup RAM, this register maintains the address at which it occurred.

SEEARA: Address 3000$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved					D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	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 to bit11] Reserved

Reading these bits returns "0".

[bit10 to bit0] D10 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 events 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 Backup RAM.
(Absolute address) = (0x00004000) + (Offset indicated by SEEARA +2b’00)

39.4.8 Double-bit ECC Error Address Register BACKUP-RAM: DEEARA

The bit configuration of the double-bit ECC error address register BACKUP-RAM is explained.

When the double-bit error detection is performed during the ECC check of Backup RAM, this register maintains the address at which it occurred.

DEEARA: Address 3002$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved					D10	D9	D8
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	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 to bit11] Reserved

Reading these bits returns "0".

[bit10 to bit0] D10 to D0: Double-bit error occurrence address bits

When 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 Backup RAM.
(Absolute address) = (0x00004000) + (Offset indicated by DEEARA +2b’00)

39.4.9 ECC False Error Generation Address Register Backup-RAM: EFEARA (ECC False Error Address Register Backup-RAM)

The bit configuration of the ECC false error (a pseudo ECC error) generation address register BACKUP-RAM is explained.

The ECC false error (a pseudo ECC error) generation address register (EFEARA) specifies the address where a false error (a pseudo error) is caused.

EFEARA: Address 3006_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved					D10	D9	D8
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	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 to bit11] Reserved

Always write "0". Reading these bits returns "0".

[bit10 to bit0] D10 to D0: False error generation address setting bits

These bits set the address where false ECC error (a pseudo ECC error) is caused.

ECC error is caused because the write access to this address is generated at EFECRA:FERR ="1", and the written data contains the error according to the setting of EFECRA by intention.

39.4.10 ECC False Error Generation Control Register Backup-RAM: EFECRA (ECC False Error Control Register Backup-RAM)

The bit configuration of the ECC false error (a pseudo ECC error) generation control register BACKUP-RAM is explained.

The ECC false error (a pseudo ECC error) generation control register (EFECRA) specifies each false error by its byte position and its bit position where the false error is generated.

EFECRA: Address 3009_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". Reading these bits returns "0".

[bit16] FERR: false error generation enable bit

FERR	Description of Setting
0	false error (a pseudo error) generation disable
1	false error (a pseudo error) enable

[bit15 to bit8] EY7 to EY0: false error generation byte setting bits

These bits specify byte position of the target that causes false ECC error (a pseudo ECC error).

EYn	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]

For example, when EY2 is filled with "1" and other EYn's are filled with "0", the target byte where a false error (a pseud 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

These bits specify bit position of the target that causes false ECC error (a pseudo ECC error).

EIn	Target Bit on Byte
EI0	bit0
EI1	bit1
EI2	bit2
EI3	bit3
EI4	bit4
EI5	bit5
EI6	bit6
EI7	bit7

For example, when both EY2 and EI4 are filled with "1", and others are filled with "0", the target bit where a false error (a pseudo 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 (a pseud 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 (a pseudo error) is generated are RAM data[28] and RAM data[20]. As a result, a single bit error can be corrected in each byte.

39.5 Operation

This section explains operations of the RAMECC.

39.5.1 ECC Generation

39.5.2 ECC Inspection

39.5.3 Interrupt by Error Detection

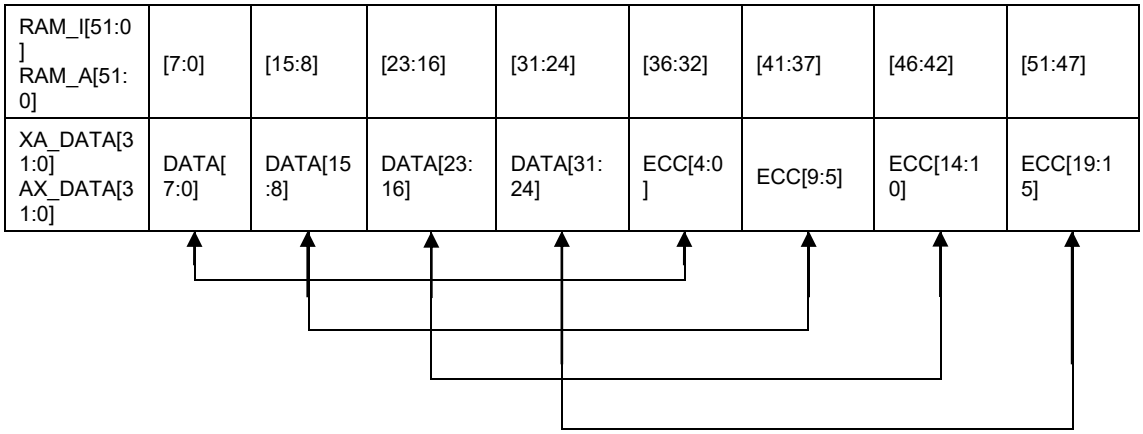
39.5.4 Test Function

39.5.1 ECC Generation

The ECC generation is explained.

ECC for the unit of 8-bit (1 byte) is generated to the data of 32-bit on the XBS interface.

Figure 39-3. Relation between XBS Data and RAM Data



The relation between the XBS data (XA_DATA[31:0] / AX_DATA[31:0]) and RAM data (RAM_I [51:0] / RAM_A[51:0]) is defined as shown in the figure above. The one connected by the arrow in the figure is a pair of DATA and ECC. (Example: 8-bit {XA_DATA[15:8] / AX_DATA[15:8]} is stored in {RAM_I[15:8] / RAM_A[15:8] } as RAM data, and ECC[9:5] corresponding to it is stored in { RAM_I[41:37]/ RAM_A[41:37]} as RAM data.)

Moreover, the (13,8) odd number weight sign shown in the table below is adopted as ECC sign matrix.

Table 39-2. ECC Sign Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	0	1	1	1	0	1	1	0	0	0	0
2	1	1	0	0	1	0	1	0	0	1	0	0	0
3	1	1	1	0	0	1	0	1	0	0	1	0	0
4	0	1	1	1	0	1	1	0	0	0	0	1	0
5	0	0	1	1	1	0	1	1	0	0	0	0	1

As a result, each bit of ECC can be calculated by requesting 5 or 6 exclusive-OR from the data of 8 bits.

This forms the inspection matrix, and the generation matrix is requested by the transposed matrix of 5 × 8 matrix up to eight rows and the combinations with the unit matrix.

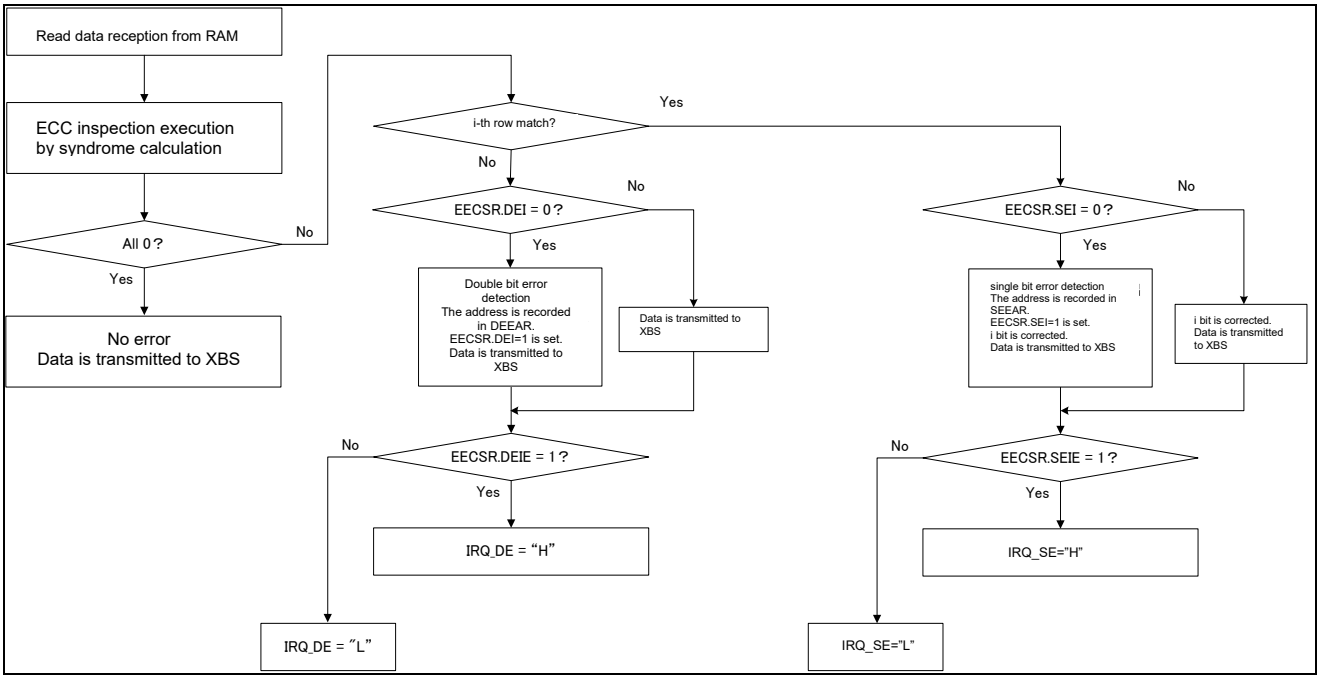
It is necessary to be going to generate ECC in one clock cycle after the XBS data is received, and to complete writing in RAM.

39.5.2 ECC Inspection

The ECC inspection is explained.

The inspection is executed by constructing the sign vector of 13-bit including ECC with the data that exists on RAM as shown in Figure 39-3, and using the inspection matrix of Table 39-2 according to the procedure of the figure below.

Figure 39-4. ECC Inspection Flow



39.5.3 Interrupt by Error Detection

This section explains the interrupt at the error detection.

When the ECC error is detected, the interrupt can be generated. Write "1" in the DEIE bit and the SEIE bit according to the usage to generate the interrupt, and set the RAMECC interrupt vector and the interrupt level.

Interrupt Factor	Interrupt Vector	Interrupt Level
SEI (RAM single-bit error interrupt)	#61($000FFF08_H$)	ICR45
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, SEI) is not automatically cleared, clear the flag forcibly with software before the status of the MCU returns from the interrupt. (Write "0" into the DEI bit and the SEI bit).

The interrupt at the NMI level is generated when a double-bit error is detected because the error data is read without any correction.

39.5.4 Test Function

Test function is explained.

An ECC false error (a pseudo ECC error) is generated in order to debug software.

The ECC false error (a pseudo ECC error) is generated in accordance with the following procedures:

1. The address where a false error (a pseudo error) is caused in ECC false error (a pseudo ECC error) generation address register (EFEARX/ EFEARA) is specified.
2. The byte and the bit are set by ECC false error (a pseudo ECC error) generation control register (EFECRX/EFECRA).
   a. Byte position in which a false error (a pseudo error) EFECRX:EY[7:0]/EFECRA:EY[7:0] is caused is specified.
   b. The bit position that causes a false error (a pseudo error) EFECRX:EI[7:0]/EFECRA:EI[7:0] is specified.
3. The FERR bit of ECC false error (a pseudo ECC error) generation control register (EFECRX/ EFECRA) is written "1".

The CPU starts writing the data, which intentionally includes error, into the address specified with FEEARX/FEEARA . The byte position and bit position in the address are specified with EY[7:0] and EI[7:0], respectively. Then the CPU reads the data subsequently, detecting the false error (a pseudo ECC error).

40. Multi Function Serial Interface

This chapter explains the multi function serial interface.

40.1 Overview

40.2 Features

40.3 Configuration

40.4 Registers

40.5 Operation of UART

40.6 Operation of CSIO

40.7 Operation of LIN-UART

40.8 Operation of I2C

40.1 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), LIN-UART (LIN Processing Hardware Attached Serial Interface), and I^2C serial communication function.

40.2 Features

This section explains features of the multi function serial interface.

This product is equipped with 4-channel multi function serial interface communication module. To use this device, you will select UART, CSIO, LIN-UART, or I^2C using the serial mode register (SMR).

Note:

The I^2C function supports only ch.0 and ch.1.

40.2.1 UART

This section explains UART of the multi function serial interface.

UART (asynchronous serial interface) is the general-purpose serial data communication interface designed to communicate with external devices asynchronously (start-stop synchronization). It supports the 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.

Name	Function
Data	■ Full-duplex double buffering(when FIFO is unused) ■ Transmission/reception FIFO (16 bytes each) (when FIFO is used)
Serial input	Execute over-sampling 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	5-9 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[1]
Interrupt request	■ Reception interrupt (Reception completed, framing error, overrun error, parity error[1]) ■ Transmission interrupt (transmission data empty, transmission bus idle) ■ Transmission FIFO interrupt (when the transmission FIFO is empty) ■ Both transmission and reception have the DMA function
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: 16 bytes, reception FIFO: 16 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

[1]: Parity error is for the normal mode only.

40.2.2 CSIO

This section explains CSIO of the multi function serial interface.

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 (16 bytes each).

Name	Function
Data buffer	■ Full-duplex double buffering(when FIFO is unused) ■ Transmission/reception FIFO (16 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 5-9 bits
Reception error detection	Overrun error
Interrupt request	■ Reception interrupt (reception completed, overrun error) ■ Transmission interrupt (transmission data empty, transmission bus idle) ■ Transmission FIFO interrupt (when the transmission FIFO is empty) ■ Both transmission and reception have the DMA transfer support function
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: 16 bytes, reception FIFO: 16 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

40.2.3 LIN-UART

This section explains LIN-UART of the multi function serial interface.

LIN-UART (LIN Communication Control UART) provides specific functions to support LIN bus. It is also equipped with the FIFO for transmission/reception (16 bytes each).

Name	Function
Data buffer	■ Full-duplex double buffering(when FIFO is unused) ■ Transmission/reception FIFO (16 bytes each) (when FIFO is used)
Serial input	Execute over-sampling for three times by the peripheral clock (PCLK) and determine the reception value by the majority of the sampling value.
Transfer mode	Asynchronous
Baud rate	■ Dedicate baud rate generator provided (comprising of 15-bit reload counter) ■ External clock can be adjusted by the reload counter
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 synch break detection) ■ Interrupt request for ICU (LIN synch field detected: LSYN) ■ Transmission FIFO interrupt (when the transmission FIFO is empty) ■ Both transmission and reception have the DMA function
LIN bus option	■ LIN protocol revision 2.1 is supported. ■ Master device operation ■ Slave device operation ■ LIN Synch break generation (can be changed to 13-16 bits) ■ Synch Delimiter generation (can be changed to 1-4 bits) ■ LIN Synch break detection ■ Detection of start/stop edges for LIN synch field connected to the input capture by input capture (See "Chapter: Input Capture".)
FIFO option	■ Transmission/reception FIFO equipped (transmission FIFO: 16 bytes, reception FIFO: 16 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

40.2.4 I²C

This section explains I²C of the multi function serial interface.

I²C interface supports buses among ICs, and runs as a master/slave device on the I²C bus. It is also equipped with the FIFO for transmission/reception (16 bytes each).

Name	Function
Data buffer	■ Full-duplex double buffering(when FIFO is unused) ■ Transmission/reception FIFO (16 bytes each) (when FIFO is used)
Serial input	The noise up to two clocks of the serial clock/serial data input is filtered out by the peripheral clock (PCLK).
Transfer mode	Synchronization
Baud rate	■ Dedicated baud rate generator provided (comprising 15-bit reload counter) ■ External clock can be adjusted by the reload counter
Data length	8-bit
Signaling system	NRZ (Non Return to Zero)
Start bit detection	Synchronize with the start bit falling edge
Interrupt request	■ Reception interrupt ■ Transmission interrupt ■ Status interrupt/interrupt request for ICU ■ Transmission FIFO interrupt (when the transmission FIFO is empty)
I²C	■ Master/slave transmission/reception function ■ Adjustment function ■ Clock synchronous function ■ Transmission direction detection function ■ Generation of iterative start condition and detection function ■ Bus error detection function ■ General call addressing function ■ 7-bit addressing as master or slave ■ Interrupt can be generated at transmission or bus error ■ 10-bit addressing function is supported by a program
FIFO	■ Transmission/reception FIFO equipped (transmission FIFO: 16 bytes, reception FIFO: 16 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: Not supported Status: Not supported

40.2.5 Notes

- In serial communications, there is a possibility of receiving the incorrect data by the noise etc.
  Therefore, design the board that suppresses the noise.
  Moreover, add the checksum of data at the end, in consideration of the case when the erroneous data is received by the influence of the noise etc. Transmit data again when you detect the error by checksum.
- The I^2C function supports only ch.0 and ch.1.

40.3 Configuration

This section explains the configuration of the multi function serial interface.

Figure 40-1. Block Diagram (UART: Operating Mode 0, 1)

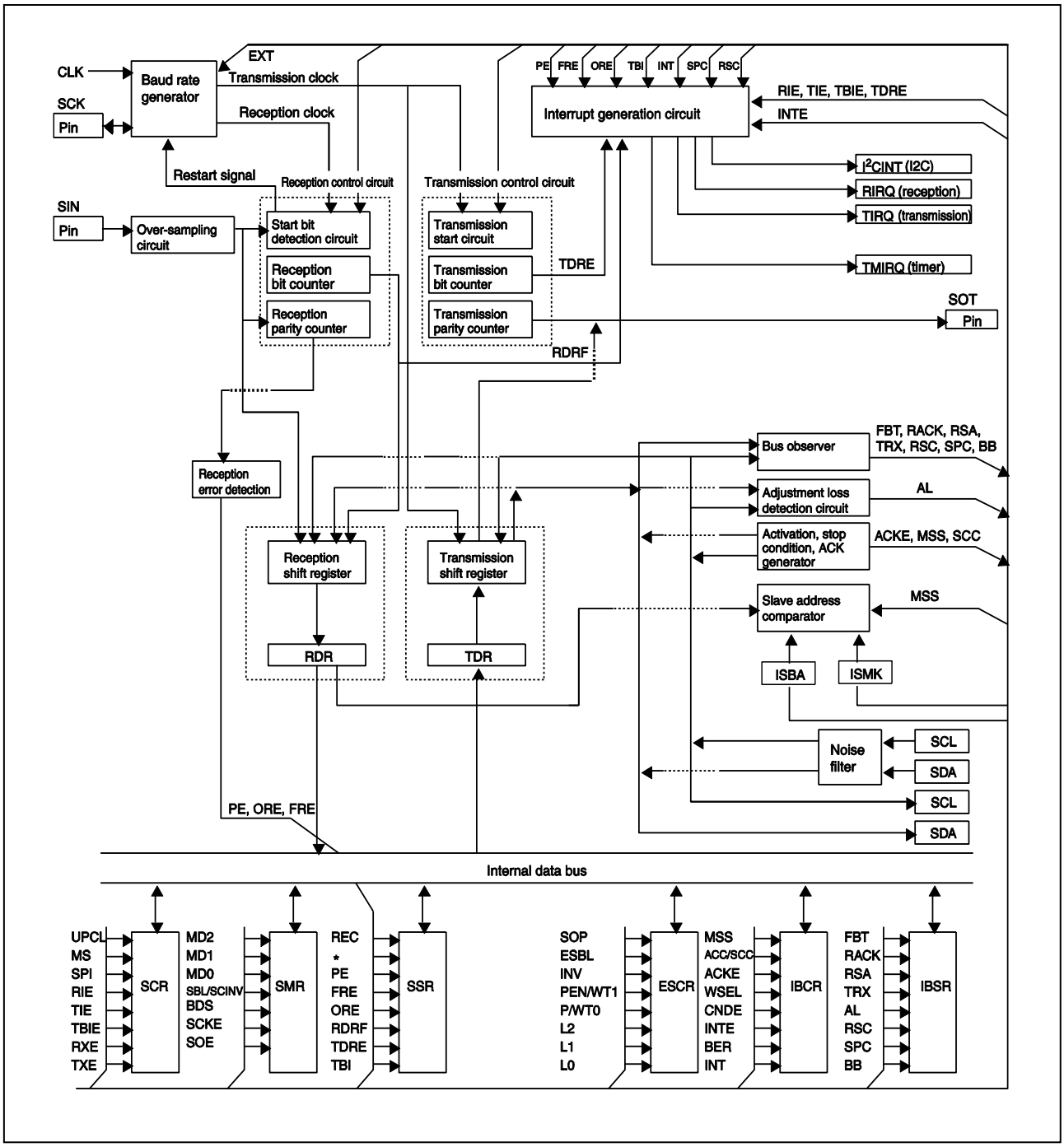


Figure 40-2. Block Diagram (CSIO: Operating Mode 2)

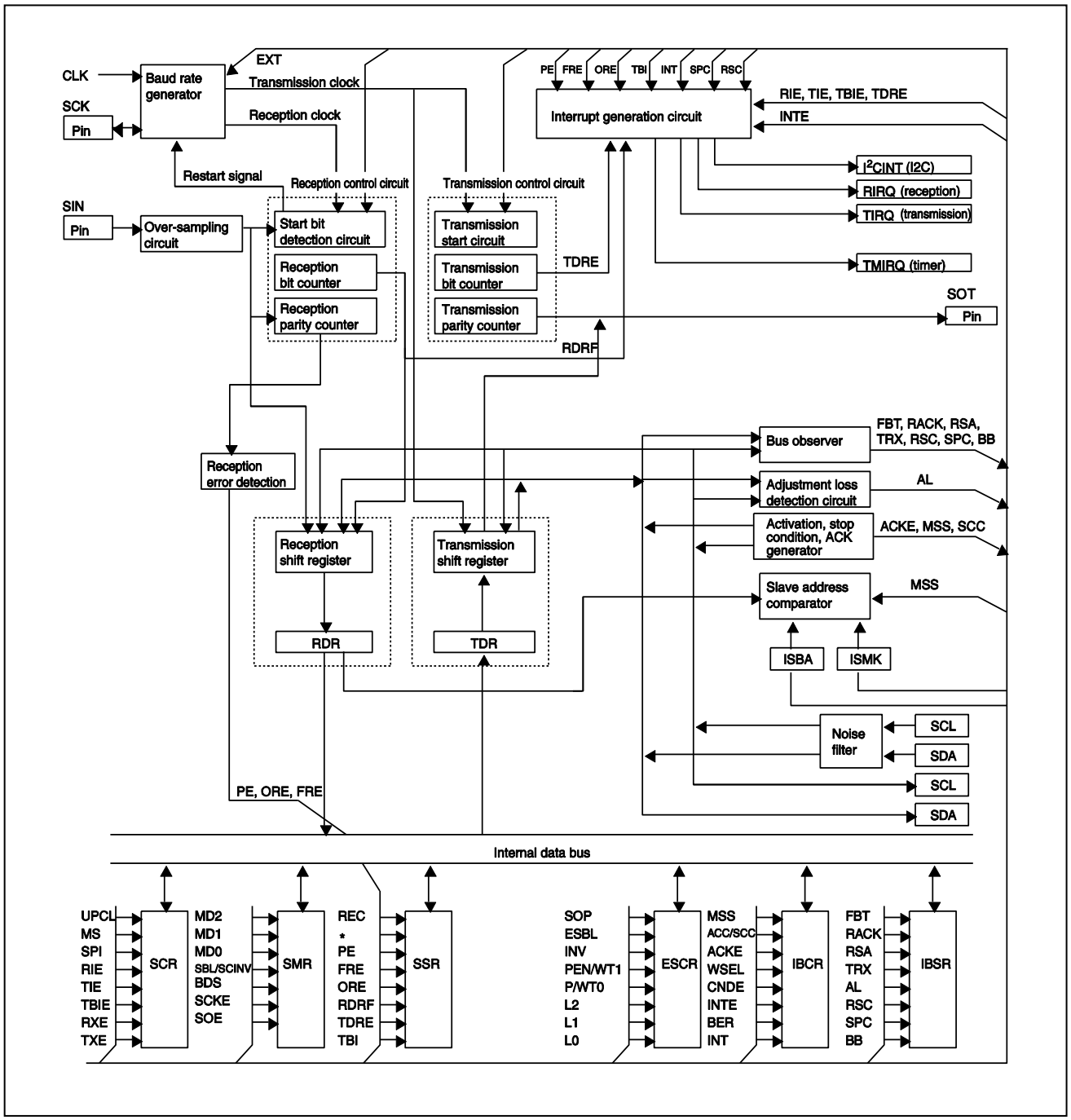


Figure 40-3. Block Diagram (LIN-UART: Operating Mode 3)

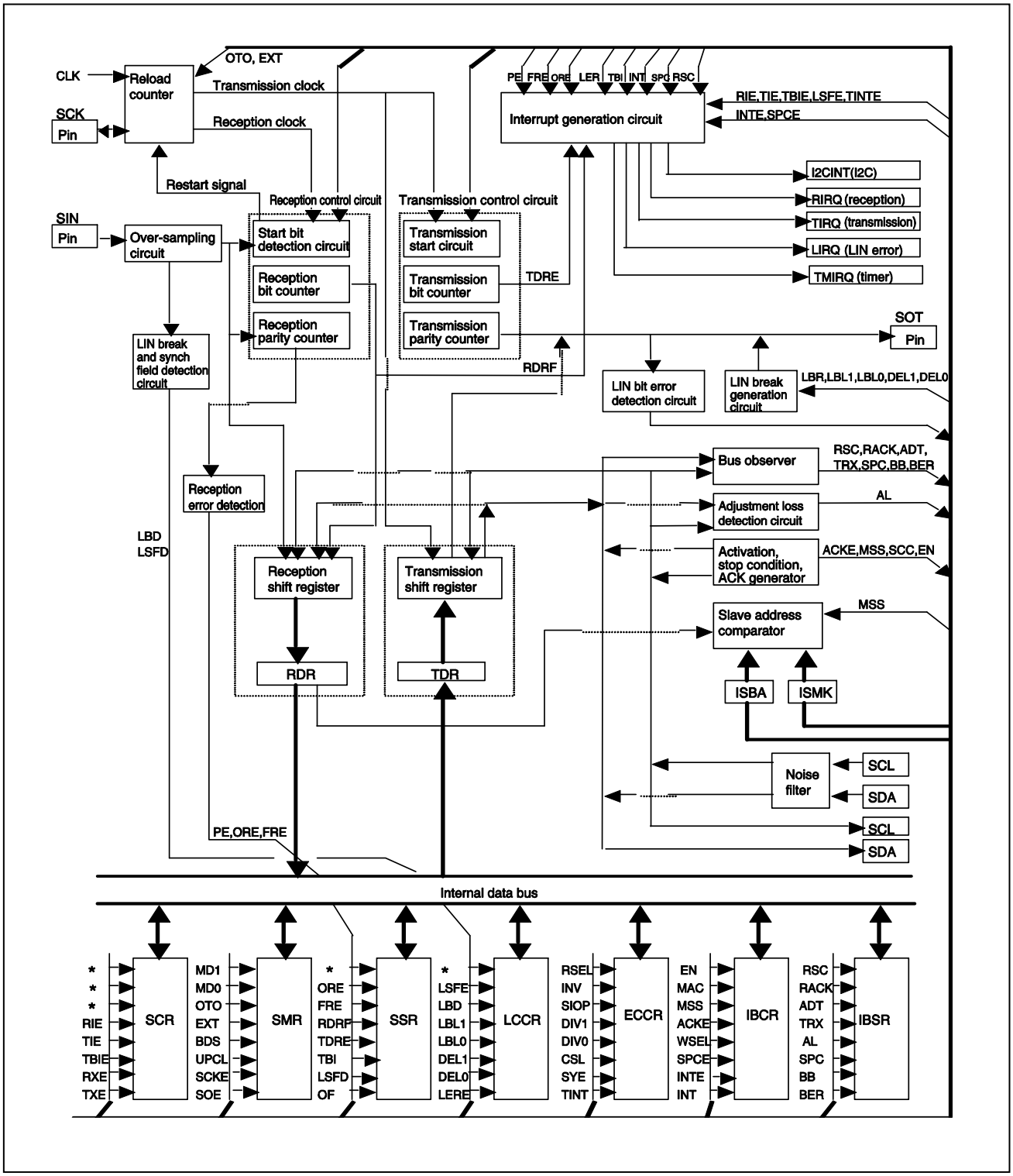
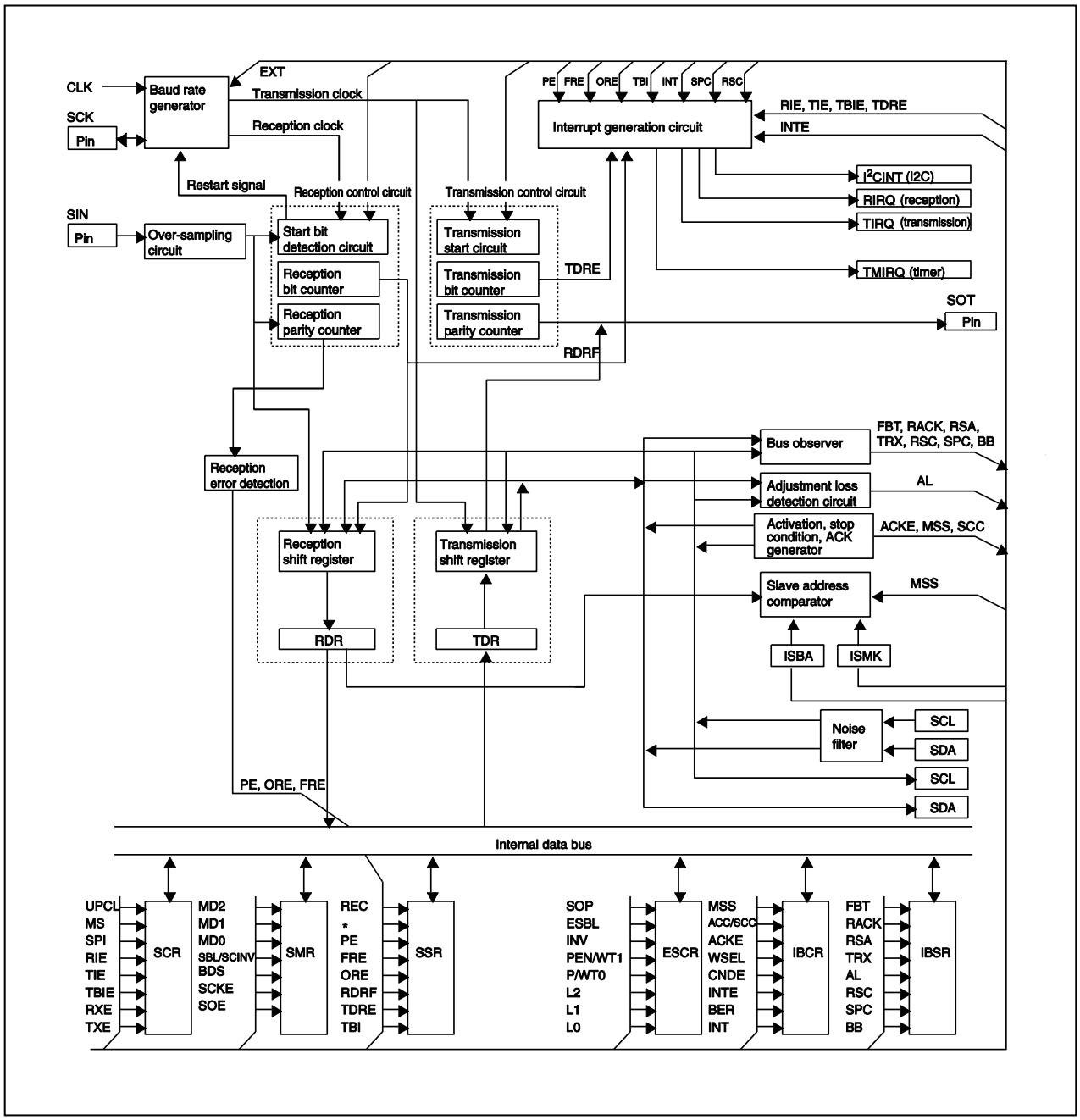


Figure 40-4. Block Diagram (I2C Operating Mode 4)



40.4 Registers

This section explains registers of the multi function serial interface.

Table of Base Addresses (Base_addr) and External Pins

Table 40-1. Table of Base Addresses (Base_addr) and External Pins

Channels	Base_addr	External Pin		
		SCK SCL[1]	SOT SDA[1]	SIN
0	0x00B0	SCK0_0/SCK0_1	SOT0_0/SOT0_1	SIN0_0/SIN0_1
1	0x00C0	SCK1_0/SCK1_1	SOT1_0/SOT1_1	SIN1_0/SIN1_1
8	0x04E0	SCK8_0/SCK8_1/ SCK8_2	SOT8_0/SOT8_1/ SOT8_2	SIN8_0/SIN8_1/ SIN8_2
9	0x04F0	SCK9_0/SCK9_1	SOT9_0/SOT9_1	SIN9_0/SIN9_1

[1]: Pin names for I^2C settings. (Only SCK0_0, SCK1_0, SOT0_0, SOT1_0 can be used for I^2C.)

Registers Map

Table 40-2. Registers Map

Address	Registers				Registers Function
	+0	+1	+2	+3	
0x00B0	[UART] SCR0 [CSIO] SCR0 [LIN-UART]SCR0 [I^2C] IBCR0	[Common] SMR0	[UART] SSR0 [CSIO] SSR0 [LIN-UART] SSR0 [I^2C] SSR0	[UART] ESCR0 [CSIO] ESCR0 [LIN-UART] ESCR0 [I^2C] IBSR0	--- ch.0 --- [UART] Serial control register [CSIO] Serial control register [LIN-UART] Serial control register [I^2C] I^2C Bus control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN-UART] Serial status register [I^2C] Serial status register [UART] Extended serial control register [CSIO] Extended serial control register [LIN-UART] Extended serial control register [I^2C] I^2C Bus status register

Address	Registers				Registers Function
	+0	+1	+2	+3	
0x00B4	[UART] RDR0/TDR0 [CSIO] RDR0/TDR0 [LIN-UART] RDR0/TDR0 [I²C] RDR0/TDR0		[UART] BGR0 [CSIO] BGR0 [LIN-UART] BGR0 [I²C] BGR0		--- ch.0 --- [UART] Transmit/receive data register [CSIO] Transmit/receive data register [LIN-UART] Transmit/receive data register [I²C] Transmit/receive data register [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN-UART] Baud rate generator register [I²C] Baud rate generator register
0x00B8	[UART] Reserved [CSIO] Reserved [LIN-UART] Reserved [I²C] ISMK0	[UART] Reserved [CSIO] Reserved [LIN-UART] Reserved [I²C] ISBA0	Reserved		--- ch.0--- [I²C] 7-bit Slave address mask register [I²C] 7-bit Slave address register
0x00BC	[Common] FCR10	[Common] FCR00	[Common] FBYTE0		--- ch.0--- [Common] FIFO Control register 1 [Common] FIFO Control register 0 [Common] FIFO Byte register
0x00C0	[UART] SCR1 [CSIO] SCR1 [LIN-UART]SCR1 [I²C] IBCR1	[Common] SMR1	[UART] SSR1 [CSIO] SSR1 [LIN-UART] SSR1 [I²C] SSR1	[UART] ESCR1 [CSIO] ESCR1 [LIN-UART] ESCR1 [I²C] IBSR1	--- ch.1--- [UART] Serial control register [CSIO] Serial control register [LIN-UART] Serial control register [I²C] I²C Bus control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN-UART] Serial status register [I²C] Serial status register [UART] Extended serial control register [CSIO] Extended serial control register [LIN-UART] Extended serial control register [I²C] I²C Bus status register
0x00C4	[UART] RDR1/TDR1 [CSIO] RDR1/TDR1 [LIN-UART] RDR1/TDR1 [I²C] RDR1/TDR1		[UART] BGR1 [CSIO] BGR1 [LIN-UART] BGR1 [I²C] BGR1		--- ch.1--- [UART] Transmit/receive data register [CSIO] Transmit/receive data register [LIN-UART] Transmit/receive data register [I²C] Transmit/receive data register [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN-UART] Baud rate generator register [I²C] Baud rate generator register

Address	Registers				Registers Function
	+0	+1	+2	+3	
0x00C8	[UART] Reserved [CSIO] Reserved [LIN-UART] Reserved [I²C] ISMK1	[UART] Reserved [CSIO] Reserved [LIN-UART] Reserved [I²C] ISBA1	Reserved		--- ch.1--- [I²C] 7-bit Slave address mask register [I²C] 7-bit Slave address register
0x00CC	[Common] FCR11	[Common] FCR01	[Common] FBYTE1		--- ch.1--- [Common] FIFO Control register 1 [Common] FIFO Control register 0 [Common] FIFO Byte register
0x04E0	[UART] SCR8 [CSIO] SCR8 [LIN-UART]SCR8	[Common] SMR8	[UART] SSR8 [CSIO] SSR8 [LIN-UART] SSR8	[UART] ESCR8 [CSIO] ESCR8 [LIN-UART] ESCR8	--- ch.8--- [UART] Serial control register [CSIO] Serial control register [LIN-UART] Serial control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN-UART] Serial status register [UART] Extended serial control register [CSIO] Extended serial control register [LIN-UART] Extended serial control register
0x04E4	[UART] RDR8/TDR8 [CSIO] RDR8/TDR8 [LIN-UART] RDR8/TDR8		[UART] BGR8 [CSIO] BGR8 [LIN-UART] BGR8		--- ch.8--- [UART] Transmit/receive data register [CSIO] Transmit/receive data register [LIN-UART] Transmit/receive data register [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN-UART] Baud rate generator register
0x04E8	Reserved		Reserved		
0x04EC	[Common] FCR18	[Common] FCR08	[Common] FBYTE8		--- ch.8--- [Common] FIFO Control register 1 [Common] FIFO Control register 0 [Common] FIFO Byte register

Address	Registers				Registers Function
	+0	+1	+2	+3	
0x04F0	[UART] SCR9 [CSIO] SCR9 [LIN-UART]SCR9	[Common] SMR9	[UART] SSR9 [CSIO] SSR9 [LIN-UART] SSR9	[UART] ESCR9 [CSIO] ESCR9 [LIN-UART] ESCR9	--- ch.9--- [UART] Serial control register [CSIO] Serial control register [LIN-UART] Serial control register [Common] Serial mode register [UART] Serial status register [CSIO] Serial status register [LIN-UART] Serial status register [UART] Extended serial control register [CSIO] Extended serial control register [LIN-UART] Extended serial control register
0x04F4	[UART] RDR9/TDR9 [CSIO] RDR9/TDR9 [LIN-UART] RDR9/TDR9		[UART] BGR9 [CSIO] BGR9 [LIN-UART] BGR9		--- ch.9--- [UART] Transmit/receive data register [CSIO] Transmit/receive data register [LIN-UART] Transmit/receive data register [UART] Baud rate generator register [CSIO] Baud rate generator register [LIN-UART] Baud rate generator register
0x04F8	Reserved		Reserved		
0x04FC	[Common] FCR19	[Common] FCR09	[Common] FBYTE9		--- ch.9--- [Common] FIFO Control register 1 [Common] FIFO Control register 0 [Common] FIFO Byte register

40.4.1 Common Registers

Common registers are shown.

40.4.1.1 Serial Mode Register: SMR
40.4.1.2 FIFO Control Register 1: FCR1
40.4.1.3 FIFO Control Register 0: FCR0
40.4.1.4 FIFO BYTE Register: FBYTE

40.4.1.1 Serial Mode Register: SMR

The bit configuration of the serial mode register is shown below.

This register selects the serial communication method (UART or I²C). Bit3 to bit0 changes their function according to the method selected (UART, CSIO, or I²C).

SMR: Address Base_addr + 01H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MD[2:0]			Reserved	SBL/ SCINV/ RIE	BDS/TIE	SCKE/ (Reserved)	SOE/ (Reserved)
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W0	R/W	R/W (R/W0)	R/W (R/W0)	R/W (R/W0)

[bit7 to bit5] MD[2:0] (Mode): Operation mode

These bits are used to set the communication method.

000B: Operating mode 0 (asynchronous normal mode) is set.
001B: Operating mode 1 (asynchronous multi-processor mode) is set.
010B: Operating mode 2 (CSIO mode) is set.
011B: Operating mode 3 (LIN communication mode) is set.
100B: Operating mode 4 (I²C mode) is set.

Notes:

- Settings other than those listed above are prohibited.
- Configure each register after setting the operation mode.
- [UART][CSIO][LIN-UART] Before changing the operation mode, execute programmable clear (SCR:UPCL=1).
- [I²C] Before changing the operation mode, disable I²C (ISMK:EN=0).
- Setting "100B" for ch.8 and ch.9 is prohibited.

[bit4] Reserved

Always write "0" to this bit.

[bit3] SBL/SCINV/RIE (Stop Bit Length/Serial Clock Inversion/Receive Interrupt Enable): Stop bit length selection bit/serial clock inversion bit, reception interrupt enable bit

[UART][LIN-UART]

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 bits are set to 2-bit.
When SBL=0 and ESCR:ESBL=1 are set: stop bits are set to 3-bit.
When SBL=1 and ESCR:ESBL=1 are set: stop bits are set to 4-bit.

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).

[CSIO]

This bit inverses the serial clock format.

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 (TXE=RXE=0).
- Set it to reception enabled (SCR:RXE=1) after setting the SCINV bit.

[I^2C]

This bit enables or disables the output of reception interrupt request to the CPU.

When the RIE bit and the reception data flag bit (SSR:RDRF) are set to "1", or any of the error flag bits (SSR:ORE) is set to "1", a reception interrupt request will be output.

Note:

To receive data using the INT bit of I^2C bus control register (IBCR), make sure to clear this bit to "0".

[bit2] BDS/TIE (Bit Direction Select/Transmit Interrupt Enable): Transfer direction selection bit/ transmission interrupt enable bit

[LIN-UART]

Always write "0" to this bit in the LIN-UART mode.

[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).

Note:

Set this bit when transmission and reception are disabled (TXE=RXE=0).

[I²C]

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.

Note:

To send data using the INT bit of I²C bus control register (IBCR), make sure to clear this bit to "0".

[bit1] SCKE (Serial Clock Enable): Serial clock output enable bit

[UART][LIN-UART]

This bit is not used in UART/LIN-UART. The reading value is "0". Always set this bit to "0".

[CSIO]

This bit controls the I/O ports of a serial clock.

When this bit is set to "0": Output serial clock is disabled.

When this bit is set to "1": Output serial clock is enabled.

When the port used as the SCK pin, do the GPIO setting..

Note:

To use the SCK pin as a serial clock input (SCKE=0), set the general-purpose I/O port as an input port. In this case, select the external clock using the external clock selection bit (BGR:EXT=1).

[I²C]

Reserved bit. Always set this bit to "0".

[bit0] SOE (Serial Output Enable): Serial output enable bit

[UART][CSIO][LIN-UART]

This bit enables/disables output of serial data.

When the serial data output is enabled (SOE="1"), the pin functions as SOT port regardless DDR setting. When it is used as SOT port, do the GPIO setting.

[I²C]

This bit is reserved. Always set this bit to "0".

40.4.1.2 FIFO Control Register 1: FCR1

The bit configuration of the FIFO control register 1 is shown below.

The 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.

FCR1: Address Base_addr + 0C$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		Reserved	FLSTE	FRIIE	FDRQ	FTIE	FSEL
Initial value	0	0	-	0	0	1	0	0
Attribute	R/W0	R/W0	R0,W0	R/W	R/W	R(RM1),W	R/W	R/W

[bit7,bit6]: Reserved

These bits must always be written to "0".

[bit5]: Reserved

Always write "0" to this bit. "0" is always read from this bit.

[bit4] FLSTE (Flag for Data Lost Detection Enable): Retransmission data lost detection enable bit

This bit is a bit that enables the FIFO retransmission data lost flag (FLST) detection.

When this bit is set to "0": the FLST bit detection disabled
When this bit is set to "1": the 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 the 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 this bit is set to "1", set this bit to "1" after setting "1" to the FSET bit.

[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

- FBYTE (for transmission) = 0 (transmission FIFO is empty)

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 you set this bit to "1", it does not affect the operation.
- If a read-modify-write instruction is executed, "1" will be read.
- When this bit is "0", changing the FSEL bit is prohibited.

[bit1] FTIE (Flag for Transmit Interrupt Enable): Transmission FIFO interrupt enable bit

This bit 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 (FCR0:FCL2, FCL1=1).
- When you change this bit, disable the FIFO operation (FCR0:FE2, FE1=0) first.

40.4.1.3 FIFO Control Register 0: FCR0

The bit configuration of the FIFO control register 0 is shown below.

The FIFO control register 0 (FCR0) is used to enable/disable FIFO operation, reset FIFO, save read pointer, and configure retransmission.

FCR0: Address Base_addr + $0D_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W0	R,WX	R/W	R/W	R/W	R/W	R/W	R/W

[bit7] Reserved

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 the 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)

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 of 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.
- Before writing "1" to this bit, clear the TIE bit and TBIE bit to "0", then set the TIE bit and TBIE bit to "1" after enabling transmission FIFO.

[bit4] FSET (FIFO pointer SET) FIFO pointer save bit

This bit is used to save the read pointer of transmission FIFO. If you save the 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 FCR1:FLST bit will be initialized while other bits of the FCR1/FCR0 register are retained.

Notes:

- Execute FIFO2 reset after disabling transmission/reception.
- Execute after clearing the transmission FIFO interrupt enable bit to "0".
- The valid data count of the 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 FCR1:FLST bit will be initialized while other bits of the FCR1/FCR0 register are retained.

Notes:

- Execute FIFO1 reset after disabling transmission/reception.
- Execute after clearing the transmission FIFO interrupt enable bit to "0".
- The valid data count of the FBYTE1 register will be 0.

[bit1] FE2 (FIFO Enable 2) FIFO2 operation enable bit

This bit enables/disables operation of FIFO2.

For mode 1,2,3:

- To use FIFO2, set this bit to "1".
- With the FIFO2 configured as transmission FIFO (FCR1:FSEL=1), if FIFO2 contains data when you write "1" to this bit and transmission is enabled (SCR:TXE=1), it immediately starts the transmission. Before writing "1" to this bit, clear the SCR:TIE bit and the SCR:TBIE bit to "0", then set the SCR:TIE bit and the SCR:TBIE bit to "1".
- When this bit 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 the transmission FIFO is used, if the transmission buffer is empty (SSR:TDRE=1), or when the reception FIFO is used, 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.

For the mode of 4:

- To use FIFO2, set this bit to "1".
- When the reception FIFO is selected by the FCR1:FSEL bit, this bit is cleared to "0" if a reception error occurs. As long as the reception error is not cleared, this bit cannot be set to "1".
- With the FIFO2 configured as transmission FIFO, write "1" or "0" to this bit when the transmission data is empty (SSR:TDRE="1").
- With the FIFO2 configured as transmission FIFO, write "1" or "0" to this bit when I2C interface is disabled (ISMK:EN="0"), or the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1", the reception data is empty (SSR:RDRF="0") and no data in the reception FIFO (FBYTE2="0").
- With the FIFO2 configured as reception FIFO, write "1" to this bit when I2C interface is disabled (ISMK:EN="0"), or the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1", and the reception data is empty (SSR:RDRF="0").
- Even if FIFO2 is disabled, the status of FIFO2 is retained.

Notes:

- Set enable/disable to this bit when the IBSR:BB bit is "0" or IBCR:INT bit is "1".
- Clear this bit to "0" and clear the IBCR:ACKE to "0", when the reception FIFO is selected, a reserved address is detected, and a slave transmission is operated.
- The reception FIFO is not disabled before the SSR:RDRF bit is from "1" to "0" when this bit cleared "1" to "0" using as the reception FIFO.
- Set this bit to "1" and set the SMR:TIE bit to "1" after clearing the SMR:TIE bit to "0" when this bit is set from "0" to "1", and data in the FIFO1 using as the transmission FIFO.

[bit0] FE1 (FIFO Enable 1) FIFO1 operation enable bit

This bit enables/disables operation of FIFO1.

For mode 1,2,3:

- To use FIFO1, set this bit to "1".
- With the FIFO1 configured as transmission FIFO (FCR1:FSEL=0), if FIFO1 contains data when you write "1" to this bit and transmission is enabled (SCR:TXE=1), it immediately starts the transmission. Before writing "1" to this bit, clear the SCR:TIE bit and the SCR:TBIE bit to "0", then set the SCR:TIE bit and the SCR:TBIE bit to "1".
- When this bit 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 the transmission FIFO is used, if the transmission buffer is empty (SSR:TDRE=1), or when the reception FIFO is used, 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.

For the mode of 4:

- To use FIFO1, set this bit to "1".
- When the reception FIFO is selected by the FCR1:FSEL bit, this bit is cleared to "0" if a reception error occurs. As long as the reception error is not cleared, this bit cannot be set to "1".
- With the FIFO1 configured as transmission FIFO, write "1" or "0" to this bit when the transmission data is empty (SSR:TDRE="1").
- With the FIFO1 configured as transmission FIFO, write "1" or "0" to this bit when I2C interface is disabled (ISMK:EN="0"), or the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1", the reception data is empty (SSR:RDRF="0") and no data in the reception FIFO (FBYTE2="0").
- With the FIFO1 configured as reception FIFO, write "1" to this bit when I2C interface is disabled (ISMK:EN="0"), or the operation flag (IBCR:ACT) is "0", or the interrupt flag (IBCR:INT) is "1", and the reception data is empty (SSR:RDRF="0").
- Even if FIFO1 is disabled, the status of FIFO1 is retained.

Notes:

- Set enable/disable to this bit when the IBSR:BB bit is "0" or IBCR:INT bit is "1".
- Clear this bit to "0" and clear the IBCR:ACKE to "0", when the reception FIFO is selected, a reserved address is detected, and a slave transmission is operated.
- The reception FIFO is not disabled before the SSR:RDRF bit is from "1" to "0" when this bit cleared "1" to "0" using as the reception FIFO.
- Set this bit to "1" and set the SMR:TIE bit to "1" after clearing the SMR:TIE bit to "0" when this bit is set from "0" to "1", and data in the FIFO1 using as the transmission FIFO.

40.4.1.4 FIFO BYTE Register: FBYTE

The bit configuration of the FIFO BYTE register is shown below.

FBYTE: Address Base_addr + $0E_H$ (Access: Byte, Half-word, Word)

The function of this register changes for reading and writing.
For reading, FIFO byte register (FBYTE) shows the 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 number of data sets.

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	FBYTE2[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	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FBYTE1[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 bit8] FBYTE2[7:0] (FIFO Byte 2) FIFO2 data count display bits

[bit7 to bit0] FBYTE1[7:0] (FIFO Byte 1) FIFO1 data count display bits

The FBYTE register indicates the valid data count written to or received at FIFO. The following table 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 FBYTE data count display match, the interrupt flag (SSR:RDRF) will be set to "1".
- If the data count contained in the reception FIFO does not reach the transfer count while the reception FIFO idle detection enable bit (FRIIE) is set to "1", the interrupt flag (RDRF) will be set to "1" after the reception idle state continues for 8 baud rate clocks or longer. 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. If the reception FIFO is disabled, the counter is reset to 0. When the reception FIFO is enabled while there is data remaining in it, the counter starts counting again.
- [CSIO] To receive data in the master operation mode (master reception), clear the TIE bit and the TBIE bit to "0", set the reception data count at the FBYTE register of transmission FIFO, and write "0" to the FDRQ bit. Then, it outputs serial clocks for the volume of data configured when the TXE bit is "1", which allows you to receive the data volume you have configured. To set the TIE bit and the TBIE bit to "1", set them to 1 after FDRQ changes to "1".
- [I^2C] To receive data in the master operation mode (master reception), clear the TIE bit to "0", set the reception data count at FBYTE of transmission FIFO, and write "0" to the FDRQ bit. It outputs the SCL clocks for the data volume configured. Then, the INT bit will be set to "1". To set the TIE bit to "1", set it to 1 after the FDRQ bit changes to "1".

Notes:

- [UART][LIN-UART] Set FBYTE of the transmission FIFO to 00H.
- [CSIO] [I2C] Other than the case of receiving data in the master operation mode, set FBYTE 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 bit and the SSR:TBIE bit are "0".
- [I2C] 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 bit is "0".
- [CSIO] Before you disable reception (SCR:RXE=0) while data is being received in the master operation mode, you need to disable the transmission/reception, after you disable the transmission FIFO.
- [I2C] Before you disable the I^2C interface (ISMK:EN=0) while data is being received in the master operation mode, you will need to disable the transmission/reception FIFO first.
- [Common] Data configured at FBYTE of the reception FIFO should be "1" or greater.
- [Common] Change the bit after disabling transmission/reception.
- [Common] You will not be able to use read-modify-write instructions for this register.
- [Common] Settings that go over the FIFO capacity are prohibited.

40.4.2 Registers for UART

Registers for UART are shown.

40.4.2.1 Serial Control Register: SCR
40.4.2.2 Serial Status Register: SSR
40.4.2.3 Extended Serial Control Register: ESCR
40.4.2.4 Receive Data Register/Transmit Data Register: RDR/TDR
40.4.2.5 Baud Rate Generator Register: BGR

40.4.2.1 Serial Control Register: SCR

The bit configuration of the serial control register is shown below.

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.

SCR: Address Base_addr + 00_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	UPCL	Reserved	Reserved	RIE	TIE	TBIE	RXE	TXE
Initial value	0	-	-	0	0	0	0	0
Attribute	R0,W	RX,WX	RX,WX	R/W	R/W	R/W	R/W	R/W

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 BGR register. ■ All the transmission and reception interrupt factors (SSR:PE,FRE,ORE,RDRF,TDRE,TBI) are initialized(000011B). 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.
bit6, bit5	Reserved	Read: The value is undefined. Write: No effect.
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 the SSR: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.

40.4.2.2 Serial Status Register: SSR

The bit configuration of the serial status register is shown below.

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.

SSR: Address Base_addr + 02_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	REC	Reserved	PE	FRE	ORE	RDRF	TDRE	TBI
Initial value	0	-	0	0	0	0	1	1
Attribute	R0,W	RX,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

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	Read: The value is undefined. Write: No effect.
bit5	PE: Parity error flag bit (Functions only in the operation mode 0)	"0" Read: No parity error "1" Read: Parity error exists ■ If a parity error occurs while a reception is in progress (ESCR:PEN=1), the 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: Framing error exists ■ 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: Overrun error exists. ■ 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 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.
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 the RDR, this flag will be set to "1" and when the receive data register (RDR) is read out, it will be cleared to "0". ■ When the RDRF bit and the 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. ■ While using reception FIFO and the reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1", if the reception FIFO contains data without receiving the specified number of data sets and the reception idle state has continued for 8 baud rate clocks or longer, the RDRF will be set to "1". 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 TDR is empty. ■ The flag indicates the state of the transmit data register (TDR). ■ When a transmit data is written to TDR, this flag turns to "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 "40.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".
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.

40.4.2.3 Extended Serial Control Register: ESCR

The bit configuration of the extended serial control register is shown below.

The extended serial control register (ESCR) allows you to set the data length of transmission/reception, enable/disable the parity bit, select a parity bit, inverse the serial data format, as well as to select the length of stop bit.

ESCR: Address Base_addr + 03H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ESBL	INV	PEN	P	L[2:0]		
Initial value	-	0	0	0	0	0	0	0
Attribute	RX,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Name		Function
bit7	Reserved	Always set this bit to "0".
bit6	ESBL: Extended stop bit length selection bit	This bit configures the bit length of stop bit (frame end mark for transmission data). When SMR:SBL=0 and ESBL=0 are set: stop bit is set to 1-bit. When SMR:SBL=1 and ESBL=0 are set: stop bits are set to 2-bit. When SMR:SBL=0 and ESBL=1 are set: stop bits are set to 3-bit. When SMR:SBL=1 and ESBL=1 are set: stop bits are set to 4-bit. **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).
bit5	INV: Invert 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

Bit Name		Function
bit2 to bit0	L2, L1, L0: Data length selection bits	These bits specify the data length of transmission/reception data. ■ 000B: Data length will be set to 8-bit. ■ 001B: Data length will be set to 5-bit. ■ 010B: Data length will be set to 6-bit. ■ 011B: Data length will be set to 7-bit. ■ 100B: Data length will be set to 9-bit. **Notes:** ■ Settings other than those shown above are prohibited. ■ In operation mode 1, set the data length to 7/8-bit. The other settings are prohibited.

40.4.2.4 Receive Data Register/Transmit Data Register: RDR/TDR

The bit configuration of receive data register/transmit data register is shown below.

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. When FIFO enabled, the address of RDR/TDR will be the address for reading/writing FIFO.

RDR/TDR: Address Base_addr + 04_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							D8
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	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

Read

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) 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 length. The AD bit received will be stored at the D8 bit.
- For the 9-bit length transfer and in operation mode 1, RDR will be read in 16-bit access mode.

Notes:

- 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 (SSR: PE, ORE, or 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

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 (SOUT).
- 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

(X is the transmission data bit)

- 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 length. The AD bit will be transmitted by writing to the D8 bit.
- For the 9-bit length 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 the 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 "40.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

40.4.2.5 Baud Rate Generator Register: BGR

The bit configuration of the baud rate generator register is shown below.

The baud rate generator register (BGR) sets the division ratio of a serial clock. It can also select an external clock as the clock source of a reload counter.

BGR: Address Base_addr + 06_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EXT	BGR[14:8]						
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
	BGR[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] EXT (External clock): External clock selection 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=1, the external clock source will be used. This may not be implemented for some product types.

[bit14 to bit0] BGR[14:0] (Baud Rate Generator)

These bits set the reload value for internal reload counter for baud rate generation.
When the reload value is written in this register, the reload counter begins counting.

Notes:

- Write to the baud rate generator register (BGR) in 16-bit access mode.
- If the value of BGR is an even number, the “H” width of a serial clock is 1 cycle shorter than that of the “L” width. If it is an odd number, the duty ratio will be 1:1.
- If you change to the setting of External clock (EXT=1) in operation of baud rate generator, you write "0" in baud rate generator(BGR) and execute a programmable clear(SCR:UPCL),then set External clock(EXT=1).
- 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 (SCR:UPCL) after you have change the setting value of BGR1/BGR0.
- Set the value of four or more to BGR. However, it is not likely to be able to receive the data normally by the error margin of the baud rate and setting the reload value.

40.4.3 Registers for CSIO

Registers for CSIO are shown.

40.4.3.1 Serial Control Register: SCR
40.4.3.2 Serial Status Register: SSR
40.4.3.3 Extended Serial Control Register: ESCR
40.4.3.4 Receive Data Register/Transmit Data Register: RDR/TDR
40.4.3.5 Baud Rate Generator Register: BGR

40.4.3.1 Serial Control Register: SCR

The bit configuration of the serial control register is shown below.

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.

SCR: Address Base_addr + 00_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 transmission and reception interrupt sources (SSR:TDRE, TBI, RDRF, and ORE) are initialized. 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.
bit6	MS: Master/slave function selection 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 to reception enable (RXE=1) after setting the MS bit.
bit5	SPI: SPI support bit	This bit is used to execute a SPI communication. When this bit is set to "0": Normal synchronous communication When this bit is set to "1": SPI communication supported.
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 (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 bit are set to "1", a transmission interrupt request will be output.

Bit Name		Function
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 the 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. ■ Set this bit to reception enable (RXE=1) after setting the MS bit and SMR:SCINV bit.
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.

40.4.3.2 Serial Status Register: SSR

The bit configuration of the serial status register is shown below.

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.

SSR: Address Base_addr + 02_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	REC	Reserved			ORE	RDRF	TDRE	TBI
Initial value	0	-	-	-	0	0	1	1
Attribute	R0,W	RX,WX	RX,WX	RX,WX	R,WX	R,WX	R,WX	R,WX

Bit Name		Function
bit7	REC: Reception error flag clear bit	This bit clears the ORE flag 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 to bit4	Reserved	Read: The value is undefined. Write: No effect.
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 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.
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 the receive data register (RDR) is read out, it will be cleared to "0". ■ When the RDRF bit and the SCR:RIE bit are set to "1", a reception interrupt request will be output. ■ While using reception FIFO, the RDRF bit will be set to "1" once the reception FIFO has received the specified number of data sets. ■ When you use reception FIFO, if the reception FIFO contains data without receiving the specified number of data sets and the reception idle state has continued for 8 baud rate clocks or longer, the RDRF bit will be set to "1". 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.

Bit Name		Function
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 turns to "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". ■ See "40.6.1.5Interrupts When Using Transmission FIFO and Flag Setting Timing" for the set/reset timing of the TDRE bit when you use transmission FIFO.
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.

40.4.3.3 Extended Serial Control Register: ESCR

The bit configuration of the extended serial control register is shown below.

The extended serial control register (ESCR) is used to set the transmission/reception data length as well as to fix the serial output at the "H" level.

ESCR: Address Base_addr + 03_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SOP	Reserved		WT[1:0]		L[2:0]		
Initial value	0	-	-	0	0	0	0	0
Attribute	R0,W	RX,WX	RX,WX	R/W	R/W	R/W	R/W	R/W

Bit Name		Function
bit7	SOP: Serial output pin set bit	■ This bit is used to set the serial output pin at the "H" level. When you write "1" to this bit, the SOUT pin will be set to "H". However, you do not need to write "0" to this bit afterward. ■ A read always results in "0". **Note:** Do not set this bit during serial data transmission.
bit6, bit5	Reserved	Read: The value is undefined. Write: No effect.
bit4, bit3	WT1, WT0: Data transmission/ reception wait selection bits	In the master mode, these bits set the number of wait 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.
bit2 to bit0	L2, L1, L0: Data length selection bits	These bits specify the data length of transmission/reception data. ■ "000B": Data length will be set to 8-bit. ■ "001B": Data length will be set to 5-bit. ■ "010B": Data length will be set to 6-bit. ■ "011B": Data length will be set to 7-bit. ■ "100B": Data length will be set to 9-bit. **Note:** Settings other than those listed above are prohibited.

40.4.3.4 Receive Data Register/Transmit Data Register: RDR/TDR

The bit configuration of receive data register/transmit data register is shown below.

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. When FIFO enabled, the address of RDR/TDR will be the address for reading/writing FIFO.

RDR/TDR: Address Base_addr + 04_H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved							D8
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	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

Read

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) 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 reception 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.
- For the 9-bit long transfer, read a value from the RDR in 16-bit access mode.

Notes:

- 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 (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.

Write

The transmit data register (TDR) is the 9-bit data buffer register for sending serial data.

- When transmission operations are enabled (SCR:TXE=1), if transmission data is written to the transmit data register (TDR), the transmission data is transferred to the transmission shift register and converted to serial data, then output from the serial data output pin (SOUT).
- 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

(X is the transmission data bit)

- 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 transmission 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.
- For the 9-bit long transfer, write a value to the TDR in 16-bit access mode.

Notes:

- The transmit data register is write-only register and the 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.
- See "40.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing" for the set timing of the transmission data empty flag (SSR:TDRE) when you use transmission FIFO.

40.4.3.5 Baud Rate Generator Register: BGR

The bit configuration of the baud rate generator register is shown below.

The baud rate generator register (BGR) sets the division ratio of a serial clock.

BGR: Address Base_addr + 06_H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	BGR[14:8]						
Initial value	0	0	0	0	0	0	0	0
Attribute	RX,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BGR[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

[bit14 to bit0] BGR[14:0] (Baud rate Generator)

These bits set the reload value for internal reload counter for baud rate generation.
When the reload value is written in this register, the reload counter begins counting.

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 peripheral clock (PCLK).
  - ☐ If SMR:SCINV=1, the "L" width of the serial clock is longer by one cycle of the peripheral clock (PCLK).
- 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 change 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.

40.4.4 Registers for LIN-UART

Registers for LIN-UART are shown.

40.4.4.1 Serial Control Register: SCR
40.4.4.2 Serial Status Register: SSR
40.4.4.3 Extended Serial Control Register: ESCR
40.4.4.4 Receive Data Register/Transmit Data Register: RDR/TDR
40.4.4.5 Baud Rate Generator Register: BGR

40.4.4.1 Serial Control Register: SCR

The bit configuration of the serial control register is shown below.

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 a function to generate LIN Synch Break and reset LIN-UART.

SCR: Address Base_addr + 00_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	UPCL	MS	LBR	RIE	TIE	TBIE	RXE	TXE
Initial value	0	0	0	0	0	0	0	0
Attribute	R0,W	R/W	R0,W	R/W	R/W	R/W	R/W	R/W

Bit Name		Function
bit7	UPCL: Programmable clear bit	This bit initializes the internal state of LIN-UART. When this bit is set to "1": ■ Directly reset LIN-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 BGR register. ■ All transmission and reception interrupt sources (SSR:TDRE, TBI, RDRF, FRE, ORE, and LBD) are 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 (FCR:FE2,FE1=0) before you execute a programmable clear
bit6	MS: Master/slave selection bit	This bit selects master or slave mode. 0: Master 1: Slave
bit5	LBR: Lin Synch Break setting bit (Functions only in the master operation)	When you write "1" to this bit, the LIN Synch Break and the LIN Sync delimiter with the length specified by the ESCR:LBL1/LBL0 bits and DEL1/DEL0 are generated. Write: Writing "0": No effect. Writing "1": Generates LIN Synch Break. For reading, "0" will be always read out. **Notes:** ■ Functions only in the master operation. ■ Do not set this bit to "1" while generating LIN Break field.

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: LER, 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 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 the 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-UART. ■ 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), LIN-UART 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 receptions are enabled (RXE=1) during LIN synch break field transmission. ■ If you disable reception (RXE=0) while a reception is in progress, it immediately stops the reception. ■ To detect a Lin synch break, enable LIN synch break detection interrupt (ESCR:LBIE=1) and then disable reception (SCR:RXE=0).
bit0	TXE: Transmission enable bit	This bit enables/disables the transmission of LIN-UART. ■ 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.

40.4.4.2 Serial Status Register: SSR

The bit configuration of the serial status register is shown below.

The serial status register (SSR) allows you to check the status of transmission/reception and the reception error flag and to detect the LIN Synch break as well as to clear the reception error flag.

SSR: Address Base_addr + 02_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	REC	Reserved	LBD	FRE	ORE	RDRF	TDRE	TBI
Initial value	0	-	0	0	0	0	1	1
Attribute	R0,W	RX,WX	R(RM1),W	R,WX	R,WX	R,WX	R,WX	R,WX

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	Read: The value is undefined. Write: No effect.
bit5	LBD: LIN Synch Break detection flag bit (Functions only in the slave operation)	"0" Read: No Synch Break "1" Read: There is a Synch Break "0" write: Clear LBD flag "1" write: No effect This bit indicates a detection of LIN Synch Break. The LBD bit is set to "1" when the serial input (SIN) is "0" for more than 11-bit width. In this case, if the LIN Synch Break interrupt enable bit (LBIE) is set to "1", a status interrupt will be generated. (For reading) "1": LIN Synch Break has been detected. "0": LIN Synch Break has not been detected. (For writing) "0": LBD bit will be cleared. "1": No effect. **Notes:** ■ This function is enabled only in the slave operation. ■ If a read-modify-write instruction is executed, "1" will be read out.

Bit Name		Function
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 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.
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 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.
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 the 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. ■ 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 (TDR) is empty ■ The flag indicates the state of the transmit data register (TDR). ■ When a transmission data is written to TDR, this flag turns to "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 "40.7.1.5 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-UART has no transmission in progress. ■ When transmission data has been written to the transmit data register (TDR), this bit will become "0". ■ If the LIN Break field is set (SCR:LBR=1), this bit will be cleared to "0". ■ When the transmit data register (TDR) is empty (TDRE=1) and no transmission is in progress, this bit will be set to "1". ■ If the LIN Break field transmission has ended, and the transmit data register is empty, this 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.

40.4.4.3 Extended Serial Control Register: ESCR

The bit configuration of the extended serial control register is shown below.

The extended serial control register (ESCR) is used to select LIN Synch Break interrupt enable/disable, LIN Synch Break detection, LIN Synch Break length, Synch delimiter length settings, and stop bit length.

ESCR: Address Base_addr + 03H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	ESBL	Reserved	LBIE	LBL[1:0]		DEL[1:0]	
Initial value	-	0	-	0	0	0	0	0
Attribute	R0,W0	R/W	RX,WX	R/W	R/W	R/W	R/W	R/W

Bit Name		Function
bit7	Reserved	This is undefined bit. The read value is always "0". Always write "0" to this bit.
bit6	ESBL: Extended stop bit length selection bit	This bit configures the bit length of stop bit (frame end mark for transmission data). When SMR:SBL=0 and ESBL=0 are set: stop bit is set to 1-bit. When SMR:SBL=1 and ESBL=0 are set: stop bits are set to 2-bit. When SMR:SBL=0 and ESBL=1 are set: stop bits are set to 3-bit. When SMR:SBL=1 and ESBL=1 are set: stop bits are set to 4-bit. **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	Reserved	Read: The value is undefined. Write: No effect.
bit4	LBIE: LIN Synch Break detection Interrupt enable bit	The bit to enable/disable LIN Synch Break detection interrupt. A reception interrupt occurs when LIN Synch Break detection flag (SSR:LBD) is set to "1" and interrupts are enabled (LBIE=1). **Note:** To detect a LIN synch break, enable LIN synch break detection interrupt (LBIE=1), and then disable reception (SCR:RXE=0).
bit3, bit2	LBL[1:0]: LIN synch break length selection bits (Functions only in the master operation)	00: 13-bit length 01: 14-bit length 10: 15-bit length 11: 16-bit length ■ These bits set the length of LIN Synch Break generation time interval (in bits). ■ Before you set the LBR bit in the serial control register (SCR) to "1" (LIN Synch Break transmission), set this bit. ■ The timing of Lin Synch Break detection is always the 11th bit at slave operation, regardless of the set value of this bit. **Note:** This function is enabled only in the master operation.

Bit Name		Function
bit1, bit0	DEL[1:0]: LIN synch delimiter length selection 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 Synch delimiter (in bits). ■ Before you set the LBR bit in the serial control register (SCR) to "1" (LIN Synch Break transmission), set this bit. **Note:** This function is enabled only in the master operation.

40.4.4.4 Receive Data Register/Transmit Data Register: RDR/TDR

The bit configuration of the receive data register/transmit data register is shown below.

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.

RDR/TDR: Address Base_addr + 04_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,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,WX	R0,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,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W

Read

The receive data register (RDR) is the data buffer register for serial data reception.

- Serial data signals sent to the serial input pin (SIN) 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 (SSR:either ORE or FRE is "1"), data in the receive data register (RDR) will become invalid.

Notes:

- 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

The transmit data register (TDR) is the data buffer register for sending serial data.

- When transmission 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 (SOUT).
- The 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.
- 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:

- The transmit data register is write-only register and the receive data register is read-only register. Because the two registers are located in the same address, the write value and read value might be different. Therefore instructions such as INC/DEC instructions which perform the 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 "40.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing".

40.4.4.5 Baud Rate Generator Register: BGR

The bit configuration of the baud rate generator register is shown below.

The 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.

BGR: Address Base_addr + 06H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	EXT	BGR[14:8]						
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
	BGR[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] EXT (External clock): External clock selection 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=1, the external clock source will be used.

[bit14 to bit0] BGR[14:0] (Baud rate Generator)

These bits set the reload value for internal reload counter for baud rate generation.
When the reload value is written in this register, the reload counter begins counting.

Notes:

- Write to the baud rate generator register (BGR) in 16-bit access mode.
- If the value of BGR is an even number, the “H” width of a serial clock is 1 cycle shorter than that of the "L" width. If it is an odd number, the duty ratio will be 1:1.
- If you change to the setting of External clock(EXT=1) in operation of baud rate generator, you write "0" in baud rate generator(BGR) and execute a programmable clear(SCR:UPCL),then set External clock(EXT=1).
- 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 (SCR:UPCL) after you have change the setting value of BGR1/BGR0.
- Set the value of three or more to BGR. However, it is not likely to be able to receive the data normally by the error margin of the baud rate and setting the reload value.

40.4.5 Registers for I²C

Registers for I²C are shown.

40.4.5.1 I2C Bus Control Register: IBCR
40.4.5.2 Serial Status Register: SSR
40.4.5.3 I2C Bus Status Register: IBSR
40.4.5.4 Receive Data Register/Transmit Data Register: RDR/TDR
40.4.5.5 Baud Rate Generator Register: BGR
40.4.5.6 I2C 7-bit Slave Address Mask Register: ISMK
40.4.5.7 I2C 7-bit Slave Bus Address Register: ISBA

40.4.5.1 I²C Bus Control Register: IBCR

The bit configuration of the I²C bus control register is shown below.

The I²C bus control register (IBCR) indicates master/slave selection, generation of repeat start condition, acknowledge enable, interrupt enable setting, and display of interrupt flag.

IBCR: Address Base_addr + 00H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MSS	ACT/SCC	ACKE	WSEL	CNDE	INTE	BER	INT
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R,WX	R(RM1),W

Bit Name		Function
bit7	MSS: Master/ slave selection bit	■ This bit selects master mode when it is set to "1" while I²C bus is in the idle state (ISMK:EN=1, IBSR:BB=0) ■ When the BB bit in IBSR register is "1" if you set "1" to this bit, this microcontroller waits for the start condition until the BB bit turns to "0". While waiting, if the slave address matches and it operates as slave, this bit will turn to "0", the AL bit in IBSR register will turn to "1". ■ When master is running (MSS=1, ACT=1) and interrupt flag (INT) is "1", if you write "0" to this bit, a stop condition occurs. The MSS bit will be cleared on the following conditions. 1. I²C interface disable (ISMK:EN bit=0) 2. When arbitration lost occurred 3. Bus error detected (BER bit=1) 4. Write "0" to the MSS bit when INT =1 5. The DMA mode is enabled (SSR:DMA=1) and "0" writing in the MSS bit at SSR:TBI =1. The relation between the MSS bit and ACT bit is as follows. MSS=0, ACT=0 idle MSS=0, ACT=1 slave address matches or responds ACK[1] to the reserved address and the slave is in operation (slave mode) MSS=1, ACT=0 master operation wait MSS=1, ACT=1 master is in operation (master mode) [1]: ACK response: indicates that SDA of I²C bus is "L" in the acknowledge interval. **Notes:** ■ If the DMA mode is disabled (SSR:DMA=0) also MSS bit is set as "1", go at MSS bit =1 and INT bit =1 when you change the MSS bit to "0". When "0" is written in the MSS bit when the ACT bit is "1", the INT bit is cleared to "0". ■ If the DMA mode is enabled (SSR:DMA=1), also MSS bit is set as "1", go when MSS bit =1, INT bit =1 or the SSR:TBI bit is "1" when the MSS bit is changed to "0". When "0" is written in the MSS bit when the ACT bit is "1", the INT bit is cleared to "0". ■ While the master is in operation, even if you write "0" to the MSS bit, "1" will still be read out while the ACT bit stays "1".

Bit Name		Function
bit6	ACT/SCC: Operation flag/repeat start condition generation bit	This bit differs in meanings between read and write. Read: ACT bit Write: SOC bit The ACT bit indicates whether the operation is in master mode or slave mode. ACT bit set conditions: 1. When outputting a start condition to I^2C bus (master mode) 2. When the slave address matches the address sent from the master (slave mode) 3. When the reserved address was detected and an acknowledge response was sent toward it (slave mode with MSS=0) ACT bit reset conditions: <Master mode> 1. A stop condition detected 2. Arbitration lost detected 3. A bus error detected 4. I^2C interface disable (ISMK:EN bit=0) <Slave mode> 1. (Repeat) start condition detected 2. A stop condition detected 3. Reserved address detected state (IBSR:RSA=1) and no acknowledge response sent 4. I^2C interface disable (ISMK:EN bit=0) 5. A bus error occurs (BER bit=1) When in master mode, writing "1" to this bit executes a repeat start. Writing "0" to this bit is ignored. **Notes:** ■ Write "1" to the SCC bit while master mode is interrupted (MSS=1, ACT=1, INT=1). If the ACT bit is "1", writing "1" to the SCC bit when the ACT bit is "1" clears the INT bit to "0". ■ Writing "1" to this bit is disabled during slave mode (MSS=0, ACT=1). ■ When you write "1" to the SCC bit and "0" to the MSS bit, the MSS bit will take precedence. ■ For read-modify-write instructions, the SCC bit will be read. ■ If "1" writing in the SCC bit and NACK is received by the ninth bit when interrupting the master mode in the eighth bit(MSS=1, ACT=1, INT=1, WSEL=1), "1" is set in the INT bit and I^2C bus wait (SCL="L"). It is necessary to write "1" in the SCC bit again to generate the repetition start condition, and to clear the INT bit. ■ If you issues the repetition start condition when the DMA mode is enabled (SSR:DMA=1), also the SSR:TBI bit is set as "1" and the INT bit is set as "0", please write the slave address in TDR and set "1" to this bit after "1" is written in the INT bit also confirm the INT bit is set in "1".

Bit Name		Function
bit5	ACKE: Data byte acknowledge enable bit	■ If you set "1" to this bit, "L" will be output at the time of acknowledgement. ■ If ACT = 1, change this bit when the INT bit is "1". ■ If you change this bit when the DMA mode is disabled (SSR:DMA=0) also ACT=1, please do it when the INT bit is "1" and DMA interrupt permission (SSM:DMA=1) also the SSR:TBI bit is "1", or DMA interrupt permission (SSM:DMA=1) and slave reception also SSM:RDRF is "1". ■ If you change this bit when the DMA mode is enabled (SSR:DMA=1) also ACT=1, please do the INT bit is "1" and the SSR:TBI bit is "1", or slave reception and SSM:RDRF bit is "1". This bit will be disabled on the following conditions. 1. Acknowledgement for address fields except for reserved address (automatic generation). 2. At data transmission (IBSR:RSA=0, IBSR:TRX=1, IBSR:FBT=0). 3. At slave reception with reception FIFO enable (FCR0:FE=1, MSS=0, ACT=1), it always responds with ACK. 4. When reception FIFO is enabled and WSEL is "0" at master reception (FCR0:FE=1, MSS=1, ACT=1, WSEL=0), when the SSR:TDRE bit is "0", it responds with ACK and when the SSR:TDRE bit is "1", it responds with NACK. 5. When reception FIFO is enabled, WSEL=0, the reserved address is detected and slave is transmitted (IBSR:RSA=1, IBSR:TRX=1, IBSR:FBT=1), it always responds with ACK. If you want to respond with NACK, at an interrupt after the detection of reserved address, disable reception FIFO and set ACKE=0. 6. When reception FIFO is enabled and WSEL is "1", the transmit data register has data on master reception (FCR0:FE=1, MSS=1, ACT=1, WSEL=1, SSR:TDRE=0).
bit4	WSEL: Wait selection bit	■ When the DMA mode is disabled (SSR:DMA=0), this bit is enable to select whether I^2C bus is waited by generating interrupt (INT=1) at before or behind of the acknowledge. ■ When the DMA mode is enabled (SSR:DMA=1), this bit is enable to select whether I^2C bus is waited by generating interrupt (INT=1, SSR:TBI=1 at the time of transmission, SSR:RDRF=1 at the time of reception) at before or behind of the acknowledge. ■ The WSEL bit will be disabled on the following condition. 1. When an interrupt to the first byte[1] is generated (INT=1) 2. When a reserved address is detected (IBSR:FBT=1,IBSR: RSA=1) 3. While the data transfer is in progress using FIFO and when NACK response[2] is detected (FCR0:FE=1, IBSR:RACK=1, ACT=1) 4. When reception FIFO is used and reception FIFO becomes FULL [1]: The first byte: indicates data after the (repeat) start condition. [2]: NACK response: indicates that SDA of I^2C bus is "H" in the acknowledge interval.
bit3	CNDE: Condition detection interrupt enable bit	This bit is used to enable interrupts when a stop condition or a repeat start condition is detected in master mode or in slave mode (ACT=1). When the RSC bit or the SPC bit in the IBSR register is "1" and this bit is "1", an interrupt occurs.
bit2	INTE: Interrupt enable bit	This bit is used to enable interrupts to the data transmission/reception and bus error in master mode or in slave mode (INT=1).

Bit Name		Function
bit1	BER: Bus error flag bit	This bit indicates that an error has been detected on I²C bus. BER bit set conditions: 1. While the first byte[1] transferring, the bit detects a start condition or a stop condition. 2. For the second byte or later, the bit detects a (repeat) start condition or a stop condition at the 2-9th (acknowledge) bit of data. BER bit reset conditions: 1. Write "0" to the INT bit when BER = 1 2. I²C interface disable (EN bit=0) [1]: The first byte: indicates data after the (repeat) start condition. **Note:** Please Check for this flag when interrupt flag (INT bit) turns "1" and if this bit is "1", please process re-transmission etc because normal send/receive operations cannot have been performed.
bit0	INT: Interrupt flag bit	Sets this flag to "1" when in master or slave mode, after 8-bit or 9-bit (ACK) of the data transmission/reception, or upon a bus error. When the INT bit is "1", the state of SCL turns to "L" and when the bit is "0", the "L" state is released except for bus errors. INT bit set conditions: **<8th bit>** <It is unrelated to the DMA mode > 1. When a reserved address is detected in the first byte 2. When WSEL is "1" and arbitration lost is detected in the second byte or later < When DMA mode is disabled (SSR:DMA=0)> 1. When DMA mode is disabled (SSR:DMA=0), WSEL is "1" and the SSR:TDRE bit is "1" in the second byte or later in master operation 2. When DMA mode is disabled (SSR:DMA=0), WSEL is "1" and reception FIFO is disabled, the SSR:TDRE bit is "1" in the second byte or later in slave operation 3. When DMA mode is disabled (SSR:DMA=0), WSEL is "1" and the SSR:TDRE bit is "1" in the second byte or later in slave transmission 4. When DMA mode is disabled (SSR:DMA=0), WSEL is "1" and reception FIFO is disabled in the slave reception <When DMA mode is enabled (SSR:DMA=1)> 1. When DMA mode is enabled (SSR:DMA=1), WSEL is "1" and the SSR:TBI bit is "1" and write the INTbit to "1" in the second byte or later in master operation **<9th bit>** < It is unrelated to the DMA mode > 1. When arbitration lost is detected in the first byte 2. When NACK is received except for stop condition setting (write "0" to MSS bit in master operation) 3. When WSEL is "0" and arbitration lost is detected in the second byte or later 4. In the first byte, no reserved address is detected in the receiving direction in master or slave mode (IBSR:TRX=0) and there are reception FIFO data at reception FIFO enable state < When DMA mode is disabled (SSR:DMA=0)> 1. When DMA mode is disabled (SSR:DMA=0), in the first byte, no reserved address is detected and the SSR:TDRE bit is "1" in the receiving direction in master or slave mode (IBSR:TRX=1) 2. When the DMA mode is disabled (SSR:DMA=0), the master mode without detecting the reservation address in the first byte or the SSR:TDRE bit is "1" when reception FIFO is disabled at mode of production is received

		3. When DMA mode is disabled (SSR:DMA=0), WSEL is "0" and the SSR:TDRE bit is "1" in the second byte or later in master operation 4. When DMA mode is disabled (SSR:DMA=0), WSEL is "0" and the SSR:TDRE bit is "1" in the second byte or later in slave transmission 5. When DMA mode is disabled (SSR:DMA=0), WSEL is "0" and reception FIFO is disabled in slave reception. However, for slave reception at the first byte where a reserved address is detected, an interrupt will not occur at the 9th bit. 6. When DMA mode is disabled (SSR:DMA=0), reception FIFO enable, for slave reception, when FIFO is Full < When DMA mode is enabled (SSR:DMA=1)> 1. When DMA mode is enabled (SSR:DMA=1), in the first byte, no reserved address is detected and the SSR:TDRE bit is "1" in the transmitting direction in slave mode (IBSR:TRX=1) 2. When DMA mode is enabled (SSR:DMA=1), in the first byte, no reserved address is detected, the SSR:TDRE bit is "1" in the receiving direction in slave mode (IBSR:TRX=0) and reception FIFO is disabled 3. When DMA mode is enabled (SSR:DMA=1), WSEL is "0" and when you write "1" in the INT bit while the master mode is operating when the SSR:TBI bit is "1" in the second byte or later in master operation **<Other>** 1. Bus error detected INT bit reset conditions: 1. write "0" to INT bit 2. INT bit is "1", write "0" to MSS bit when ACT bit is "1" 3. INT bit is "1", write "1" to SCC bit when ACT bit is "1" When DMA mode is disabled (SSR:DMA=0), writing "1" to this bit will not be effective. **Notes:** ■ When the DMA mode is enabled (SSR:DMA=1) and the INT bit is set to "1" while the SSR:TBI bit is "1" after the second byte in the master mode, status interrupt (SIRQ=1) is not generated. ■ When the DMA mode is enabled (SSR:DMA=1) and if the SSR:TBI bit issues the repetition start condition when it is "1" and the INT bit is "0", after "1" is written in the INT bit, write the slave address in TDR after confirming the INT bit is set in "1" and set "1" to the SCC bit. ■ If "0" is written in the INT flag when the INT flag is set in "1", the wait of the I^2C bus is released. ■ When the ISMK:EN bit is "0", the SSR:RDRF bit and the INT bit might be "1" depending on the reception timings. In this case, read the received data and clear the INT bit. ■ For read-modify-write instructions, "1" will be read. ■ When reception FIFO is enabled, even if reception FIFO is Full on the master reception operation, "1" will not be set to the INT bit.

40.4.5.2 Serial Status Register: SSR

The bit configuration of the serial status register is shown below.

The serial status register (SSR) checks for the transmission/reception states.

SSR: Address Base_addr + 02_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	REC	TSET	DMA	TBIE	ORE	RDRF	TDRE	TBI
Initial value	0	0	0	0	0	0	1	1
Attribute	R0,W	R0,W	R/W	R/W	R,WX	R,WX	R,WX	R,WX

Bit Name		Function
bit7	REC: Reception error flag clear bit	This bit clears the ORE bit of the serial status register (SSR) ■ Writing "1" clears the ORE bit. ■ Writing "0" does not affect anything. A read always results in "0".
bit6	TSET: Transmission buffer empty flag set bit	This bit sets the TDRE bit in the serial status register (SSR) ■ Writing "1" sets the TDRE bit and also sets the TBI bit if the DMA mode is enabled (SSR:DMA=1).. ■ Writing "0" does not affect anything. A read always results in "0". **Note:** Write "1" in this bit when the IBCR:INT bit is "1".
bit5	DMA: DMA mode enable bit	This bit enables/disables the DMA mode. ■ When this bit is set in "1", it becomes an interrupt condition corresponding to the DMA Transfer. ■ When this bit is set to "0", it becomes an interrupt condition during the normal transfer. See "Table 40-13" for details. **Note:** Only when ISMK:EN=0, this bit can be changed.
bit4	TBIE: Transmission bus idle enabled bit (Only the DMA mode enabled is effective.)	■ This bit enables/disables the transmission bus idle interrupt demand output to CPU. ■ When the DMA mode is enabled (DMA=1) and the TBIE bit and the TBI bit are "1", the DMA mode outputs the transmission bus idle interrupt request. ■ When the DMA mode is disabled (DMA=0),this bit becomes "0" and no matter what this bit is written, writing is ignored and this bit keep the state of "0".

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 SMR:RIE bit are set to "1", a reception interrupt request will be output. ■ If this flag is set, the receive data register (RDR) will be disabled. ■ When you are using the reception FIFO, if this flag is set, the received 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 the RIE bit and the reception data flag bit (RDRF) are "1", a reception interrupt request will be output. ■ 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". ■ Set at the SCL falling timing in 8th bit of the data. ■ Also set at the NACK response[1]. ■ 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. ■ When you use reception FIFO, if the reception FIFO contains data without receiving the specified number of data sets and the reception idle state has continued for 8 baud rate clocks or longer, the RDRF will be set to "1" with the IBCR:BER bit set to "0". 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. [1]: NACK response: indicates that SDA of I^2C bus is "H" in the acknowledge interval. **Note:** ■ In case that the reception FIFO is unused, DMA mode is enabled (DMA=1), the RDRF bit is "1", and the WSEL bit is "0", SCL turns to "L" after ACK transmission. It is released after the RDRF bit become "0". ■ In case that the reception FIFO is unused, DMA mode is enabled (DMA=1), receiving the 2nd byte of data, the RDRF bit is "1", and the WSEL bit is "1", SCL turns to "L" after 1 byte data reception. It is released after the RDRF bit become "0". ■ In case that the reception FIFO is used, DMA mode is enabled (DMA=1), SCL turns to "L" when the reception FIFO is full. It is released after the reception FIFO becomes not full.

Bit Name		Function
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 the TDRE bit and the TIE bit are set to "1", a transmission interrupt request will be output. ■ When a transmission data is written to TDR, this flag turns to "0", which indicates that a valid data exists in the TDR. Once a transmission starts after data being loaded to the transmission shift register, the bit will be set to "1", which indicates that the TDR does not contain any valid data. ■ Writing "1" to the TSET bit on the serial status register (SSR) results in a setting. Use this flag for setting "1" to the TDRE bit when detecting an arbitration lost or a bus error.
bit0	TBI: Transmission bus idle flag bit (Only the DMA mode enabled is effective.)	This bit is a bit that shows that I^2C does not do the transmission operation when the DMA mode is enabled (DMA=1). When the DMA mode is enabled (DMA=1) and the TBI bit becomes "1" after the second byte, the SCL becomes "L". The SCL is released from the state of "L" when the TBI bit becomes "0". Set condition of TBI bit : **< 8th bit >** 1. In the 2nd or subsequent byte , the TDRE bit is "1" while WSEL is "1" and the master is operating 2. In the 2nd or subsequent byte , the SSR:TDRE bit is "1" while WSEL is "1" and the slave is transmitting **< 9th bit>** 1. The SSR:TDRE bit is "1" without detecting the reservation address in the first byte while the master is operating 2. In the 2nd or subsequent byte , the SSR:TDRE bit is "1" while IBCR:WSEL is "0" and the master is operating 3. In the 2nd or subsequent byte , the SSR:TDRE bit is "1" while IBCR:WSEL is "0" and the slave is transmitting **<other>** When the transmission buffer empty flag set bit (TSET) is set to "1" Reset condition of TBI bit : 1. If the transmission data is written to the transmission data register (TDR) When this bit is "1" and the transmission bus idle interrupt is enabled (SCR:TBIE=1), this bit outputs the transmission interrupt request. ■ This bit is undefined when the DMA mode is disabled (DMA=0).

40.4.5.3 I²C Bus Status Register: IBSR

The bit configuration of the I²C bus status register is shown below.

The I²C bus status register (IBSR) indicates that repeat starts, acknowledges, data directions, arbitration lost, stop conditions, I²C bus states, and bus errors have been detected.

IBSR: Address Base_addr + 03H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
Initial value	0	0	0	0	0	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R(RM1),W	R(RM1),W	R,WX

Bit Name		Function
bit7	FBT: First byte bit	"0" Read: Other than the first byte "1" Read: Transmitting/receiving the first byte This bit indicates the first byte. FBT bit set conditions: 1. When (repeat) starts condition detected FBT bit clear conditions: 1. Transmission/reception of the 2nd byte 2. A stop condition detected 3. I²C interface disable (ISMK:EN bit=0) 4. Bus error detected (IBCR:BER bit=1)
bit6	RACK: Acknowledge flag bit	"0" Read: "L" Reception "1" Read: "H" Reception This bit indicates the acknowledges received on the first byte, in master or slave mode. Update condition for RACK bit 1. Acknowledgement at the first byte 2. Acknowledgement of the data in master or slave mode Clear condition of RACK bit (RACK bit=0) 1. (Repeat) start condition detected 2. I²C interface disable (ISMK:EN bit=0) 3. Bus error detected (IBCR:BER bit=1)

Bit Name		Function
bit5	RSA: Reserved address detection bit	"0" Read: No reserved address detected "1" Read: Reserved address detected This bit indicates that a reserved address was detected. RSA bit set condition (RSA=1) 1. The first byte is (0000xxxx) or (1111xxxx). "x" represents "0" or "1". RSA bit set condition (RSA=0) 1. A (repeat) start condition detected 2. A stop condition detected 3. I^2C interface disable (ISMK:EN bit=0) 4. Bus error detected (IBCR:BER bit=1) When the RSA bit is "1" at the first byte, the interrupt flag (IBCR:INT) turns to "1" and SCL turns to "L" at SCL falling edge of the 8th bit on the first byte, regardless of the FIFO enable/disable state. Read the received data and if you want to make it perform as slave, set IBCR:ACKE to "1" and set the interrupt flag (IBCR:INT) to "0". After that, if the TRX bit is "0", the data is received as the slave. When you are planning not to receive data at a relay point, set "0"to the IBCR:ACKE bit. After that, no data is received. **Notes:** ■ When you turn IBCR:ACKE to "0" while data transfer is going on, do not set IBCR:ACKE to "1" until a stop condition or a repeat start condition is detected. ■ When a reserved address detection interrupt occurs and you identify a slave transmission, if the reception FIFO is enabled, it would respond with ACK, so disable the reception FIFO and turn to IBCR:ACKE=0.
bit4	TRX: Data direction bit	"0" Read: Reception direction "1" Read: Transmission direction This bit indicates the direction of data. TRX bit set conditions: 1. Send a (repeat) start condition in master mode 2. When the 8th bit of the first byte is "1" in slave mode (transmission direction as a slave) TRX bit reset conditions: 1. Arbitration lost is generated(AL=1) 2. When the 8th bit of the first byte is "0" in slave mode (reception direction as a slave) 3. When the 8th bit of the first byte is "1" in master mode (reception direction as a master) 4. A stop condition detected 5. Detect a (repeat) start condition in a mode other than master mode 6. I^2C interface disable (ISMK:EN bit=0) 7. Bus error detected (IBCR:BER bit=1)

Bit Name		Function
bit3	AL: Arbitration lost bit	"0" Read: No arbitration lost occurred "1" Read: Arbitration lost occurred This bit indicates an arbitration lost. AL bit set conditions: 1. When the data output in master mode and received data are different. 2. You set "1" to the MSS bit but the operation is still in slave mode. 3. A repeat start condition was detected at the first bit of the second byte or later in master mode. 4. A stop condition was detected at the first bit of the second byte or later in master mode. 5. Trying to generate a repeat start condition but cannot do so in master mode. 6. Trying to generate a stop condition but cannot do so in master mode. AL bit reset conditions: 1. Writing "1" to the MSS bit 2. Writing "0" to the INT bit 3. Writing "0" to SPC bit when AL=1 and SPC=1 4. I^2C interface disable (ISMK:EN bit=0) 5. Bus error detected (IBCR:BER bit=1)
bit2	RSC: Repeat start condition check bit	"0" Read: No repeated start condition detected "1" Read: Repeated start condition detected This bit indicates that repeat start condition was detected in master mode or slave mode. RSC bit set conditions 1. A repeat start condition was detected after acknowledgement in master mode or slave mode RSC bit reset conditions: 1. Writing "0" to the RSC bit 2. Writing "1" to the IBCR:MSS bit 3. I^2C interface disable (ISMK:EN bit=0) There will be no effect on the operation of writing "1" to this bit. **Notes:** ■ If you do not respond with acknowledge when receiving data as the slave mode due to the detection of the reserved address, this bit will not be set "1" even if a repeat start condition is detected at the next time because it has already exited the slave mode. ■ For read-modify-write instructions, "1" will be read.

Bit Name		Function
bit1	SPC: Stop condition check bit	"0" Read: No stop condition detected "1" Read: (Master) stop condition detected or generation of arbitration lost at stop condition output "1" Read: (Slave) stop condition detected This bit indicates that stop condition was detected in master mode or slave mode. SPC bit set conditions: 1. A stop condition was detected in master mode or slave mode 2. An arbitration lost is generated on the stop condition generation in master mode SPC bit reset conditions: 1. Writing "0" to this bit 2. Writing "1" to the IBCR:MSS bit 3. I^2C interface disable (ISMK:EN bit=0) There will be no effect on the operation of writing "1" to this bit. **Notes:** ■ If you do not respond with acknowledge when receiving data as the slave mode due to the detection of the reserved address, this bit will not be set "1" even if a stop condition is detected at the next time because it has already exited the slave mode. ■ For read-modify-write instructions, "1" will be read.
bit0	BB: Bus state bit	"0" Read: Bus idle state "1" Read: Bus transmission/reception state This bit indicates the bus state. BB bit set conditions: 1. When "L" was detected at SDA or SCL on I^2C bus BB bit reset conditions: 1. When a stop condition detected 2. I^2C interface disable (ISMK:EN bit=0) 3. Bus error detected (IBCR:BER bit=1)

40.4.5.4 Receive Data Register/Transmit Data Register: RDR/TDR

The bit configuration of receive data register/transmit data register is shown below.

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. When FIFO enabled, the address of RDR/TDR will be the address for reading/writing FIFO.

RDR/TDR: Address Base_addr + 04_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	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Read

The receive data register (RDR) is the data buffer register for serial data reception.

- Serial data signals sent to the serial data line (SDA) are converted in the shift register and stored in the receive data register (RDR).
- When you receive the first byte[1], the least significant bit (RDR:D0) is the data direction 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".
- The reception data full flag bit (SSR:RDRF) will be automatically cleared to "0" when the receive data register (RDR) has been read out.

[1]: The first byte: indicates data after the (repeat) start condition.

Notes:

- 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".

Write

The transmit data register (TDR) is the data buffer register for sending serial data.

- Output to serial data line (SDA pin) at the MSB first on transmit data register (TDR).
- When you send the first byte, the least significant bit (TDR:D0) is the data direction bit.
- 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" when transferred to the transmit shift register.
- Following transmission data should be written on the following conditions:
  1. Interrupt flag (IBCR:INT bit) is "1".
  2. No bus error detected (IBCR:BER bit=0).
  3. Acknowledge is ACK response ("0" is received as acknowledge).
- If transmission FIFO is disabled and the transmission data empty flag (SSR:TDRE) is "0", the transmission data cannot be written to the transmit data register (TDR).
- When using transmission FIFO, the transmission data can be written to the amount of transmission FIFO, even if the transmission data empty flag (SSR:TDRE) is "0".

Note:

The transmit data register is write-only register and the receive data register is read-only register. Because the two registers are located in the same address, the write value and read value might be different. Therefore instructions such as INC/DEC instructions which perform read-modify-write (RMW) operations cannot be used.

40.4.5.5 Baud Rate Generator Register: BGR

The bit configuration of the baud rate generator register is shown below.

The baud rate generator register (BGR) sets the division ratio of a serial clock.

BGR: Address Base_addr + 06H (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved	BGR[14:8]						
Initial value	-	0	0	0	0	0	0	0
Attribute	RX,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BGR[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

[bit14 to bit0] BGR[14:0] (Baud rate Generator)

These bits set the reload value for the internal reload counter for baud rate generation.

When the reload value is written in this register, the reload counter begins counting.

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- Configure the baud rate generator when the EN bit of the ISMK register is "0".
- Configure baud rate regardless of the master mode or slave mode.
- The peripheral clock (PCLK) should be set with 8 MHz or more in operating mode 4 (I^2C mode) and baud rate generator configured in 400 kbps or more should not be used.

40.4.5.6 I²C 7-bit Slave Address Mask Register: ISMK

The bit configuration of the 7-bit slave address mask register is shown below.

The 7-bit slave address mask register (ISMK) compares and configures bits of slave address.

ISMK: Address Base_addr + 08H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EN	SM[6:0]						
Initial value	0	1	1	1	1	1	1	1
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Name		Function
bit7	EN: I²C Interface enable bit	This bit enables/disables I²C interface operation. If this bit is set to "0", I²C interface becomes disabled. If this bit is set to "1", I²C interface becomes enabled. **Notes:** ■ When the BER bit of the IBSR register is set to "1", this bit will not be cleared to "0". ■ Configure the baud rate generator when this bit is "0". ■ When this bit is "0", configure 7-bit slave address and 7-bit slave mask register. ■ If the I²C interface is disabled (EN=0), transmission/reception become disabled immediately. ■ When you disable the I²C interface operation after generating a stop condition by writing "0" to the IBCR:MSS bit, disable it (EN=0) after checking for the generation of the stop condition. ■ Setting "0" to the EN bit during transmission could generate SDA/SCL pulse on the I2C bus. **Note:** For FIFO enable, write "0" to the EN bit after disabling FIFO.
bit6 to bit0	SM6 to SM0: Slave address mask bits	This bit configures whether to exclude the 7-bit slave address and received address as the comparison targets. If these bits are set to "1": compare If these bits are set to "0": treat as matched **Note:** Configure this register when the EN bit is "0".

40.4.5.7 I²C 7-bit Slave Bus Address Register: ISBA

The bit configuration of the 7-bit slave bus address register is shown below.

The 7-bit slave address register (ISBA) sets slave addresses.

ISBA: Address Base_addr + 09H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SAEN	SA[6:0]						
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

Bit Name		Function
bit7	SAEN: Slave address enable bit	This bit enables slave address detection. Setting "0": Does not detect a slave address. Setting "1": Compares the ISBA and ISMK values with the first byte received.
bit6 to bit0	SA6 to SA0: 7-bit slave address	■ The 7-bit slave address register (ISBA), if the slave address detect is enabled (SAEN=1), compares the 7-bit data received after a (repeat) start condition detected with this register, and if all the bits are matched, it will operate as a slave and output ACK. At that time, the slave address received will be set to this register. (If SAEN=0, ACK will not be output.) ■ The address bits with "0" set on the ISMK register will be excluded from the comparison. **Notes:** ■ The reserved address cannot be set. ■ Set this register when the EN bit of the ISMK register is "0".

40.5 Operation of UART

The operation of UART is shown.

40.5.1 Interrupt of UART

40.5.2 Operation of UART

40.5.3 Setup Procedure and Program Flow

40.5.1 Interrupt of UART

Interrupt of UART is shown below.

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

40.5.1.1 List of Interrupt of UART

The list of interrupt of UART is shown below.

The following table indicates how UART interrupt control bits relate to interrupt factors.

Table 40-3. 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	○	○	Flaming 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)[1]
	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)[1]
	FDRQ	FCR1	○	○	Transmission FIFO is empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or the transmission FIFO is full

[1]: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

40.5.1.2 Reception Interrupts and Flag Setting Timing

Reception interrupts and flag setting timing are shown below.

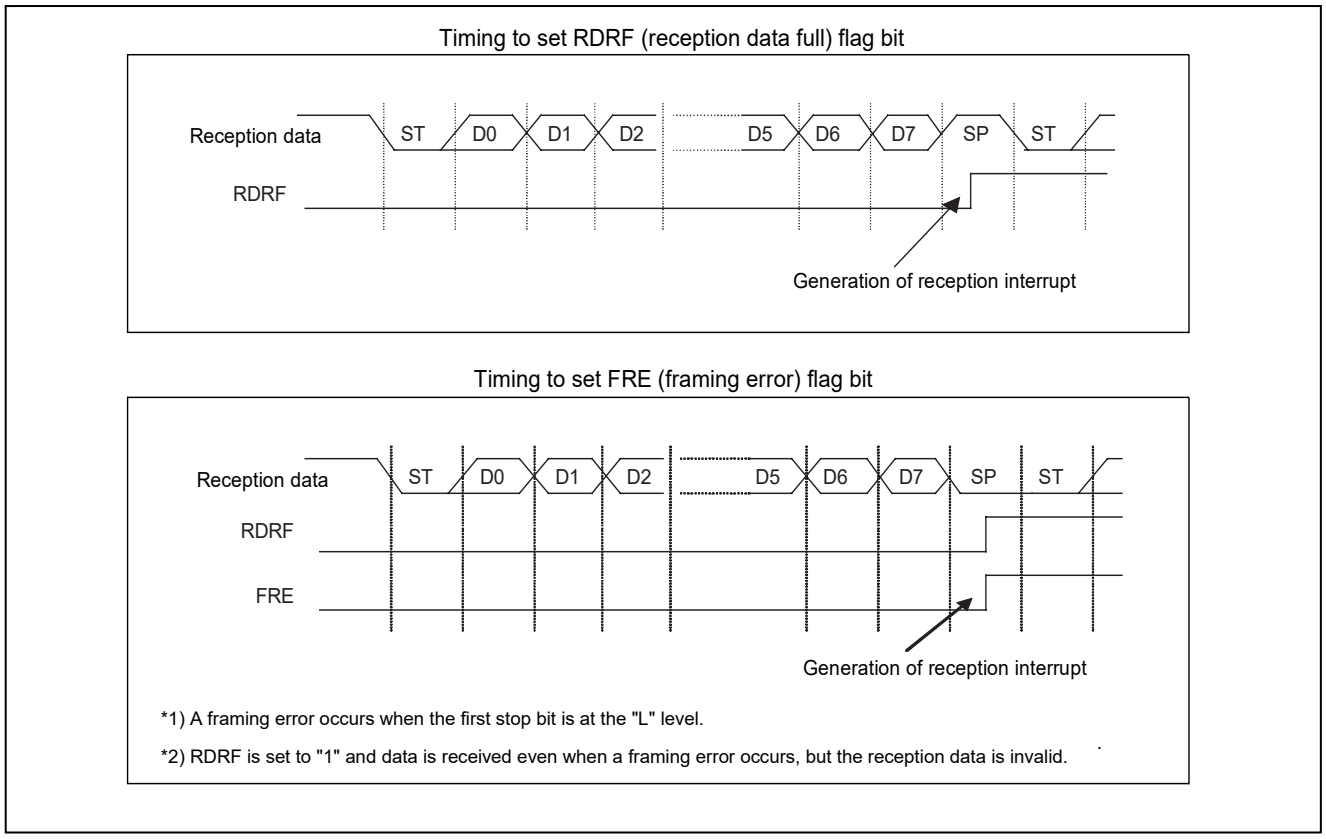
Reception interrupts occur either when the reception is completed (SSR:RDRF) or when a reception error occurs (SSR:PE, ORE, FRE).

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 at this time (SCR:RIE=1), a reception interrupt occurs.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

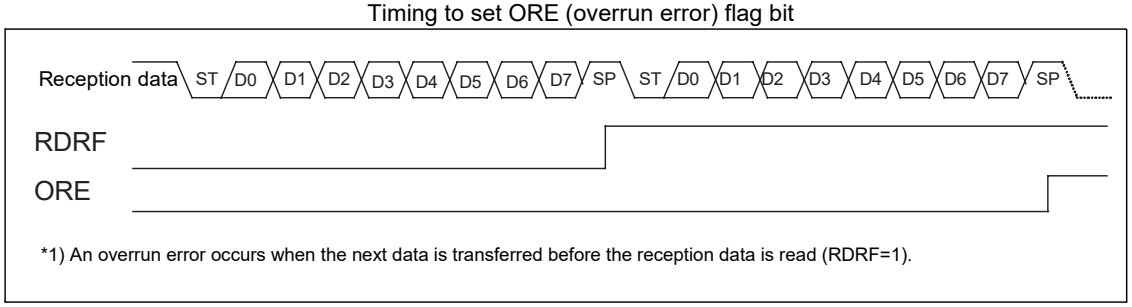
Figure 40-5. Timing of Flag Bit Setting



Note:

It becomes impossible to receive by invalidating the edge when the falling edge (ESCR:INV=0) or the rising edge (ESCR:INV =1) of the serial data is detected at the same clock or 1 or 2-machine clock earlier than the sampling point of the stop bit when it receives it. When the frame is continuously output, the interval of the frame is recommended to be opened.

Timing to set ORE (overrun error) flag bit



*1) An overrun error occurs when the next data is transferred before the reception data is read (RDRF=1).

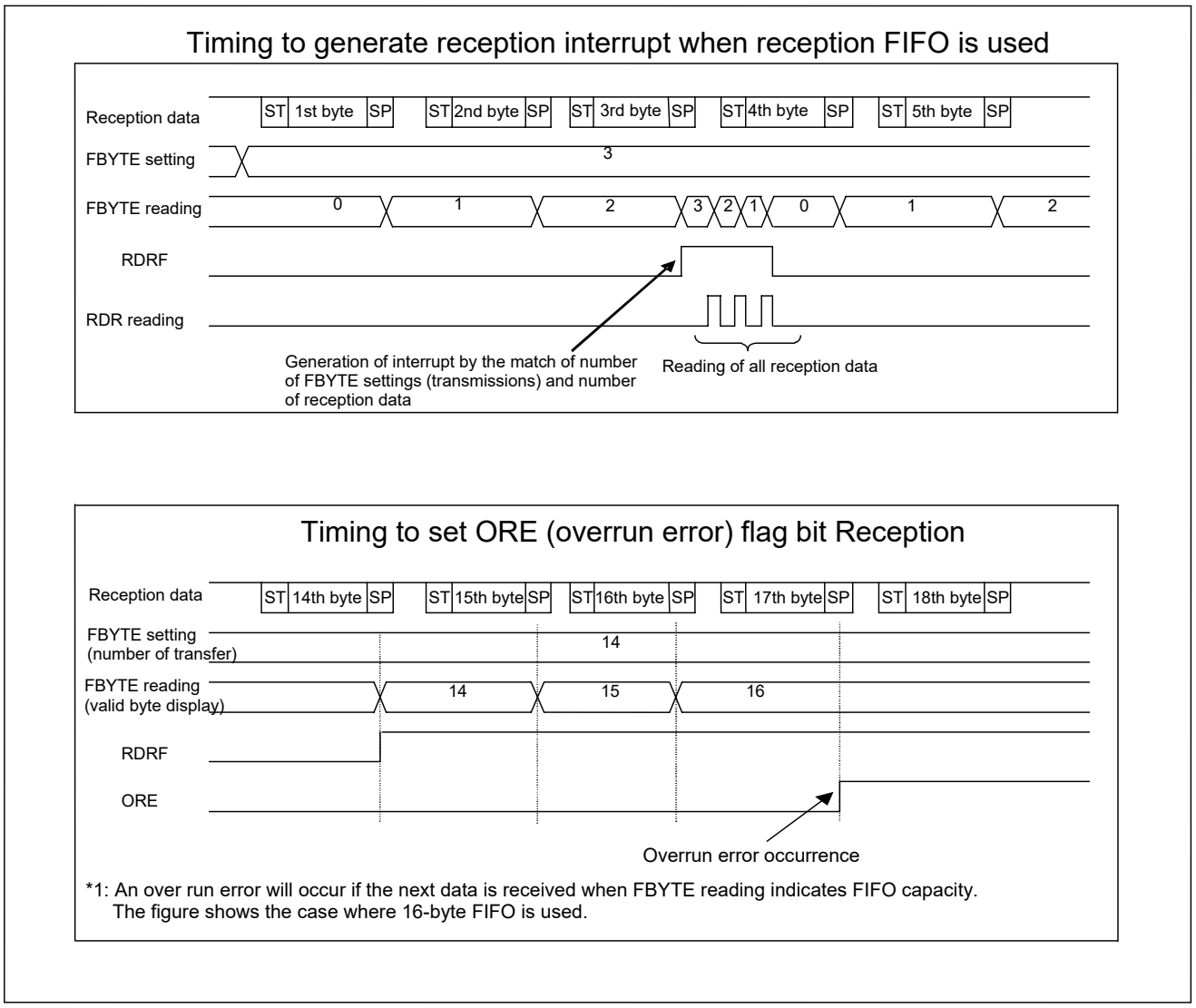
40.5.1.3 Interrupts When Using Reception FIFO and Flag Setting Timing

Interrupts when using reception FIFO and flag setting timing are shown below.

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.
- If the data count contained in the reception FIFO does not reach the transfer count while reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1", the interrupt flag (SSR:RDRF) will be set to "1" after the reception idle state continues for 8 baud rate clocks or longer. 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. If the reception FIFO is disabled, the counter is reset to 0. When the reception FIFO is enabled while there is data remaining in it, the counter starts counting 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 40-6. Timing of Using FIFO



40.5.1.4 Trans Mission Interrupts and Flag Setting Timing

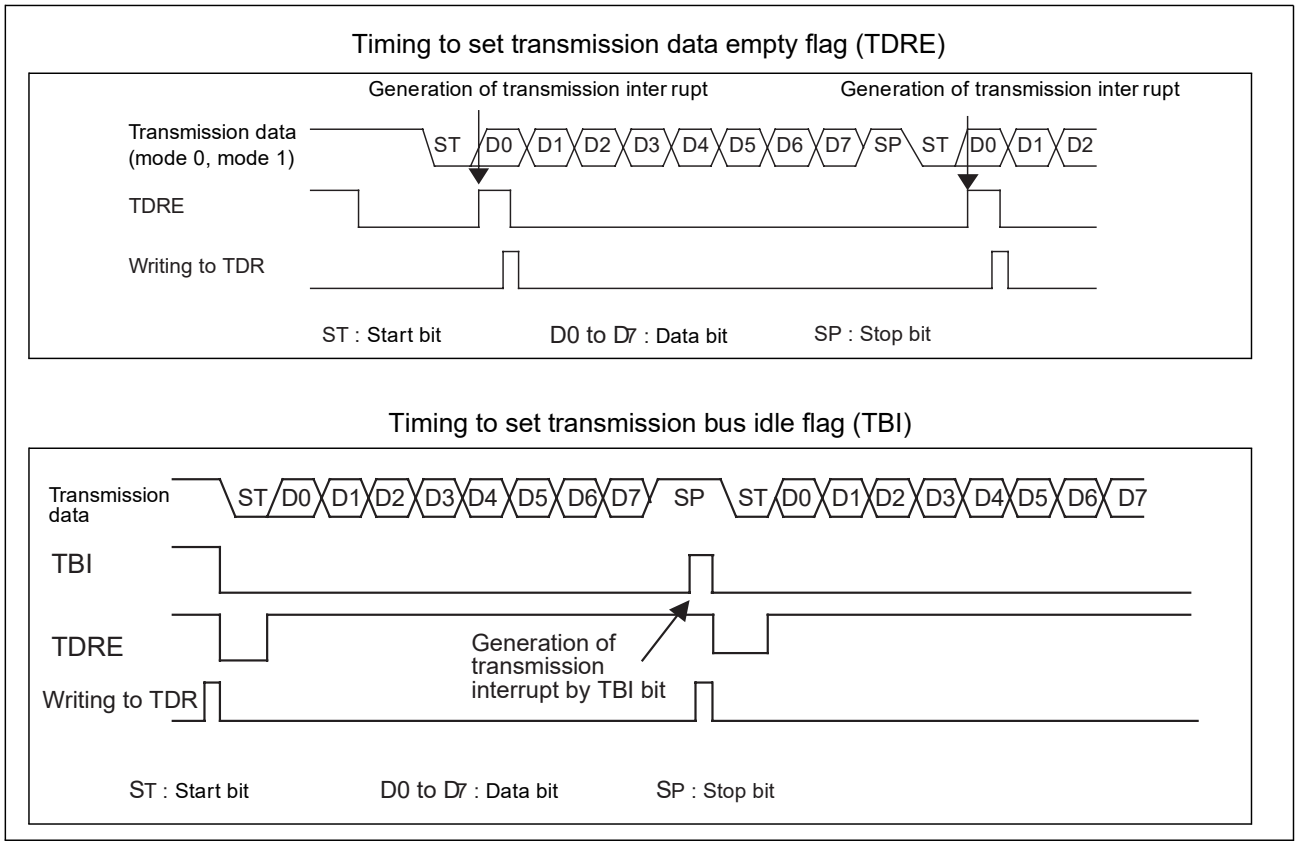
Transmission interrupts and flag setting timing are shown below.

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 SSR: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 40-7. Timing of Transmission Interrupt Flag



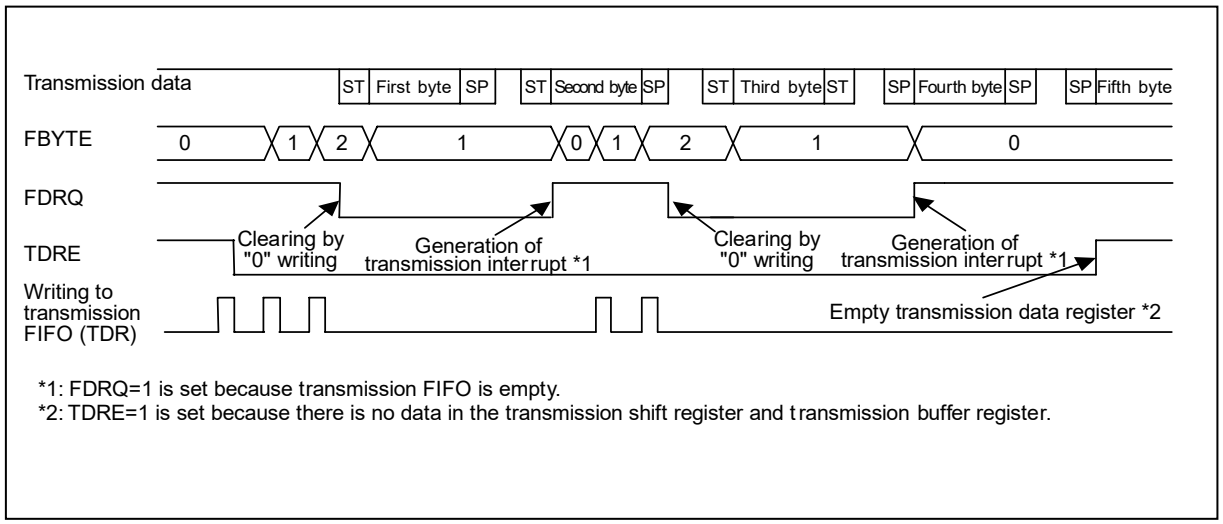
40.5.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

Interrupts when using transmission FIFO and flag setting timing are shown below.

When the transmission FIFO is used, an interrupt occurs when there is no data in the transmission FIFO.

- When there is no data in the transmission FIFO, 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).
- When FBYTE=00_H, there is no data in the transmission FIFO.

Figure 40-8. 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.

40.5.2 Operation of UART

The operation of UART is shown below.

UART operates with the mode 0 bidirectional serial asynchronous communication and the mode 1 master/slave multiprocessor communication.

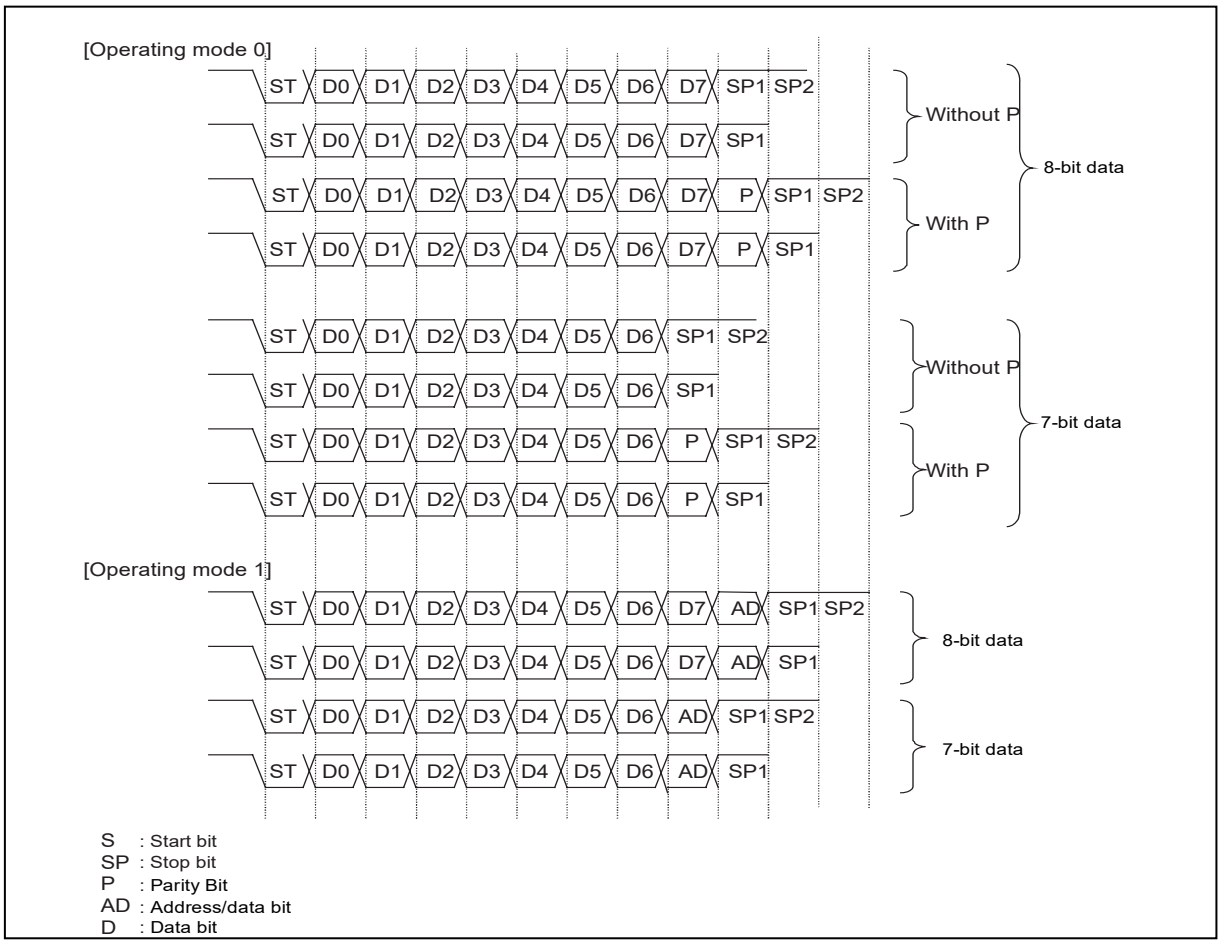
40.5.2.1 Transmission/Reception Data Format

Transmission/reception data format is shown below.

- 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 an operation with parity, 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 below:

Figure 40-9. 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 the 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 transmit/receive data register (RDR/TDR) will be enabled.

40.5.2.2 Transmission Operation

The transmission operation is shown below.

- 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 (TDRE) becomes "0".
- When the transmission operation enable bit (SCR:TXE) of the serial control register 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. Following transmission data can be written to the transmit data register when processing interrupts.

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 will occur, because the FIFO transmission data request bit (FCR1:FDRQ) has the initial value "1".

40.5.2.3 Reception Operation

The reception operation is shown below.

- 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 serial control register (ESCR:PEN, P, L2, L1, L0) and serial mode register (SMR:BDS). Detection of the start bit is as follows; falling edge (in case of ESCR:INV=0) or rising edge (in case of ESCR:INV=1) is detected after passing through the noise filter (which samples serial data input in 3 machine clock and decides the value by majority), and the data "L" is detected after the noise filter 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.
- When you read a reception data, do it 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".
- If the data count contained in the reception FIFO does not reach the transfer count while reception FIFO idle detection enable bit (FRIIE) is set to "1", the interrupt flag (RDRF) will be set to "1" after the reception idle state continues for 8 baud rate clocks or longer. 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. If the reception FIFO is disabled, the counter is reset to 0. When the reception FIFO is enabled while there is data remaining in it, the counter starts counting again.
- When the reception FIFO is enabled, if the error flag of the serial status register 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 data count which have 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 exhausted of 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 (The serial data input is sampled three times with the machine clock and decision by majority) is built into. Design the board so that the noise should not pass this filter as the measures or communicate by noise passing so as not to become a problem (For instance, when the error occurs adding the checksum of data at the end, send it again).
- It becomes impossible to receive by making the edge invalidity etc. when the falling edge (ESCR:INV =0) or the rising edge (ESCR:INV =1) of the serial data is detected at the same clock or 1 or 2-machine clock earlier than the sampling point of the stop bit when it receives it. When the frame is continuously output, the interval of the frame is recommended to be opened.

40.5.2.4 Clock Selection

The clock selection is shown below.

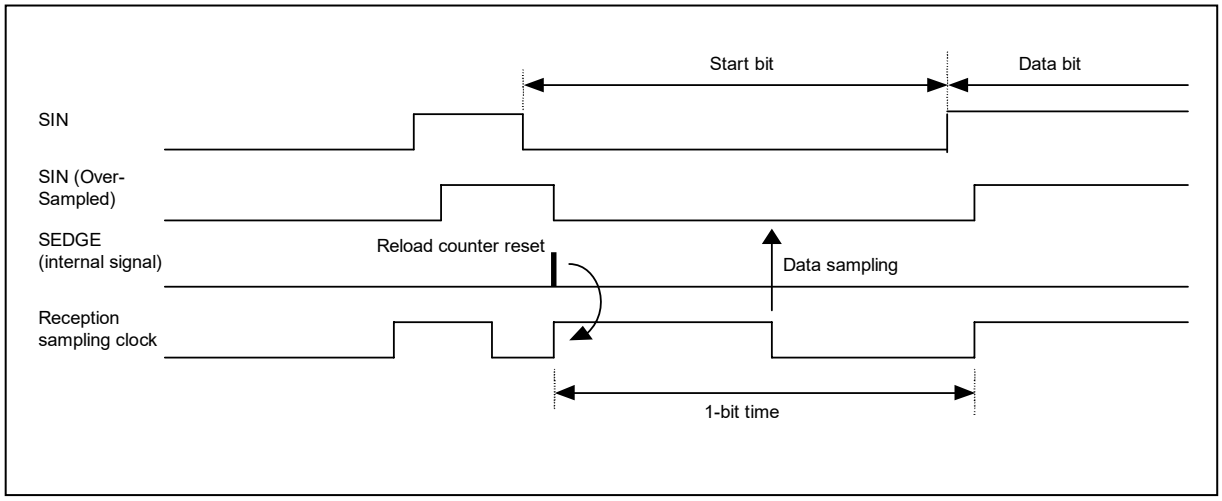
- Internal clocks or external clocks can be used.
- When you use an external clock, set BGR:EXT=1. In this case, the external clock is divided in the baud rate generator.

40.5.2.5 Start Bit Detection

The start bit detection is shown below.

- 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 40-10. Start Bit Detection



40.5.2.6 Stop Bit

The stop bit is shown below.

- You can select 1bit to 4bit length.
- The reception data full flag bit (SSR:RDRF) will be set to "1" when the first stop bit is detected.

40.5.2.7 Error Detection

The error detection is shown below.

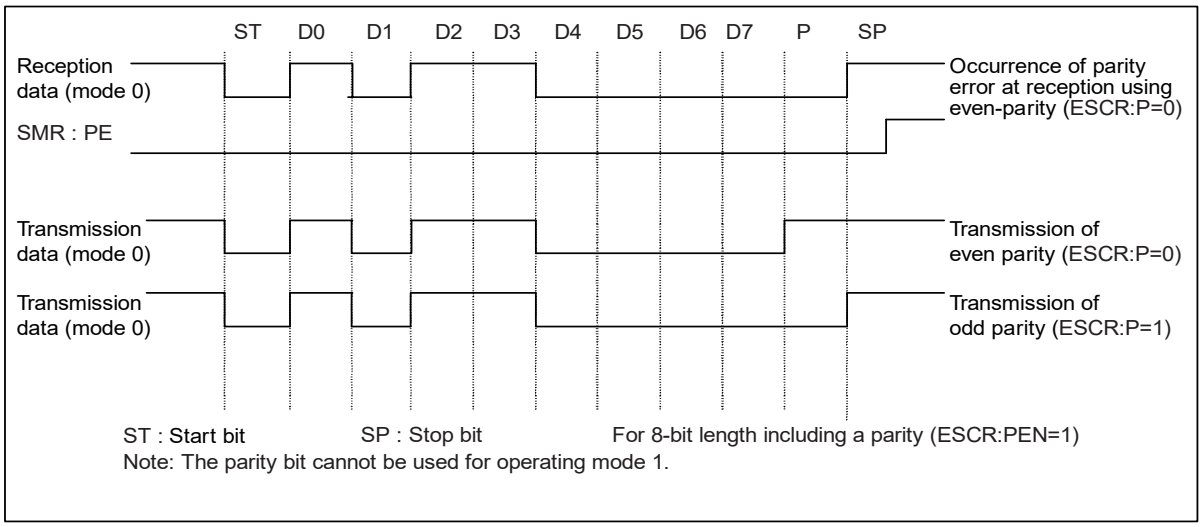
- In operation mode 0, parity errors, overrun errors, frame errors can be detected.
- In operation mode 1, overrun errors and frame errors can be detected. Parity errors cannot be detected.

40.5.2.8 Parity Bit

The parity bit is shown below.

- 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.
- The parity can be used in operation mode 1.

Figure 40-11. Operation with Parity Enabled

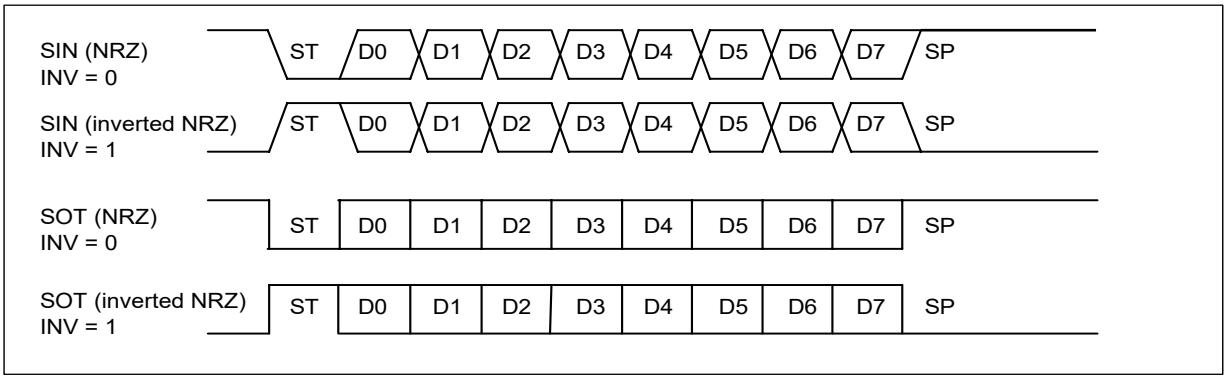


40.5.2.9 Data Signaling Method

The data signaling method is shown below.

- The INV bit setting of the extended serial 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 40-12. NRZ (Non Return to Zero) Signaling Method and Inverted NRZ Signaling Method



40.5.2.10 Data Transfer Method

The data transfer method is shown below.

LSB first or MSB first can be selected on the data bit transfer method.

40.5.2.11 UART Baud Rate Selection/Setting

The UART baud rate selection/setting is shown below.

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 resonator 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 peripheral clocks (PCLK) 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:

- Reload value

  $V = \varphi / b - 1$

  V: Reload value
  b: Baud rate
  φ: Internal clock (peripheral clock (PCLK)) or external clock frequency

- Example of calculation

  Followings are the calculation of the reload value if the internal clock (peripheral clock (PCLK)) frequency is 16 MHz and the baud rate is set to 19200 bps:

  $V = (16 \times 1{,}000{,}000) / 19200 - 1$
  $= 832$

  The baud rate when using this reload value is:

  $b = (16 \times 1{,}000{,}000) / (832 + 1)$
  $= 19208$ bps

- Baud rate error

  The baud rate error can be obtained using the following formula:

  Error (%) = (calculated value - desired value) / desired value × 100

  (Example) Internal clock 20 MHz, Target baud rate value 153600 bps

Reload value	= (20 × 10000000) / 153600 - 1	= 129
Baud rate (calculated value)	= (20 × 10000000) / (129 + 1)	= 153846 bps
Error (%)	= (153846 - 153600) / 153600 × 100	= 0.16(%)

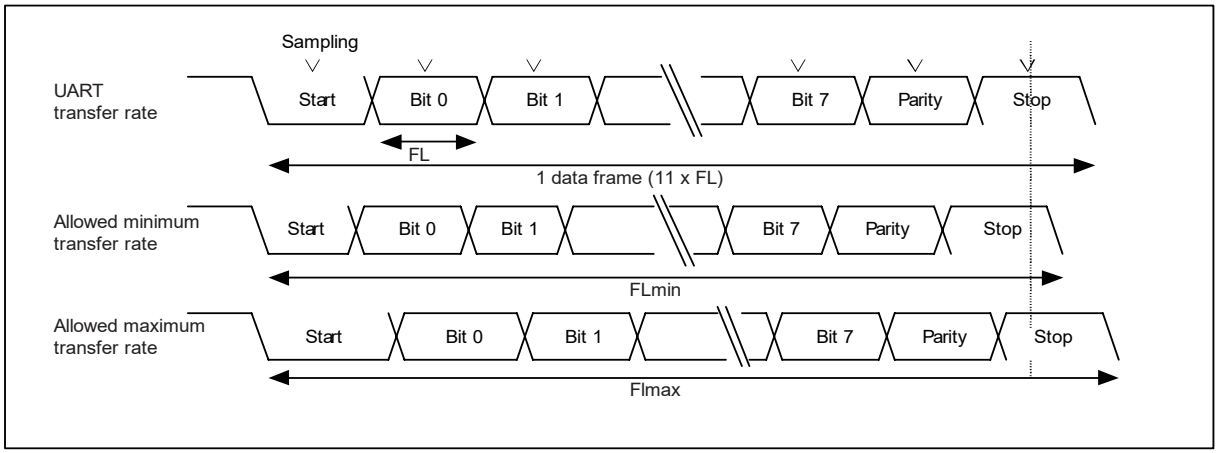
Notes:

- 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 peripheral clock (PCLK) 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 for 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 40-13. 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 on time at 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 peripheral clock (PCLK) (φ), the allowed minimum transfer rate (FLmin) would be calculated as follows:

$$FLmin = (11bit \times (V+1) - (V+1) / 2 + 2) / \varphi = (21V+25) / 2\varphi \ (s)$$

V: Reload value
φ: Internal clock (peripheral clock (PCLK))(Hz)

Therefore, the allowed maximum baud rate (BGmax) at the destination would be calculated as follows:

$$BGmax = 11/FLmin = 22\varphi/(21V+25) \ (bps)$$

V: Reload value
φ: Internal clock (peripheral clock (PCLK))(Hz)

When the allowance and maximum transfer rate (FLmax) receives the data, sampling is done in the starting point of received data in the 11th bit. Therefore, the allowance and maximum transfer rate (FLmax) is as follows:

$$10/11 \times FLmax = (11bit \times (V+1) - (V+1)/2)/\varphi$$

V: Reload value
φ: Internal clock (peripheral clock (PCLK))(Hz)

FLmax = (21/20×11×(V+1))/φ

When margin (φ) of the sampling timing is made two clocks, the allowance and maximum transfer rate (FLmax) is as follows:

FLmax = (21/20×11× (V+1) - 2)/ φ= (231V+191)/20φ(s)

V: Reload value
φ: Internal clock (peripheral clock (PCLK))(Hz)

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: Internal clock (peripheral clock (PCLK))(Hz)

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 40-4. 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, internal clock (peripheral clock (PCLK)), and the reload value. The higher the internal 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 40-5. 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		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4M	-	-	-	-	-	0	4	0	5	0	7	0
2.5M	-	-	-	0	-	-	-	-	-	-	-	-
2M	-	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
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	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, and 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

  Set "1" to the programmable clear bit (SCR:UPCL bit)

- Reception reload counter

  Detection of a start bit falling edge in asynchronous mode

40.5.3 Setup Procedure and Program Flow

The setup procedure and program flow are shown.

40.5.3.1 Operation Mode 0 (One-to-One Connection)

40.5.3.2 Operation Mode 1 (One-to-N Connection)

40.5.3.1 Operation Mode 0 (One-to-One Connection)

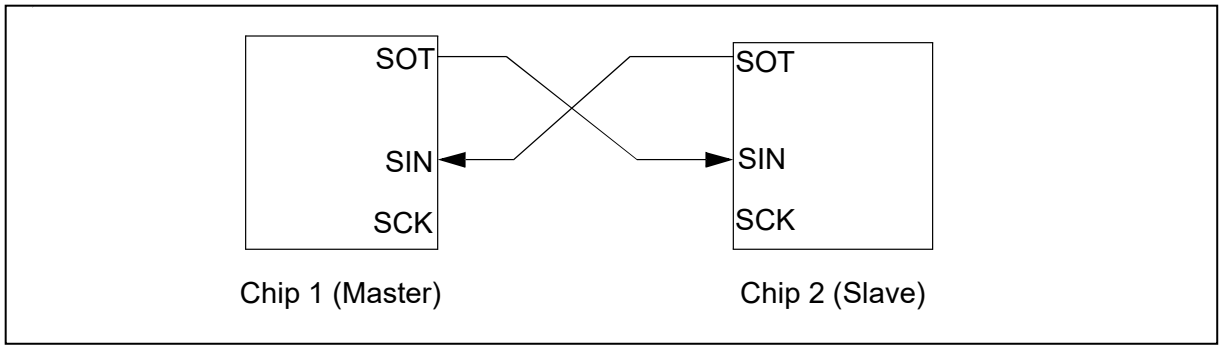
Operation mode 0 (One-to-One Connection) is shown below.

In operation mode 0, asynchronous serial bidirectional communications can be performed.

Connections between Chips

In operation mode 0 (normal mode), select bidirectional communications. Two CPUs are inter-connected as shown below:

Figure 40-14. Example of Connection for Bidirectional Communications in UART Operation Mode 0



Flowchart

Figure 40-15. Example of Settings for Bidirectional Communications (FIFO Not Used)

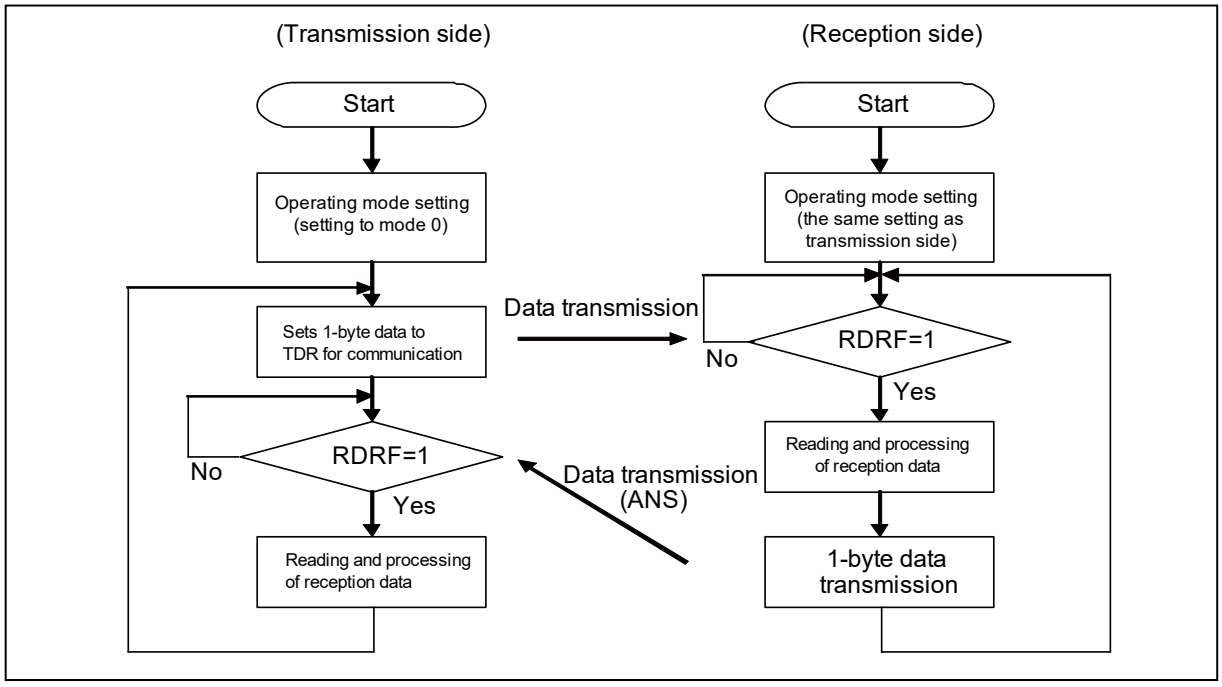
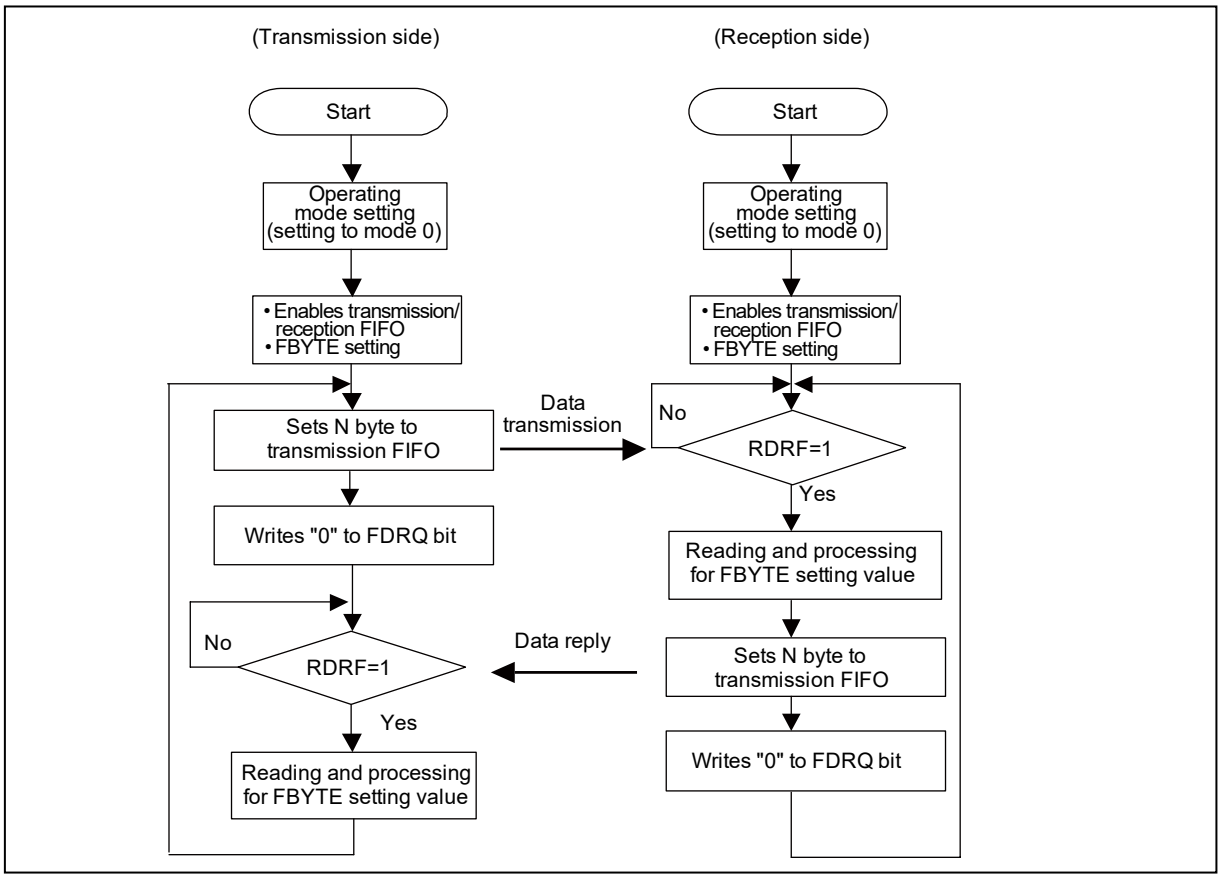


Figure 40-16. Example of Settings for Bidirectional Communications (FIFO Used)



40.5.3.2 Operation Mode 1 (One-to-N Connection)

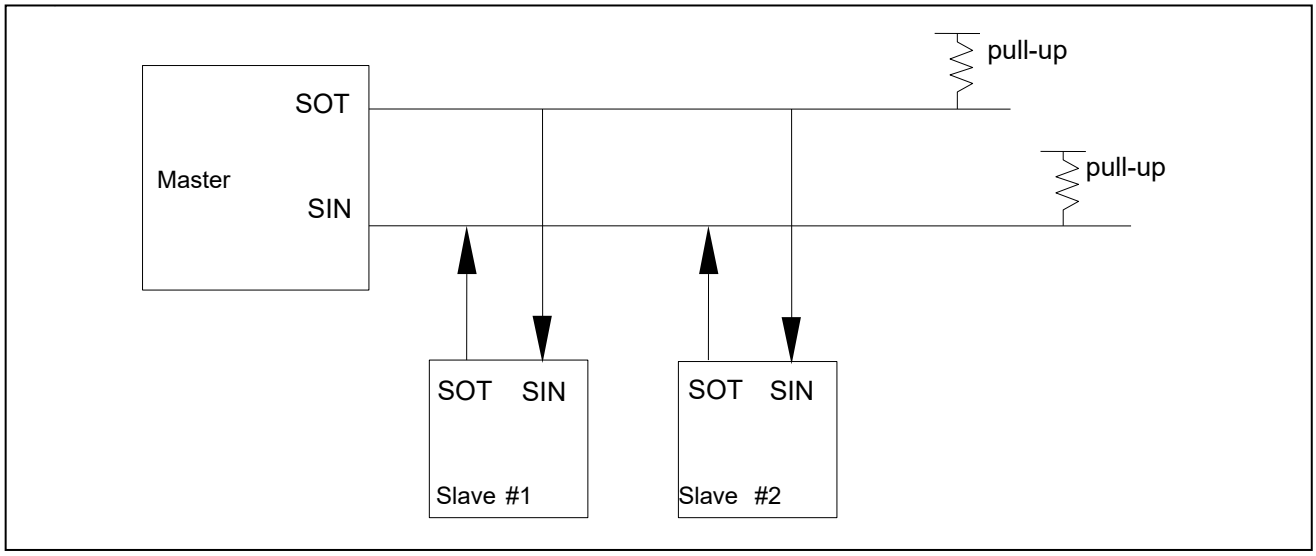
Operation mode 1 (One-to-N Connection) is shown below.

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.

Connections between Chips

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 40-17. 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 follows:

Table 40-6. Selection of Master-Slave Communication Function

	Operation mode	Data	Parity	Stop bit	Bit direction	Operation mode
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:

Access the transmit and receive data (TDR/RDR) in word access 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. The following shows a flowchart of master-slave communications (multi-processor mode).

Figure 40-18. Example of Flowchart of Master-Slave Communications (FIFO Not Used)

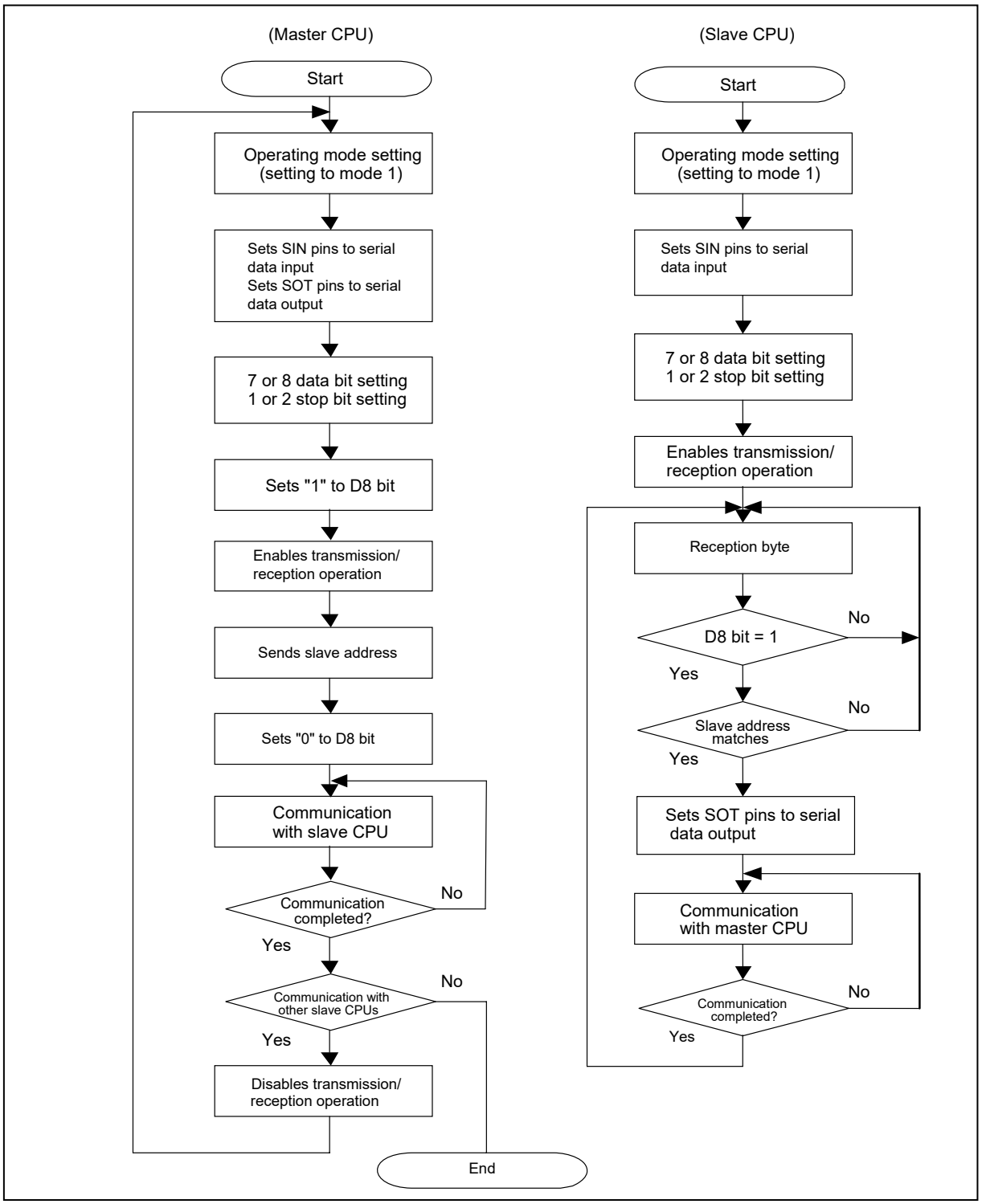
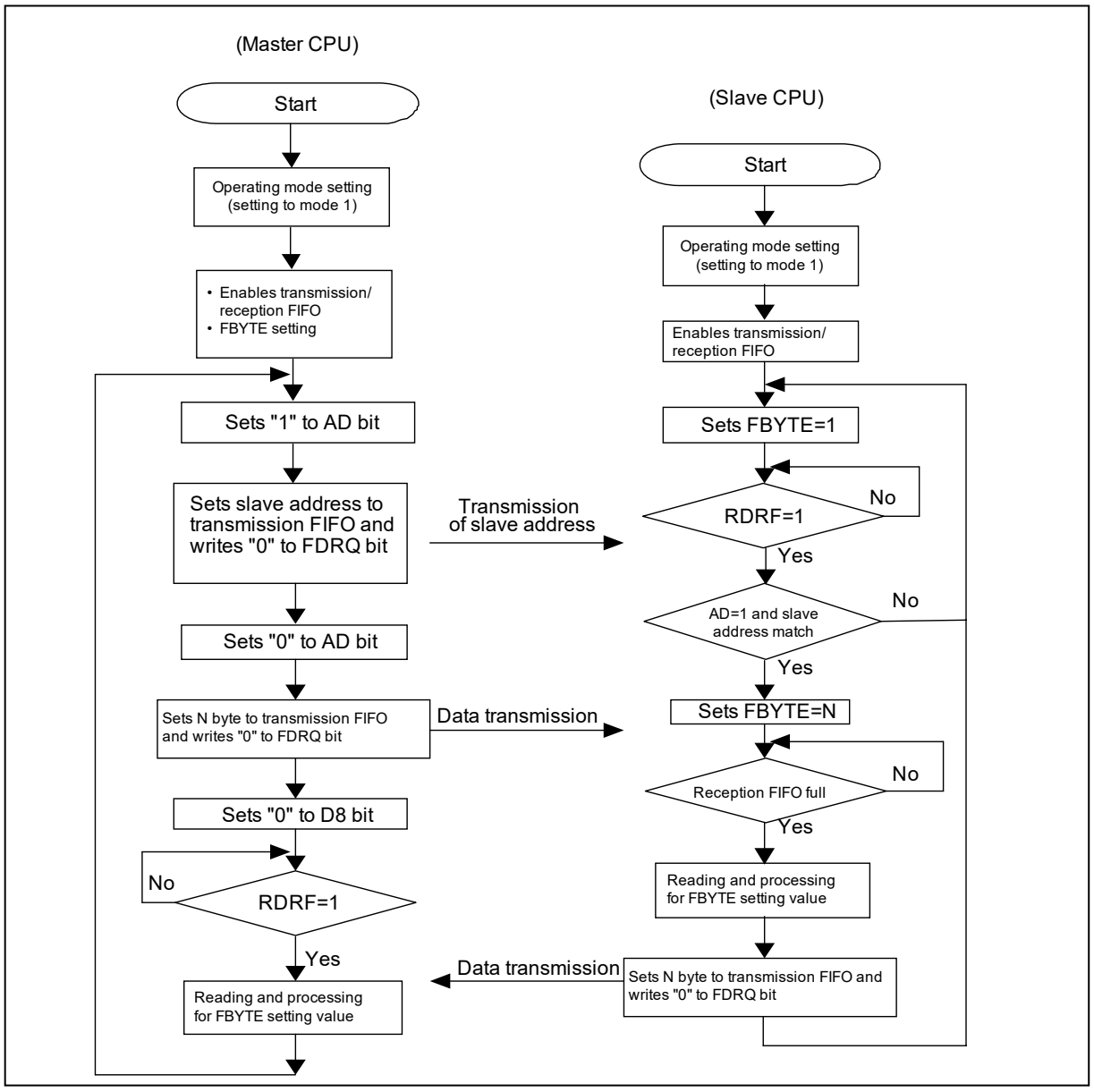


Figure 40-19. Example of Flowchart of Master-Slave Communications (FIFO Used)



40.6 Operation of CSIO

The operation of CSIO is shown.

40.6.1 Interrupts of CSIO
40.6.2 Operation of CSIO
40.6.3 Setup Procedure and Program Flow

40.6.1 Interrupts of CSIO

Interrupts of CSIO are shown below.

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

40.6.1.1 List of Interrupts of CSIO

The list of Interrupts of CSIO is shown below.

Table 40-7. Interrupt Control Bits and Interrupt Factors of CSIO

Interrupt Type	Interrupt request flag bit	Flag register	Interrupt factor	Interrupt source 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)[1]
	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)[1]
	FDRQ	FCR1	Transmission FIFO is empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ) or transmission FIFO is full

[1]: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

40.6.1.2 Reception Interrupts and Flag Setting Timing

Reception interrupts and flag setting timing are shown below.

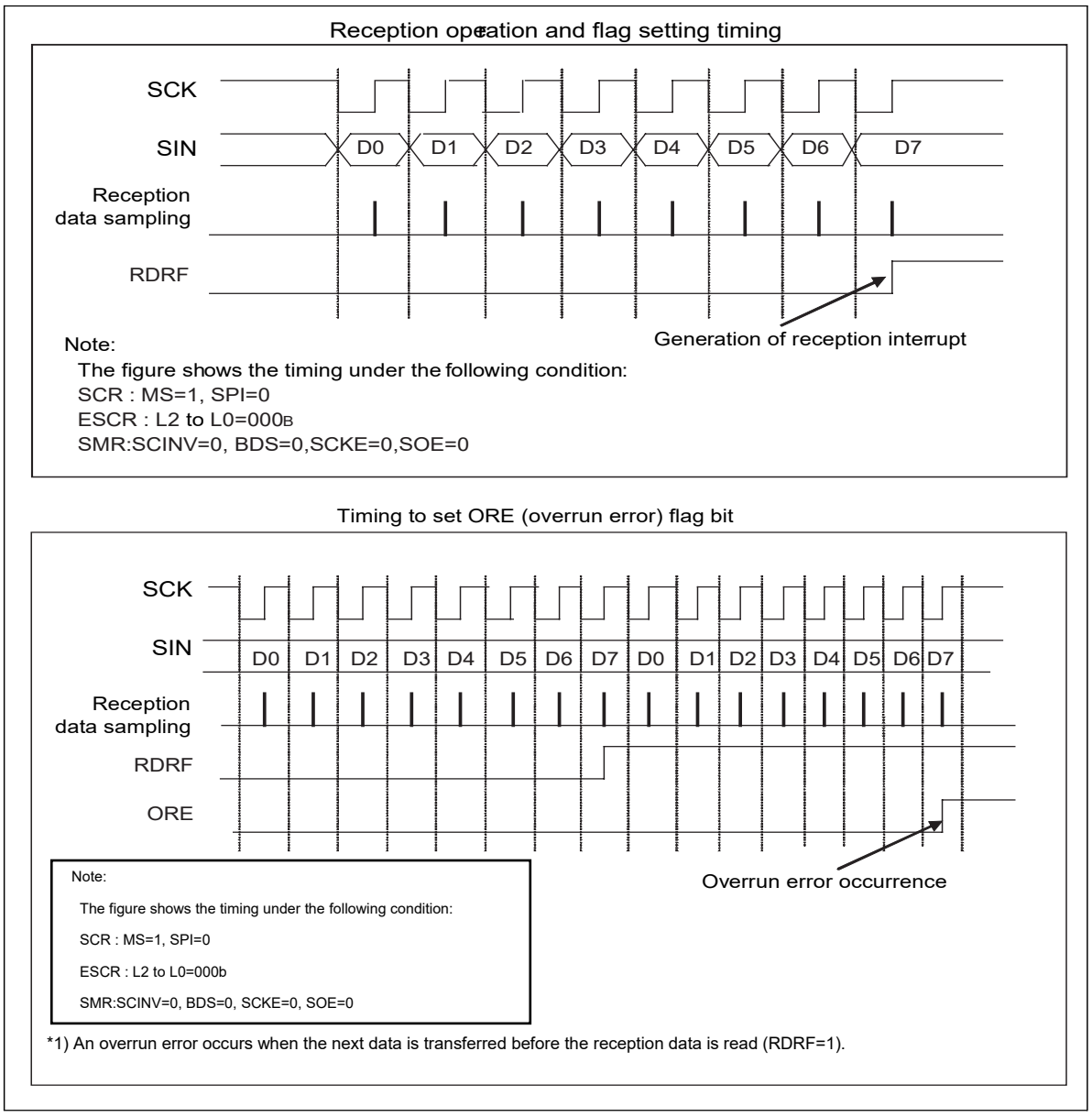
Reception interrupts occur either when the reception is completed (SSR:RDRF) or when a reception error occurs (SSR:ORE).

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 at this time (SCR:RIE=1), a reception interrupt occurs.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

Figure 40-20. Timing of Flag Setting



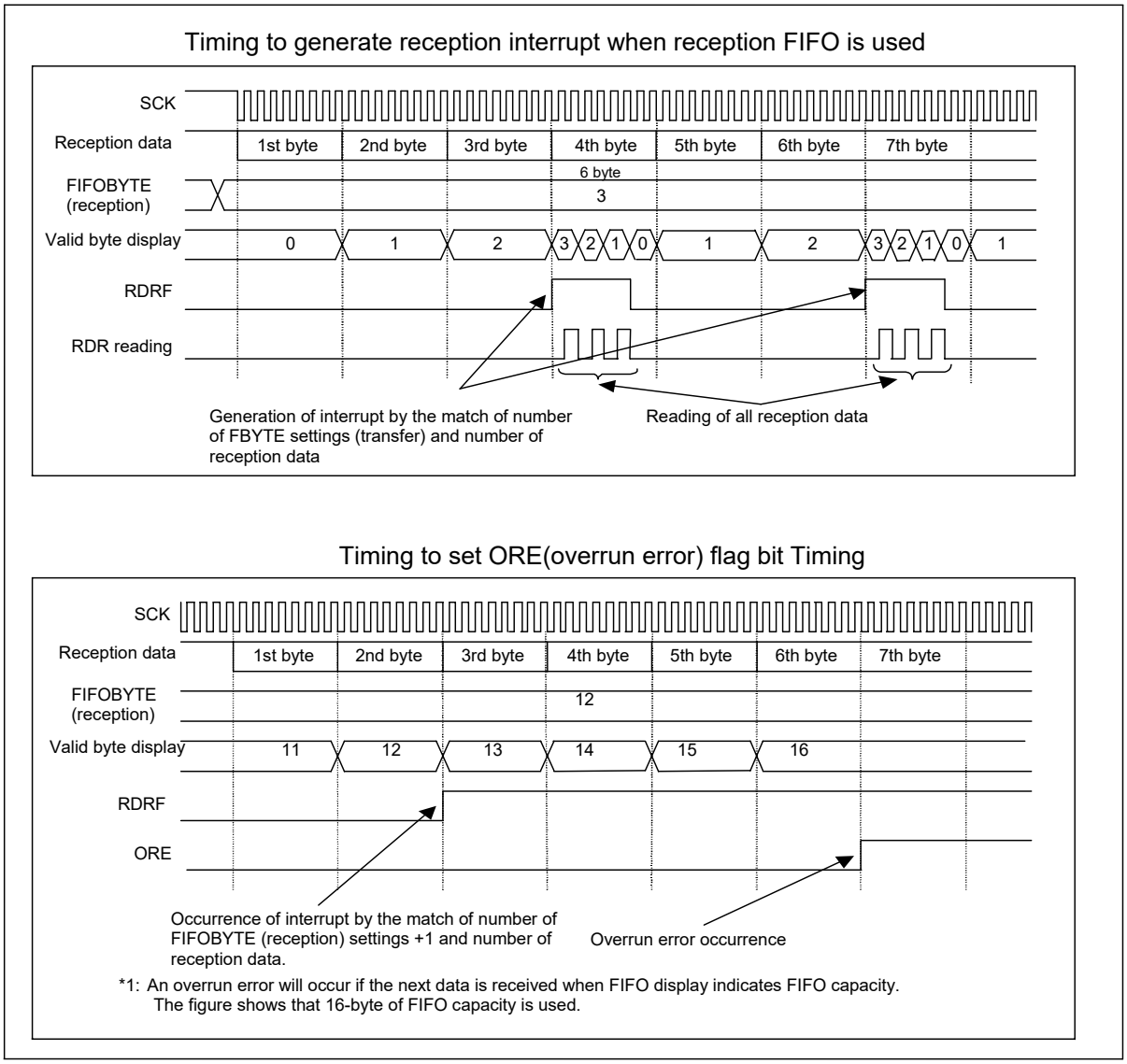
40.6.1.3 Interrupts when Using Reception FIFO and Flag Setting Timing

Interrupts when Using Reception FIFO and Flag Setting Timing are shown below.

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.
- If the data count contained in the reception FIFO does not reach the transfer count while reception FIFO idle detection enable bit (FRIIE) is set to "1", the interrupt flag (RDRF) will be set to "1" after the reception idle state continues for 8 baud rate clocks or longer. 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. If the reception FIFO is disabled, the counter is reset to 0. When the reception FIFO is enabled while there is data remaining in it, the counter starts counting 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 40-21. Timing of Interrupts and Flag Setting



40.6.1.4 Transmission Interrupts and Flag Setting Timing

Transmission interrupts and flag setting timing are shown below.

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).

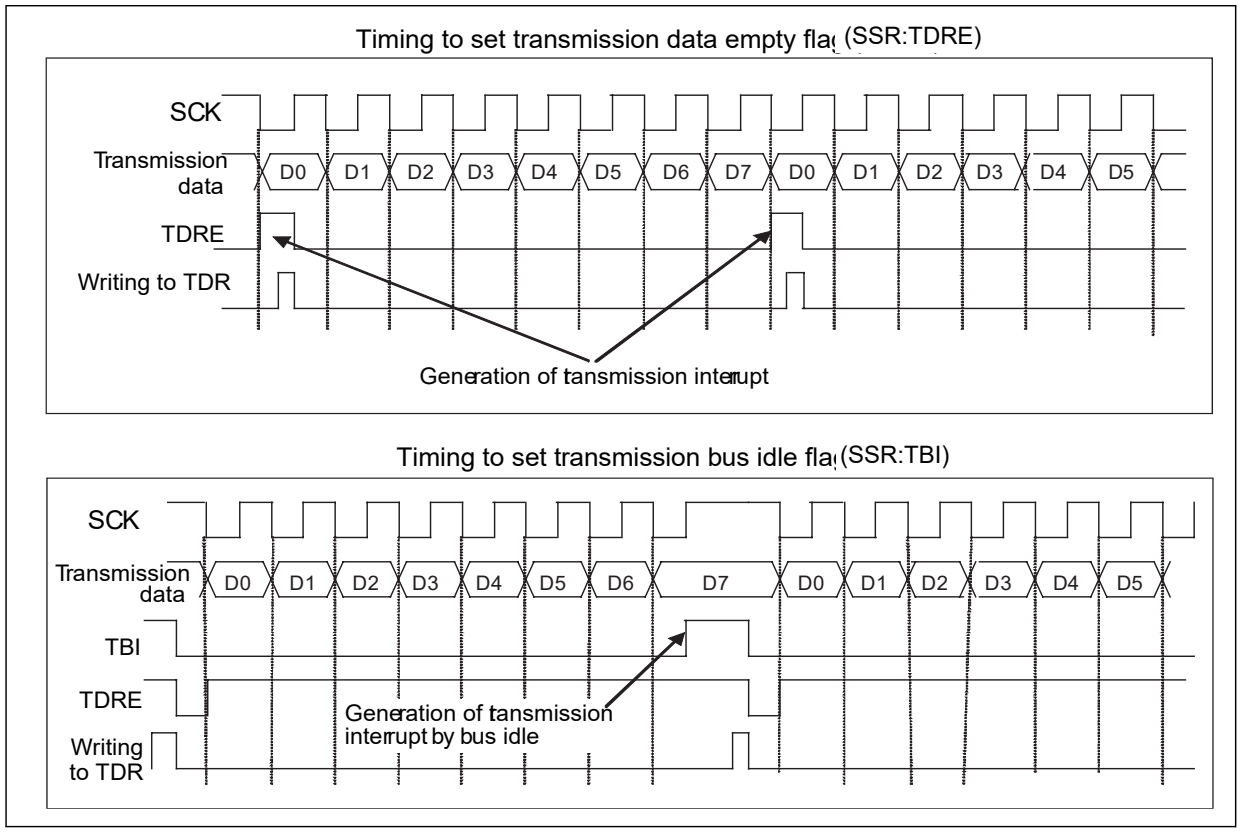
Timing of transmission data empty flag (SSR:TDRE) setting

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 (SSR:TBI) setting

When the transmit data register is empty (SSR: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 40-22. Timing of Flag Setting



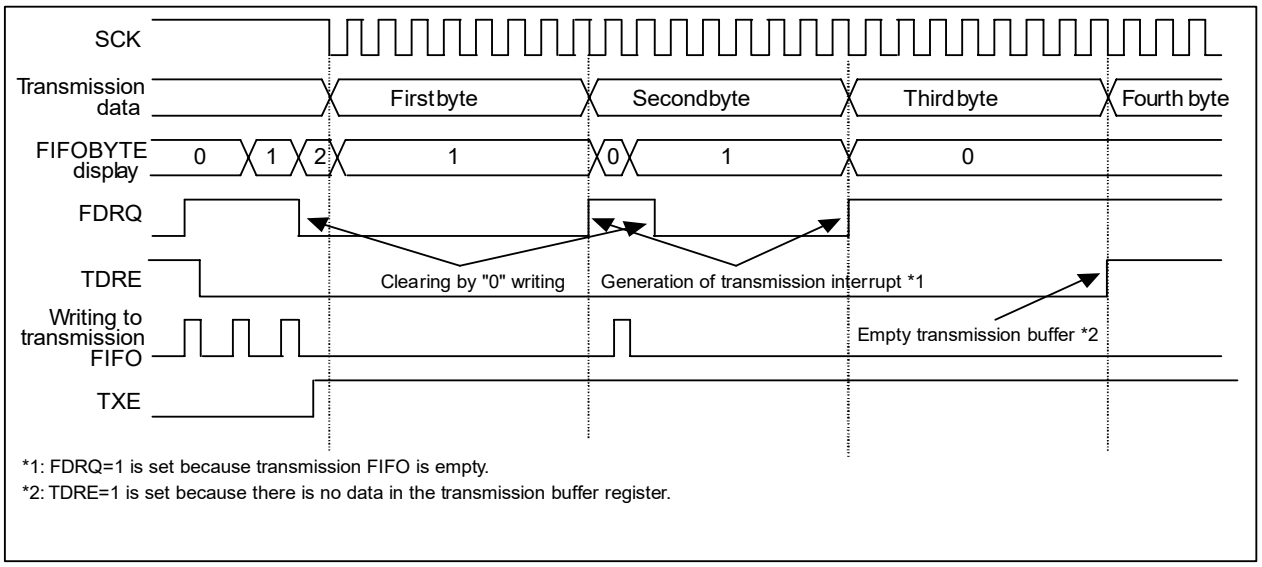
40.6.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

Interrupts when using transmission FIFO and flag setting timing are shown below.

When the transmission FIFO is used, an interrupt occurs when there is no data in the transmission FIFO.

- When there is no data in the transmission FIFO, 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). When FBYTE=00_H, there is no data in the transmission FIFO.

Figure 40-23. Timing of Interrupt Generation



40.6.2 Operation of CSIO

The operation of CSIO is shown.

40.6.2.1 Normal Transfer (I)
40.6.2.2 Normal Transfer (II)
40.6.2.3 SPI Transfer (I)
40.6.2.4 SPI Transfer (II)
40.6.2.5 Baud Rate Generation

40.6.2.1 Normal Transfer (I)

Normal Transfer (I) is shown below.

Features

	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 9 bits

Register Settings

The following table lists the register settings required for normal transfer (I).

Table 40-8. Normal Transfer (I) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	[1]	[1]	[1]	[1]	[1]	0	1	0	-	0	[1]	1/0	1/0
SSR/ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	[1]	[1]	[1]	[1]	[1]
TDR/RDR								D8	D7	D6	D5	D4	D3	D2	D1	D0
								[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
BGR	-	BGR[14:8]							BGR[7:0]							
	-	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

1: Set to "1"

0: Set to "0"

*: User-configurable setting

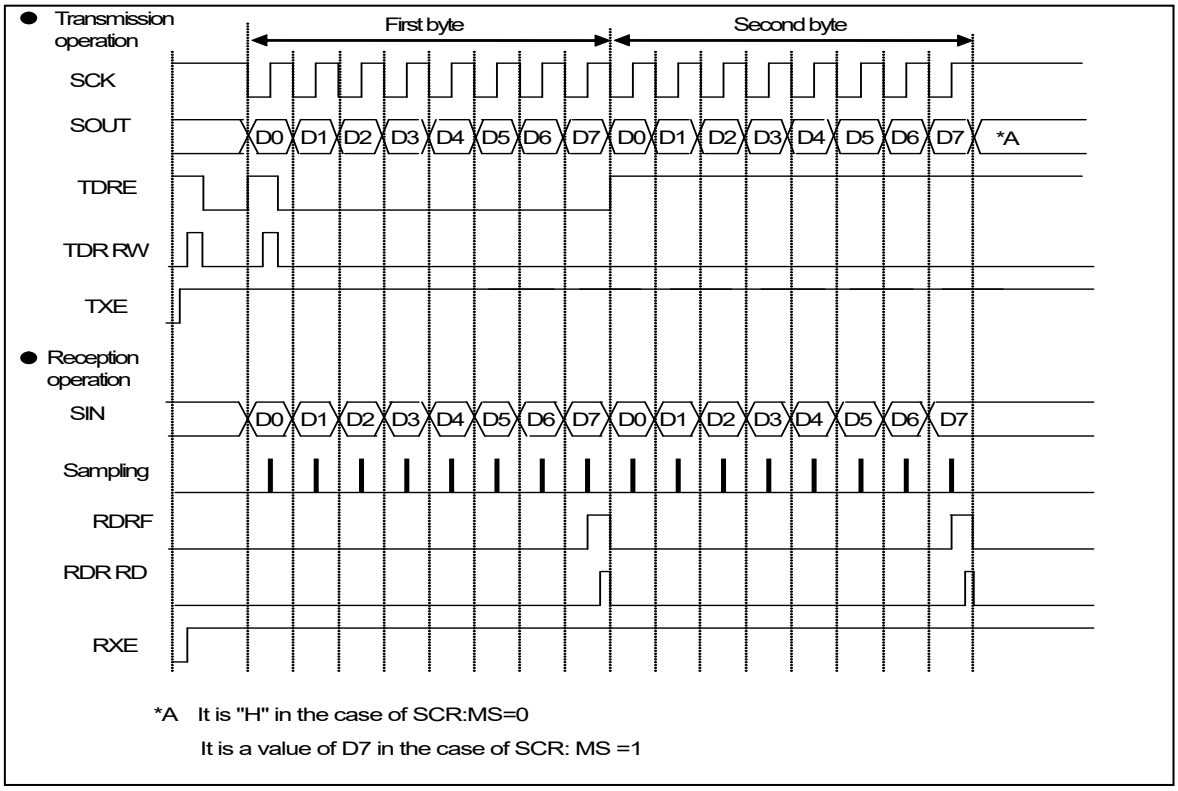
Note:

The above bit settings (1 or 0) are different between the master and slave operations. Set the bits as follows:

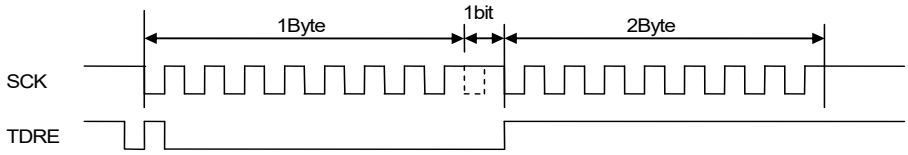
Master transmission: SCR:MS=0, SMR:SCKE=1, SOE=1
Master reception : SCR:MS=0, SMR:SCKE=1, SOE=0
Slave transmission : SCR:MS=1, SMR:SCKE=0, SOE=1
Slave reception : SCR:MS=1, SMR:SCKE=0, SOE=0

Normal Transfer (I) Timing Chart

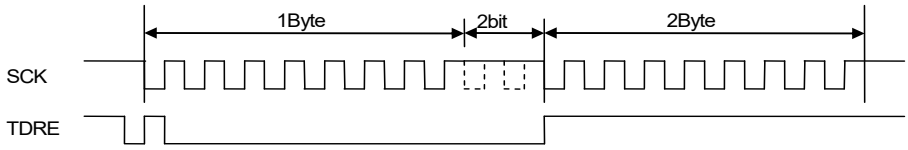
Figure 40-24. Normal Transfer (I) Timing Chart



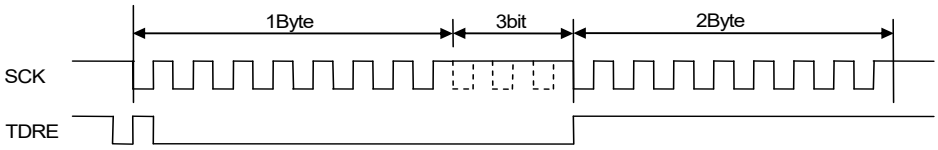
■ ESCR:WT1=0, ESCR:WT0=1 (for master)



■ ESCR:WT1=1, ESCR:WT0=0 (for master)



■ ESCR:WT1=1, ESCR:WT0=1 (for master)



Operation Explanation

Master operation (Set SCR:MS=0, SMR:SCKE=1.)

- Transmission operation
  - 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 transmission data in synchronization with a falling edge of the serial clock (SCK) output.
  - 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
  - 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).
  - 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.
  - 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
  - When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - If the transmission data is written in TDR, the SSR:TDRE is set as 0 also synchronization with the falling edge of serial clock (SCK) output and the transmission data is output. If the transmission data of the first bit outputted, the SSR:TDRE bit is set as 1 and when transmission interrupt enabled (SCR:TIE=1) is done, the transmission interrupt enabled is output. At this time, the transmission data of the second byte can be written.
  - The reception data is sampled at the rising edge of serial clock (SCK) output. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".
- Successive data transmission or reception wait operation

  If setting other than (ESCR:WT1, ESCR:WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.

Slave operation (Set SCR:MS=1, SMR:SCKE=0.)

- Transmission operation
  - With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0 and outputs the transmission data in synchronization with a falling edge of the serial clock (SCK) input.
  - 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 is outputted. At this time, the transmission data in the second byte can be written.
- Reception operation
  - 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).
  - Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is outputted. At this time, the receive data (RDR) can be read.
  - Reading the receive data (RDR) clears SSR:RDRF to "0".
- Transmission/ Reception operation
  - When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - If the transmission data is written in TDR, the SSR:TDRE is set as 0 also synchronization with the falling edge of serial clock (SCK) input and the transmission data is output. If the transmission data of the first bit outputted, the SSR:TDRE bit is set as 1 and when transmission interrupt enabled (SCR:TIE=1) is done, the transmission interrupt enabled is output. At this time, the transmission data of the second byte can be written.
  - The reception data is sampled by the rising edge of serial clock (SCK) input. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, the receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".

40.6.2.2 Normal Transfer (II)

Normal Transfer (II) is shown below.

Features

	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 9 bits

Register Settings

The following table lists the register settings required for normal transfer (II).

Table 40-9. Normal Transfer (II) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	[1]	[1]	[1]	[1]	[1]	0	1	0	-	1	[1]	1/0	1/0
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	[1]	[1]	[1]	[1]	[1]
TDR/ RDR								D8	D7	D6	D5	D4	D3	D2	D1	D0
								[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
BGR	-	BGR[14:8]							BGR[7:0]							
	-	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

1: Set to "1"

0: Set to "0"

[1]: User-configurable setting

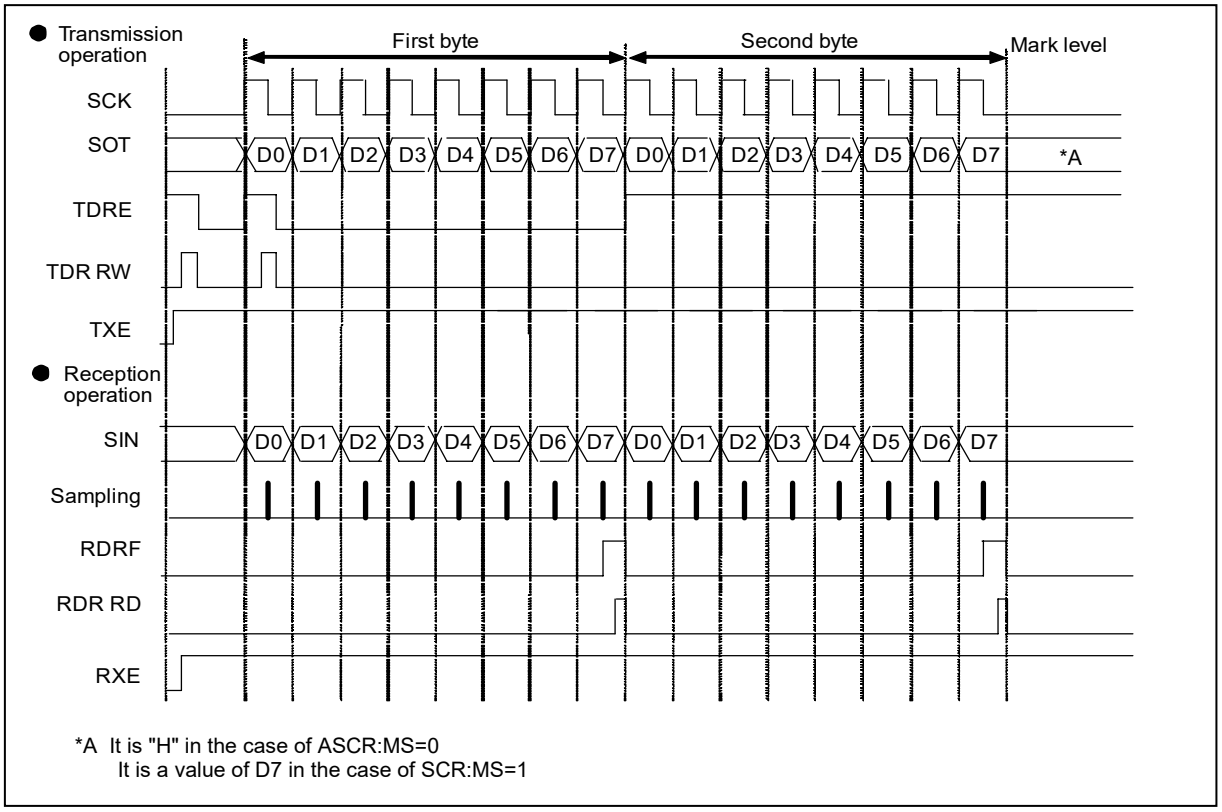
Note:

The above bit settings (1 or 0) are different between the master and slave operations. Set the bits as follows:

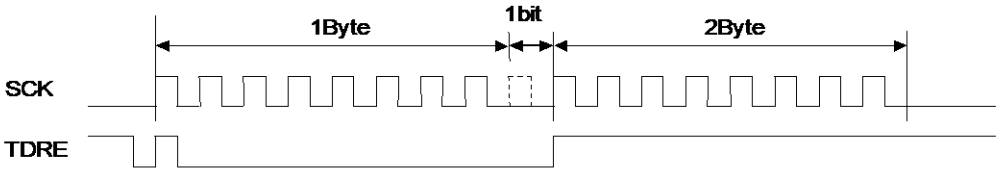
Master transmission : SCR:MS=0, SMR:SCKE=1, SOE=1
Master reception : SCR:MS=0, SMR:SCKE=1, SOE=0
Slave transmission : SCR:MS=1, SMR:SCKE=0, SOE=1
Slave reception : SCR:MS=1, SMR:SCKE=0, SOE=0

Normal Transfer (II) Timing Chart

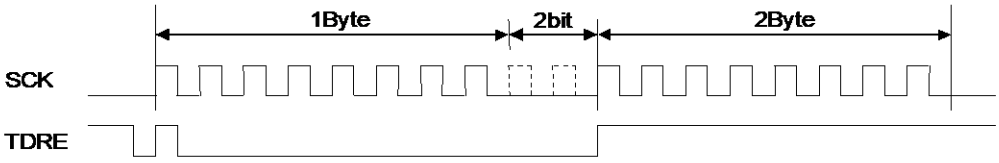
Figure 40-25. Normal Transfer (II) Timing Chart



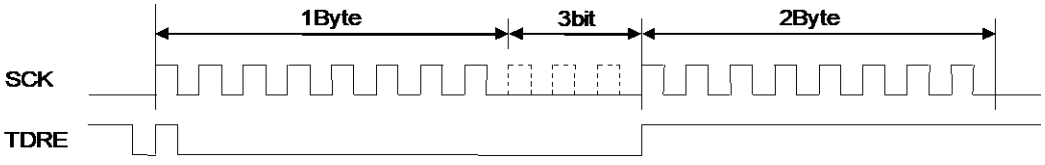
■ ESCR:WT1=0, ESCR:WT0=1 (for master)



■ ESCR:WT1=1, ESCR:WT0=0 (for master)



■ ESCR:WT1=1, ESCR:WT0=1 (for master)



Operation

Master operation (Set SCR:MS=0, SMR:SCKE=1.)

- Transmission operation
  - 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 transmission data in synchronization with a rising edge of the serial clock (SCK) output.
  - 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 is outputted. At this time, the transmission data in the second byte can be written.
- Reception operation
  - 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).
  - Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is outputted. At this time, the receive data (RDR) can be read.
  - 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
  - When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - If the transmission data is written in TDR, the SSR:TDRE is set as 0 also synchronization with the rising edge of serial clock (SCK) output and the transmission data is output. If the transmission data of the first bit outputted, the SSR:TDRE bit is set as 1 and when transmission interrupt enabled (SCR:TIE=1) is done, the transmission interrupt enabled is output. At this time, the transmission data of the second byte can be written.
  - The reception data is sampled by the falling edge of serial clock (SCK) output. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, the receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".
- Successive data transmission or reception wait operation

  If setting other than (ESCR:WT1, WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.

Slave operation (Set SCR:MS=1, SMR:SCKE=0.)

- Transmission operation
  - With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0 and outputs the transmission data in synchronization with a rising edge of the serial clock (SCK) input.
  - 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 is outputted. At this time, the transmission data in the second byte can be written.
- Reception operation
  - 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).
  - Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is outputted. At this time, the receive data (RDR) can be read.
  - Reading the receive data (RDR) clears SSR:RDRF to "0".
- Transmission/ Reception operation
  - When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - If the transmission data is written in TDR, the SSR:TDRE is set as 0 also synchronization with the rising edge of serial clock (SCK) input and the transmission data is output. If the transmission data of the first bit outputted, the SSR:TDRE bit is set as 1 and when transmission interrupt enabled (SCR:TIE=1) is done, the transmission interrupt enabled is output. At this time, the transmission data of the second byte can be written.
  - The reception data is sampled by the falling edge of serial clock (SCK) input. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, the receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".

40.6.2.3 SPI Transfer (I)

SPI Transfer (I) is shown below.

Features

	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).

Table 40-10. SPI Transfer (I) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	[1]	[1]	[1]	[1]	[1]	0	1	0	-	0	[1]	1/0	1/0
SSR/ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	[1]	[1]	[1]	[1]	[1]
TDR/RDR								D8	D7	D6	D5	D4	D3	D2	D1	D0
								[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
BGR	-	BGR[14:8]							BGR[7:0]							
	-	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]

1: Set to "1"

0: Set to "0"

[1]: User-configurable setting

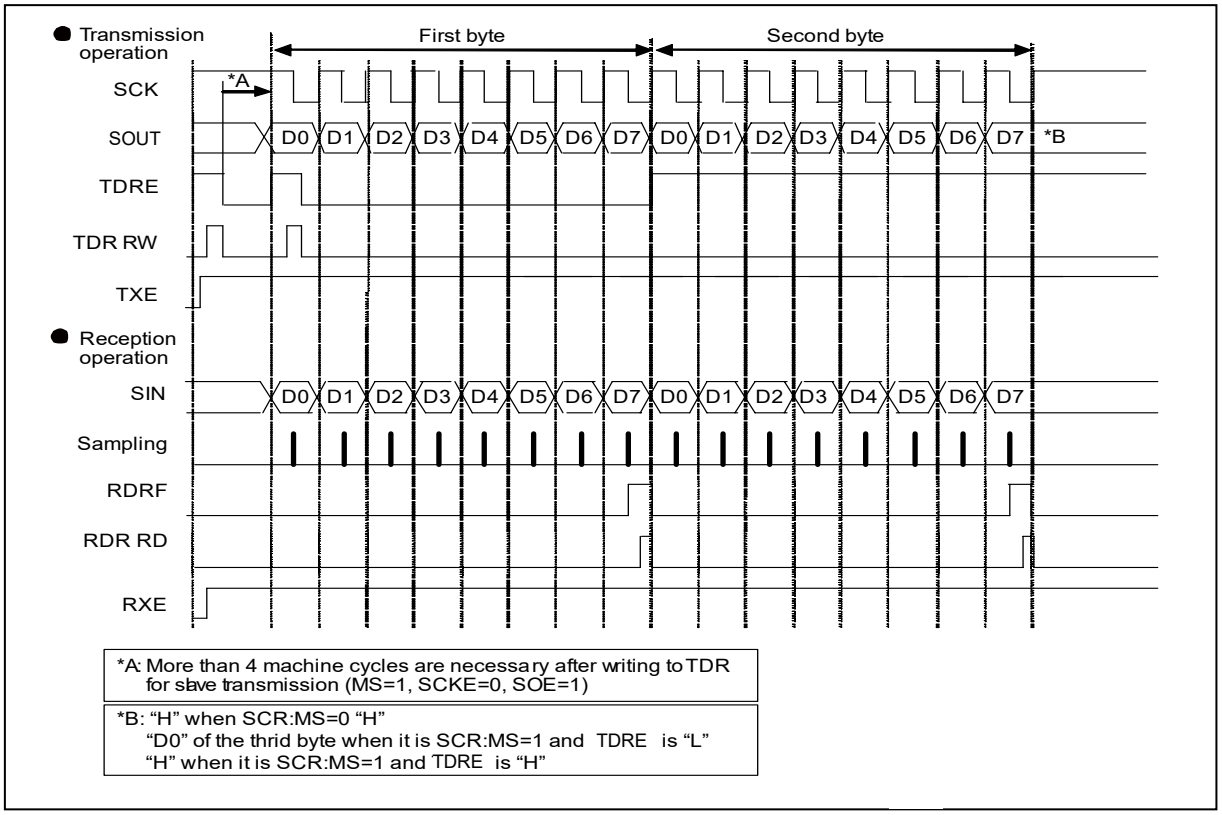
Note:

The above bit settings (1 or 0) are different between the master and slave operations. Set the bits as follows:

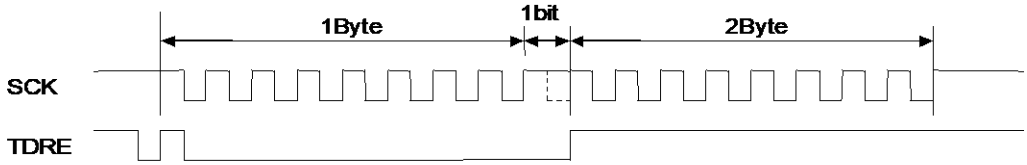
Master transmission : SCR:MS=0, SMR:SCKE=1, SOE=1
Master reception : SCR:MS=0, SMR:SCKE=1, SOE=0
Slave transmission : SCR:MS=1, SMR:SCKE=0, SOE=1
Slave reception : SCR:MS=1, SMR:SCKE=0, SOE=0

SPI Transfer (I) Timing Chart

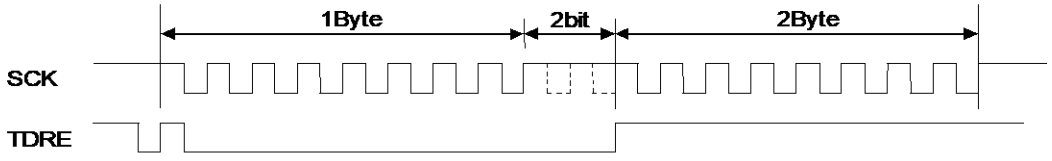
Figure 40-26. SPI Transfer (I) Timing Chart



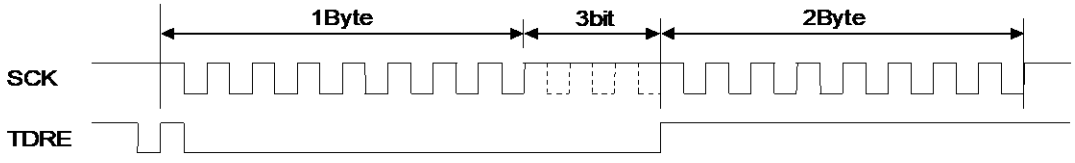
■ ESCR: WT1=0, ESCR: WT0=1 (for master)



■ ESCR: WT1=1, ESCR: WT0=0 (for master)



■ ESCR: WT1=1, ESCR: WT0=1 (for master)



Operation

Master operation (Set SCR:MS=0, SMR:SCKE=1)

- Transmission operation
  - 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.
  - Half a cycle before a falling 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.
- Reception operation
  - 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).
  - Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is outputted. At this time, the receive data (RDR) can be read.
  - 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
  - When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - If the transmission data is written in TDR, the SSR:TDRE is set as 0 also the first bit is outputted. After synchronization with the rising edge of serial clock (SCK) output and the transmission data is output. The SSR:TDRE bit is set as 1 at before half cycle of the falling edge of first serial clock and transmission interrupt enabled is outputted when transmission interrupt enabled(SCR:TIE=1) is done. At this time, the transmission data of the second byte can be written.
  - The reception data is sampled at the falling edge of serial clock (SCK) output. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, the receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".
- Successive data transmission or reception wait operation

  If setting other than (ESCR:WT1, WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.

Slave operation (Set SCR:MS=1, SMR:SCKE=0)

- Transmission operation
  - With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), 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.
  - It becomes SSR:TDRE=1 if the first bit of the transmission data is output, and when transmission interrupt enabled (SCR:TIE=1) is done, the transmission interrupt request is output. At this time, the transmission data of the second byte can be written.
- Reception operation
  - 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).
  - Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is outputted.

  At this time, the receive data (RDR) can be read.
  - Reading the receive data (RDR) clears SSR:RDRF to "0".
- Transmission/Reception operation
  - When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - If the transmission data is written in TDR, the SSR:TDRE is set as 0 also the first bit is outputted. After synchronization with the rising edge of serial clock (SCK) input and the transmission data is output. The SSR:TDRE bit is set as 1 when the first byte of the transmission data is outputted and a transmission interrupt request is outputted when transmission interrupt enabled(SCR:TIE=1) is done. At this time, the transmission data of the second byte can be written.
  - The reception data is sampled at the falling edge of serial clock (SCK) input. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, the receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".
- Continuous change from reception operation to transmission operation
  - Serial data output is disabled(SMR:SOE=0),reception interrupt is enabled(SCR:RIE=1), reception operation is enabled(SCR:RXE=1) and transmission operation is enabled(SCR:TXE=1). When serial clock (SCK) writes the dummy data in TDR at the mark level, the reception data is sampled by the falling edge of serial clock input (SCK).
  - Write the dummy data in TDR by rising edge of the following serial clock (SCK)after the reception interrupt request, when you continue the reception operation.
  - To switch from the reception operation to the transmission operation, serial data output set enabled(SMR:SOE=1), reception interrupt set disabled(SCR:RIE=0), and reception operation set disabled(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, the transmission data is outputted in synchronization with the rising edge of the serial clock.

40.6.2.4 SPI Transfer (II)

SPI Transfer (II) is shown below.

Features

	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 9 bits

Register Settings

The following table lists the register settings required for SPI Transfer (II).

Table 40-11. SPI Transfer (II) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	*	*	*	*	*	0	1	0	-	1	*	1/0	1/0
SSR/ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
TDR/RDR								D8	D7	D6	D5	D4	D3	D2	D1	D0
								*	*	*	*	*	*	*	*	*
BGR	-	BGR[14:8]							BGR[7:0]							
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

[1]: User-configurable setting

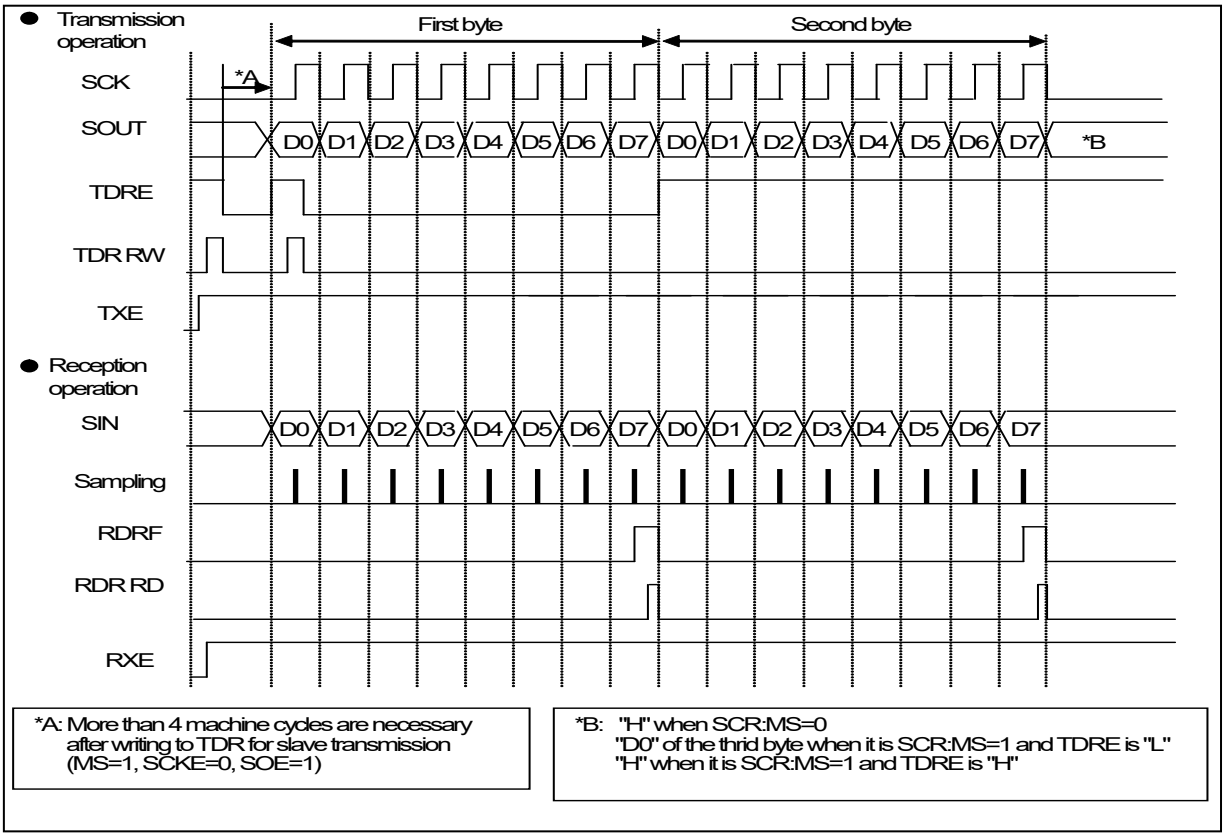
Note:

The above bit settings (1 or 0) are different between the master and slave operations. Set the bits as follows:

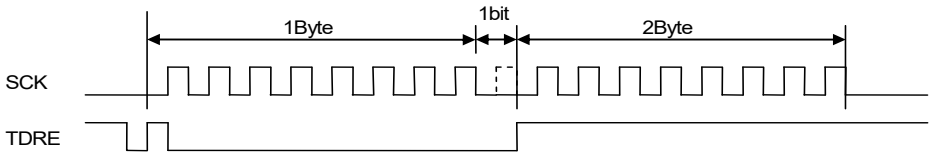
Master transmission : SCR:MS=0, SMR:SCKE=1, SOE=1
Master reception : SCR:MS=0, SMR:SCKE=1, SOE=0
Slave transmission : SCR:MS=1, SMR:SCKE=0, SOE=1
Slave reception : SCR:MS=1, SMR:SCKE=0, SOE=0

SPI Transfer (II) Timing Chart

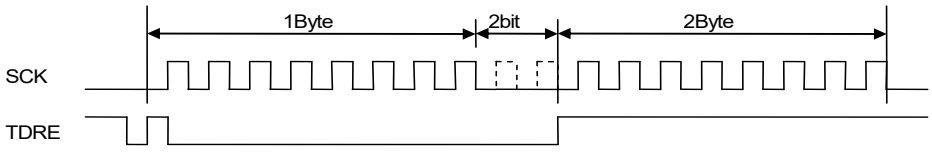
Figure 40-27. SPI Transfer (II) Timing Chart



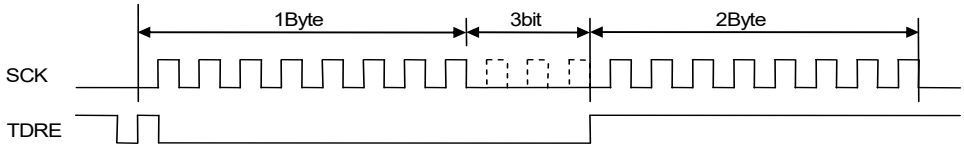
■ ESCR:WT1=0, ESCR:WT0=1 (for master)



■ ESCR:WT1=1, ESCR:WT0=0 (for master)



■ ESCR:WT1=1, ESCR:WT0=1 (for master)



Operation

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 and outputs the transmission data in synchronization with a falling edge of the serial clock (SCK) output.
  - (2) The SSR:TDRE bit is set as 1 at before half cycle of the rising edge of first serial clock (SCK) and transmission interrupt enabled is outputted when transmission interrupt enabled(SCR:TIE=1) is done. 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 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is outputted. 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) When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - (2) If the transmission data is written in TDR, the SSR:TDRE is set as 0 also the first bit is outputted. After synchronization with the falling edge of serial clock (SCK) output and the transmission data is output. The SSR:TDRE bit is set as 1 at before half cycle of the rising edge of first serial clock and transmission interrupt enabled is outputted when transmission interrupt enabled(SCR:TIE=1) is done.. At this time, the transmission data of the second byte can be written.
  - (3)The reception data is sampled at the rising edge of serial clock (SCK) output. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, the receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".
- Successive data transmission or reception wait operation

  If setting other than (ESCR:WT1, WT0)= (0,0) is specified for successive data transmission or reception, a wait is inserted between frames.

Slave operation (Set SCR:MS=1, SMR:SCKE=0)

- Transmission operation
  - With serial data output enabled (SMR:SOE=1) and transmission operation enabled (SCR:TXE=1), writing transmission data to TDR sets SSR:TDRE=0 and outputs the transmission data in synchronization with a falling edge of the serial clock (SCK) output.
  - 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 is outputted. At this time, the transmission data in 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 data is not output and the transmission is not operated normally. After the transmission operation is enabled (SCR:TXE=1), write the transmission data to the first TDR when serial clock (SCK) is at the mark level.
- Reception operation
  - 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).
  - Receiving the last bit sets SSR:RDRF=1 and, if the reception interrupt is enabled (SCR:RIE=1), a reception interrupt request is outputted. At this time, the receive data (RDR) can be read.
  - Reading the receive data (RDR) clears SSR:RDRF to "0".
- Transmission/ Reception operation
  - When the transmission and reception operation is done at the same time, set serial data output enabled (SMR:SOE=1) and set reception operation enabled (SCR:TXE, RXE=1).
  - If the transmission data is written in TDR, the SSR:TDRE is set as 0 also the first bit is outputted. After synchronization with the falling edge of serial clock (SCK) input and the transmission data is output. The SSR:TDRE bit is set as 1 when the first byte of the transmission data is outputted and a transmission interrupt request is outputted when transmission interrupt enabled(SCR:TIE=1) is done. At this time, the transmission data of the second byte can be written.
  - The reception data is sampled at the rising edge of serial clock (SCK) input. It changes to SSR:RDRF=1 if the last bit of the reception data is received, and when reception interrupt enabled (SCR:RIE=1) is done, the reception interrupt request is output. At this time, the receive data (RDR) can be read. When the reception data is read, SSR:RDRF is cleared to "0".
- Continuous change from reception operation to transmission operation
  - Serial data output is disabled(SMR:SOE=0),reception interrupt is enabled(SCR:RIE=1), reception operation is enabled(SCR:RXE=1) and transmission operation is enabled(SCR:TXE=1). When serial clock (SCK) writes the dummy data in TDR at the mark level, receive data is sampled by the falling edge of serial clock input (SCK).
  - Write the dummy data in TDR by rising edge of the following serial clock (SCK) after the reception interrupt request, when you continue the reception operation.
  - To switch from the reception operation to the transmission operation, set serial data output enabled(SMR:SOE=1), set reception interrupt disabled(SCR:RIE=0), and set reception operation disabled(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, The transmission data is outputted in synchronization with the rising edge of the serial clock.

40.6.2.5 Baud Rate Generation

The baud rate generation is shown below.

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.

- 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.

- 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:

- Reload value

  $V = \varphi / b - 1$

  V: Reload value φ: peripheral clock (PCLK) frequency b: Baud rate

- Example of calculation

  If the peripheral clock (PCLK) of 16MHz, use of the internal clock, baud rate of 19200bps are to be set,

  Reload value:

  $V = (16 \times 1{,}000{,}000)/19200-1=832$

  Therefore, the baud rate is

  $b = (16 \times 1{,}000{,}000)/(832+1)= 19208$ bps

- Baud rate error

  The baud rate error can be obtained using the following formula:

  Error (%) = (calculated value - desired value) / desired value × 100

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 SCINV=0, the "H" width of the serial clock is longer by one cycle of the peripheral clock (PCLK).
  - If SCINV=1, the "L" width of the serial clock is longer by one cycle of the peripheral clock (PCLK).
- Set the reload value to 3 or higher.

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)

40.6.3 Setup Procedure and Program Flow

The setup procedure and program flow are shown.

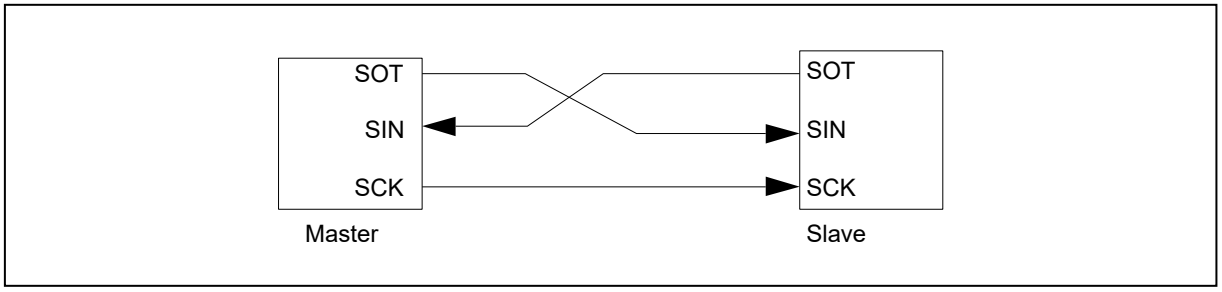
40.6.3.1 Connections between Chips

40.6.3.2 Flowchart

40.6.3.1 Connections between Chips

Connections between chips are shown below.

Figure 40-28. Example of Connection between CSIO Chips



40.6.3.2 Flowchart

The flowchart is shown below.

Figure 40-29. Flowchart Example (FIFO Not Used)

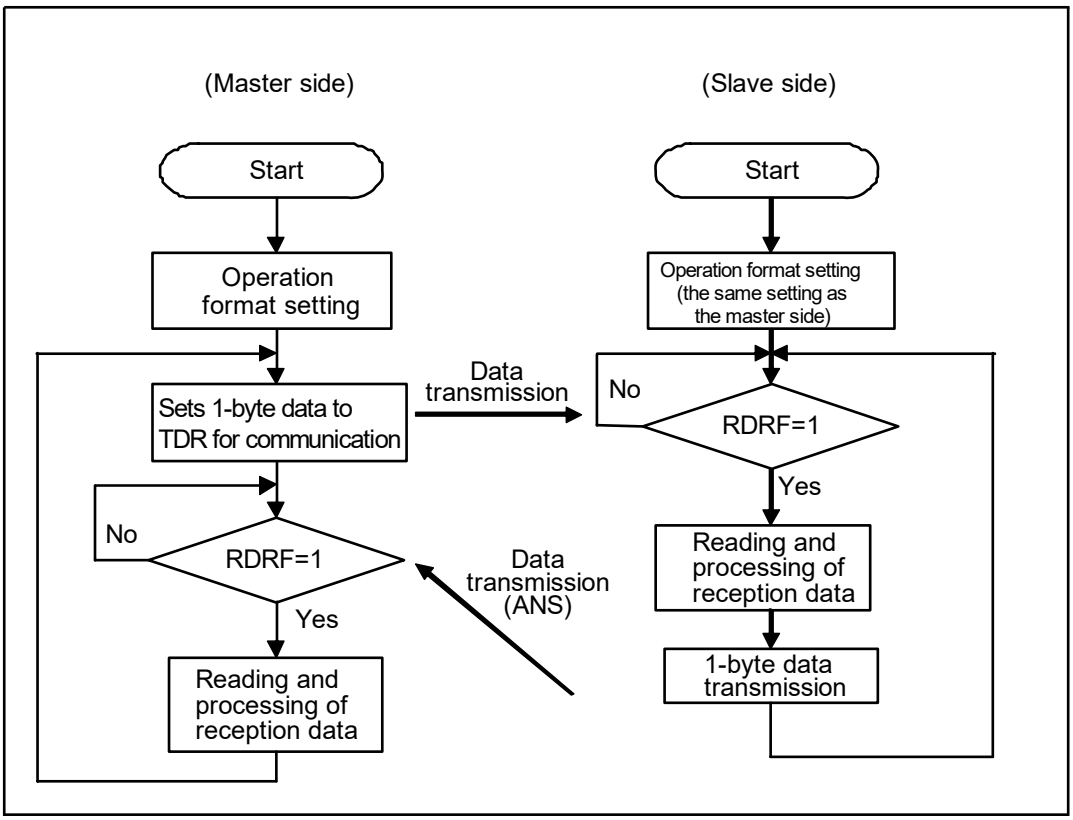
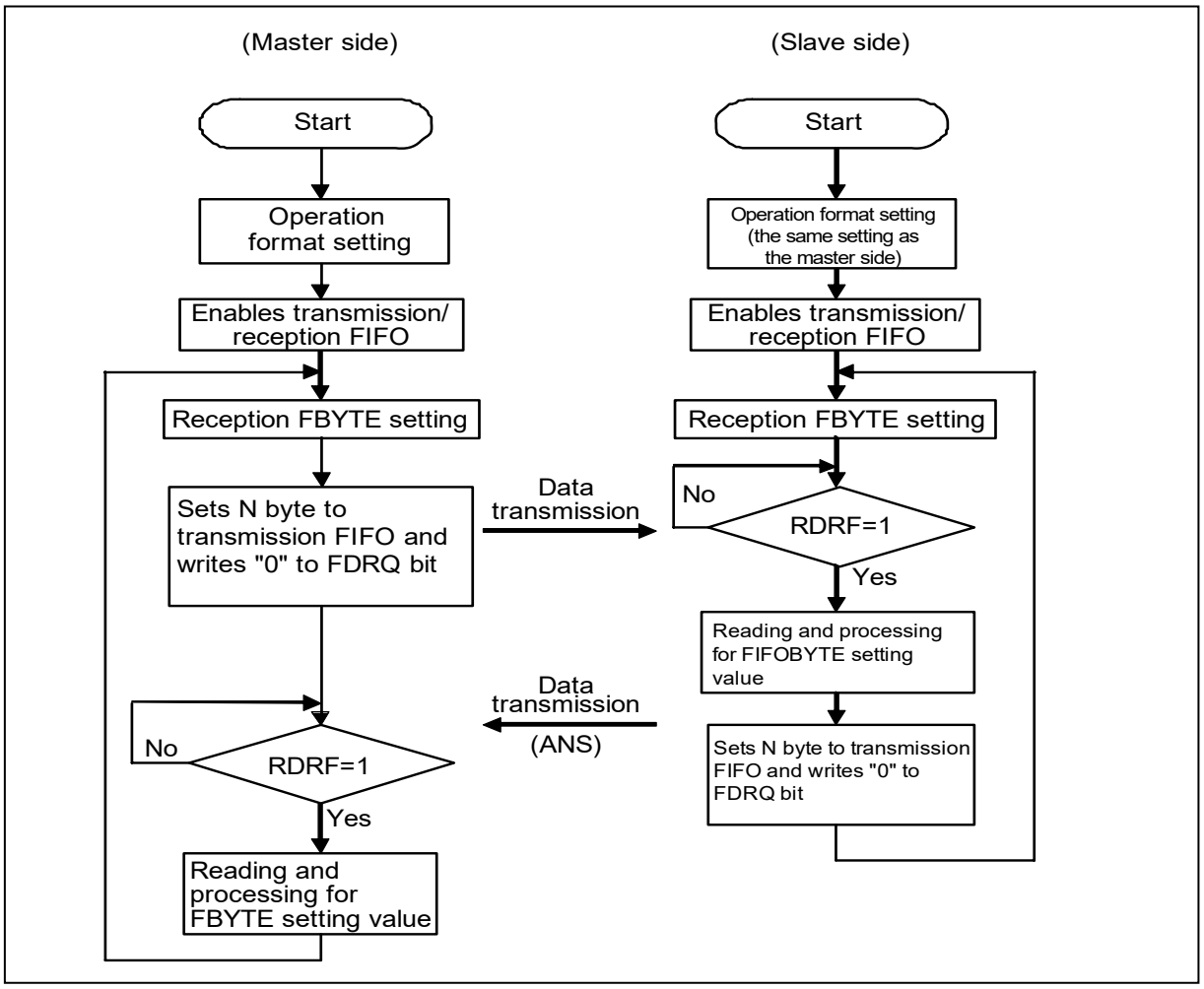


Figure 40-30. Flowchart Example (FIFO Used)



40.7 Operation of LIN-UART

The operation of LIN-UART is shown.

40.7.1 Interrupts of LIN-UART
40.7.2 Operation of LIN-UART
40.7.3 Setup Procedure and Program Flow

40.7.1 Interrupts of LIN-UART

Interrupts of LIN-UART are shown below.

The LIN-UART 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 Synch Break detection

40.7.1.1 List of Interrupts of LIN-UART Interface

The list of interrupts of LIN-UART interface is shown below.

The following table indicates how LIN-UART interrupt control bits relate to interrupt factors.

Table 40-12. Interrupt Control Bits and Interrupt Sources of LIN-UART

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request
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		
Transmission	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)[1]
	TBI	SSR	No transmission operation	SCR:TBIE	Write to the transmit data (TDR), write "1" to the LIN Synch Break 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)[1]
	FDRQ	FCR1	Transmission FIFO is empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO is full

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request
Status	LBD	SSR	Lin Synch Break detection	ESCR:LBIE	Writing "0" to the SSR:LBD
Input capture[2]	ICP	ICS	1st falling edge of Lin Synch Field	ICS:ICE	Disabling of ICP
	ICP	ICS	5th falling edge of Lin Synch Field		

[1]: Set the TIE bit after the TDRE bit is set to "0".

[2]: For registers, see "Chapter: Input Capture".

Notes:

- DMA transfer triggered by a status interrupt is not supported.
- To detect a LIN synch break, disable reception (SCR:RXE=0) after enabling LIN synch break detection interrupt(LBIE=1).

40.7.1.2 Reception Interrupts and Flag Setting Timing

Reception interrupts and flag setting timing are shown below.

Reception interrupts occur when the reception is completed (SSR:RDRF), when a reception error occurs (SSR:ORE, FRE), or when LIN Synch Break is detected.

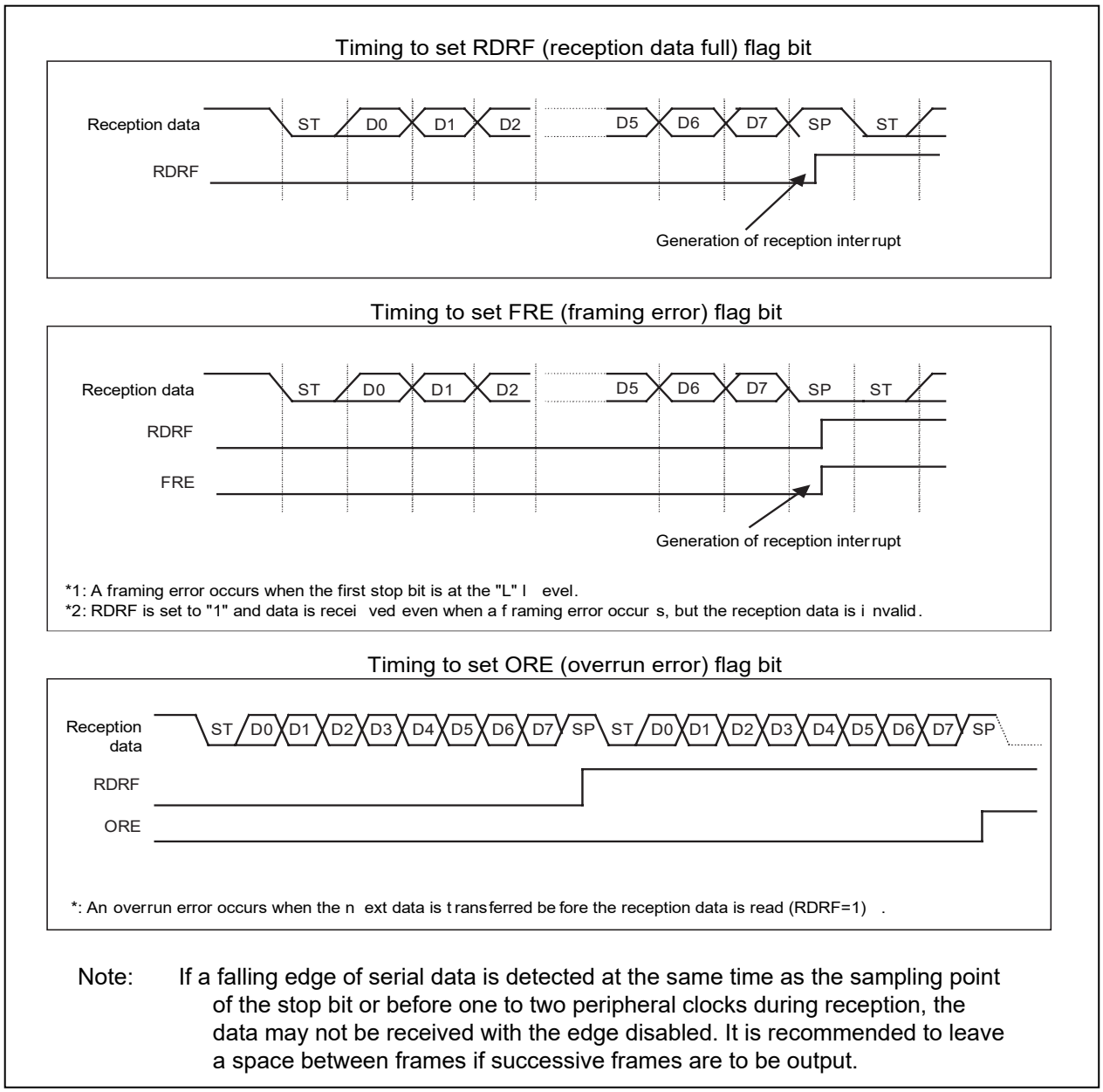
Reception Interrupts and Flag Setting Timing

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 at this time (SCR:RIE=1), a reception interrupt will occur.

Note:

When a reception error occurs, the data in the receive data register (RDR) becomes invalid.

Figure 40-31. Timing of Flag Bit Setting

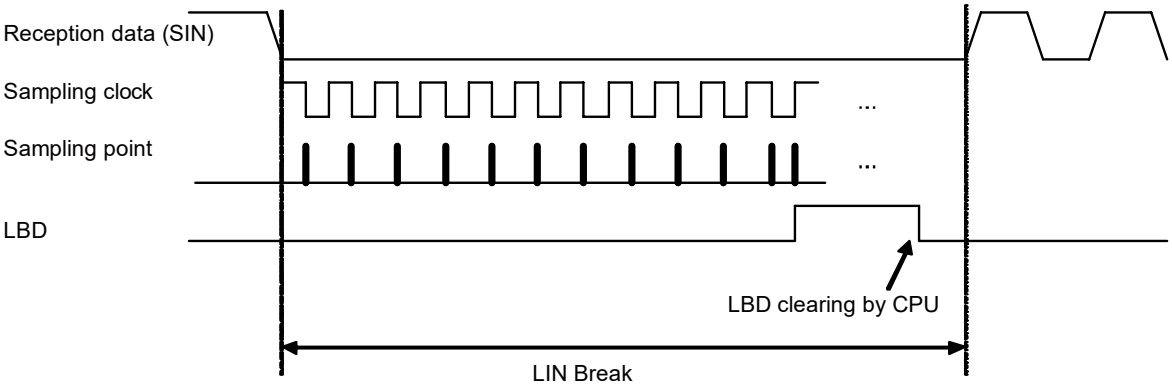


Note: If a falling edge of serial data is detected at the same time as the sampling point of the stop bit or before one to two peripheral clocks during reception, the data may not be received with the edge disabled. It is recommended to leave a space between frames if successive frames are to be output.

Timing of LIN Synch Break Detection Flag (LBD) Setting

In slave operation (SCR:MS=1), the LDB bit is set to "1" when the serial input (SIN) is "0" for more than 11-bit width. If the LIN Synch Break interrupt is enabled (ESCR:LBIE=1) at this time, a reception interrupt occurs.

Figure 40-32. Timing of LBD (LIN Synch Break Detection) Flag Setting



After 11-bit "L" of reception data is detected with the falling edge of the sampling clock, LIN break is detected with the rising edge of the sampling clock. LBD is set to "1" when LIN break is detected.

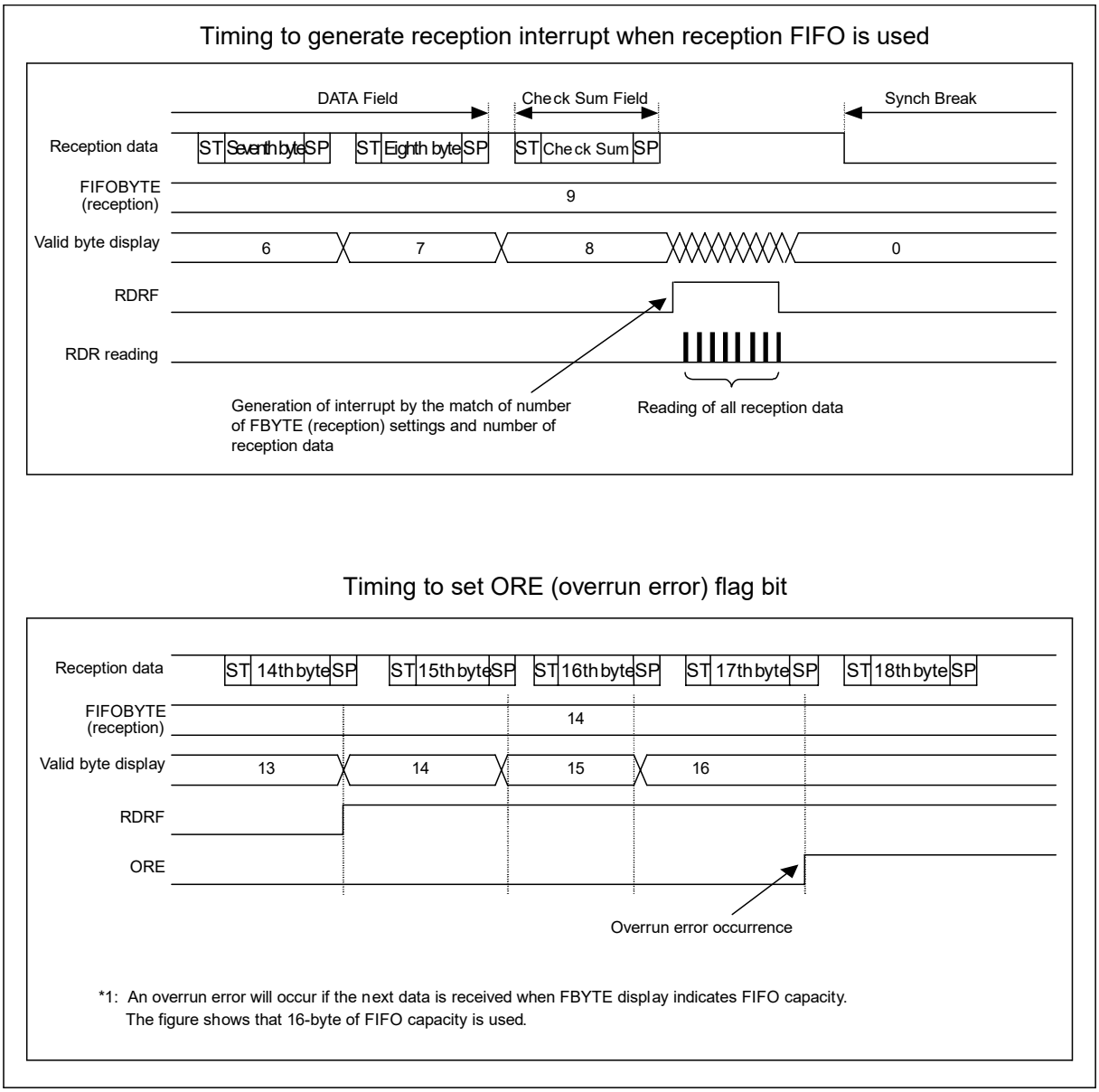
40.7.1.3 Interrupts When Using Reception FIFO and Flag Setting Timing

Interrupts when using reception FIFO and flag setting timing are shown below.

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.
- If the data count contained in the reception FIFO does not reach the transfer count while reception FIFO idle detection enable bit (FRIIE) is set to "1", the interrupt flag (RDRF) will be set to "1" after the reception idle state continues for 8 baud rate clocks or longer. 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. If the reception FIFO is disabled, the counter is reset to 0. When the reception FIFO is enabled while there is data remaining in it, the counter will start counting 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 40-33. Timing of Interrupt Generation



40.7.1.4 Transmission Interrupts and Flag Setting Timing

Transmission interrupts and flag setting timing are shown below.

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).

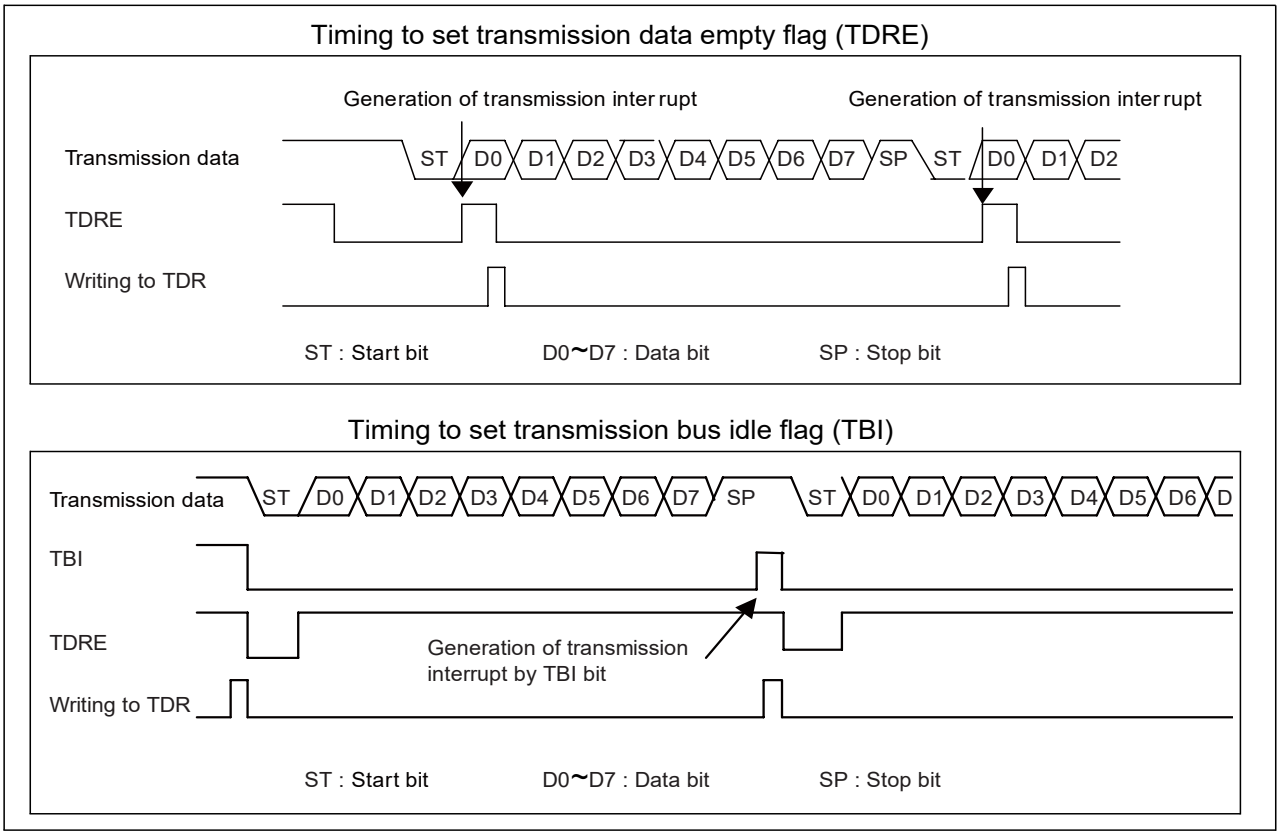
- Timing of transmission data empty flag (SSR:TDRE) setting

  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 TBI bit and the transmission interrupt request will be cleared.

Figure 40-34. Timing of TDRE and TBI Setting



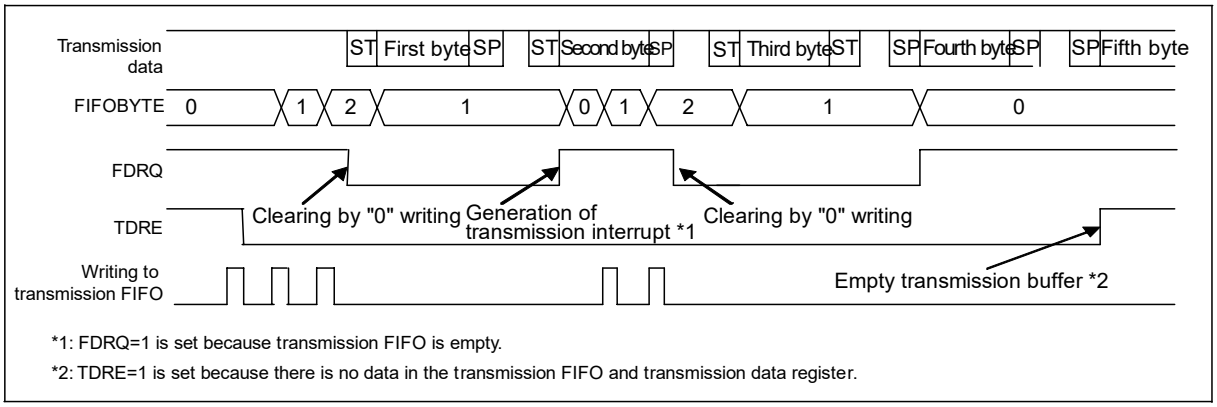
40.7.1.5 Interrupts When Using Transmission FIFO and Flag Setting Timing

Interrupts when using transmission FIFO and flag setting timing are shown below.

When the transmission FIFO is used, an interrupt will occur if there is no data in the transmission FIFO.

- When there is no data in the transmission FIFO, 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). When FBYTE=00_H, there is no data in the transmission FIFO.

Figure 40-35. Timing of Transmission Interrupts when Using Transmission FIFO



40.7.2 Operation of LIN-UART

The operation of LIN-UART is shown below.

The LIN-UART operates for the master/slave bidirectional LIN communication.

40.7.2.1 Master Device Operation

The Master device operation is shown below.

Device Selection

To make the LIN-UART work as the master device, set the SCR:MS bit to "0".

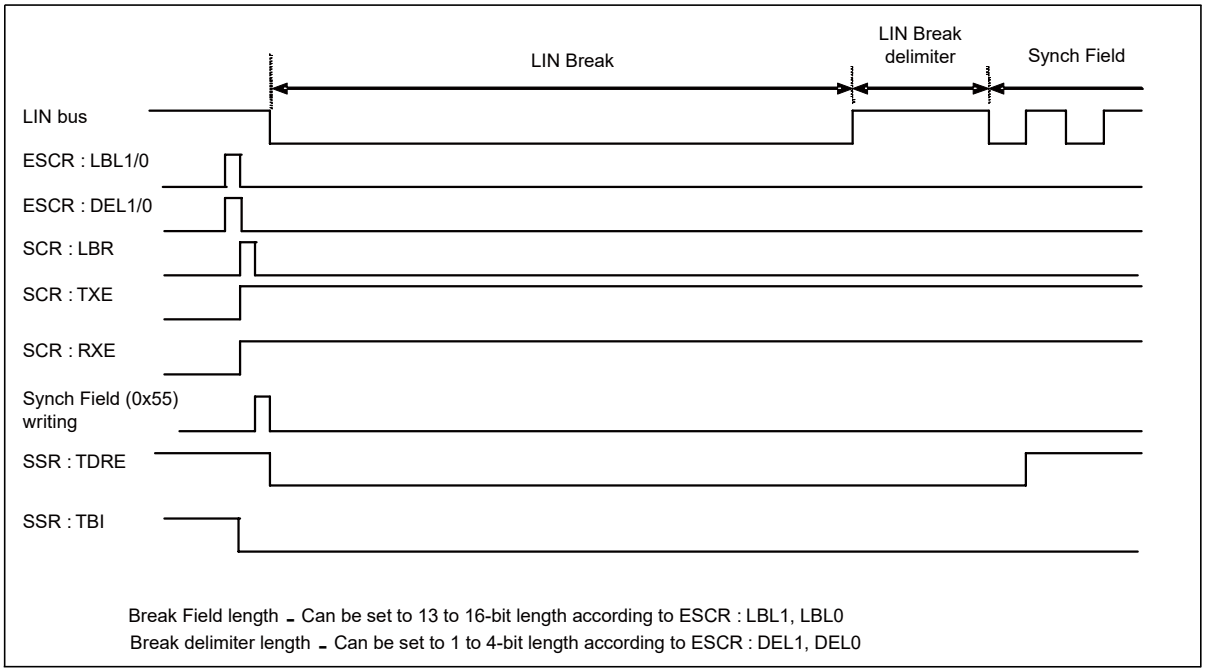
From Synch Break Transmission to Synch Field Transmission

- Selection of the Synch Break length (ESCR:LBL1, LBL0) and selection of the Synch Break delimiter length (ESCR:DEL1, DEL0) are performed.
- A Synch Break is transmitted by enabling transmission (SCR:TXE=1) and setting the SCR:LBR bit (LIN Synch Break 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 Synch Break setting bit) to "1".
- Even if the SCR:RXE bit (reception enable bit) is set to "1", the Synch Break part dose not perform the reception operation.

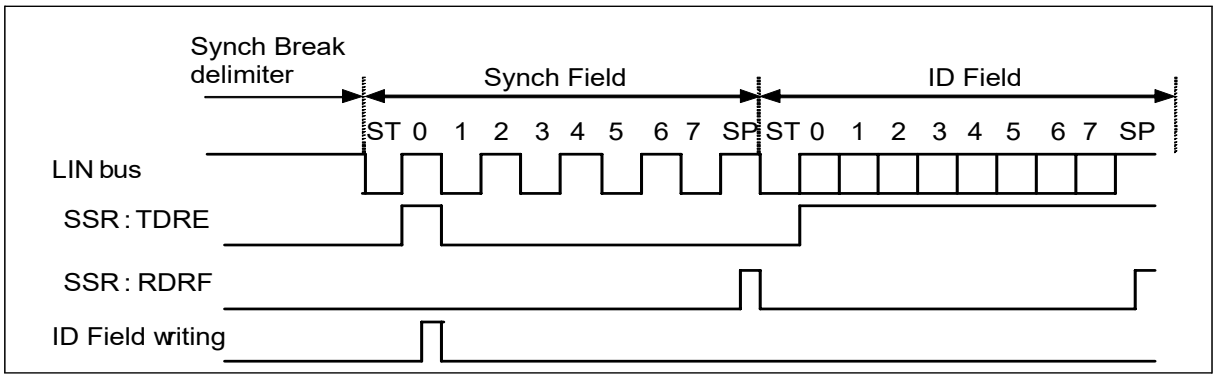
Figure 40-36. From Synch Break to Synch Field Transmission



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 received data will be compared with the transmitted 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 40-37. 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.

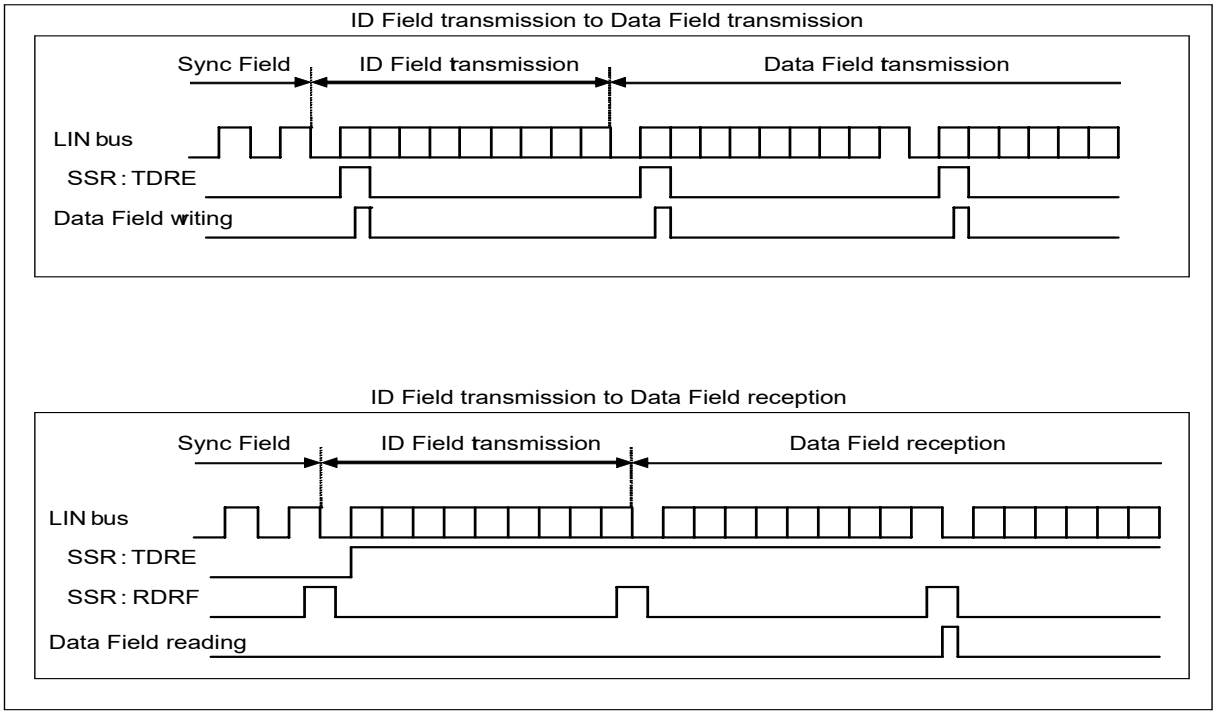
- In the case of Data Field reception:

  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 at this time (SCR:RIE=1), a reception interrupt will occur.

  Detection of the start bit is as follows; falling edge is detected after the noise filter(The serial data input is sampled three times with the machine clock and decision by majority) is passed, and the data after passage detects "L" with the sampling point.

Figure 40-38. From ID Field Transmission to Data Field Transmission/Reception



Notes:

- The board is designed so that the noise should not pass this filter or communicate by noise passing so as not to become a problem (For instance, when the error occurs adding the checksum of data at the end, send it again) though the noise filter (The serial data input is sampled three times with the machine clock and decision by majority) is built into.
- It becomes impossible to receive by making the edge invalidity etc. when the falling edge of the serial data is detected at the same time as the sampling point of the stop bit or before 1 to 2 machine clocks when it receives it. When the frame is continuously output, the interval of the frame is recommended to be opened.

Timing Chart When FIFO Is Not Used

Figure 40-39. LIN Bus Timing (at the Time of Data Field Transmission without Using FIFO)

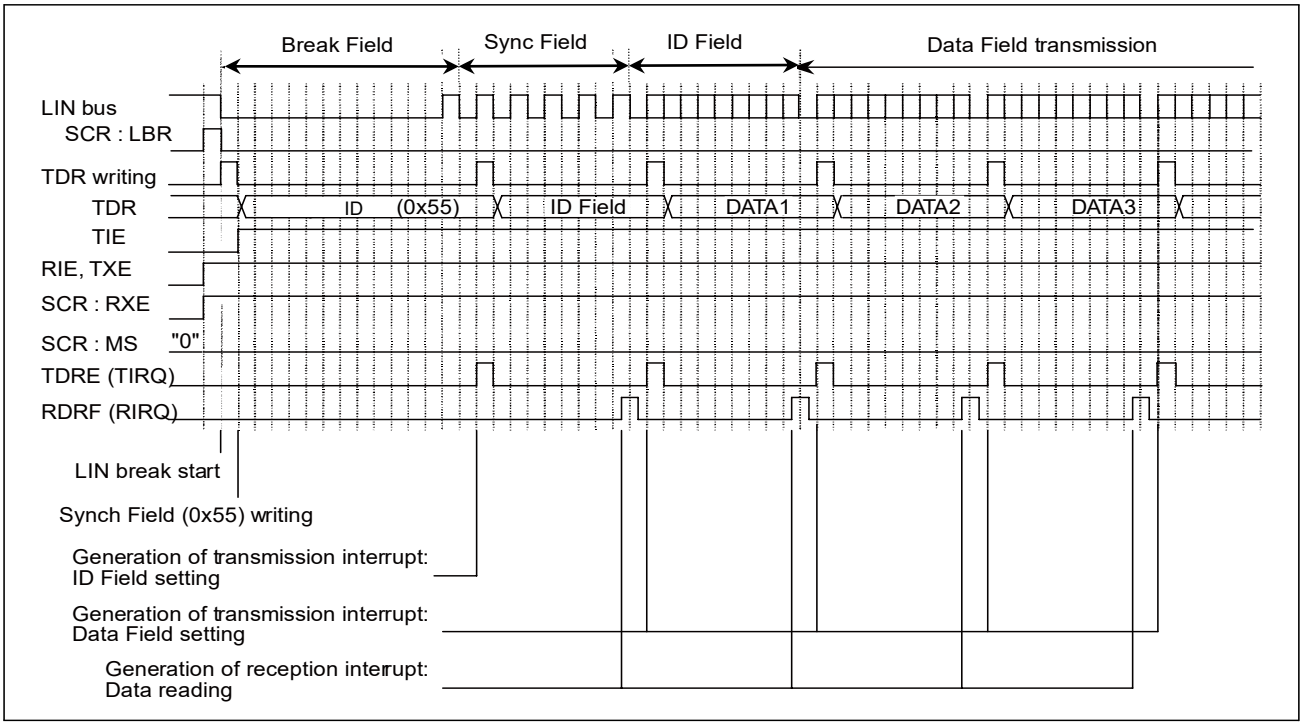
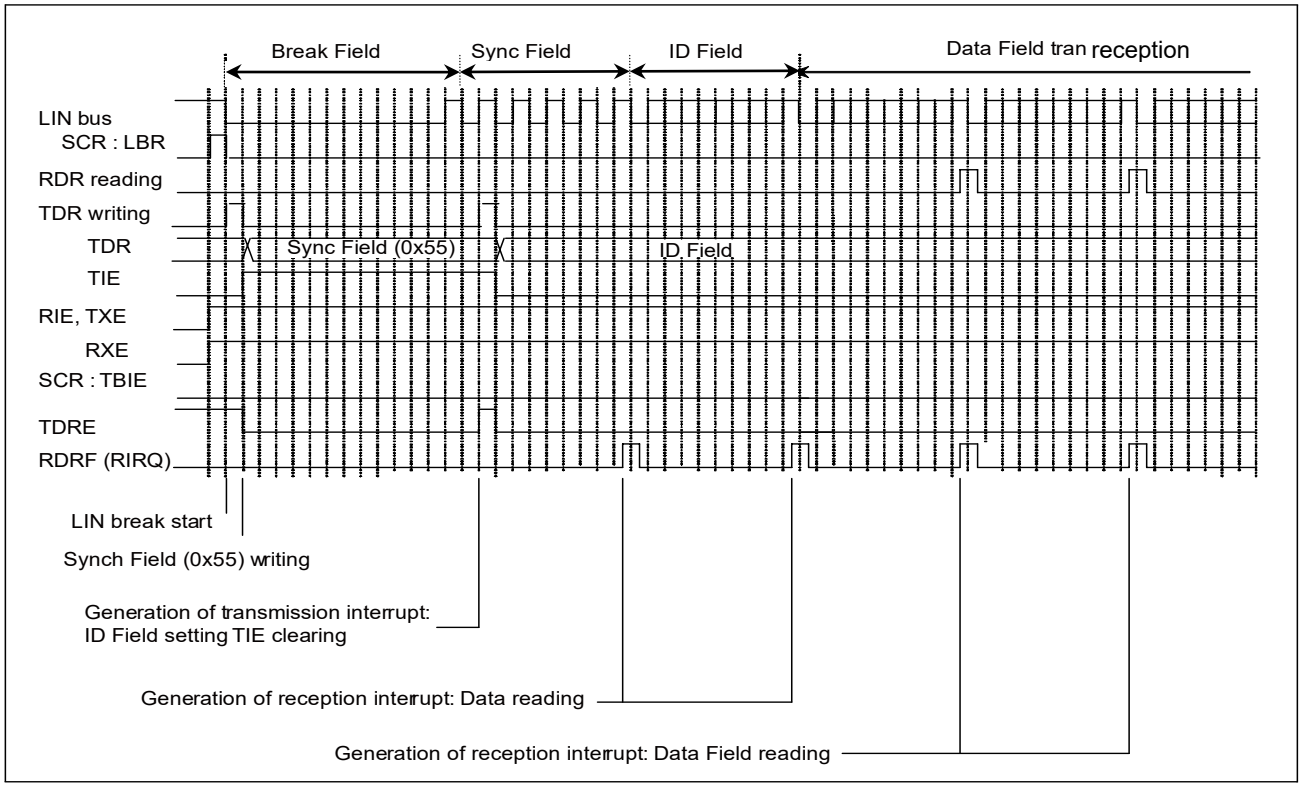


Figure 40-40. LIN Bus Timing (at the Time of Data Field Reception without Using FIFO)



Timing Chart when FIFO Is Used

Figure 40-41. LIN Bus Timing (at the Time of Data Field Transmission when Using FIFO)

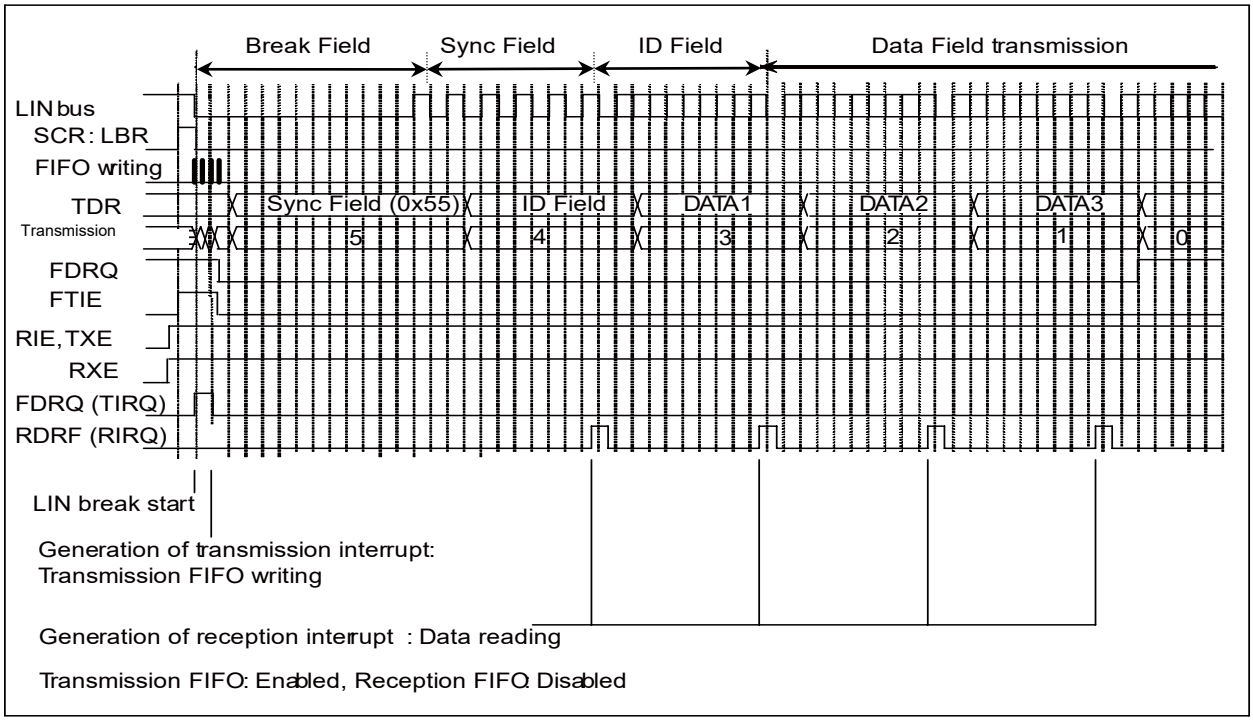
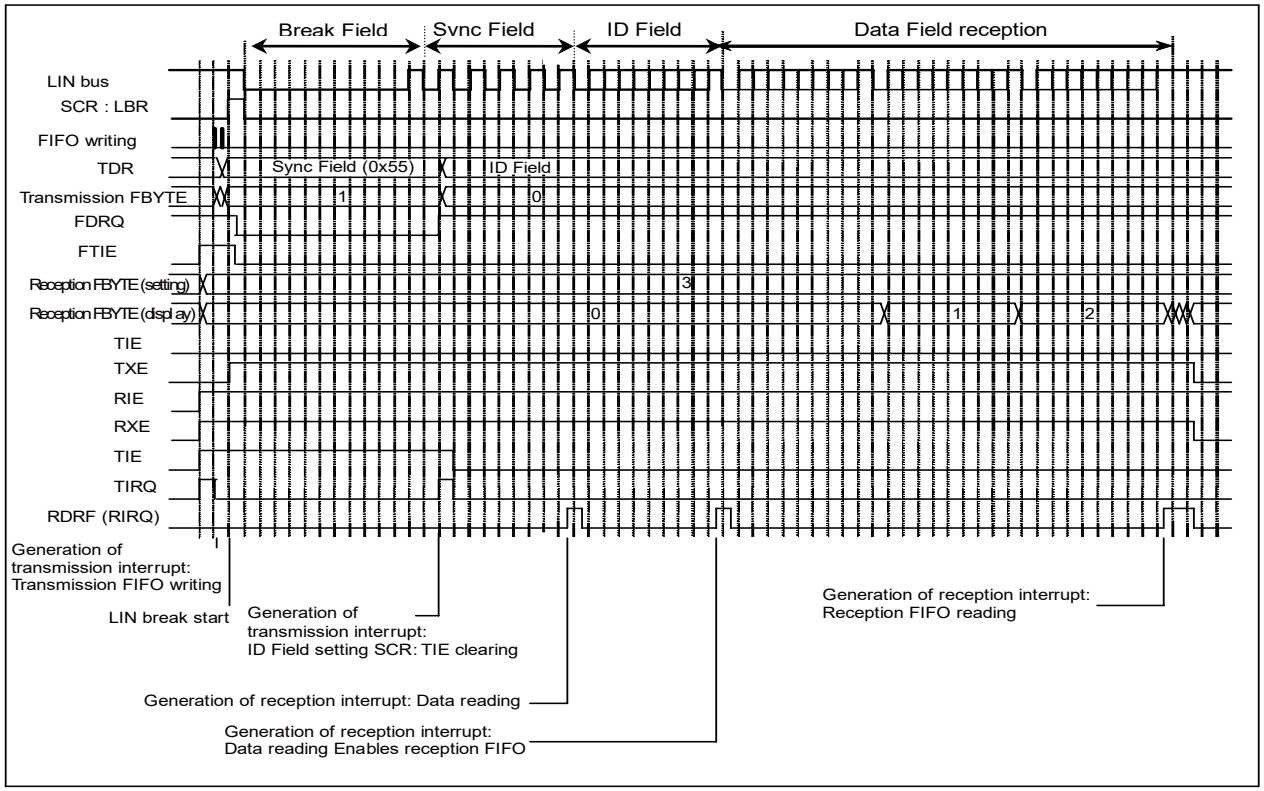


Figure 40-42. LIN Bus Timing (at the Time of Data Field Reception when Using FIFO)



40.7.2.2 Slave Device Operation

The slave device operation is shown below.

Device Selection

To make the LIN-UART work as the slave device, set the SCR:MS bit to "1".

From Synch Break Field Reception to Synch Field Reception

1. Once Synch Break input begins, the Synch Break will be detected (SSR:LBD=1) when the 11th bit is reached. If the ESCR:LBIE bit is set to "1" at this time, a reception interrupt will occur.
2. Enable ICU interrupts and set the detection mode to both edges.
3. When the LIN-UART detects the first falling edge of Synch Field, it sets the internal signal (LSYN) to be input to the ICU to "H" to start the ICU. This internal signal (LSYN) becomes "L" on the fifth falling edge.
4. The "H" duration of the internal signal (LSYN) input to the ICU becomes 8 times the baud rate. The baud rate setting is as follows:

   If the free-run timer has not overflowed:

   $$BGR\ value = (b - a) \times Fe / (8 \times \varphi) - 1$$

   If the free-run timer has overflowed:

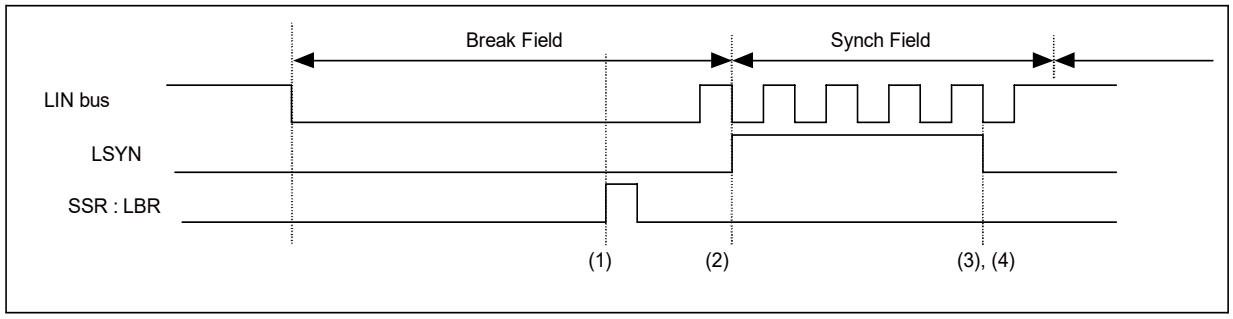
   $$BGR\ value = (max + 1 + b - a) \times Fe / (8 \times \varphi) - 1$$

   max: Free-run timer maximum value
   a: ICU data register value after the first interrupt
   b: ICU data register value after the second interrupt
   φ: Peripheral clock (PCLK) frequency (MHz)
   Fe: External clock frequency (MHz). It is assumed that the internal clock is used (EXT=0) and Fe=φ.

Note:

For Synch Break and Synch Field, disable reception (SCR:RXE=0).

Figure 40-43. From Synch Break Field Reception to Synch Field Reception



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:

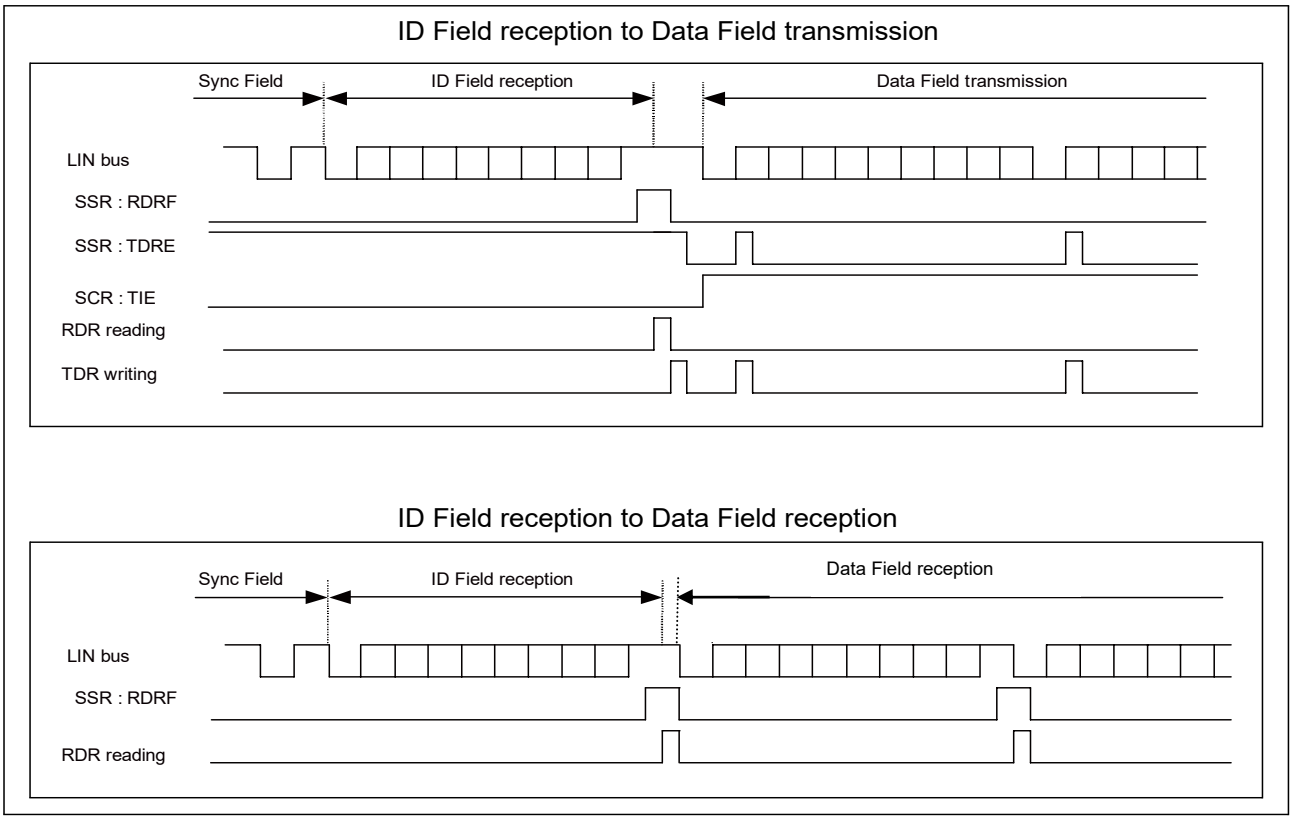
  After the ID Field is received, write data in the transmit data register (TDR). At this time, transmission interrupts must be enabled (SCR:TIE=1).

- 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.

  When falling edge is detected after the noise filter (Decision by majority sampling the serial data input with the machine clock three times) is passed, and the data after it passes of that detects "L" with the sampling point, the detection condition of the start bit becomes it.

Figure 40-44. From ID Field Reception to Data Field Transmission/Reception



Notes:

- The board is designed so that the noise should not pass this filter or communicate by noise passing so as not to become a problem (For instance, when the error occurs adding the checksum of data at the end, send it again) though the noise filter (The serial data input is sampled three times with the machine clock and decision by majority) is built into.
- It becomes impossible to receive by making the edge invalidity etc. when the falling edge of the serial data is detected in reception, at the same time of sampling point of stop bit or before 1 to 2 machine clock. When the frame is continuously output, it is recommended to be opened the interval of the frame.

Timing Chart when FIFO Is Not Used

Figure 40-45. LIN Bus Timing (at the Time of Data Field Transmission without Using FIFO)

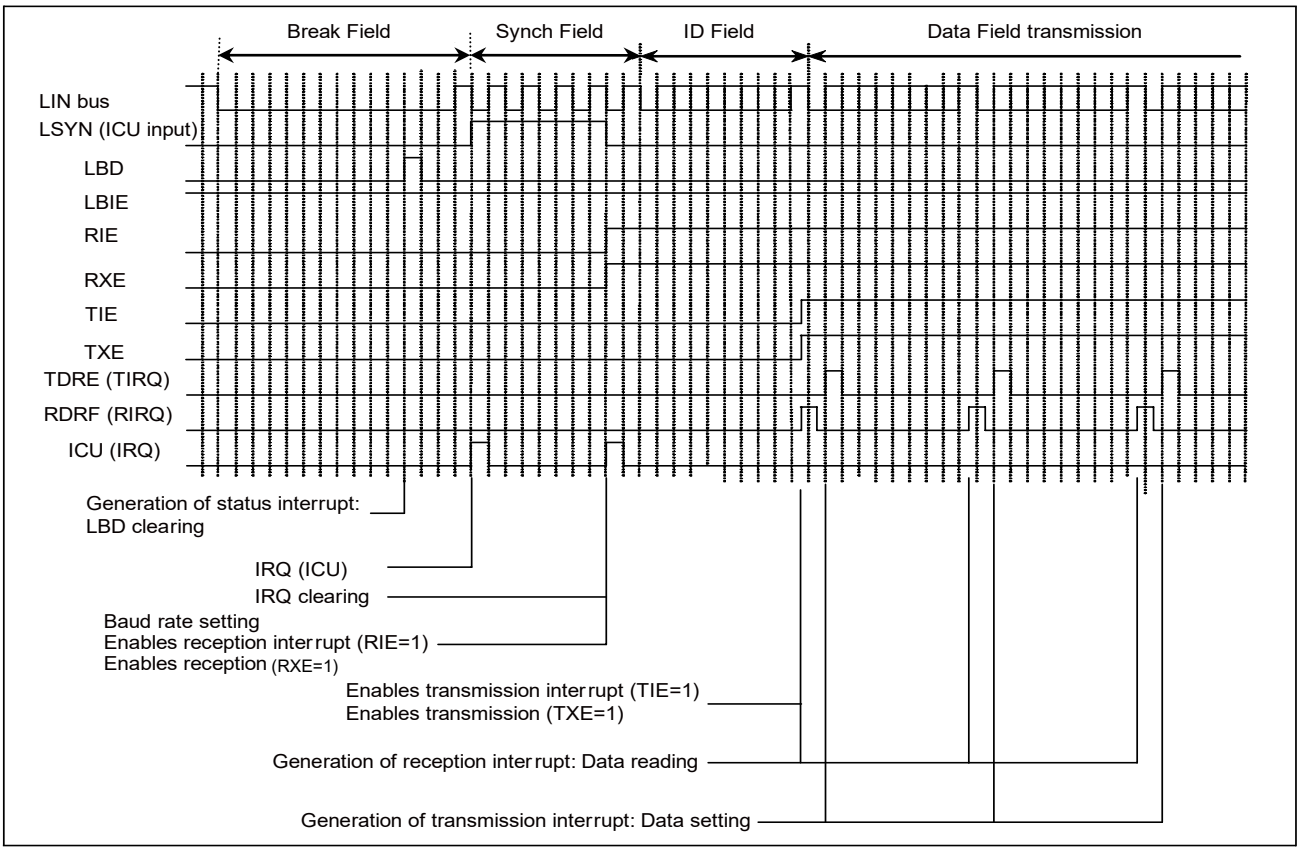
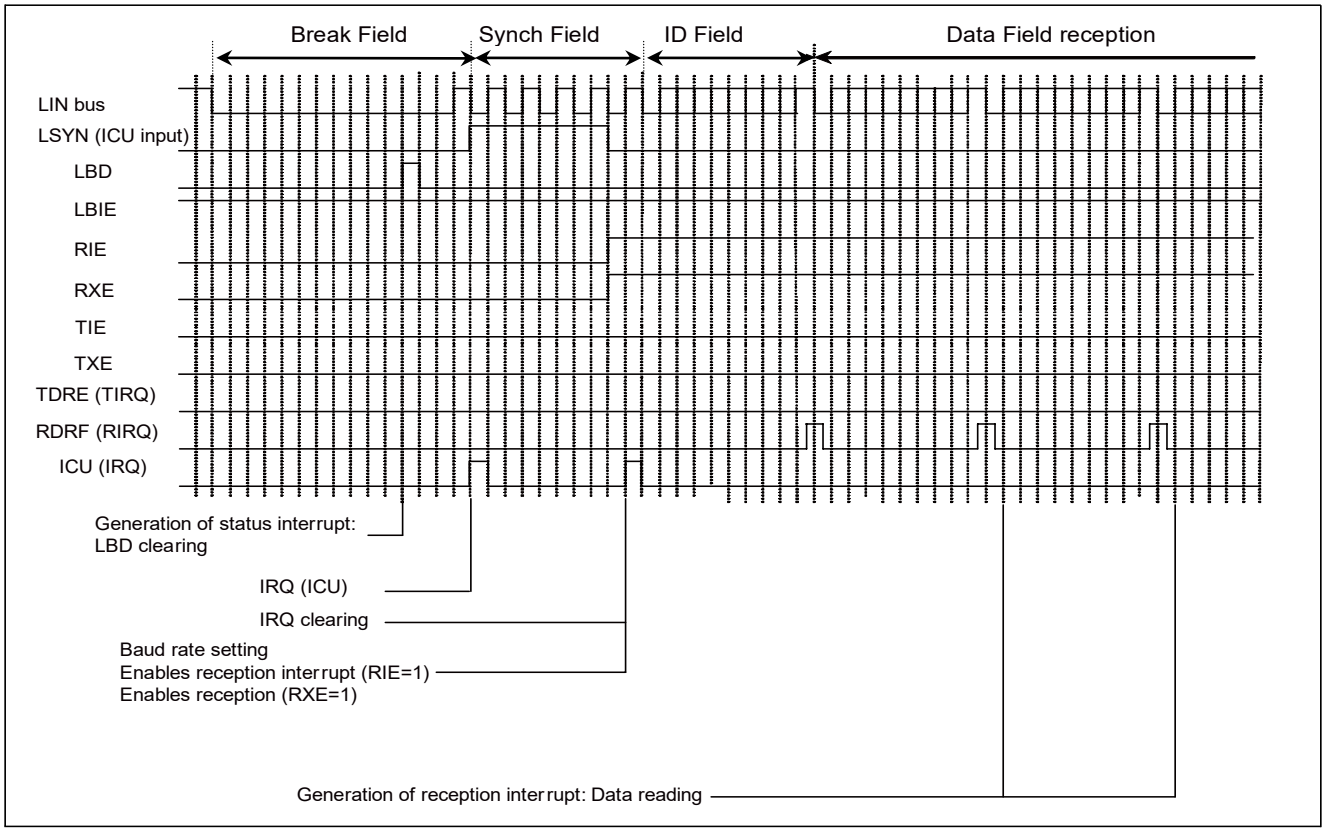


Figure 40-46. LIN Bus Timing (at the Time of Data Field Reception without Using FIFO)



Timing Chart When FIFO Is Used

Figure 40-47. LIN Bus Timing (at the Time of Data Field Transmission When Using FIFO)

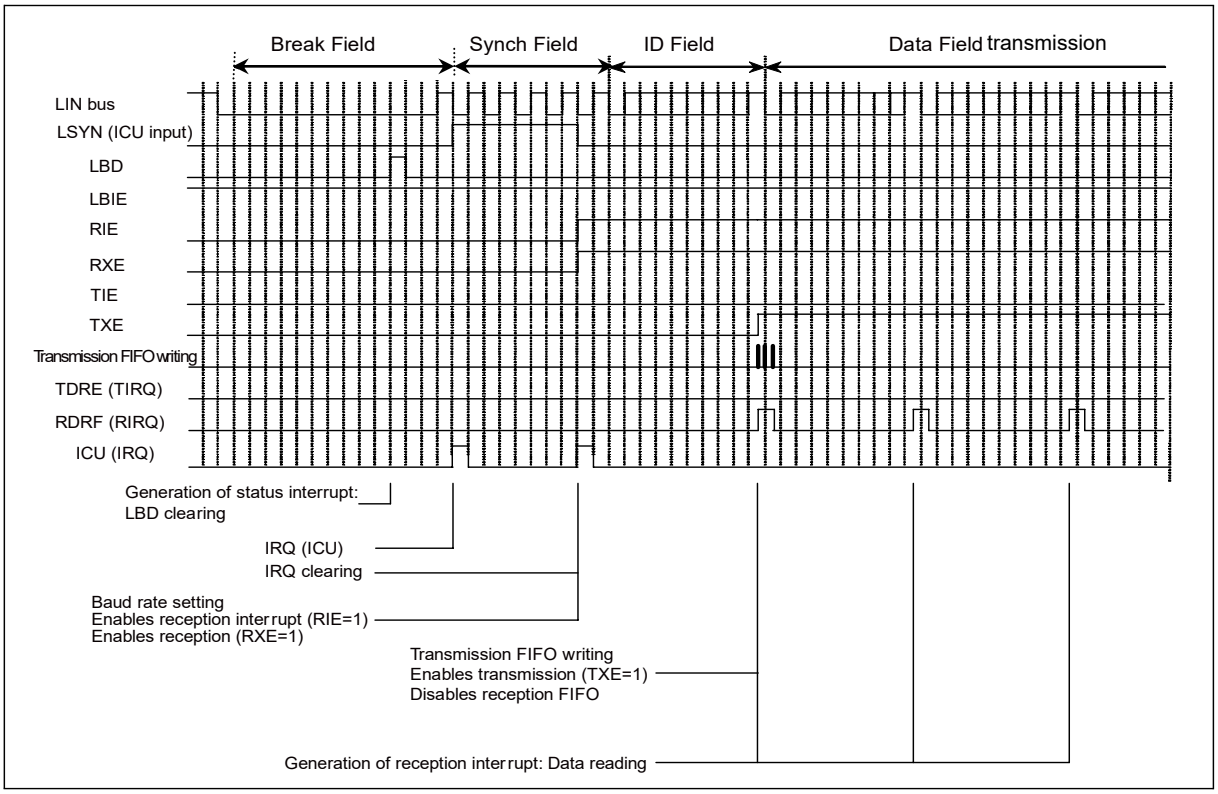
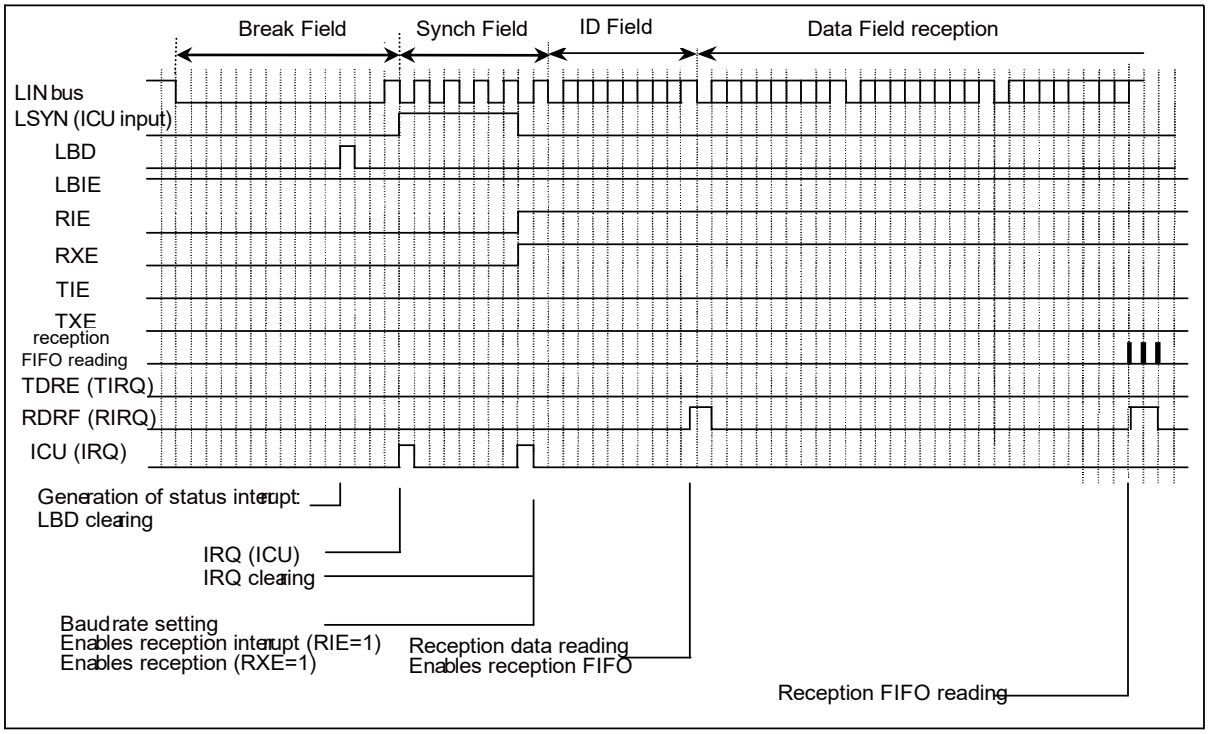


Figure 40-48. LIN Bus Timing (at the Time of Data Field Reception When Using FIFO)



40.7.2.3 LIN-UART Baud Rate Selection/Setting

The LIN-UART Baud Rate Selection/Setting is shown below.

The LIN-UART 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 "40.5.2.11. UART Baud Rate Selection/Setting".

40.7.3 Setup Procedure and Program Flow

The setup procedure and program flow are shown below.

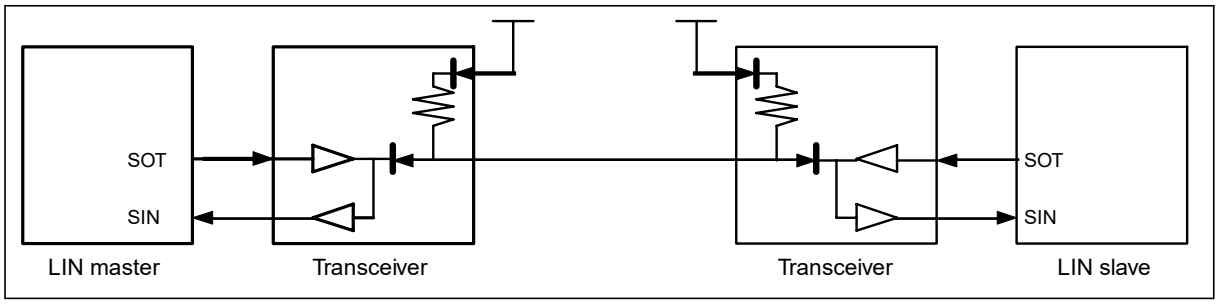
In operation mode 3 (LIN communication mode), the LIN-UART can be used for the LIN master system or LIN slave system.

40.7.3.1 Inter-CPU Connection

The Inter-CPU connection is shown below.

The following figure shows a communication system that contains one LIN master and one LIN slave. The multi function serial interface can work as a LIN master or LIN slave.

Figure 40-49. Example of LIN Bus System Communication



40.7.3.2 Flowchart Example

The flowchart Example is shown below.

Figure 40-50. Example of a Flowchart in LIN Communication Master Mode (without Using FIFO)

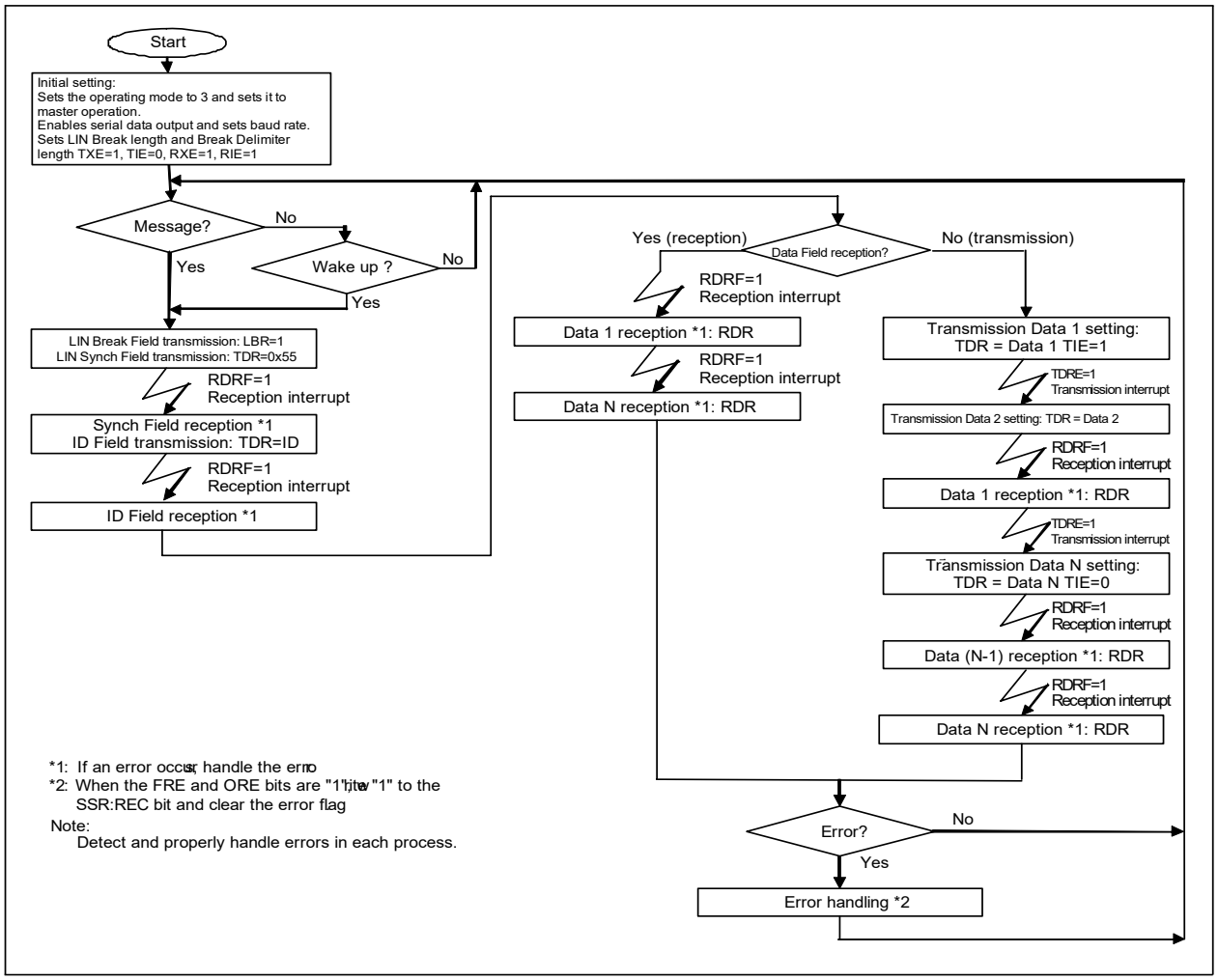
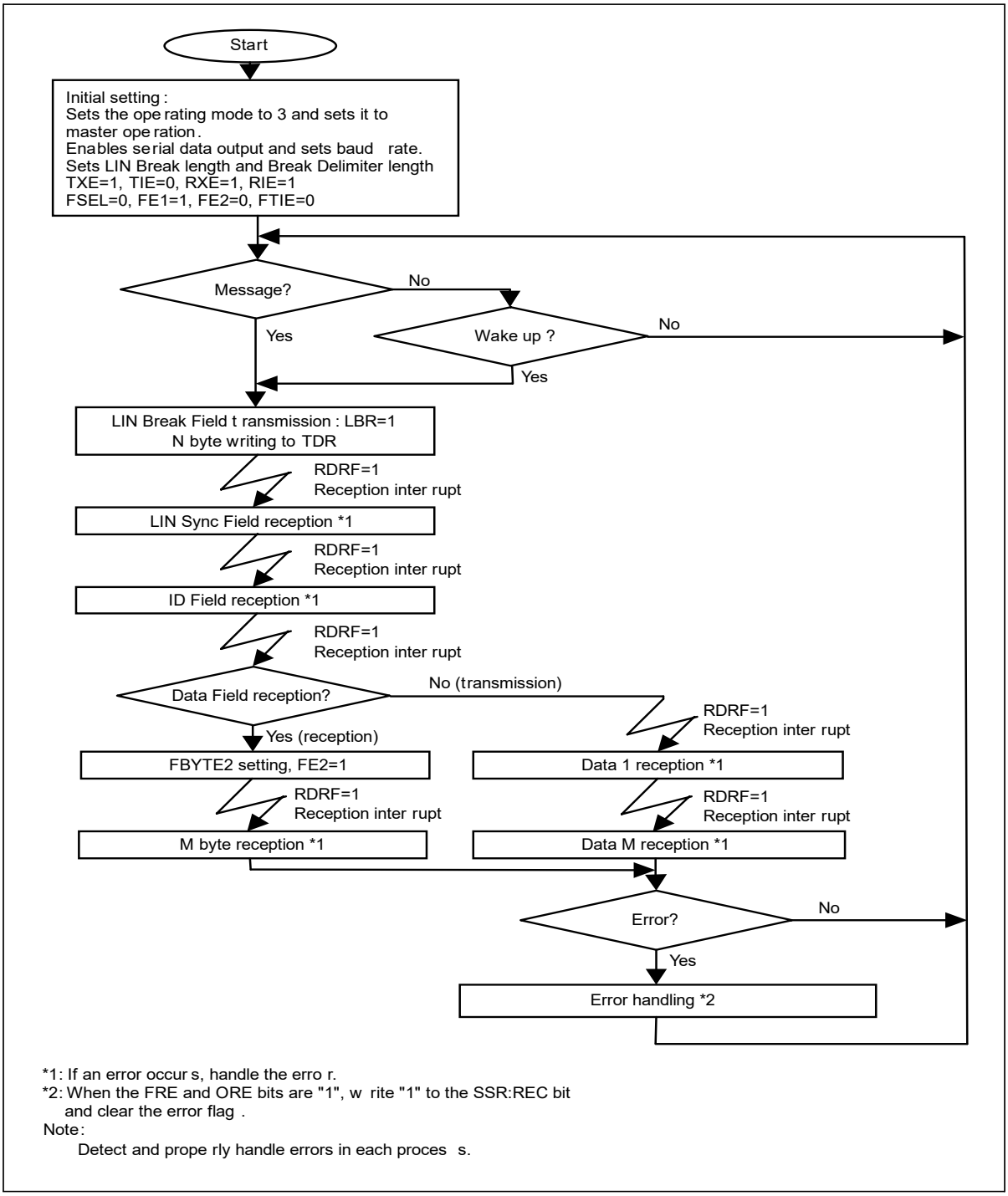


Figure 40-51. Example of a Flowchart in LIN Communication Master Mode (Using FIFO)



*1: If an error occur s, handle the erro r.
*2: When the FRE and ORE bits are "1", w rite "1" to the SSR:REC bit and clear the error flag .
Note:
Detect and prope rly handle errors in each proces s.

Figure 40-52. Example of a Flowchart in LIN Communication Slave Mode (without Using FIFO)

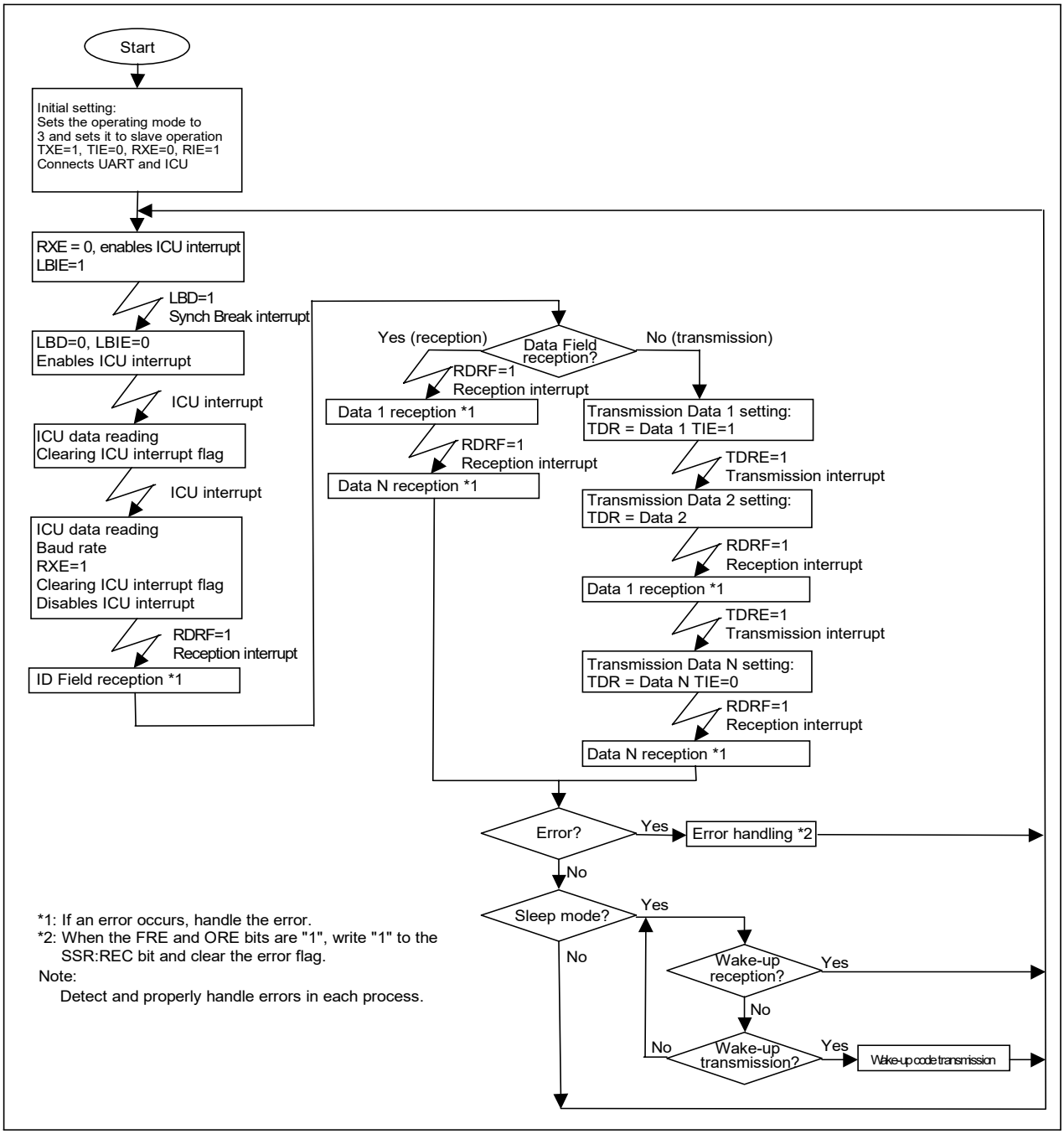
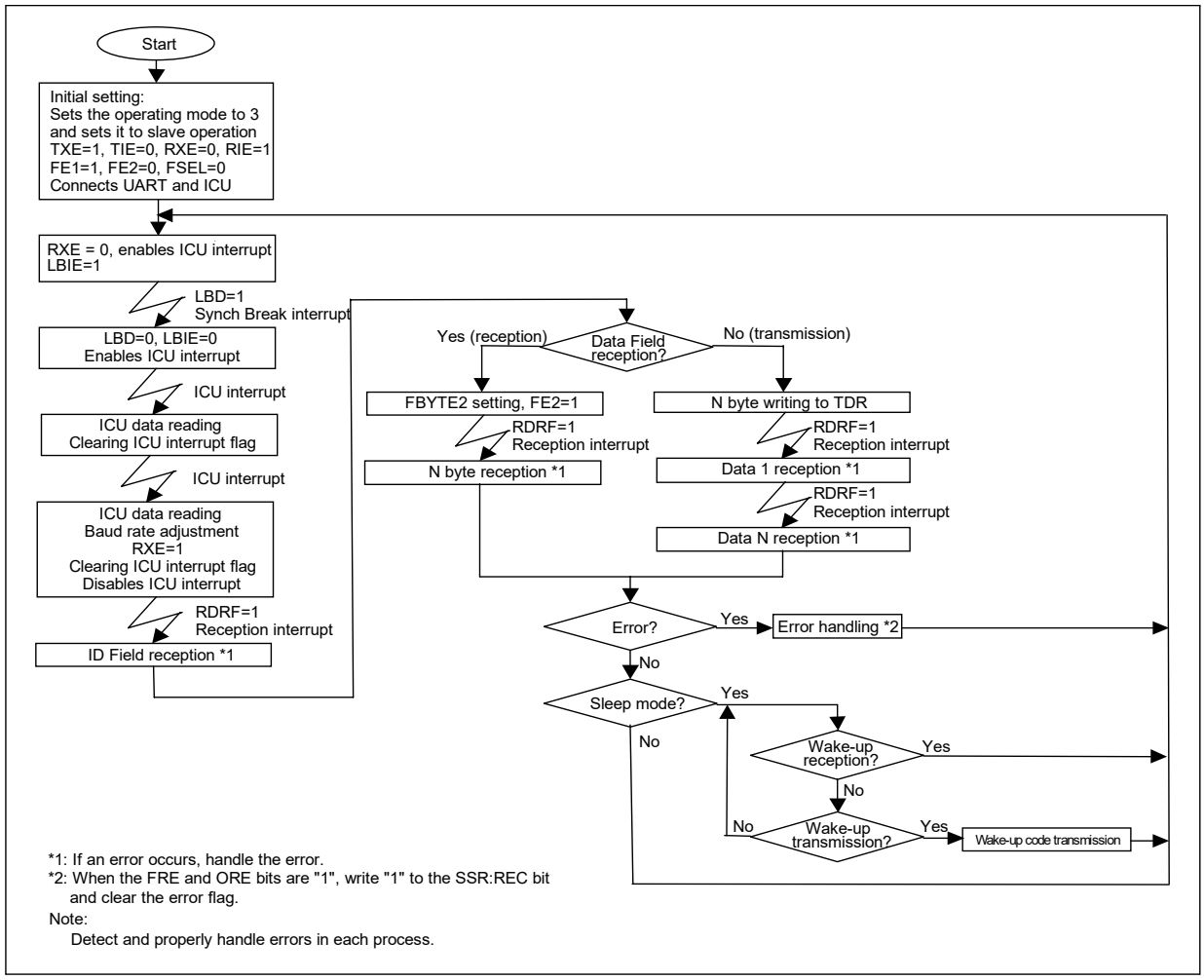


Figure 40-53. Example of a Flowchart in LIN Communication Slave Mode (Using FIFO)



40.8 Operation of I²C

The operation of I²C is shown.

40.8.1 Interrupts of I2C
40.8.2 Operation for I2C Interface Communication
40.8.3 I2C Master Mode
40.8.4 I2C Slave Mode
40.8.5 Bus Error
40.8.6 Example of I2C Flowchart

40.8.1 Interrupts of I²C

Interrupts of I²C are shown below.

The I²C interface can generate interrupt requests caused by the following factors:

- After transmission and reception of the first byte/after data transmission and reception
- Stop condition
- Repeated start condition
- FIFO transmission data request
- FIFO reception data completion

List of Interrupts of I²C Interface

The following table indicates how I²C interface interrupt control bits relate to interrupt factors.

Table 40-13. I²C Interface Interrupt Control Bits and Interrupt Factors

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Status	INT	IBCR	After transmission and reception of the first byte[1] (except Master Mode of SSR:DMA=1)	IBCR:INTE	Writing "0" to the interrupt flag bit (IBCR:INT)
			After data transmission and reception[1] (in the case of SSR:DMA=0)		
			Bus error detected		
			Arbitration lost detected		
			Reserved address detected		
			NACK reception		
			Reception FIFO full during slave reception		Writing "0" to INT after reading the reception data till the reception FIFO becomes empty
	SPC	IBSR	Stop condition	IBCR:CNDE	Writing "0" to SPC
	RSC		Repeated start detected		Writing "0" to RSC

Interrupt Type	Interrupt Request Flag Bit	Flag Register	Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request Flag
Reception	RDRF	SSR	Reserved address received	SMR:RIE	Reading of receive data (RDR)
			After data reception		
			Reception of as much data as specified by FBYTE		Reading of receive data (RDR) until the reception FIFO is emptied
			Reception idle detected by FBIIE="1"		
	ORE	SSR	Overrun error		Writing of "1" to the reception error flag bit (SSR:REC)
Transmission	TDRE	SSR	Transmission register is empty	SMR:TIE	Write to the transmit data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission)[2]
			Writing of "1" to the transmission buffer empty flag set bit (SSR:TSET)		
	FDRQ	FCR1	Transmission FIFO is empty	FCR1:FTIE	Writing of "0" to the FIFO transmission data request bit or the transmission FIFO is full
	TBI(SSR:DMA=1)	SSR	No transmission operation	SCR:TBIE	Write to the transmit data (TDR), or write "1" to the transmission FIFO operation enable bit when it is "0" and the transmission FIFO has a valid data (retransmission)[3]
			Writing of "1" to the transmission buffer empty flag set bit (SSR:TSET)		

[1]: No interrupt occurs if normal data can be transmitted/received and TDRE is "0". The purpose of this is to support DMA transfer. If you want to the INT flag to be set when data is transmitted or received, it is necessary that the TDRE bit will become "1" before the INT flag is set.

[2]: Set the TIE bit to "1" after the TDRE bit is cleared to "0".

[3]: Set the SSR:TBIE bit to "1" after the SSR:TBI bit is cleared to "0".

Note:

The DMA transfer triggered by data reception and a status interrupt is not supported.

40.8.2 Operation for I²C Interface Communication

The operation for I²C interface communication is shown below.

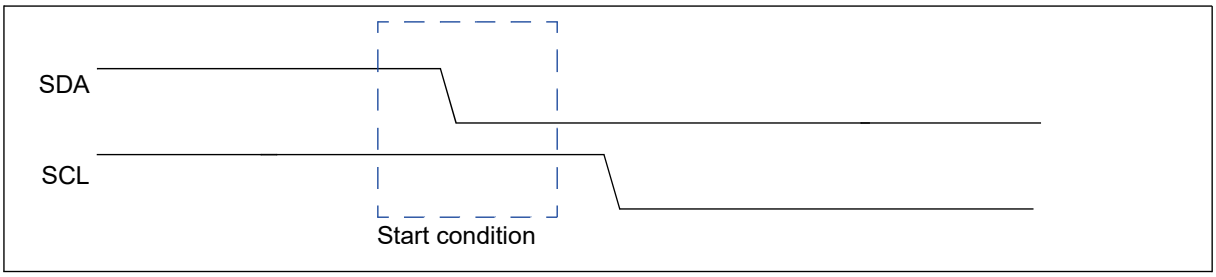
The I²C interface handles communication using two bidirectional bus lines, a serial data line (SDA), and a serial clock line (SCL).

40.8.2.1 I²C Bus Start Condition

The I²C bus start condition is shown below.

The condition for the I²C bus to be activated is as follows:

Figure 40-54. Start Condition

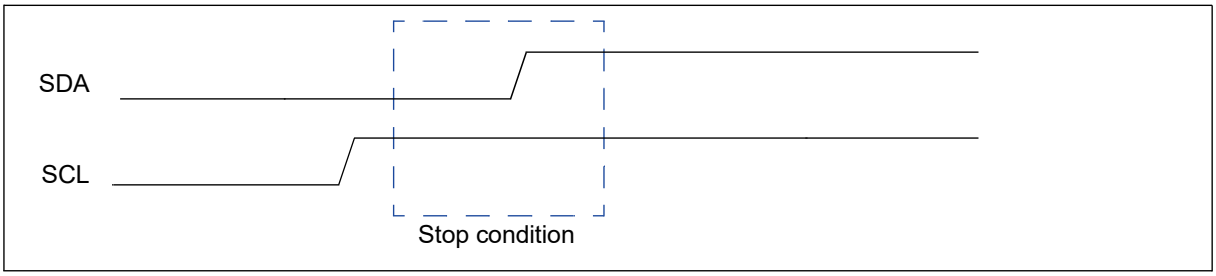


40.8.2.2 I²C Bus Stop Condition

The I²C Bus Stop Condition is shown below.

The condition for the I²C bus to stop is as follows:

Figure 40-55. Stop Condition

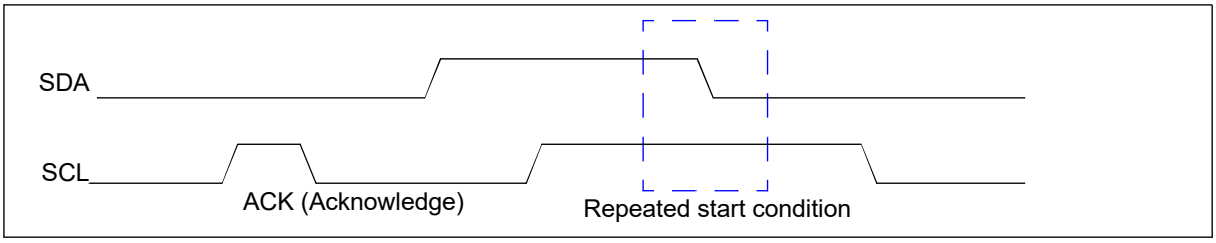


40.8.2.3 I²C Bus Repeated Start Condition

The I²C Bus repeated start condition is shown below.

The condition for the I²C bus to initiate a repeated start is as follows:

Figure 40-56. Repeated Start Condition



40.8.2.4 I²C Bus Error

The I²C bus error is shown below.

If a stop condition or (repeated) start condition is detected during data transmission/reception over the I²C bus, it is treated as a bus error.

- Bus Error Occurrence Condition

  A bus error sets the IBCR:BER bit to "1" in one of the following conditions:

  - ☐ Detection of a (repeated) start or stop condition during the transfer of the first byte
  - ☐ Detection of a (repeated) start or stop condition at the second to ninth (acknowledge) bits of the data

- Bus Error Operation

  If the interrupt flag (IBCR:INT) becomes "1" due to transmission or reception, check the IBCR:BER bit.

  If the IBCR:BER bit is "1", perform error handling. The IBCR:BER bit is cleared by writing "0" to the IBCR:INT bit. A bus error sets the IBCR:INT bit to "1", but does not bring the I²C bus to a wait state by setting SCL to "L".

40.8.2.5 Baud Rate Generation

The baud Rate Generation is shown below.

The dedicated baud rate generator sets a serial clock frequency.

Baud Rate Selection

- 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 frequency of the internal clock by the specified value.

Baud Rate Calculation

The two 15-bit reload counters are set using the baud rate generator register (BGR). The baud rate calculation formulas are as follows:

- Reload value

  $V = \varphi / b - 1$

  V: Reload value b: Baud rate φ: Internal clock (peripheral clock (PCLK)) frequency

  However, the specified baud rate may not be generated depending on the rising time of SCL on the I^2C bus. Adjust the reload value as required.

- Example of calculation

  The reload value is as follows if the internal clock (peripheral clock (PCLK)) frequency is 16 MHz, the baud rate is to be 400kbps:

  Reload value:

  $V = (16 \times 1{,}000{,}000) / 400{,}000 - 1 = 39$

  Therefore, the baud rate is

  $b = (16 \times 1{,}000{,}000) / (39 + 1) = 400$ kbps

Notes:

- Write to the baud rate generator (BGR) in 16-bit access mode.
- Configure the baud rate generator when the EN bit of the ISMK register is "0".
- Peripheral clock (PCLK) should be set with 8 MHz or more in operating mode 4 (I^2C mode) and baud rate generator configured in 400 kbps or more should not be used.
- Set the reload value to "0" to stop the reload counter.

Reload Values Relating to Baud Rates and Internal Clock Frequencies

Table 40-14. Reload Values Relating to Baud Rates and Internal Clock Frequencies

Baud Rate [bps]	Internal Clock (Peripheral Clock (PCLK))					
	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32 MHz
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

In the number value, SCL rising of the I^2C bus is a case of 0s. When SCL rising of the I^2C bus is slow, it becomes a baud rate that is slower than the above-mentioned numerical value.

Reload Counter Functions

The reload counter consists of a 15-bit register for reload values and generates a transmission/reception clock from the internal clock. In addition, the count value of the transmission reload counter can be read via the baud rate generator register (BGR).

Count Start

When a reload value is written to the baud rate generator register (BGR), the reload counter will start counting.

40.8.3 I²C Master Mode

I²C master mode is shown below.

In master mode, a start condition is generated on the I²C bus, which then output the clock to the I²C bus. If I²C bus is in the idle state (SCL="H", SDA="H"), the master mode is selected when "1" is set to the MSS bit in the IBCR register, and the ACT bit in the IBCR register becomes "1".

40.8.3.1 Start Condition Generation

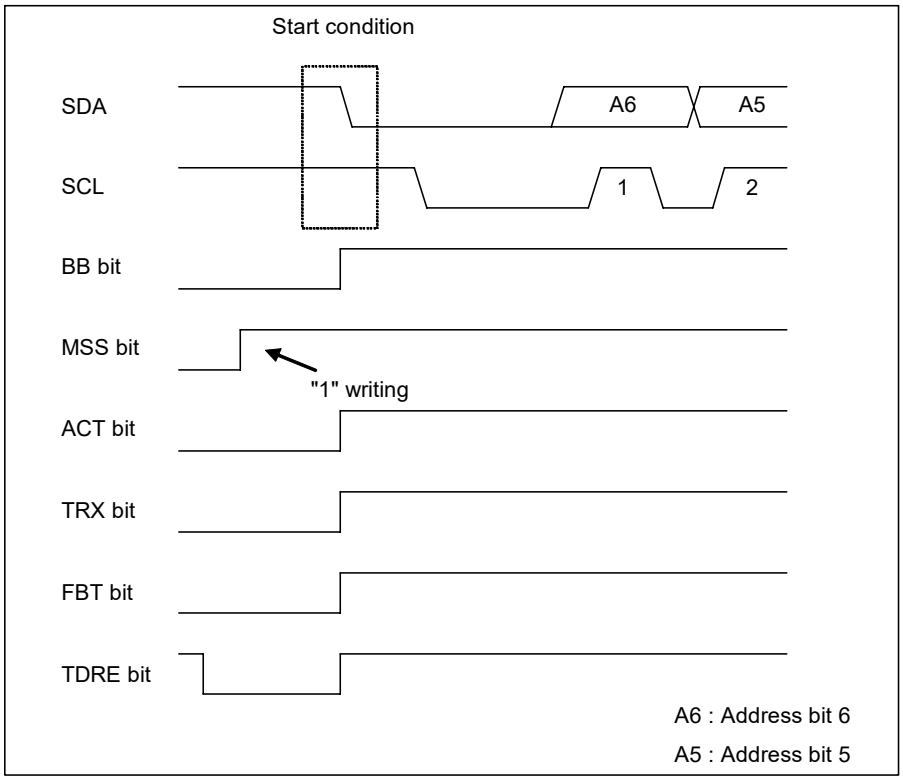
The start condition generation is shown below.

A start condition is output when:

- If SDA="H", SCL="H", EN=1, and BB=0, write "1" to the IBCR:MSS bit

If a start condition is output to I2C bus the IBCR:ACT bit is set to "1". Then, once the start condition is received, the IBSR:BB bit is set to "1", indicating that I^2C bus is on the communication.

Figure 40-57. Relationship between Start Condition Output and Various Bits



Note:

The peripheral clock (PCLK) should be set with 8MHz or more in operating mode 4 (I^2C mode) and baud rate generator configured in 400 kbps or more should not be used.

40.8.3.2 Slave Address Output

The slave address output is shown below.

When a start condition is output, the data contained in the TDR register is output as the address, beginning with bit7. If FIFO is enabled, the data first written in the TDR register is output. Bit0 is used to indicate the data direction bit (R/W). If the data direction bit (R/W) is "0", the data is in the write direction (from master to slave). Set the address for the TDR register before "1" is written to IBCR:MSS or IBCR:SCC.

Figure 40-58. Address and Data Direction (when FIFO Is Disabled)

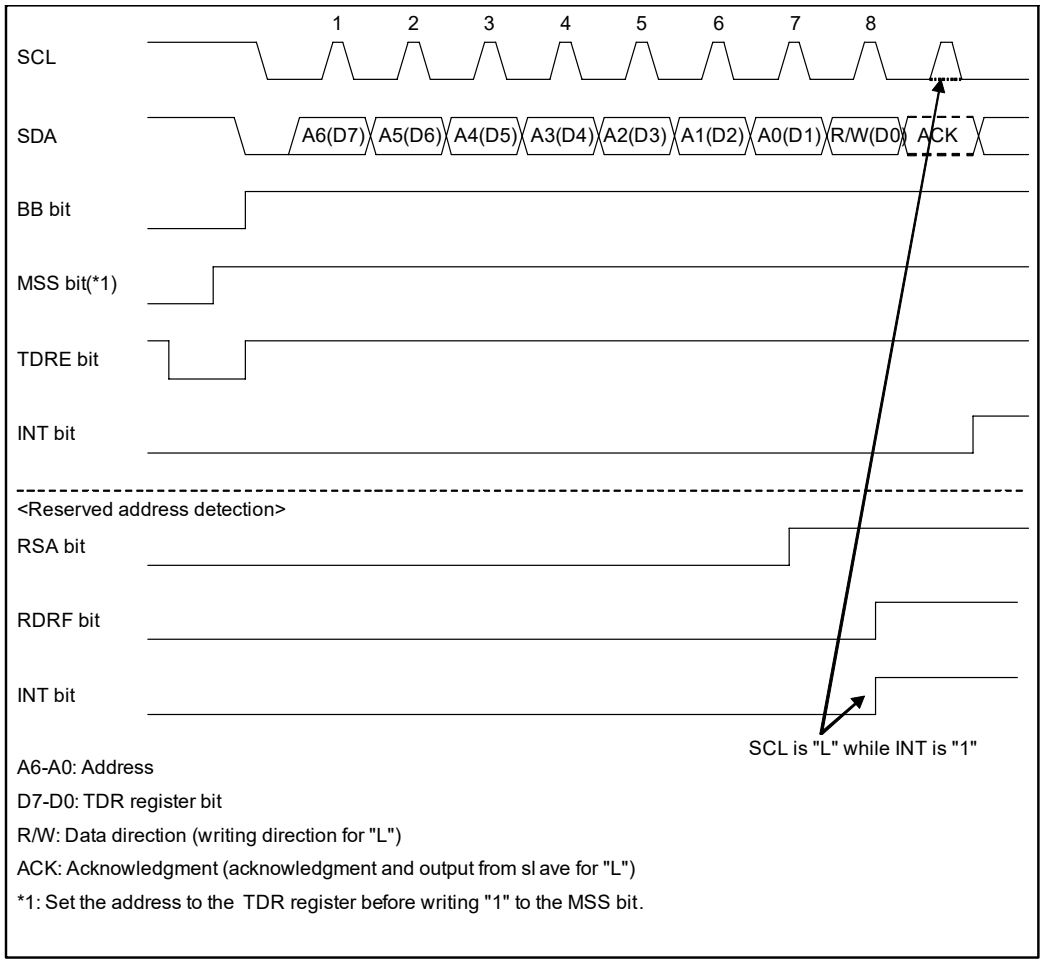
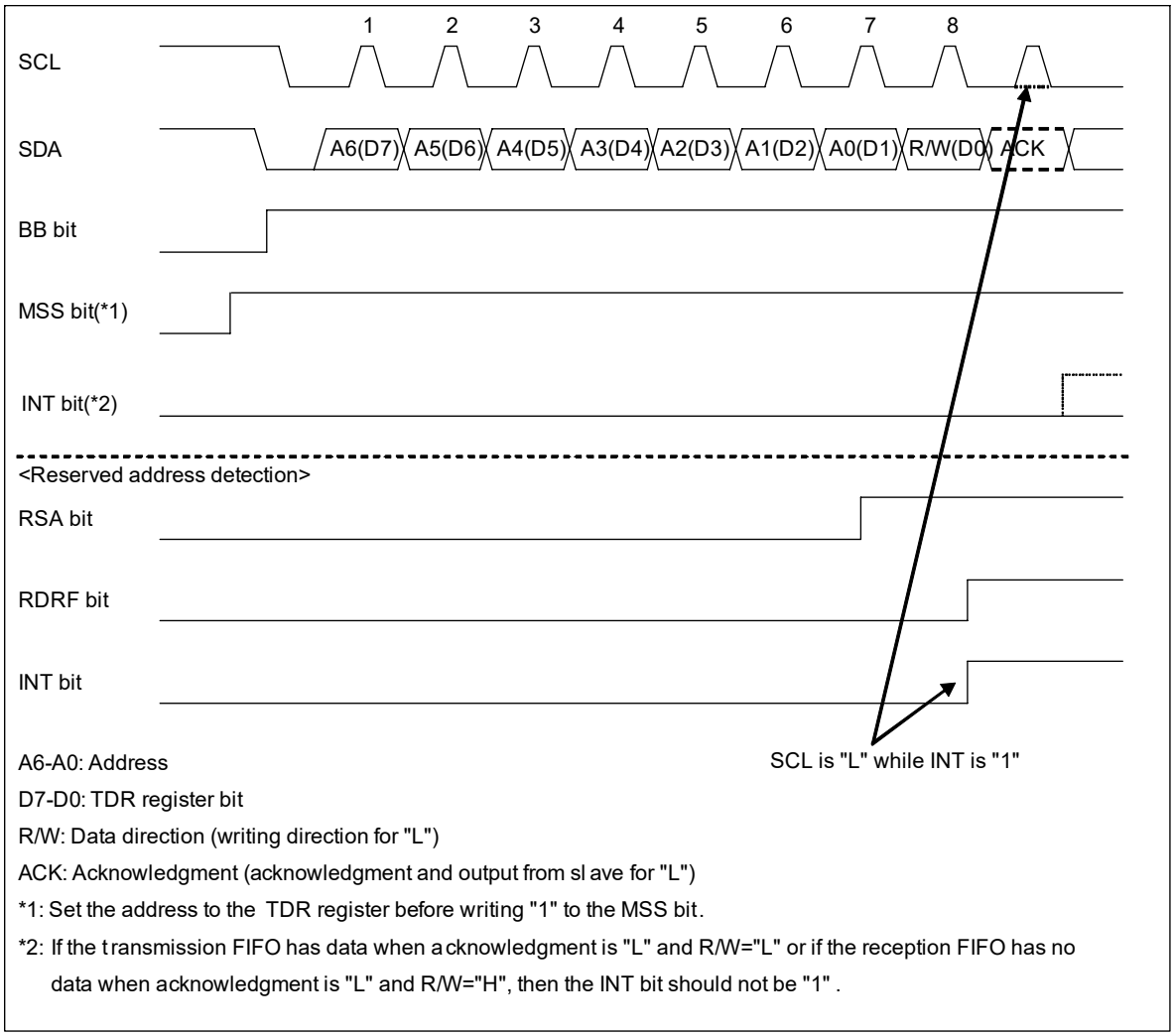


Figure 40-59. Address and Data Direction (when Transmission/Reception FIFO Is Enabled)



A6-A0: Address
D7-D0: TDR register bit
R/W: Data direction (writing direction for "L")
ACK: Acknowledgment (acknowledgment and output from sl ave for "L")
*1: Set the address to the TDR register before writing "1" to the MSS bit.
*2: If the transmission FIFO has data when acknowledgment is "L" and R/W="L" or if the reception FIFO has no data when acknowledgment is "L" and R/W="H", then the INT bit should not be "1" .

40.8.3.3 Acknowledge Reception by Transmitting First Byte

Acknowledge reception by transmitting first byte is shown below.

When the data direction bit (R/W) is output, the I²C interface receives an acknowledge from the slave. The operation varies depending on whether FIFO is enabled or disabled, as indicated in the following table:

Table 40-15. Operation after Acknowledge Reception When DMA Mode is Disabled (IBSR:RSA =0, SSR:DMA=0)

Transmission FIFO Operation	Reception FIFO Operation	Transmission FIFO Status	Reception FIFO Status	Data Direction Bit (R/W)	Operation Immediately after Acknowledge Reception	
					Acknowledge is ACK	Acknowledge is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Disabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Enabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	

Table 40-16. Operation after Acknowledge Reception(When DMA Mode is Enabled) (IBSR:RSA=0, SSR:DMA=1)

Transmission FIFO Operation	Reception FIFO Operation	Transmission FIFO Status	Reception FIFO Status	Data Direction Bit (R/W)	Operation Immediately after Acknowledge Reception	
					Acknowledge is ACK	Acknowledge is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Disabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
				1		
Enabled	Enabled	-	Without data	0	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	The IBCR:INT bit is set to "1" and waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the SSR:TBI bit is set to "1" and waited. If the SSR:TDRE bit is "0", the SSR:TBI bit is held to "0" and not waited.	

FIFO Disabled (Both Transmission and Reception FIFOs Disabled, When DMA Mode is Disabled (SSR:DMA=0))

- If the IBSR:RSA bit is "0", the interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited if the SSR:TDRE bit is "1" after acknowledge reception. To release the wait, write "0" to the interrupt flag. If the SSR:TDRE bit is "0", the reception of ACK causes clock generation on SCL without setting the interrupt flag to "1".
- If the IBSR:RSA bit is "1", the interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited after reserved address reception (before acknowledge). After the RDR register is read, the interrupt flag becomes "0" to release the wait when you set the IBCR:ACKE bit and the transmission data, and write "0" to the interrupt flag.
- The received acknowledge is set to the IBSR:RACK bit. The IBSR:RACK bit is checked during wait state. If it is NACK, "0" will be written to the IBCR:MSS bit or "1" is written to the IBCR:SCC bit to generate a stop condition or a repeated start condition. At this time, the IBCR:INT bit will be automatically cleared to "0".

FIFO Enabling (When DMA Mode is Disabled (SSR:DMA=0))

- Before setting the IBCR:MSS bit to "1", it is necessary to configure the following FIFO settings:
  - ☐ For transmission to the slave (data direction bit =0), set data including the slave address, etc. in the transmission FIFO.
  - ☐ For data reception from the slave (data direction bit =1), configure the FIFO byte count register to specify the number of bytes to be received, write to the transmit data register using the slave address, data direction bit, and number of dummy data to be received.
- If the IBSR:RSA bit is "0", the master, after receiving ACK as an acknowledge, does not set the interrupt flag (IBCR:INT) to "1", but transmits/receives data according to the data direction bit (not waited). If NACK is received, the interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited.
- The received acknowledge is set to the IBSR:RACK bit. The IBSR:RACK bit is checked during wait state. If it is NACK, "0" is written to the IBCR:MSS bit or "1" is written to the IBCR:SCC bit to generate a stop condition or a repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

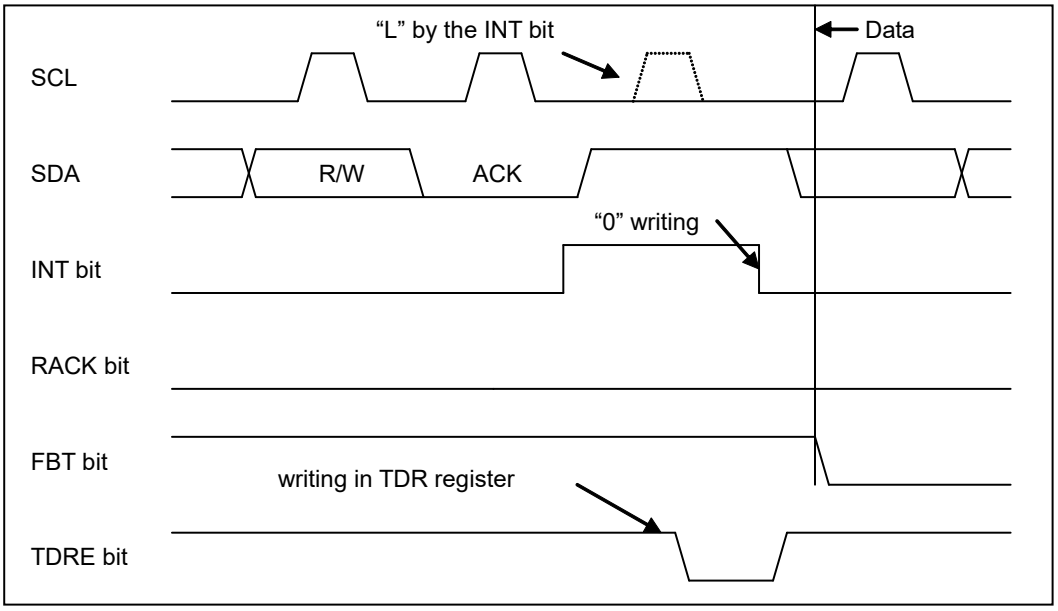
FIFO Disabled (Both Transmission and Reception FIFOs Disabled, When DMA Mode is Enabled (SSR:DMA=1))

- If the IBSR:RSA bit is "0", the transmission bus idle flag (SSR:TBI) is set to "1" and SCL is held to "L" and waited if the SSR:TDRE bit is "1" after acknowledge reception. If the transmission data is written to TDR register, the transmission bus idle flag becomes "0" and wait is released. If the SSR:TDRE bit is "0", the reception of ACK causes clock generation on SCL without setting the transmission bus idle flag (SSR:TBI) to "1".
- If the IBSR:RSA bit is "1", the interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited after reserved address reception (before acknowledge). After the RDR register is read, the interrupt flag becomes "0" to release the wait when you set the IBCR:ACKE bit and the transmission data, and write "0" to the interrupt flag.
- The received acknowledge is set to the IBSR:RACK bit. The IBSR:RACK bit is checked during wait state. If it is NACK, "0" will be written to the IBCR:MSS bit or "1" is written to the IBCR:SCC bit to generate a stop condition or a repeated start condition. At this time, the IBCR:INT bit will be automatically cleared to "0".

FIFO Enabling (When DMA Mode is Enabled (SSR:DMA=1))

- Before setting the IBCR:MSS bit to "1", it is necessary to configure the following FIFO settings:
  - For transmission to the slave (data direction bit =0), set data including the slave address, etc. in the transmission FIFO.
  - For data reception from the slave (data direction bit =1), configure the FIFO byte count register to specify the number of bytes to be received, write to the transmit data register using the slave address, data direction bit, and number of dummy data to be received.
- If the IBCR:RSA bit is "0", after receiving ACK as an acknowledge, the interrupt flag (IBCR:INT) is not set to "1", but transmits/receives data according to the data direction bit (not waited). If NACK is received, the interrupt flag (IBCR:INT) is set to "1" and SCL is held to "L" and waited.
- The received acknowledge is set to the IBSR:RACK bit. The IBSR:RACK bit is checked during wait state. If it is NACK, "0" is written to the IBCR:MSS bit or "1" is written to the IBCR:SCC bit to generate a stop condition or a repeated start condition. At this time, the IBCR:INT bit is automatically cleared to "0".

Figure 40-60. Acknowledge (If FIFO Is Disabled, IBSR:RSA=0, and the Response Is ACK)



Waiting to the address is as follows.

- After receiving the acknowledge when the IBSR:RSA bit is "0"
- Before receiving the acknowledge when the IBSR:RSA bit is "1"

It does not depend on the setting of IBCR:WSEL.

Figure 40-61. Acknowledge (If FIFO Is Disabled, IBSR:RSA=0, and the Response Is NACK)

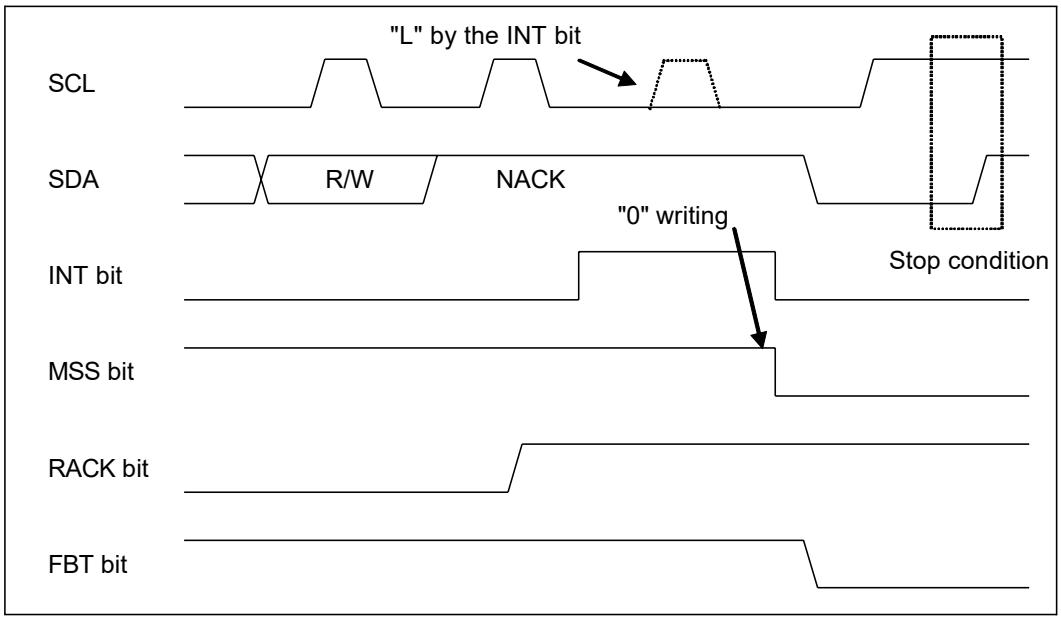


Figure 40-62. Acknowledge (If FIFO Is Disabled, IBSR:RSA=1, and the Response Is ACK)

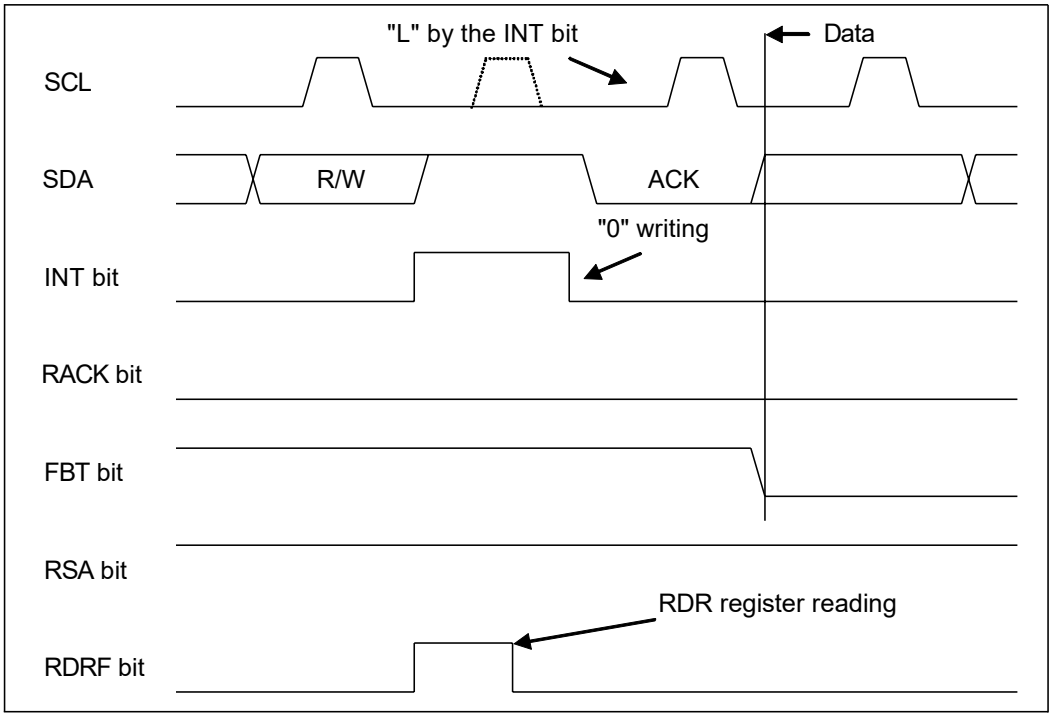


Figure 40-63. Acknowledge (If FIFO Is Disabled, IBSR:RSA=1, and the Response Is NACK)

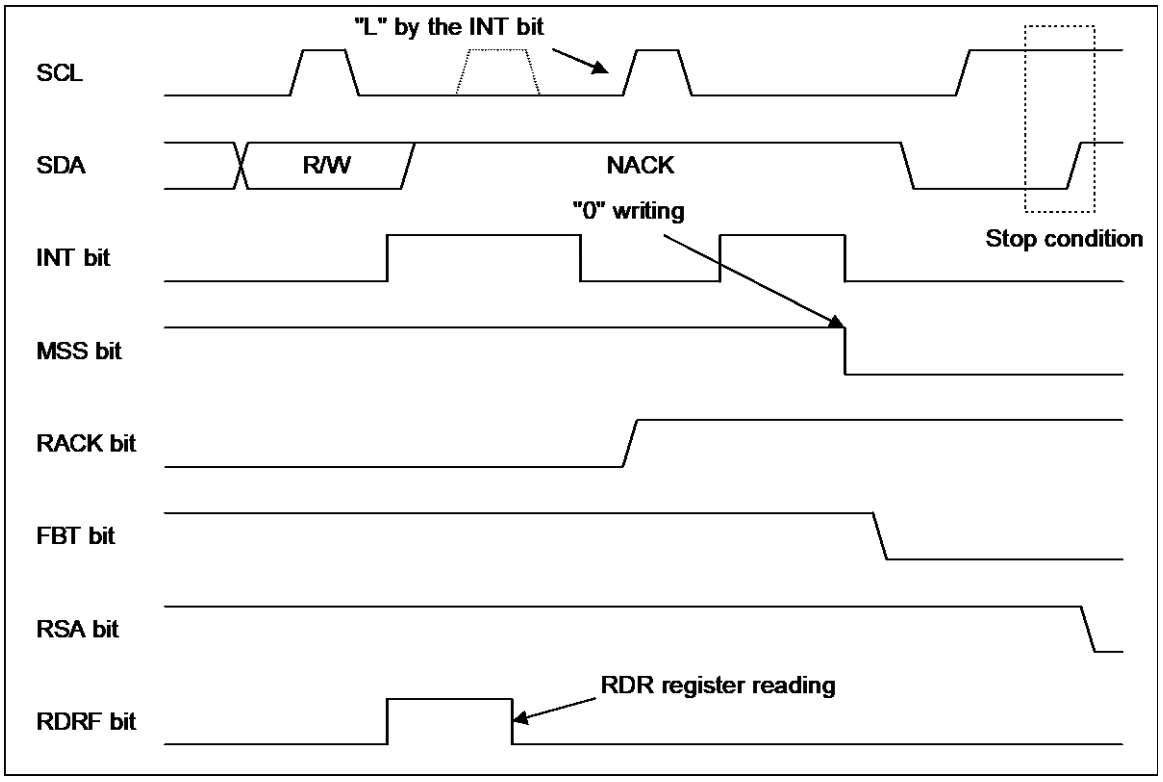
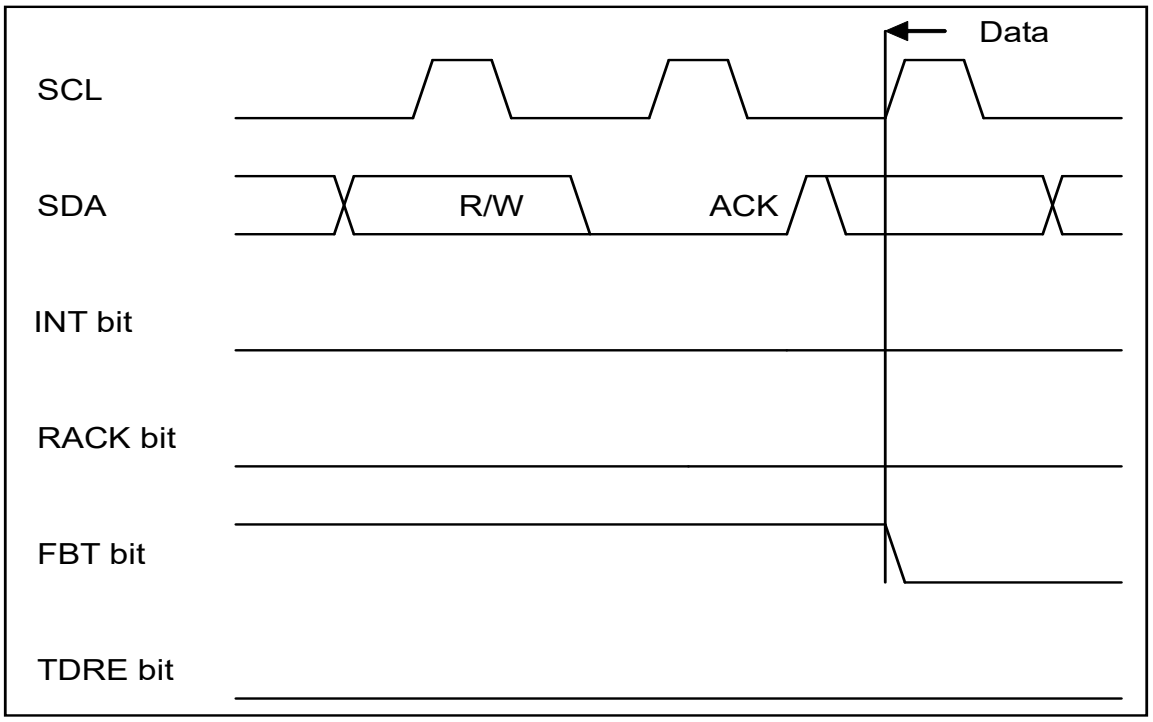


Figure 40-64. Acknowledge (If FIFO Is Enabled, Transmission FIFO Data Exists, No Reception FIFO Data Exists, IBSR:RSA=0, and the Response Is ACK)



40.8.3.4 Data Transmission by Master

Data transmission by master is shown below.

If the data direction bit (R/W) is "0", data is sent from the master. The slave responds with ACK or NACK each time one byte is transmitted. The location where a wait condition develops varies depending on the IBCR:WSEL bit setting as follows:

Table 40-17. IBCR:WSEL Bit at the Time of Master Data Transmission (When DMA Mode is Disabled(SSR:DMA=0))

WSEL	Operation
0	In the 2nd or subsequent byte, the interrupt flag bit (IBCR:INT) is set to "1" and SCL is set to "L" to go into the wait state when the SSR:TDRE bit is "1" or after the acknowledgment on the arbitration lost detection. If FIFO is enabled, the master, after receiving acknowledge, will set the post-acknowledge interrupt flag (IBCR:INT) to "1" to wait, when it detects an arbitration lost or finds no valid data in the transmit data register (SSR:TDRE=1).
1	In the 2nd or subsequent byte, the interrupt flag bit (IBCR:INT) will be set to "1" and SCL will be set to "L" to go into the wait state when the SSR:TDRE bit is "1" or after the master transmitted 1 byte data on the arbitration lost detection. If FIFO is enabled, the master will transmit data, and then set the interrupt flag (IBCR:INT) to "1" and wait, when it detects arbitration lost or finds no valid data in the transmit data register (SSR:TDRE=1).

Table 40-18. IBCR:WSEL Bit at the Time of Master Data Transmission (When DMA mode is enabled(SSR:DMA=1))

WSEL	Operation
0	In the 2nd or subsequent byte, after the acknowledge by the SSR:TDRE bit to "1", transmission bus idle flag (SSR:TBI) makes "1" and SCL "L" into the wait state. Moreover, transmission bus idle flag (SSR:TBI) is made "1" after the acknowledge at (SSR:TDRE=1) of lost of effective data for the transmission data register after the acknowledge at the FIFO enabled and it puts it into the wait state .
1	In the 2nd or subsequent byte, after the master transmits the data of one byte by the SSR:TDRE bit to "1", transmission bus idle flag (SSR:TBI) makes "1" and SCL "L" into the wait state. Moreover, transmission bus idle flag (SSR:TBI) is made "1" after the master transmits the data of one byte at (SSR:TDRE=1) of lost of effective data for the transmission data register after the acknowledge at the FIFO permission and it puts it into the wait state.

However, the master sets the interrupt flag (IBCR:INT) after receiving acknowledge regardless of the IBCR:WSEL setting in one of the following cases:

- If NACK is received except for stop condition setting (IBCR:MSS=0, ACT=1)

  The following gives an example of procedure used to transmit data to the slave:

  - Data transmission to slave of when DMA mode is disabled (SSR:DMA=0)

    Transmission to a destination that is not at the reserved address

    If transmission FIFO is disabled:

    1. Set the slave address (including the data direction bit) in the TDR register, and set the IBCR: MSS bit to "1".
    2. Transmit the slave address and receive ACK. The interrupt flag (IBCR:INT) becomes "1".
    3. Write transmission data in the TDR register.
    4. Update the IBCR:WSEL bit and set the interrupt flag (IBCR:INT) to "0" to release the I^2C bus from waiting state.
    5. Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when IBCR:WSEL is set to "1". Repeat steps (3) to (5) until the specified number of data have been transmitted. However, if NACK is received after the bus is released from waiting state when IBCR:WSEL=1, another interrupt will occur after acknowledge reception to make the bus wait.
    6. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.

    If transmission FIFO is enabled:

    1. Write the slave address (including the data direction bit) and transmission data in the TDR register.
    2. Set the IBCR:WSEL bit and write "1" to the IBCR:MSS bit.
    3. If NACK is received during transmission, set the interrupt flag (IBCR:INT) to "1" immediately to make the I^2C bus wait. If all responses received are ACK, set the interrupt flag to"1" after transmitting the last byte, according to the IBCR:WSEL setting to make the I^2C bus wait.
    4. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.

Transmission to the reserved address

If transmission FIFO is disabled:

1. Set the reserved address as the slave address in the TDR register and set the IBCR:MSS bit to "1".
2. Transmit the slave address. The interrupt flag (IBCR:INT) becomes "1".
3. Read the RDR register and confirm the reserved address.[1]
4. Write transmission data in the TDR register.
5. Update the IBCR:WSEL bit and set the interrupt flag (IBCR:INT) to "0" to release the I2C bus from waiting state.
6. Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when IBCR:WSEL is set to "0", or immediately after one byte has been transmitted when IBCR:WSEL is set to "1". Repeat steps (4) to (6) until the specified number of data have been transmitted. However, if NACK is received after the bus is released from waiting state when IBCR:WSEL=1, another interrupt will occur after acknowledge reception to make the bus wait.
7. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.

If transmission FIFO is enabled:

1. Set the reserved address as the slave address in the TDR register and set the IBCR:MSS bit to "1".
2. Transmit the slave address. The interrupt flag (IBCR:INT) becomes "1".
3. Read the RDR register and confirm the reserved address.[1]
4. Write all transmission data in the TDR register (until the transmission FIFO becomes full if it can).
5. If NACK is received during transmission, set the interrupt flag (IBCR:INT) to "1" immediately to make the I^2C bus wait. If all responses received are ACK, set the interrupt flag to"1" after transmitting the last byte, according to the IBCR:WSEL setting to make the I^2C bus wait.
6. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.

[1]: If the reserved address is a general-call address in a multi-master configuration or if an arbitration lost is detected and the device may work as the slave, it is necessary to set the IBCR:ACKE bit and IBCR:WSEL bit to "1" and determine whether the device is to work as the master or slave for subsequent data.

☐ Data transmission to slave of when DMA mode is enabled (SSR:DMA=1)

Transmission to a destination that is not at the reserved address

If transmission FIFO is disabled:

1. Slave Address (The data direction bit is included) is set in the TDR register and set the IBCR: MSS bit to "1".
2. Transmit the slave address and receive ACK. The transmission bus idle flag (SSR:TBI) becomes "1".
3. The data transmitted to the TDR register is written and release the I^2C bus from waiting state.
4. Put the I^2C bus in a wait by setting the transmission bus idle flag (SSR:TBI) to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when IBCR:WSEL is set to "1".
5. The data transmitted to the TDR register is written and release the I^2C bus from waiting state.
6. Put the I^2C bus in a wait by setting the transmission bus idle flag (SSR:TBI) to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when IBCR:WSEL is set to "1". Repeat steps (5) to (6) until the specified number of data have been transmitted. However, if NACK is received after the bus is released from waiting state when IBCR:WSEL=1, interrupt flag (IBCR:INT) is set to "1" after acknowledge reception to make the bus wait.
7. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.[2]

If transmission FIFO is enabled:

1. Slave Address (The data direction bit is included) and the transmission data is written to the TDR register.
2. Update the IBCR:WSEL bit and set the IBCR:MSS bit to "1".
3. If NACK is received during transmission, set the interrupt flag (IBCR:INT) to "1" immediately to make the I^2C bus wait. If all responses received are ACK, set the transmission bus idle flag (SSR:TBI) to"1" after transmitting the last byte, according to the IBCR:WSEL setting to make the I^2C bus wait.
4. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.[2]

Transmission to the reserved address

If transmission FIFO is disabled:

1. Set the reserved address as the slave address in the TDR register and set the IBCR:MSS bit to "1".
2. Transmit the slave address. The interrupt flag (IBCR:INT) becomes "1".
3. Read the RDR register and confirm the reserved address.[1]
4. Write transmission data in the TDR register.
5. Update the IBCR:WSEL bit and set the interrupt flag (IBCR:INT) to "0" to release the I2C bus from waiting state.
6. Put the I2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when IBCR:WSEL is set to "0", or immediately after one byte has been transmitted when IBCR:WSEL is set to "1".
7. The data transmitted to the TDR register is written and release the I^2C bus from waiting state.
8. When the IBCR:WSEL bit is 0 after one byte is transmitted, acknowledge reception also when the IBCR:WSEL=1 is 1, one byte is transmitted and the transmission bus idle flag (SSR:TBI) is set as "1" and I2C bus is wait. Repeat steps (7) to (8) until the specified number of data have been transmitted. However, if NACK is received after the bus is released from waiting state when IBCR:WSEL=1, another interrupt will occur after acknowledge reception to make the bus wait.
9. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.[2]

If transmission FIFO is enabled:

1. Set the reserved address as the slave address in the TDR register and set the IBCR:MSS bit to "1".
2. Transmit the slave address. The interrupt flag (IBCR:INT) becomes "1".
3. Read the RDR register and confirm the reserved address.[1]
4. Write all transmission data in the TDR register (until the transmission FIFO becomes full if it can).
5. If NACK is received during transmission, set the interrupt flag (IBCR:INT) to "1" immediately to make the I2C bus wait. If all responses received are ACK, set the interrupt flag (IBCR:INT) to"1" after transmitting the last byte, according to the IBCR:WSEL setting to make the I^2C bus wait.
6. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" to generate a stop condition or a repeated start condition.[2]

   [1]: If the reserved address is a general-call address in a multi-master configuration or if an arbitration lost is detected and the device may work as the slave, it is necessary to set the IBCR:ACKE bit and IBCR:WSEL bit to "1" and determine whether the device is to work as the master or slave for subsequent data.

   [2]: The DMA mode must write the slave address in TDR after confirming the IBCR:INT bit is set in "1" after "1" is written in the IBCR:INT bit, and set "1" to the IBCR:SCC bit when the SSR:TBI bit issues the repetition start condition by permission (SSR:DMA=1) when it is "1" and the IBCR:INT bit is "0".

Notes:

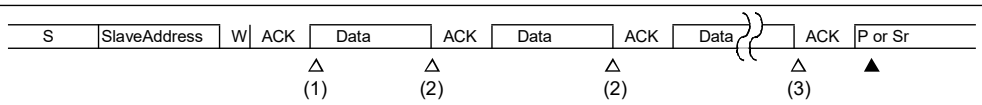
- If the 7-bit slave address detection is enabled (ISBA:SAEN=1), you cannot specify a 7-bit slave address in the master mode.
- If you need to change the IBCR register during data sending or receiving, change it only when the interrupt flag (IBCR:INT) is "1".
- If you change the IBCR:WSEL bit, it is used interrupt flag of following data (IBCR:INT) and generation condition of the transmission bus idle flag (SSR:TBI) when the DMA mode is enabled (DMA=1).
- In DMA mode prohibition (DMA=0), if transmission data is written to the TDR register when SSR:TDRE is "1" during data transmission, the detection of an ACK response triggers the transmission of the written data without setting the interrupt flag (IBCR:INT) to "1".
- In DMA mode prohibition (DMA=0),if transmission data is written to the TDR register when TDRE is "1" during data reception, also there is an ACK response, the interrupt flag (IBCR:INT) is not set as "1", but only sets RDRF to "1" (if reception FIFO is enabled and as much data as specified by the FBYTE register is received).
- In DMA mode permission, if transmission data is written to the TDR register when TDRE is "1" during data reception, also there is an ACK response, the transmission bus idle flag(SSR:TBI) is not set as "1", but only keep written data.
- In DMA mode permission, if transmission data is written to the TDR register when TDRE is "1" during data reception, also there is an ACK response, the transmission bus idle flag(SSR:TBI) is not set as "1", but only sets RDRF to "1" (if reception FIFO is enabled and as much data as specified by the FBYTE register is received).

Figure 40-65. Master Transmission Interrupt (1)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=0 and IBSR:RSA=0)



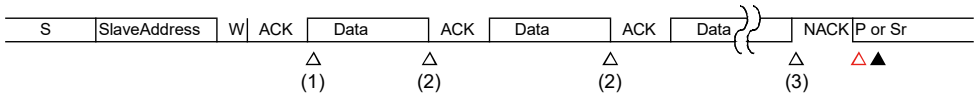
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register
(3) An interrupt generated by 1 byte transmission + acknowledgment reception
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 40-66. Master Transmission Interrupt (2)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0, ACK Response)



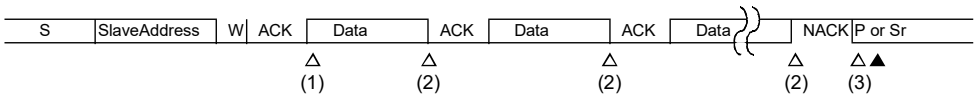
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission
- Write INT = "0" after the transmission data is written to the TDR register
(3) An interrupt generated by 1 byte transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 40-67. Master Transmission Interrupt (3)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0, NACK Response)



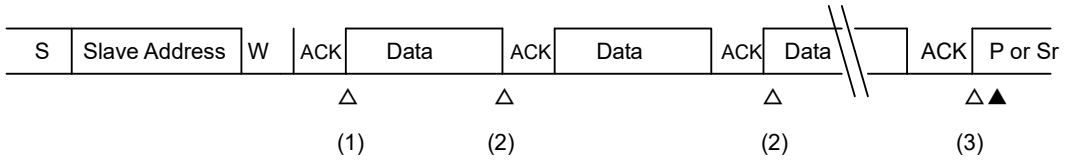
S: Start condition
W: Data direction bit (Write direction)
P: Stop condition Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission
- Write INT = "0" after the transmission data is written to the TDR register
(3) An interrupt generated by 1 byte transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 40-68. Master Transmission Interrupt (4)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0, Intermediate NACK Response)



S: Start condition
W: Data direction bit (Write direction)
P: Stop condition Sr: Repeated start condition
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register
(2) An interrupt generated by 1 byte transmission
- Write INT = "0" after the transmission data is written to the TDR register
(3) An interrupt generated by NACK response
- Set MSS = "0" or MSS = "1" and SCC = "1"
*: The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 40-69. Master Transmission Interrupt (5)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=1->0, IBSR:RSA=0, ACK Response)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

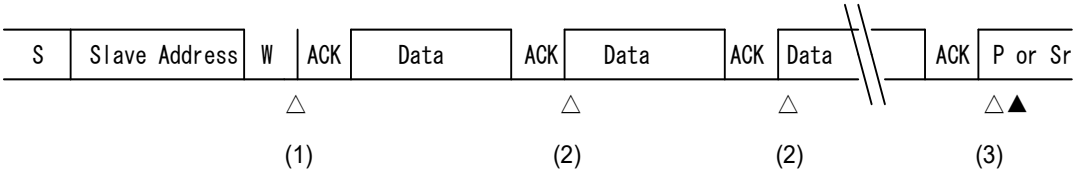
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register

(2) An interrupt generated by 1 byte transmission
- Write WSEL="0", INT = "0" after the transmission data is written to the TDR register

(3) An interrupt generated by 1 byte transmission
- Set MSS = "0" or MSS = "1" and SCC = "1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 40-70. Master Transmission Interrupt (6)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=0, IBSR:RSA=1)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

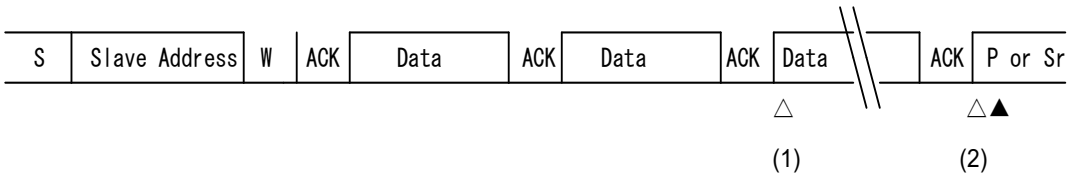
(1) An interrupt generated by slave address (reserved address) transmission + direction bit transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register

(2) An interrupt generated by 1 byte transmission + acknowledgment reception
- Write INT = "0" after the transmission data is written to the TDR register

(3) An interrupt generated by 1 byte transmission + acknowledgment reception
- Set MSS = "0" or MSS = "1" and SCC = "1"

*) The TDRE bit is "1" upon the generation of the interrupt flag (INT)

Figure 40-71. Master Transmission Interrupt (7)-When FIFO is Enabled (SSR:DMA=0, IBCR:WSEL=0, IBSR:RSA=0, ACK Response)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

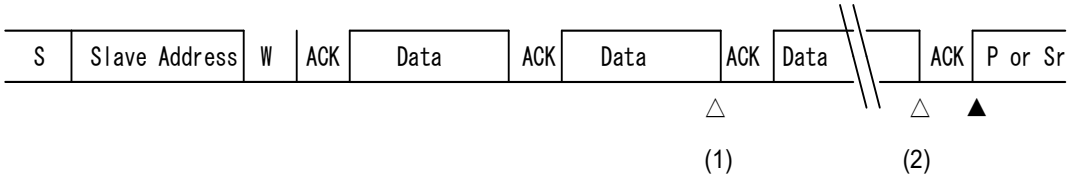
(1) An interrupt generated because the transmission FIFO is empty

- Write INT = "0" after the transmission data is written to the transmission FIFO

(2) An interrupt generated by the last byte transmission (transmission FIFO is empty) + acknowledgment reception

- Set MSS = "0" or MSS = "1" and SCC = "1"

Figure 40-72. Master Transmission Interrupt (8)-When FIFO is Enabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

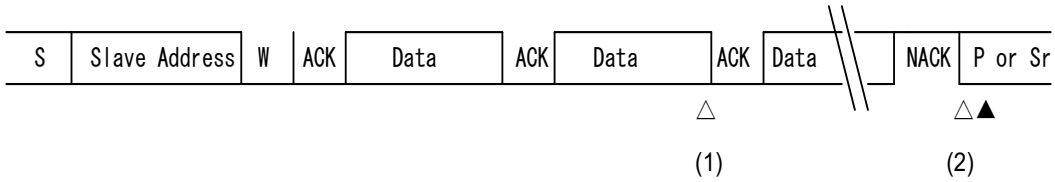
(1) An interrupt generated because the transmission FIFO is empty

- Write INT = "0" after the transmission data is written to the transmission FIFO

(2) An interrupt generated by the last byte transmission (transmission FIFO is empty)

- Set MSS = "0" or MSS = "1" and SCC = "1"

Figure 40-73. Master Transmission Interrupt (9)-When FIFO is Enabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0, NACK Response)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

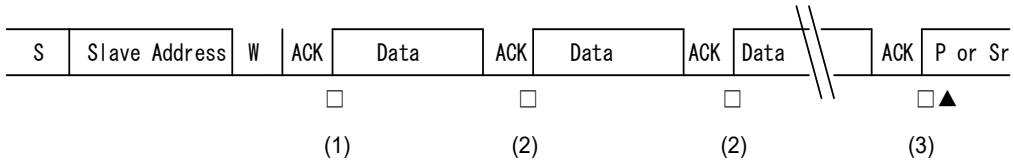
(1) An interrupt generated because the transmission FIFO is empty

- Write INT = "0" after the transmission data is written to the transmission FIFO

(2) An interrupt generated by NACK response

- Set MSS = "0" or MSS = "1" and SCC = "1"

Figure 40-74. Master Transmission Interrupt (10)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=0, IBSR:RSA=0)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

(1) An interrupt generated by slave address transmission ＋ direction bit transmission＋acknowledgment reception

－ the transmission data is written to the TDR register

(2) An interrupt generated by 1 byte transmission＋acknowledgment reception

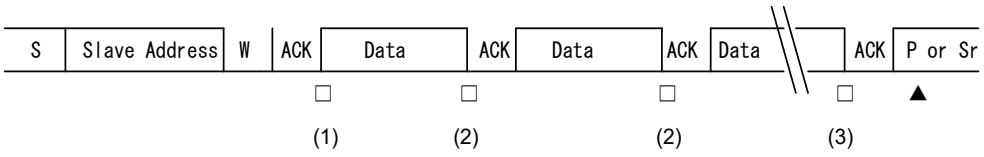
－ the transmission data is written to the TDR register

(3) An interrupt generated by 1 byte transmission＋acknowledgment reception

－ Sets MSS="0" or MSS="1" and SCC="1"

*) When interrupt flag(TBI) generated, TDRE bit is "1"

Figure 40-75. Master Transmission Interrupt (11)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0, ACK Response)



S : Start condition

W : Data direction bit(write direction)

P : Stop condition

Sr : Repeated start condition

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

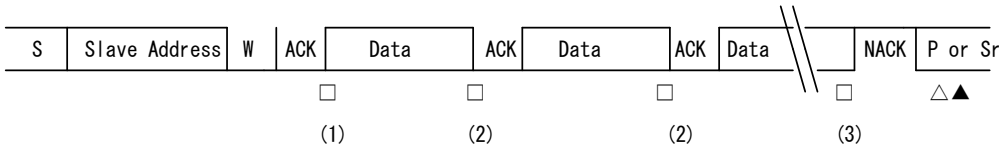
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- the transmission data is written to the TDR register

(2) An interrupt generated by 1 byte transmission
- the transmission data is written to the TDR register

(3) An interrupt generated by 1 byte transmission
- Sets MSS="0" or MSS="1" and SCC="1"

*) Interrupt flag (INT) generated, TDRE bit is "1"

Figure 40-76. Master Transmission Interrupt (12)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0, NACK Response)



S : Start condition

W : Data direction bit(Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

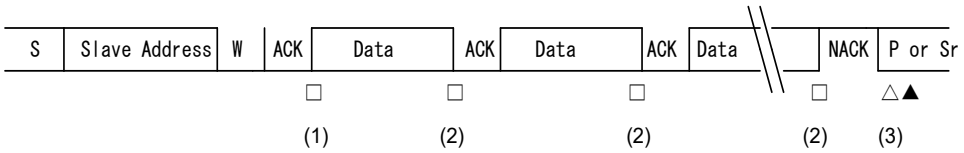
(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception
- The transmission data is written to the TDR register

(2) An interrupt generated by 1 byte transmission
- The transmission data is written to the TDR register

(3) An interrupt generated by 1 byte transmission
- Set MSS="0" or MSS="1" and SCC="1"

*) Interrupt flag (INT,TBI) generated, TDRE bit is"1"

Figure 40-77. Master Transmission Interrupt (13)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0, Intermediate NACK Response)



S : Start condition

W : Data direction bit(Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

— the transmission data is written to the TDR register

(2) An interrupt generated by 1 byte transmission

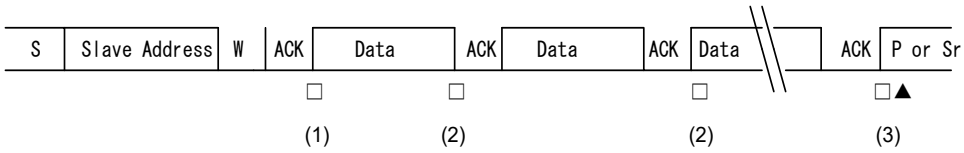
— the transmission data is written to the TDR register

(3) An interrupt generated by NACK response

— Sets MSS="0" or MSS="1" and SCC="1"

*) Interrupt flag(INT,TBI) generated, TDRE bit is "1"

Figure 40-78. Master Transmission Interrupt (14)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=1->0, IBSR:RSA=0, ACK Response)



S : Start condition

W : Data direction bit(Write direction)

P : Stop condition

Sr : Repeated start condition

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

(1) An interrupt generated by slave address transmission + direction bit transmission + acknowledgment reception

— The transmission data is written to the TDR register

(2) An interrupt generated by 1 byte transmission

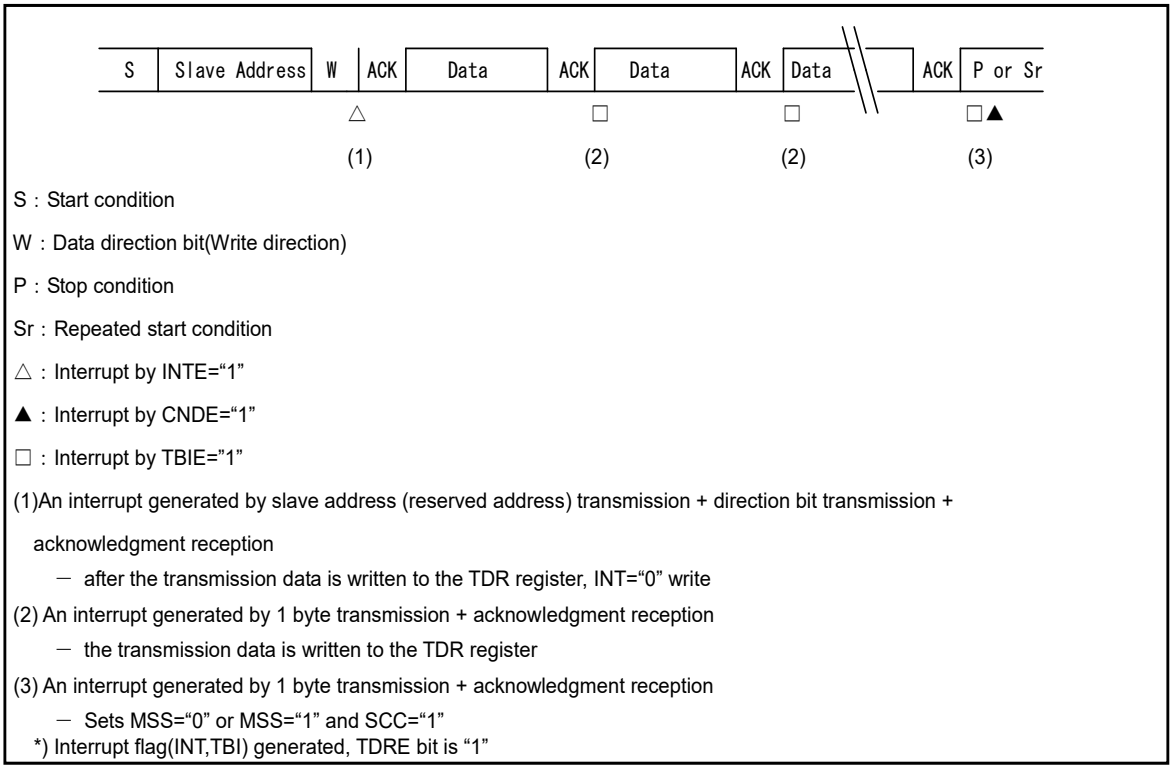
— The transmission data is written to the TDR register after write WSEL = "0"

(3) An interrupt generated by 1 byte transmission

— Sets MSS="0" or MSS="1" and SCC="1"

*) Interrupt flag(TBIE) generated, TDRE bit is "1"

Figure 40-79. Master Interrupt (15)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=0, IBSR:RSA=1)



S : Start condition

W : Data direction bit(Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

(1)An interrupt generated by slave address (reserved address) transmission + direction bit transmission + acknowledgment reception

- after the transmission data is written to the TDR register, INT="0" write

(2) An interrupt generated by 1 byte transmission + acknowledgment reception

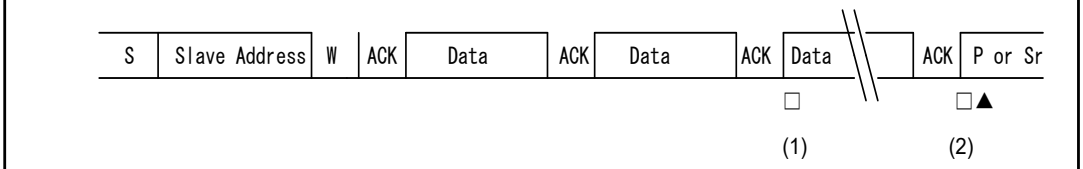
- the transmission data is written to the TDR register

(3) An interrupt generated by 1 byte transmission + acknowledgment reception

- Sets MSS="0" or MSS="1" and SCC="1"

*) Interrupt flag(INT,TBI) generated, TDRE bit is "1"

Figure 40-80. Master Transmission Interrupt (16)-When FIFO is Enabled (SSR:DMA=1, IBCR:WSEL=0, IBSR:RSA=0, ACK Response)



S : Start condition

W : Data direction bit(Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

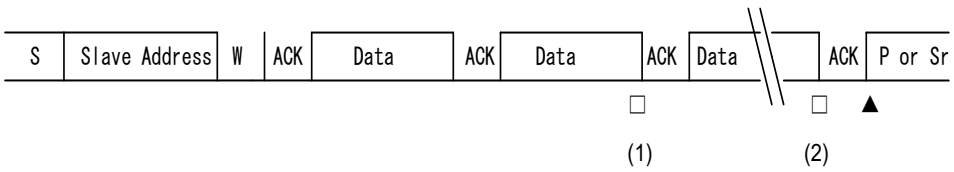
(1) An interrupt generated because the transmission FIFO is empty

- The transmission data is written to the transmission FIFO

(2) An interrupt generated by the last byte transmission (transmission FIFO is empty) + acknowledgment reception

- Sets MSS="0" or MSS="1" and SCC="1"

Figure 40-81. Master Transmission Interrupt (17)-When FIFO is Enabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0)



S : Start condition

W : Data direction bit(Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

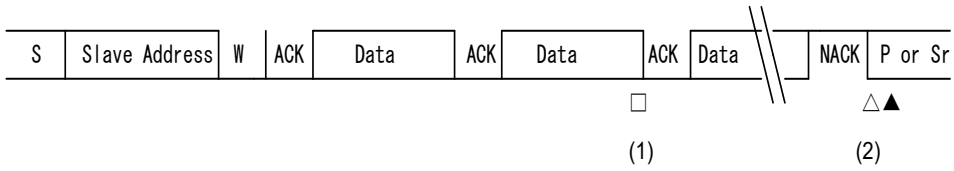
(1) An interrupt generated because the transmission FIFO is empty

- The transmission data is written to the transmission FIFO

(2) An interrupt generated by the last byte transmission (transmission FIFO is empty)

- Sets MSS="0" or MSS="1" and SCC="1"

Figure 40-82. Master Transmission Interrupt (18)-When FIFO is Enabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0, NACK Response)



S : Start condition

W : Data direction bit(Write direction)

P : Stop condition

Sr : Repeated start condition

△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TDRE="1"

(1) An interrupt generated because the transmission FIFO is empty

- The transmission data is written to the transmission FIFO

(2) An interrupt generated by NACK response

- Sets MSS="0" or MSS="1" and SCC="1"

40.8.3.5 Data Reception by Master System

Data reception by the master system is shown below.

- When DMA mode is disable (SSR:DMA=0)

  If the data direction bit (R/W) is "1", data sent from the slave device will be received.

  If FIFO operation is disabled and if the SSR:TDRE bit is set to "1", the master device will generate a wait (IBCR:INT=1, SSR:RDRF=1) each time it receives 1 byte of data. Also, the master device responds with ACK or NACK to the ACKE bit setting of IBCR register based on the IBCR:WSEL bit setting. If the SSR:TDRE bit is "0" and if the ACKE bit setting of IBCR register is responded with ACK, no wait will be generated (IBCR:INT=0) and the next data will be received. If responded with NACK, a wait is generated (IBCR:INT=1).

  If FIFO operation is enabled, the SSR:RDRF bit is set when the same number of bytes as the received data bytes is received. The interrupt flag is set if the SSR:TDRE bit is "1", and the I^2C bus is waited. If IBCR:WSEL=0 and if the SSR:TDRE bit is set to "1", a NACK response is made and the interrupt flag is set to "1". When IBCR:WSEL=1, a wait is generated after the last byte is received. Set the IBCR:ACKE bit during the wait, and then the ACK or NACK response is performed according to the setting for the IBCR:ACKE bit once the interrupt flag is cleared to "0". The received data is stored in the receive FIFO memory even if a NACK response is made.

  The following explains the waiting by interrupt.

Table 40-19. IBCR:WSEL Bit during Master Data Reception (When DMA Mode is Disable (SSR:DMA=0))

WSEL	Operation
0	In the 2nd or subsequent byte, the interrupt flag bit (IBCR:INT) is set to "1" and SCL is set to "L" to go into the wait state when the SSR:TDRE bit is "1" after the acknowledgment.
1	In the 2nd or subsequent byte, the interrupt flag bit (IBCR:INT) will be set to "1" and SCL will be set to "L" to go into the wait state when the SSR:TDRE bit is "1"after the master receives 1 byte data.

The following gives an example of procedure to receive data from the slave device.

☐ If the receive FIFO operation is disabled

1. Set the slave address (including the data direction bit) in the TDR register, and set the IBCR:MSS bit to "1".
2. Transmit the slave address and receive ACK. The interrupt flag (IBCR:INT) becomes "1".
3. Update the IBCR:WSEL bit and set the interrupt flag bit (IBCR:INT) to "0" to release the I2C bus from waiting state.
4. Put the I2C bus in a wait by setting the interrupt flag to "1", after transmitting an acknowledge upon the reception of one byte when IBCR:WSEL is set to "0", or immediately after one byte has been received when IBCR:WSEL is set to "1". Repeat Steps (3) to (4) until the specified number of data sets are received.
5. After reception of the last data, send a NACK response, set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" in order to generate a stop condition or a repeated start condition.

☐ If the send/receive FIFO operation is enabled

1. Set the receive data count to the FBYTE register.
2. Write the slave address (including the data direction bit) and the dummy data (for the receive data size) into the TDR register.
3. Set the IBCR:MSS bit to "1".
4. Respond with an ACK and continue data reception when the SSR:TDRE bit is kept "0". After receiving the specified bytes of data (set by FBYTE), set the SSR:RDRF bit to "1". When the SSR:RDRF bit is set to "1", read the RDR register.
5. If the SSR:TDRE bit is set to "1" and if IBCR:WSEL=0, send a NACK response. If IBCR: WSEL=1, set the interrupt flag to "1" immediately after 1 byte of data reception in order to wait the I^2C bus.
6. If IBCR:WSEL=1, set the IBCR:ACKE bit to "0". If IBCR:WSEL=0, the IBCR:ACKE bit needs not be set. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" in order to generate a stop condition or a repeated start condition.

- When DMA mode is enable (SSR:DMA=1)

  If the data direction bit (R/W) is "1", data sent from the slave device will be received.

  If FIFO operation is disabled and if the SSR:TDRE bit is set to "1", the master device will generate a wait (SSR:TBI=1, SSR:RDRF=1) each time it receives 1 byte of data. Also, the master device responds with ACK or NACK to the ACKE bit setting of IBCR register based on the IBCR:WSEL bit setting. If the SSR:TDRE bit is "0" and if the ACKE bit setting of IBCR register is responded with ACK, no wait will be generated and the next data will be received. If responded with NACK, a wait is generated (IBCR:INT=1).

  If FIFO operation is enabled, the SSR:RDRF bit is set when the same number of bytes as the received data bytes is received. The transmission bus idle flag (SSR:TBI) is set if the SSR:TDRE bit is "1", and the I^2C bus is waited. If IBCR:WSEL=0 and if the SSR:TDRE bit is set to "1" if it is NACK by the ACKE bit setting, a NACK response is made and the interrupt flag (IBCR:INT) and transmission bus idle flag (SSR:TBI) are set to "1". When IBCR:WSEL=1, a wait (SSR: TBI=1) is generated after the last byte is received. Set the IBCR:ACKE bit during the wait, and then the ACK or NACK response is performed according to the setting for the IBCR:ACKE bit once the transmission bus idle flag (SSR:TBI) is cleared. The received data is stored in the reception FIFO memory even if a NACK response is made.

  The following explains the waiting by interrupt.

Table 40-20. IBCR:WSEL Bit during Master Data Reception (When DMA Mode is Enabled (SSR:DMA=1))

WSEL	Operation
0	In the 2nd or subsequent byte, the transmission bus idle flag (SSR:TBI) is set to "1" and SCL is set to "L" to go into the wait state when the SSR:TDRE bit is "1" after the acknowledgment. In the 2nd or subsequent byte, the reception data full flag (SSR:RDRF) is set to "1" and SCL is set to "L" to go into the wait state when the reception FIFO is not used after the acknowledgment.
1	In the 2nd or subsequent byte, the interrupt flag (SSR:TBI) will be set to "1" and SCL will be set to "L" to go into the wait state when the SSR:TDRE bit is "1"after the master receives 1 byte data. In the 2nd or subsequent byte, the reception data full flag (SSR:RDRF) is set to "1" and SCL is set to "L" to go into the wait state after the data reception when the reception FIFO is not used after the acknowledgment.

The following gives an example of procedure to receive data from the slave device.

☐ If the reception FIFO operation is disabled

1. Set the slave address (including the data direction bit) in the TDR register, and set the IBCR:MSS bit to "1".
2. Transmit the slave address and receive ACK. The transmission bus idle flag (SSR:TBI) becomes "1".
3. The data transmitted to the TDR register is written, and release the I²C bus from waiting state.
4. Put the I²C bus in a wait by setting the transmission bus idle flag (SSR:TBI) and reception data full flag (SSR:RDRF)[2] to "1", after transmitting an acknowledge upon the reception of one byte when IBCR:WSEL is set to "0", or immediately after one byte has been received when IBCR:WSEL is set to "1".
5. Update the IBCR:WSEL bit and RDR register is read, and the data of the dummy is written in the TDR register.
6. Put the I²C bus in a wait by setting the transmission bus idle flag (SSR:TBI) and reception data full flag (SSR:RDRF)[2] to "1", after transmitting an acknowledge upon the reception of one byte when IBCR:WSEL is set to "0", or immediately after one byte has been received when IBCR:WSEL is set to "1". Repeat Steps (5) to (6) until the specified number of data sets are received.
7. After reception of the last data, send a NACK response, set the IBCR:MSS bit to "0" or set the IBCR: SCC bit to "1" in order to generate a stop condition or a repeated start condition.

☐ If the transmission/reception FIFO operation is enabled

1. Set the reception data count to the FBYTE register.
2. Write the slave address (including the data direction bit) and the dummy data (for the reception data size) into the TDR register.
3. If IBCR:WSEL=0, it makes to NACK by setting the ACKE bit, and "1" is written in the IBCR:MSS bit.
4. Respond with an ACK and continue data reception when the SSR:TDRE bit is kept "0". After receiving the specified bytes of data (set by FBYTE), set the SSR:RDRF bit to "1". When the SSR:RDRF bit is set to "1", read the RDR register.
5. If the SSR:TDRE bit is set to "1" and if IBCR:WSEL=0, send a NACK response the interrupt flag is set as "1" in order to wait the I2C bus. If IBCR:WSEL=1, set the transmission bus idle flag (SSR:TBI) to "1" immediately after 1 byte of data reception in order to wait the I²C bus.
6. If IBCR:WSEL=1, set the IBCR:ACKE bit to "0". If IBCR:WSEL=0, the IBCR:ACKE bit needs not be set. Set the IBCR:MSS bit to "0" or set the IBCR:SCC bit to "1" in order to generate a stop condition or a repeated start condition.

[1]: The DMA mode must write the slave address in TDR after confirming the IBCR:INT bit is set in "1" after "1" is written in the IBCR:INT bit, and set "1" to the IBCR:SCC bit if you issues the repetition start condition when the DMA mode is enabled (SSR:DMA=1), also the SSR:TBI bit is set as "1" and the INT bit is set as "0".

[2]: Reception data full flag (SSR:RDRF) is set in "1" regardless of the setting of IBCR:WSEL immediately after the reception one byte. When the reception data full flag is set as "1" since the second byte, if the IBCR:WSEL bit is 0, after acknowledge transmission also IBCR:WSEL=1, I²C bus is waited at one byte reception.

Notes:

- If the 7-bit slave address detection is enabled (ISBA:SAEN=1), you cannot specify a 7-bit slave address in the master mode.
- If SSR:TDRE is "0", an acknowledge signal will be sent based on the IBCR:ACKE bit setting and the subsequent process will be executed even if an overrun error occurs.
- If you need to change the IBCR register during data sending or receiving, change it only when the interrupt flag (IBCR:INT) is "1" or when the transmission bus idle flag (SSR:TBI1) is "1" during the DMA mode is enabled (SSR:DMA=1).
- When the master device is receiving data and if the DMA mode is disabled (SSR:DMA=0), dummy data is written in the TDR register and if the SSR:TDRE bit is set as "0" at the interrupt flag (IBCR:INT) is set to "1", the interrupt flag (IBCR:INT) will be held to "0" and the next data will be received.
- When the master device is receiving data and if the DMA mode is enable (SSR:DMA=1), dummy data is written in the TDR register and if the SSR:TDRE bit is set as "0" at the transmission bus idle flag (SSR:TBI) is set to "1", the transmission bus idle flag (SSR:TBI) will be held to "0" and the next data will be received.
- If data is received when the reception FIFO is enabled and IBCR:WSEL=0, the SSR:RDRF bit becomes "1" after the last bit is received and the interrupt flag (IBCR:INT) becomes "1" after ACK is transmitted.

Figure 40-83. Master Reception Interrupt (1)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=0, IBSR:RSA=0)

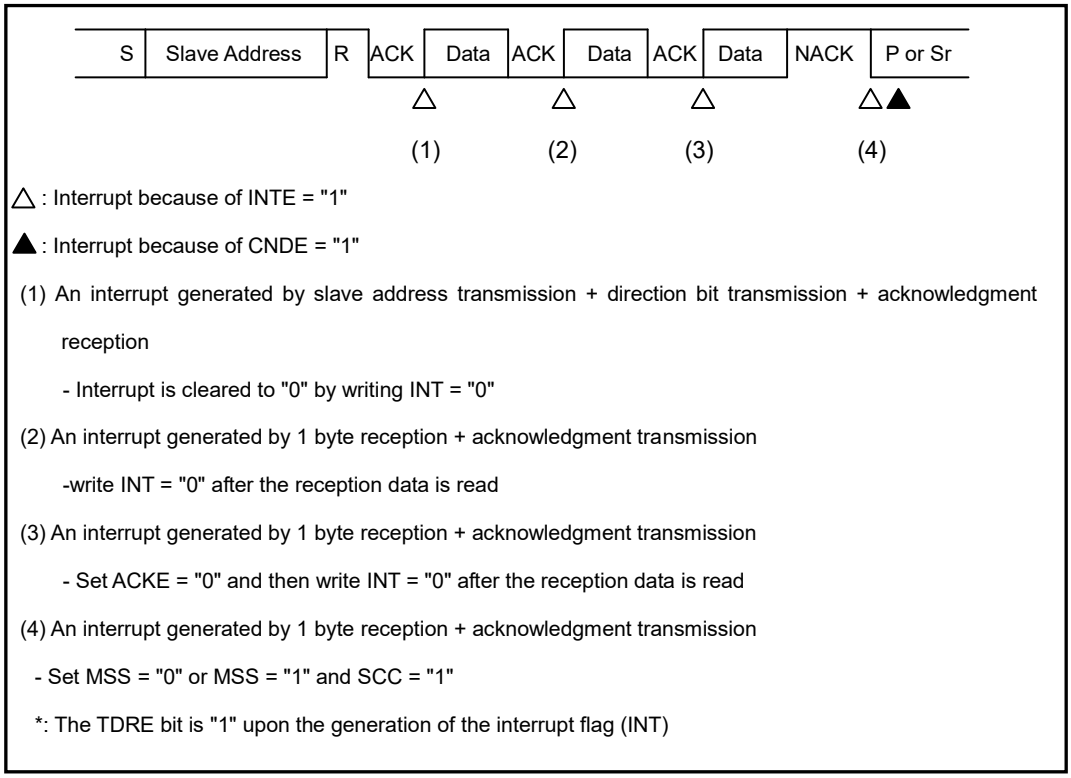
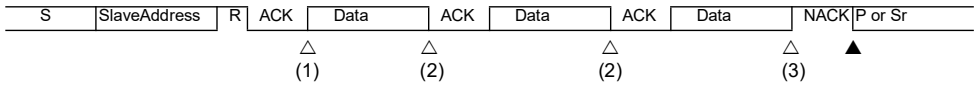


Figure 40-84. Master Reception Interrupt (2)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0)



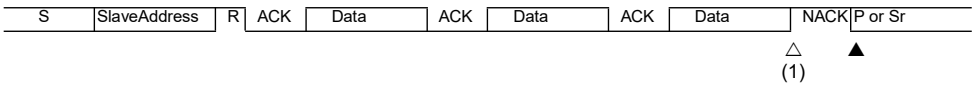
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
(1) Generation of interrupt by slave address transmission, direction bit transmission and acknowledgement reception:
Clears interrupt to "0" after writing "0" to INT
(2) Generation of interrupt by 1-byte reception.
Writes "0" to INT after reading reception data
(3) Generation of interrupt by 1-byte reception
Set "0" to ACKE and sets MSS="0" or MSS="1" and SCC="1" after reading reception data
*) TDRE bit is "1" when the interrupt flag (INT) is set

Figure 40-85. Master Reception Interrupt (3)-when FIFO is Enabled (SSR:DMA=0, IBCR:WSEL=0, IBCR:ACKE=0, IBSR:RSA=0)



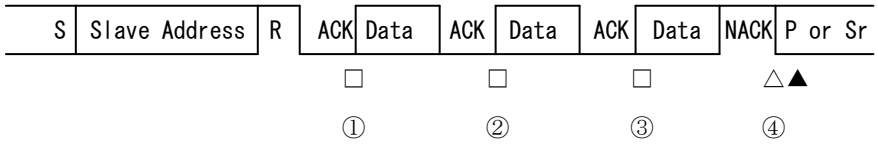
△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
(1) Generation of interrupt by TDRE="1"
Set MSS="0" or MSS="1" and SCC="1" after reading all data from reception FIFO

Figure 40-86. Master Reception Interrupt (4)-When FIFO is Enabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0)



△ : Interrupt by INTE="1"
▲ : Interrupt by CNDE="1"
(1) Generation of interrupt by TDRE="1"
Set "0" to ACKE and set MSS="0" or MSS="1" and SCC="1" after reading all data from reception FIFO

Figure 40-87. Master Reception Interrupt (5)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=0, IBSR:RSA=0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

(1) An interrupt generated by slave address transmission＋direction bit transmission＋acknowledgment reception

— The data of the dummy is written in the TDR register.

(2)An interrupt generated by 1byte reception＋acknowledgment transmission

— The data of the dummy is written in the TDR register after reading reception data

(3) An interrupt generated by 1byte reception＋acknowledgment transmission

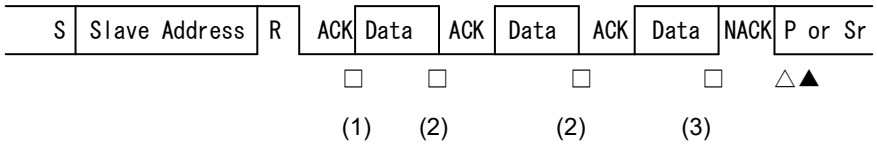
— The data of the dummy is written in the TDR register after reading reception data and set ACKE="0"

(4) An interrupt generated by 1byte reception＋acknowledgment transmission

— Set MSS="0" or MSS="1" and SCC="1"

*) When the Interrupt flag(INT,TBI) is generated, the TDRE bit is"1"

Figure 40-88. Master Reception Interrupt (6)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

□ : Interrupt by TBIE="1"

(1) Generation of interrupt by slave address transmission＋direction bit transmission＋acknowledgment reception

— The data of the dummy is written in the TDR register.

(2) An interrupt generated by 1 byte reception

— The data of the dummy is written in the TDR register after reading reception data

(3) An interrupt generated by 1 byte reception

— Set ACKE="0", MSS="0" or MSS="1" and SCC="1" after reception data reading

*) When the Interrupt flag(INT,TBI) is generated, the TDRE bit is "1"

Figure 40-89. Master Reception Interrupt (7)-When FIFO is Enabled (SSR:DMA=1, IBCR:WSEL=0, IBCR:ACKE=0, IBSR:RSA=0)

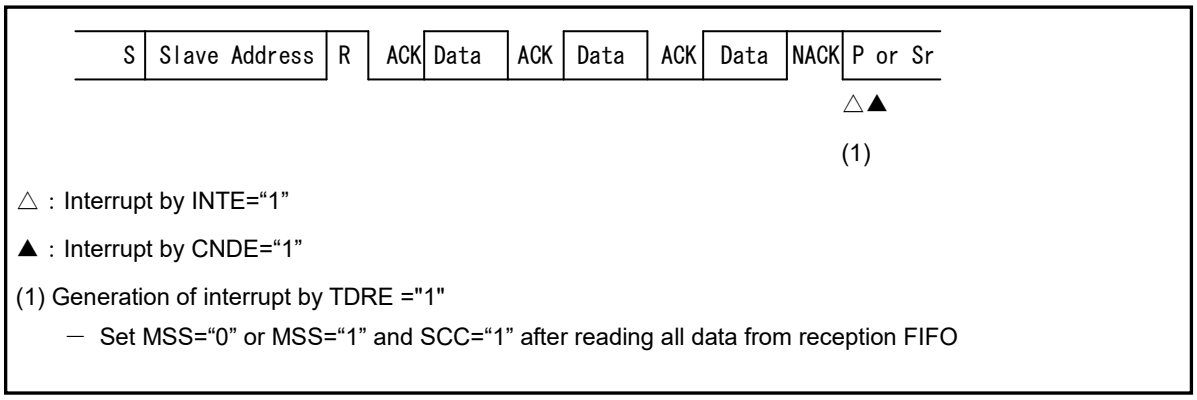
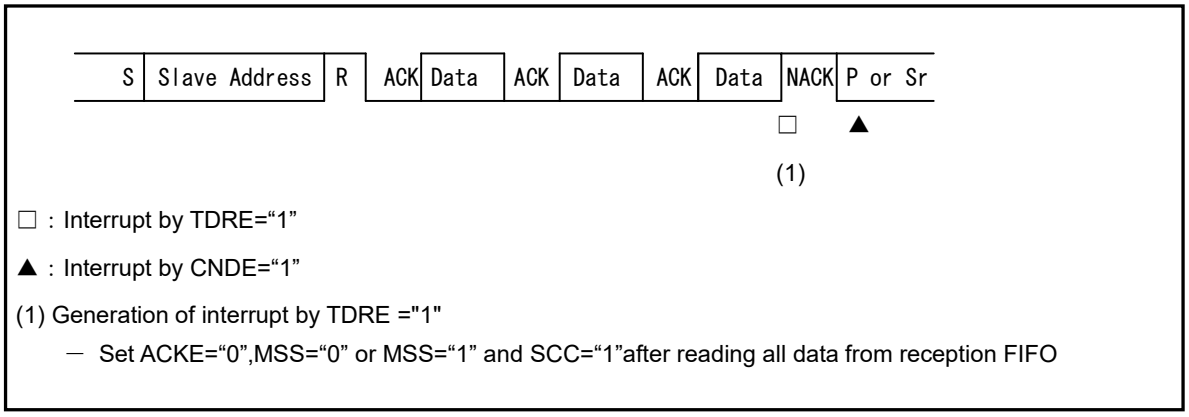


Figure 40-90. Master Reception Interrupt (8)-When FIFO is Enabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0)



40.8.3.6 Arbitration Lost

The arbitration lost is shown below.

If data of the master device is hit by data from another master device and if the data that is different from the transmitted data is received, it will be determined to be an arbitration lost. The IBCR:MSS bit is set to "0" and the IBSR:AL bit is set to "1" so that the device can operate in the slave mode. The IBSR:AL bit is cleared to "0" if:

- The IBCR:MSS bit is set to "1"
- The IBCR:INT bit is set to "0".
- The IBSR:SPC bit is set to "0" when IBSR:AL=1 and IBSR:SPC=1.
- The I^2C interface is disabled (ISMK:EN bit=0).

If an arbitration lost occurs, the interrupt flag (IBCR:INT) will be set to "1" and the SCL of I^2C bus will be set to "L" based on the IBCR:WSEL setting.

40.8.3.7 Wait of the Master Mode

Wait of the master mode is shown below.

When the IBSR:BB bit is "1" and the IBCR:MSS bit is set to "1", if the slave mode is not operated, wait the master mode as long as the IBSR:BB bit is "1". The start condition is transmitted when the IBSR:BB bit becomes "0". You can judge whether the master mode is in wait state or not using IBCR:MSS and IBCR:ACT bits (in wait state if IBCR:MSS=1 and IBCR:ACT=0). To operate as the slave mode after the IBCR:MSS bit is set to "1", set the IBSR:AL and IBCR:ACT bits to "1" and the IBCR:MSS bit to "0".

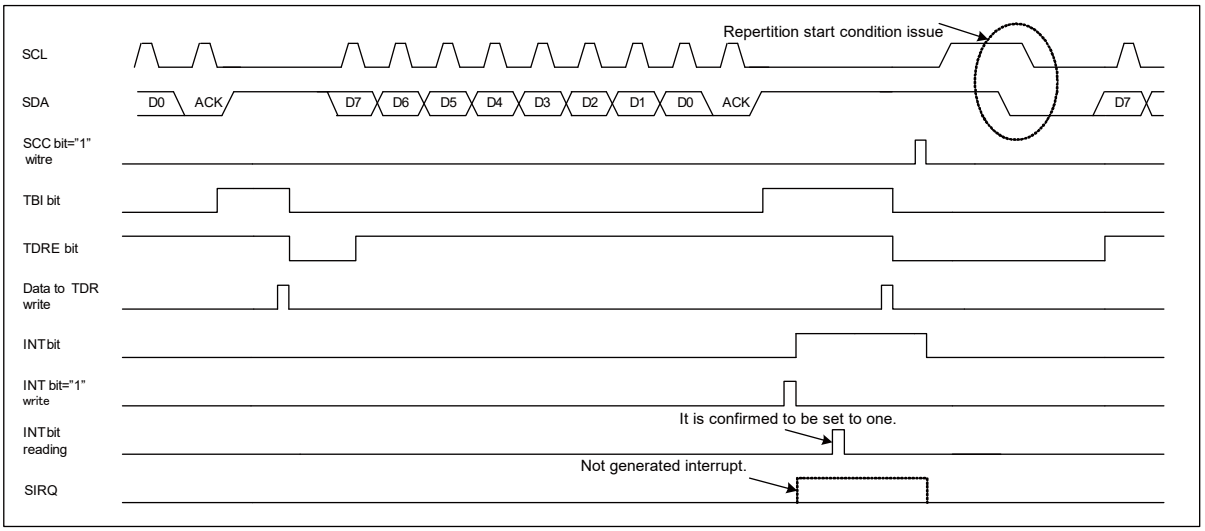
40.8.3.8 Repetition Start Condition Issue When DMA Mode Enabled (SSR:DMA=1)

The repetition start condition issue when DMA mode is enabled (SSR:DMA=1) is shown below.

The transmission operation begins when the slave address is written in the TDR register when the transmission bus idle is in progress (SSR:TBI=1) and interrupt flag (IBCR:INT) is "0", and it is impossible to issue the repetition start condition.

Therefore, if the repetition start condition is issued when the transmission bus idle is in progress (SSR:TBI=1) and the interrupt flag (IBCR: INT) is "0", it is confirmed that "1" is written in the interrupt flag (IBCR:INT) first, and the interrupt flag (IBCR: INT) is set in "1" afterwards. At this time, the SIRQ interrupt is not generated. Next, the slave address is written in the TDR register, and the repetition start is issued afterwards (IBCR: SCC=1).

Figure 40-91. Repetition Start Condition Issue When DMA Mode is Enabled (SSR:DMA=1, IBCR:WSEL=0, IBSR:RSA=0, ACK Response)



40.8.4 I²C Slave Mode

The I²C Slave mode is shown below.

If the (repeated) start condition is detected and if a combination of ISBA register and ISMK register settings match the received address, an ACK response is sent and the slave mode operation starts.

Note:

- If a start condition is detected again in the following cases, next address data cannot be received because reception of the data is discontinued due to the bus error being detected (IBCR:BER=1):
  - ☐ During the time that address data is transmitted after the start condition is detected.
  - ☐ During the time that data of bit2 to bit9 (acknowledge bits) are transmitted.
- In both cases, it is necessary to send a start condition again from the master after the interrupt flag (IBCR:INT) is cleared.

40.8.4.1 Detection of Slave Address Matching

Detection of slave address matching is shown below.

When the (repeated) start condition is detected, the 7 bits of the next data are received as the address. If the bit is set to "1" in the ISMK register, it is compared with each bit of the ISBA register and the received address. If they match, an ACK signal is output.

Table 40-21. Operations Immediately after Slave Address Acknowledgment

Transmission FIFO	Reception FIFO	Transmission FIFO Status	Reception FIFO Status	Data Direction bit (R/W)	Operation Immediately after Acknowledgement	
					Acknowledge is ACK	Acknowledge is NACK
Disabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
				1		
Disabled	Enabled	-	Without data	0	The IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	
Enabled	Disabled	-	-	0	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
				1		
Enabled	Enabled	-	Without data	0	The IBCR:INT bit is held to "0" and not waited.	The IBCR:INT bit is held to "0" and not waited.
			With data		The IBCR:INT bit is set to "1" and waited.	
			-	1	If the SSR:TDRE bit is "1", the IBCR:INT bit is set to "1" and waited. If the SSR:TDRE bit is "0", the IBCR:INT bit is held to "0" and not waited.	

- Reserved address detected

  If the first byte already matches the reserved address ("0000xxxx" or "1111xxxx"), the IBCR:INT bit is set to "1" and the I^2C bus is waited after receiving the 8-th bit of data. These operations are not associated with a permission of transmission or reception FIFO operation. During this time, if you read the received data and operate the device as a slave one, set the IBCR:ACKE bit to "1" and check the data direction bit (IBSR:TRX). If it is the transmission direction, write the transmission data in the TDR register and clear the IBCR:INT bit. Then, the device operates as a slave one. If the IBCR:ACKE bit is set to "0", the device does not operate as a slave device after acknowledgement.

40.8.4.2 Data Direction Bit

The data direction bit is shown below.

After the address reception, a data direction bit that determines the data transmission or reception is received. If this bit is "0", it shows the data transmission from the master device while this device receives data as a slave device.

40.8.4.3 Reception by Slave Device

Data reception by the slave device is shown below.

If the slave address matches and if the data direction bit is "0", it indicates the data reception in the slave mode. The following gives an example of procedure of data reception in the slave mode.

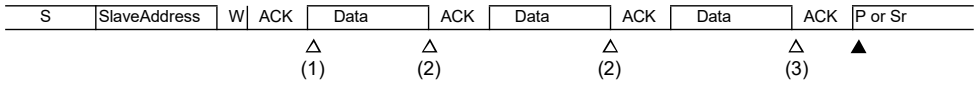
- When DMA mode is disable (SSR:DMA=0)
  - ☐ If the reception FIFO operation is disabled
    1. After sending an ACK signal, set the interrupt flag (IBCR:INT) to "1" to wait the I²C bus. You can determine the interrupt occurrence due to slave address matching shown by IBCR:MSS bit, IBCR:ACT bit and IBSR:FBT bit. Set the IBCR:ACKE bit to "1" and set the interrupt flag (IBCR:INT) to "0" to release the I²C bus from the waiting state. See Table 40-21.
    2. After receiving one byte of data, set the interrupt flag (IBCR:INT) to "1" based on the IBCR:WSEL setting, and wait the I²C bus.
    3. Read the received data from the RDR register, set the IBCR:ACKE bit, and set the interrupt flag (IBCR:INT) to "0" to release the I²C bus from the waiting state.
    4. Repeat Steps (2) and (3) until the stop condition or the repeated start condition is detected.
  - ☐ If the reception FIFO operation is enabled
    1. When a NACK signal is detected or when the reception FIFO memory is full, the interrupt flag (IBCR:INT) is set to "1" and the I2C bus is waited. When the stop condition or the repeated start condition is detected, the IBSR:SPC bit and IBSR:RSC bit are set to "1" but the interrupt flag (IBCR:INT) is not set to "1" (and the I2C bus is not waited). If the value set in the FBYTE register matches the number of received data, the reception FIFO sets the SSR:RDRF bit to "1". During this time, if the SMR:RIE bit is "1", a reception interrupt occurs.
    2. If the interrupt flag (IBCR:INT) is set to "1", the received data is read from the RDR register. After reading all data sets, set the interrupt flag to "0" and release the I2C bus from the waiting state. When the stop condition or the repeated start condition is detected, all of the received data sets are read from the RDR register, and the IBSR:SPC bit or IBSR:RSC bit is cleared to "0".
- When DMA mode is enable (SSR:DMA=1)
  - ☐ If the reception FIFO operation is disabled
    1. After sending an ACK signal, set the interrupt flag (IBCR:INT) to "1" to wait the I2C bus. You can determine the interrupt occurrence due to slave address matching shown by IBCR:MSS bit, IBCR:ACT bit and IBSR:FBT bit. Set the IBCR:ACKE bit to "1" and set the interrupt flag (IBCR: INT) to "0" to release the I2C bus from the waiting state. See "Table 8-9 Operations Immediately After Slave Address Acknowledgment".
    2. Reception data full flag (SSR:RDRF) is set in "1" immediately after the reception 1 byte after the data of 1 byte is received. The I2C bus is waited for IBCR:WSEL=1 immediately after the reception 1 byte after the acknowledge is transmitted for IBCR:WSEL=0 in the place where reception data full flag (SSR:RDRF) is set in "1".
    3. Reading the data received from the RDR register after the IBCR:ACKE bit is set clears the reception data full flag (SSR:RDRF) to "0" and the I2C bus from the waiting state is released.
    4. Repeat Steps (2) and (3) until the stop condition or the repeated start condition is detected.
  - ☐ If the reception FIFO operation is enabled
    1. The interrupt flag (IBCR:INT) becomes "1" and waits for the I²C bus by detecting NACK. The I²C bus is waited for when reception FIFO becomes full. The IBSR:SPC bit and the IBSR:RSC bit are made "1" when the stop condition and the repetition start condition are detected and the interrupt flag (IBCR:INT) does not become "1" (none waiting of the I²C bus). When a set value of the FBYTE register is corresponding to the received number of data, reception FIFO makes the SSR:RDRF bit "1". When the SMR:RIE bit is "1" at that time, the reception interrupt is generated.
    2. If the interrupt flag (IBCR:INT) is set to "1", the received data is read from the RDR register. After reading all data sets, set the interrupt flag to "0" and release the I2C bus from the waiting state. If the data received from the RDR register even once is read when reception FIFO becomes full, the I²C bus from the waiting state is released. When the stop condition or the repeated start condition is detected, all of the received data sets are read from the RDR register, and the IBSR:SPC bit or IBSR:RSC bit is cleared to "0".

Figure 40-92. Slave Reception Interrupt (1)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=0, IBSR:RSA=0)



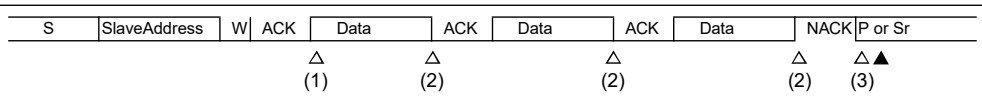
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT = "0"
(2) An interrupt generated by 1 byte reception + ACK response
- Read the reception data from the reception buffer, and write INT = "0"
(3) An interrupt generated by 1 byte reception + NACK response
- Read the reception data from the reception buffer, and then write INT = "0"

Figure 40-93. Slave Reception Interrupt (2)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0)



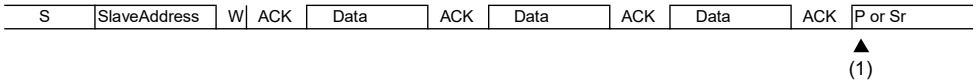
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT = "0"
(2) An interrupt generated by 1 byte reception
- Read the reception data from the reception buffer, and write INT = "0"
(3) An interrupt generated by 1 byte reception
- Read the reception data from the reception buffer, and write INT = "0"

Figure 40-94. Slave Reception Interrupt (3)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=1, IBSR:RSA=0)



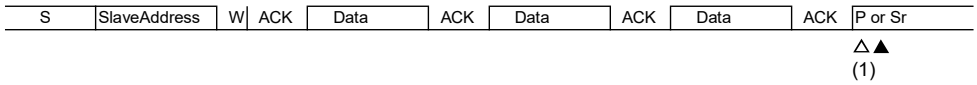
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by ACK output due to the match with the slave address
- Write ACKE="1", INT = "0"
(2) An interrupt generated by 1 byte reception
- Read the reception data from the reception buffer, and write INT = "0"
(3) An interrupt generated by NACK response
- Write INT = "0"

Figure 40-95. Slave Reception Interrupt (4)-When FIFO is Enabled (SSR:DMA=0, IBSR:RSA=0)



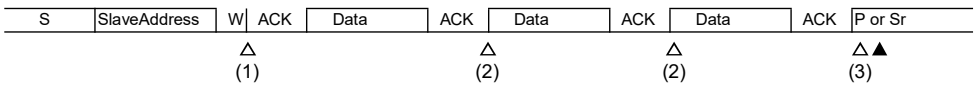
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by the detection of a stop condition or a repeated start condition
- Read all data from the reception FIFO

Figure 40-96. Slave Reception Interrupt (5)-When FIFO is Enabled (SSR:DMA=0, IBSR:RSA=0)



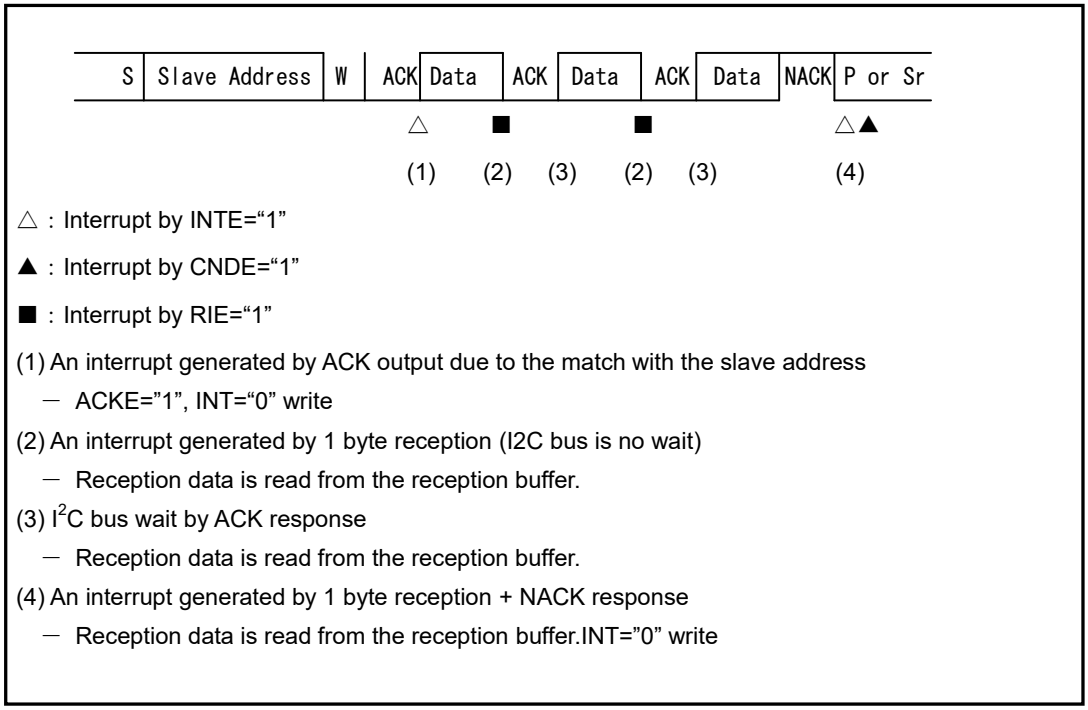
△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated because the reception FIFO gets full
- Read all data from the reception FIFO, and write INT = "0"

Figure 40-97. Slave Reception Interrupt (6)-When FIFO is Disabled (SSR:DMA=0, IBCR:WSEL=0, IBSR:RSA=1)



△: Interrupt because of INTE = "1"
▲: Interrupt because of CNDE = "1"
(1) An interrupt generated by matching the reserved address ("0000XXXX" or "1111XXXX")
- Read the reception data, and write ACKE = "1" and INT = "0"
(2) An interrupt generated by 1 byte reception + acknowledgment output
- Write INT = "0"
(3) An interrupt generated by 1 byte reception + acknowledgment output
- Interrupt by writing INT = "0"

Figure 40-98. Slave Reception Interrupt (7)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=0, IBSR:RSA=0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

■ : Interrupt by RIE="1"

(1) An interrupt generated by ACK output due to the match with the slave address

— ACKE="1", INT="0" write

(2) An interrupt generated by 1 byte reception (I2C bus is no wait)

— Reception data is read from the reception buffer.

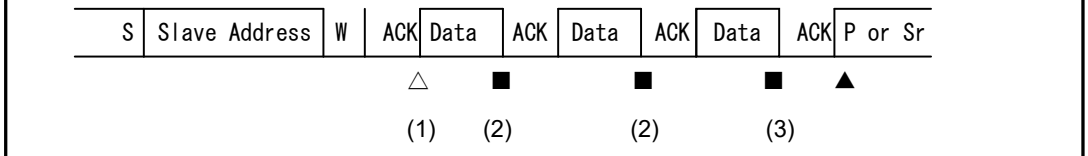
(3) I^2C bus wait by ACK response

— Reception data is read from the reception buffer.

(4) An interrupt generated by 1 byte reception + NACK response

— Reception data is read from the reception buffer.INT="0" write

Figure 40-99. Slave Reception Interrupt (8)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

■ : Interrupt by RIE="1"

(1) An interrupt generated by ACK output due to the match with the slave address

— ACKE="1", INT="0" write

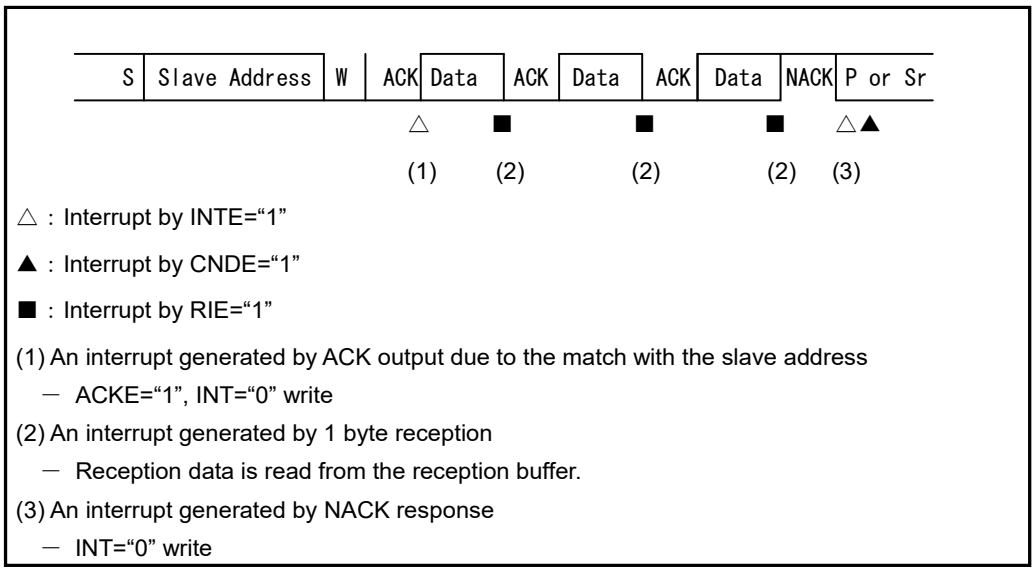
(2) An interrupt generated by 1 byte reception

— Reception data is read from the reception buffer.

(3) An interrupt generated by 1 byte reception

— Reception data is read from the reception buffer.

Figure 40-100. Slave Reception Interrupt (9)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=1, IBSR:RSA=0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

■ : Interrupt by RIE="1"

(1) An interrupt generated by ACK output due to the match with the slave address

— ACKE="1", INT="0" write

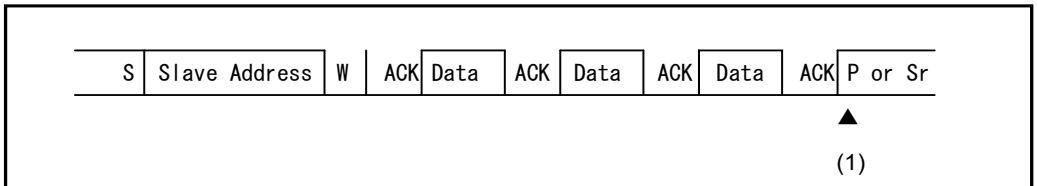
(2) An interrupt generated by 1 byte reception

— Reception data is read from the reception buffer.

(3) An interrupt generated by NACK response

— INT="0" write

Figure 40-101. Slave Reception Interrupt (10)-When Reception FIFO is Enabled (SSR:DMA=1, IBSR:RSA=0)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

(1) An interrupt generated by the detection of a stop condition or a repeated start condition

— Read all data from reception FIFO

Figure 40-102. Slave Reception Interrupt (11)-When Reception FIFO is Enabled (SSR:DMA=1, IBSR:RSA=0)



▲ : Interrupt by CNDE="1"

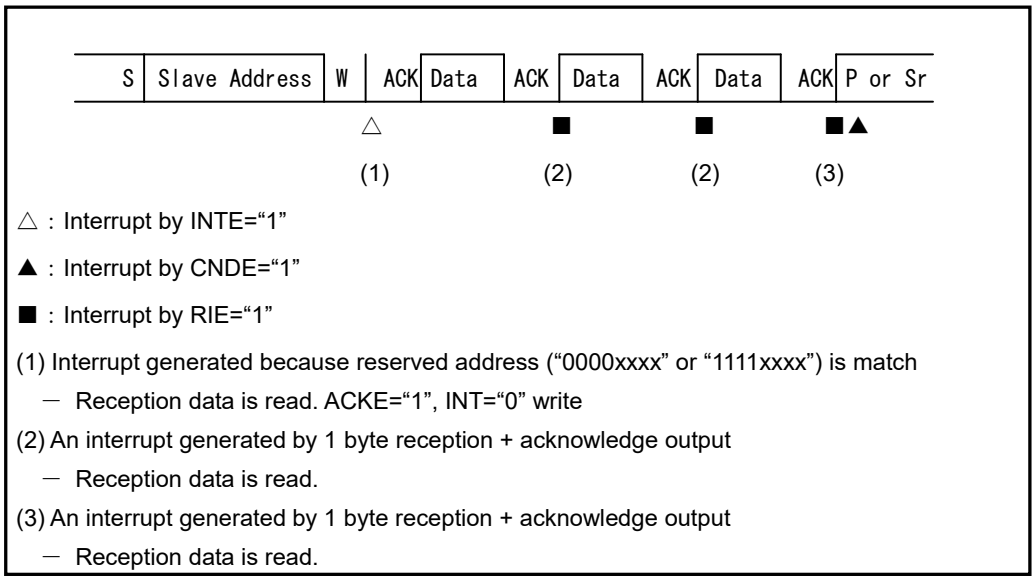
(1) The I2C bus wait according to becoming of reception FIFO Full.

— Releases waiting if it reads out data from reception FIFO even once.

(2) An interrupt generated by the detection of a stop condition or a repeated start condition

— Read all data from reception FIFO

Figure 40-103. Slave Reception Interrupt (12)-When FIFO is Disabled (SSR:DMA=1, IBCR:WSEL=0, IBSR:RSA=1)



△ : Interrupt by INTE="1"

▲ : Interrupt by CNDE="1"

■ : Interrupt by RIE="1"

(1) Interrupt generated because reserved address ("0000xxxx" or "1111xxxx") is match

— Reception data is read. ACKE="1", INT="0" write

(2) An interrupt generated by 1 byte reception + acknowledge output

— Reception data is read.

(3) An interrupt generated by 1 byte reception + acknowledge output

— Reception data is read.

40.8.4.4 Transmission by Slave Device

Data transmission by the slave device is shown below.

If the slave address matches and if the data direction bit is "1", it indicates the data transmission in the slave mode. If the FIFO operation is disabled, the interrupt flag (IBCR:INT) is set to "1" and a wait is generated based on the IBCR:WSEL setting after sending one byte of data or after acknowledgement. (See Table 40-21).

The IBSR:RACK bit is used to check an acknowledgement by the master device. If the master returns a NACK response, it indicates that the master has failed to receive data or the data reception has completed. If a NACK signal is detected when the IBCR:WSEL bit is "1", an interrupt will occur and wait will be generated.

40.8.5 Bus Error

The bus error is shown below.

When the stop condition and the start (repetition) condition are detected while sending and receiving data on the I²C bus, it is treated as an bus error.

40.8.5.1 Bus Error Generation Condition

The bus error generation condition is shown below.

The bus error makes the IBCR:BER bit "1" on the following conditions.

- The start (repetition) condition or the stop condition is detected while forwarding the first byte.
- The start (repetition) condition or the stop condition is detected by 2nd to 9th (acknowledge) bit of data.

40.8.5.2 Bus Error Operation

The bus error Operation is shown below.

Confirm the IBCR:BER bit when the interrupt flag (IBCR:INT) by sending and receiving becomes "1", and do error processing when the IBCR:BER bit is "1". The IBCR:BER bit is cleared by writing "0" in the IBCR:INT bit.

SCL of the I²C bus is made "L" and it doesn't do to the wait state though the IBCR:INT bit is set in "1" by the bus error.

40.8.6 Example of I²C Flowchart

The example of I²C Flowchart is shown below.

40.8.6.1 Example of I²C Flowchart (FIFO Memory Not Used) (When DMA Mode is Disable (SSR: DMA=0))

Figure 40-104. Example of I²C Flowchart (FIFO Memory Not Used) (When DMA Mode is Disable (SSR: DMA=0))1/3

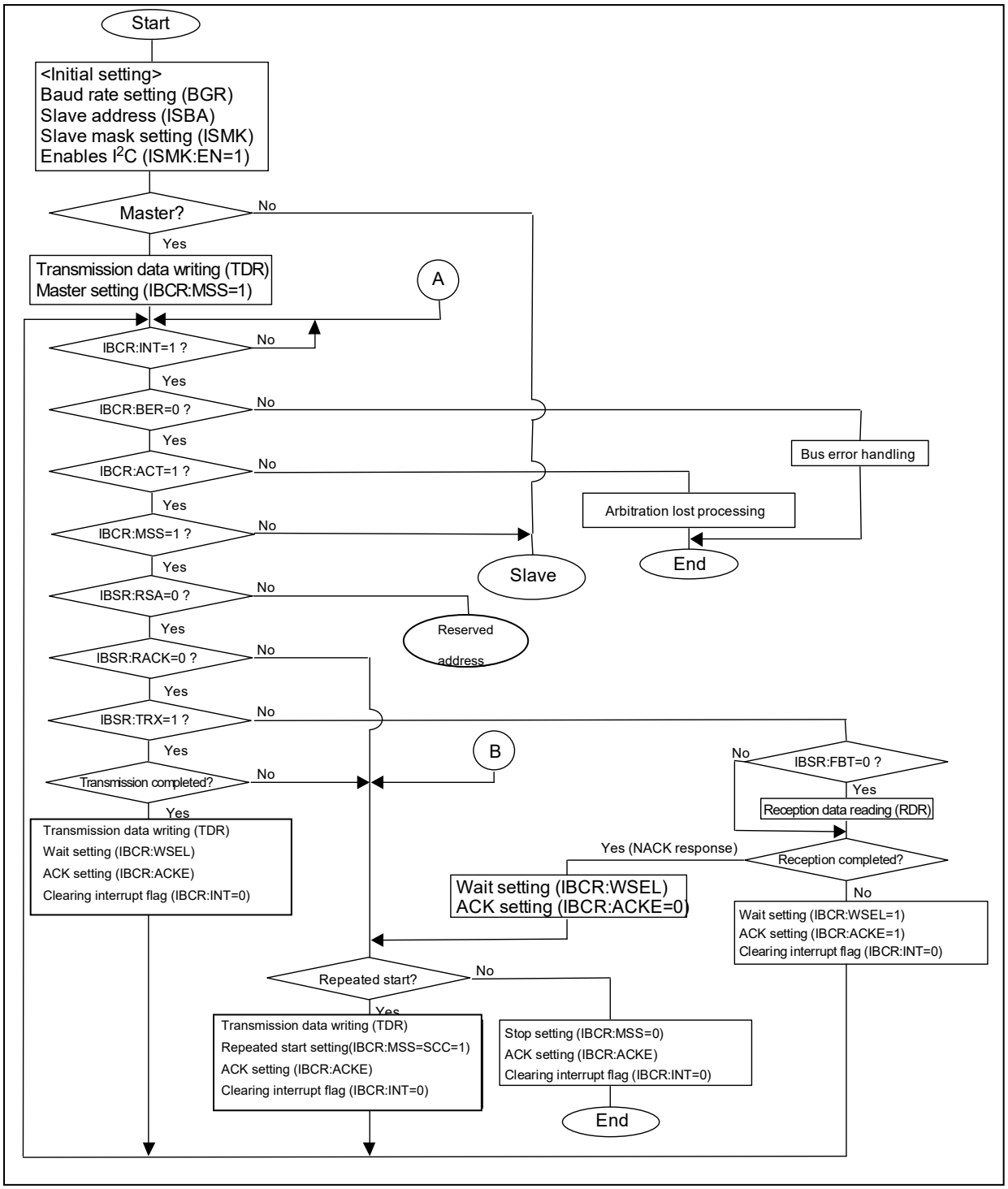
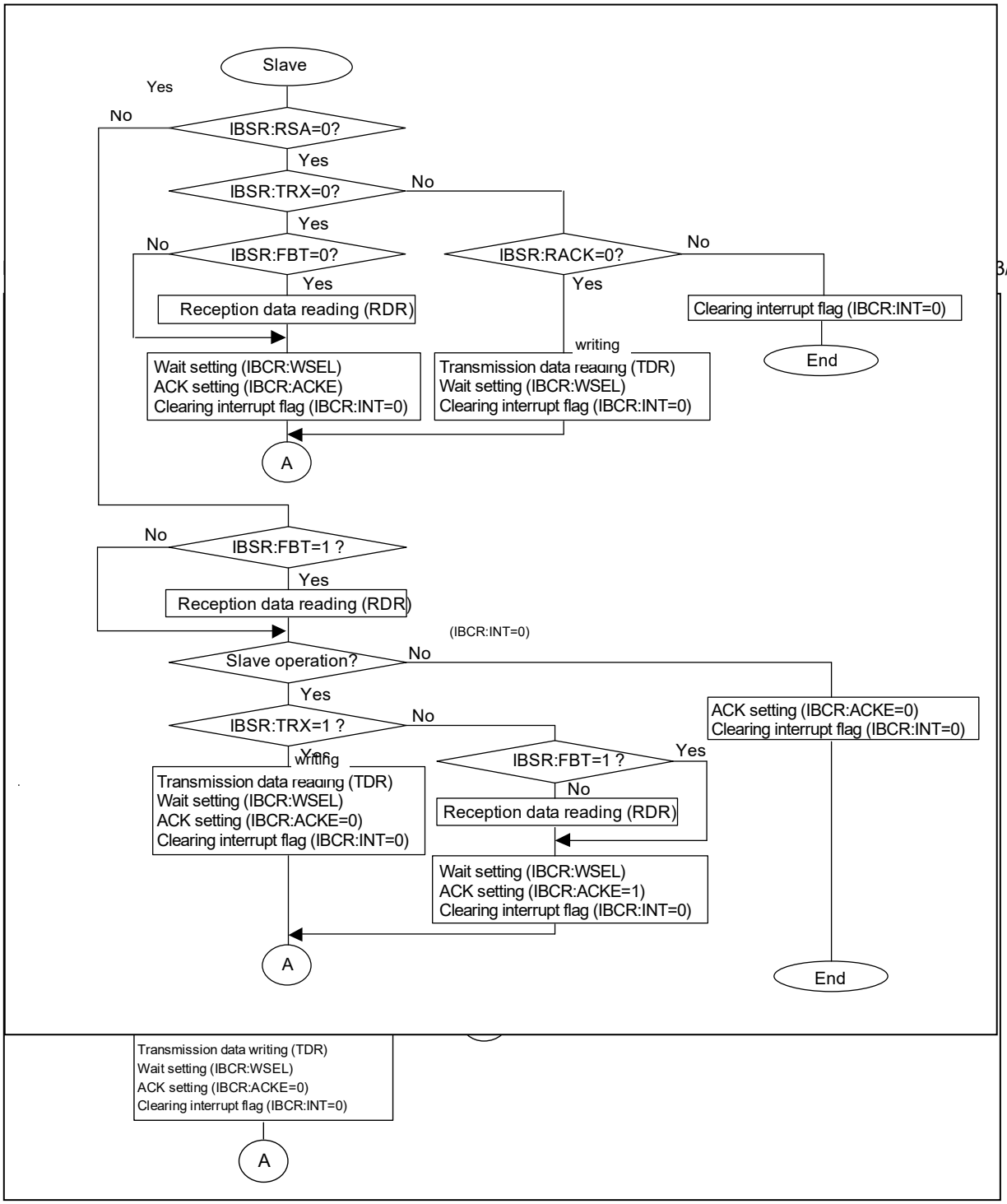


Figure 40-105. Example of I²C Flowchart (FIFO Memory Not Used) (When DMA Mode is Disable (SSR: DMA=0)) 2/3



40.8.6.2 Example of I²C Flowchart (FIFO Memory Not Used) (When DMA Mode is Enable (SSR: DMA=1))

Figure 40-107. Example of I²C flowchart (FIFO Not Used) (When DMA Mode is Enable (SSR: DMA=1)1/4

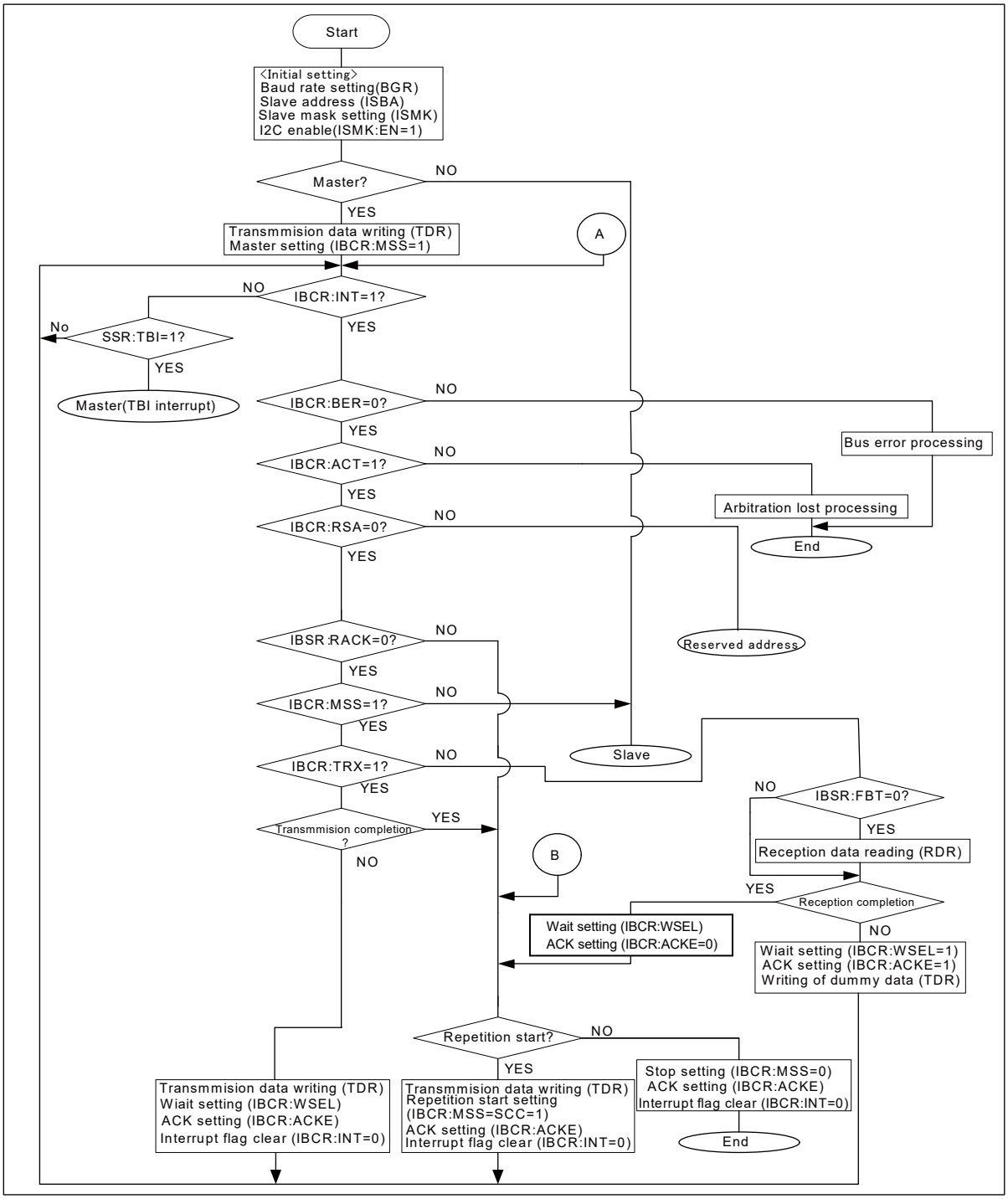


Figure 40-108. Example of I²C flowchart (FIFO Not Used) (When DMA Mode is Enable (SSR: DMA=1) 2/4

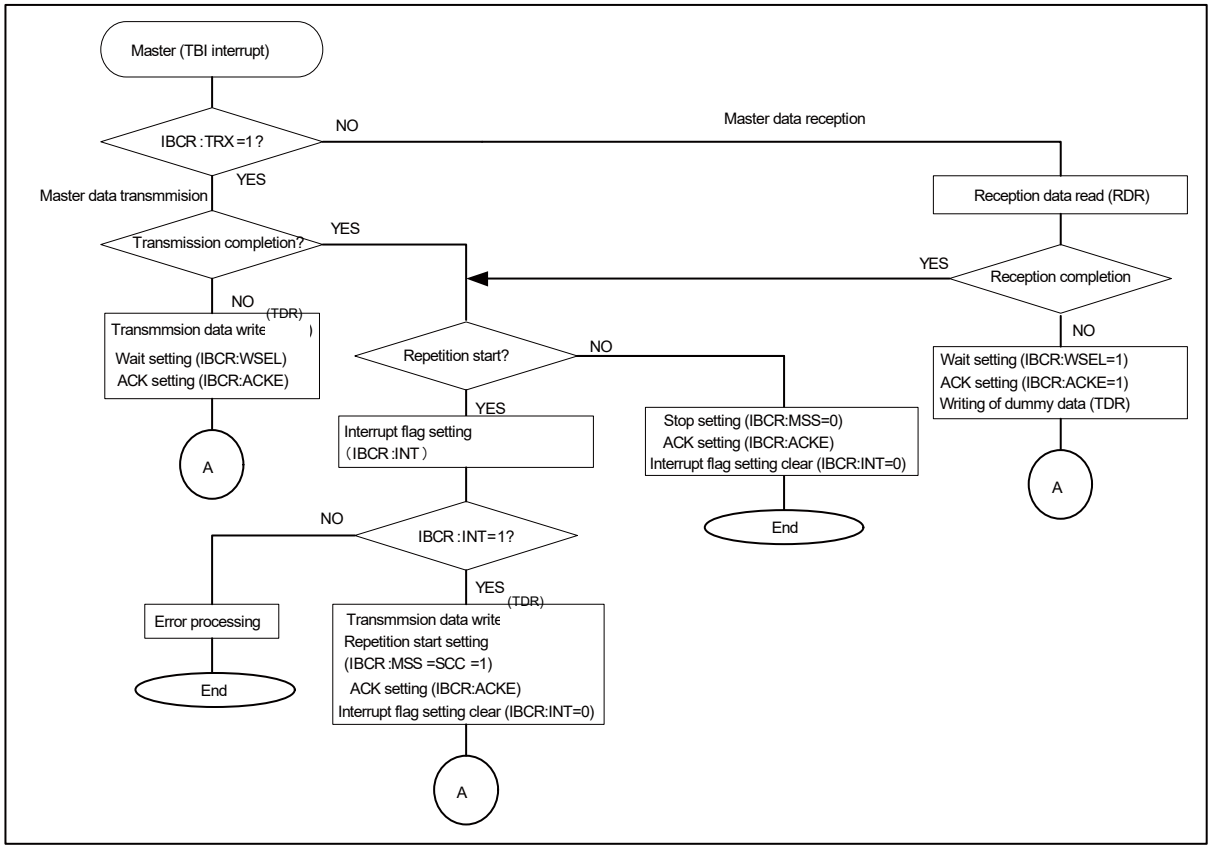


Figure 40-109. Example of I²C flowchart (FIFO Not Used) (When DMA Mode is Enable (SSR: DMA=1)) 3/4

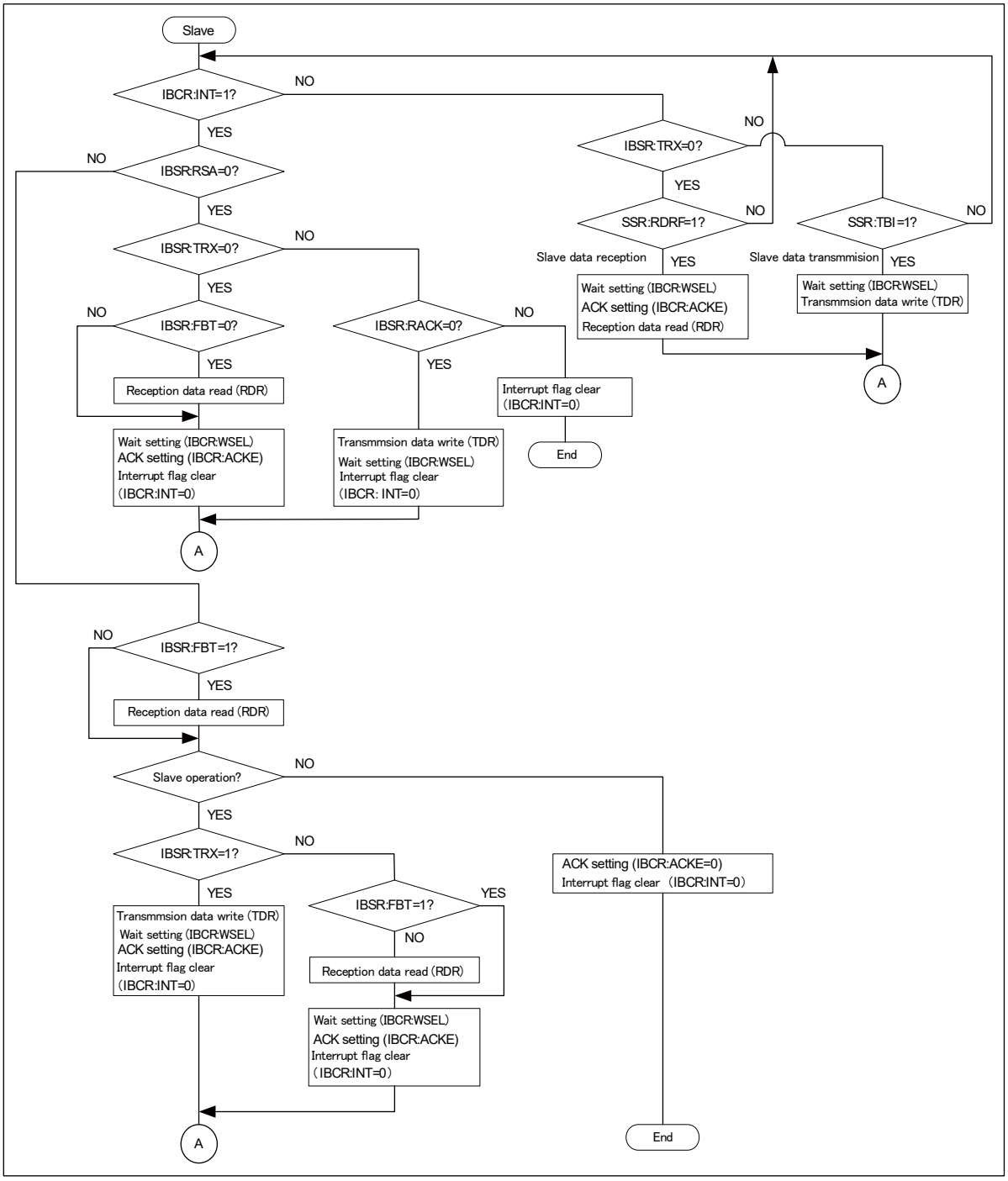
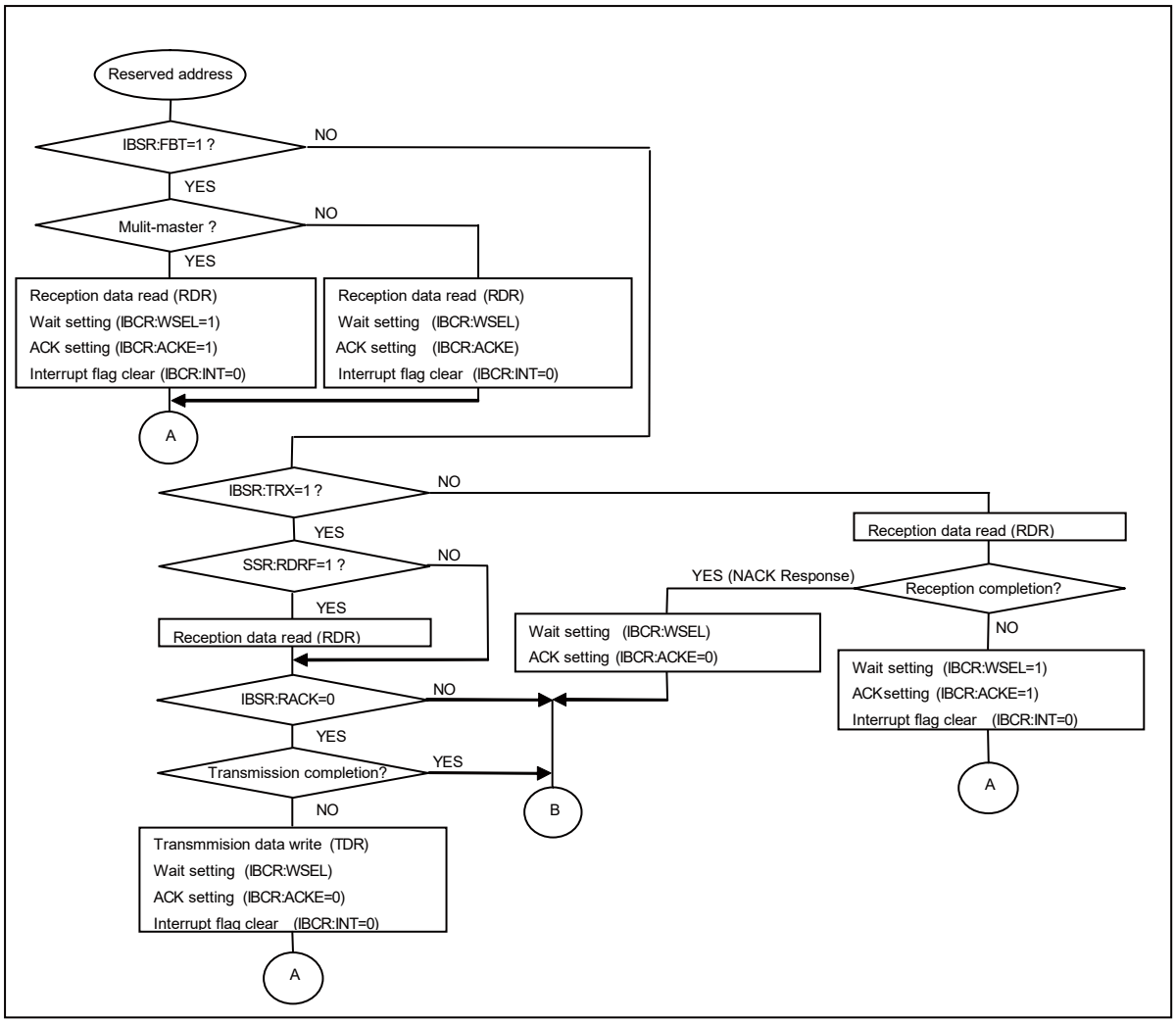


Figure 40-110. Example of I²C flowchart (FIFO Not Used) (When DMA Mode is Enable (SSR: DMA=1)) 4/4



Note:

Flow is flow that shows the operation setting outline by the I²C mode. It is necessary to do processing that considers error processing etc. to the application.

41. LIN-UART

This chapter explains the LIN-UART.

41.1 Overview

41.2 Features

41.3 Configuration

41.4 Registers

41.5 Interrupts

41.6 Baud Rates

41.7 Operation

41.8 Notes on Usage

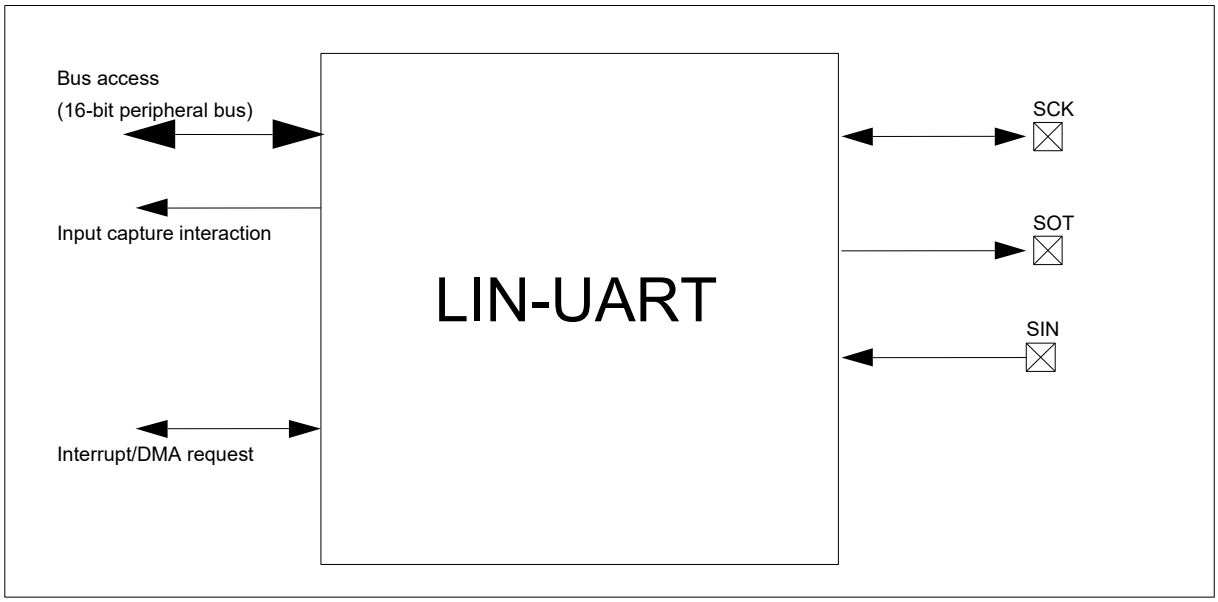
41.9 Notes on DMAC Linkage Operation

41.1 Overview

This section explains the overview of the LIN-UART.

The LIN (Local Interconnect Network) supported UART (Universal Asynchronous Receiver and Transmitter) is a general-purpose serial data communication interface to allow synchronous or asynchronous communication with external devices. It supports the bidirectional communication function (normal mode), the master/slave type communication function (multi-processor mode: both master and slave are supported) and the LIN bus system (operable for both master and slave).

Figure 41-1. Block Diagram (Overview for 1 Channel)



41.2 Features

This section explains the features of the LIN-UART.

LIN-UART is the general-purpose serial data communication interface used to transmit/receive data with another CPU or external devices, especially a LIN device.

41.2.1 Functions

This section explains functions of the LIN-UART.

Table 41-1. LIN-UART Functions

Item	Function
Data buffer	Full-duplex buffering
Serial input	Execute over-sampling for five times and determine the reception value by the majority of the sampling value.(Asynchronous mode only)
Transfer mode	■ Clock synchronous (Start/stop synchronous, start/stop bit select) ■ Clock asynchronous (Start/stop bit available)
Baud rate	■ Dedicated baud rate generator provided (comprising of 15-bit reload counter) ■ An external clock can be entered. It can also be adjusted by the reload counter.
Data length	■ 7 bits (except for synchronous or LIN mode) ■ 8 bits
Signaling system	NRZ (Non Return to Zero)
Start bit timing	Synchronized with a falling edge of the start bit in asynchronous mode
Reception error detection	■ Framing error ■ Overrun error ■ Parity error
Interrupt request	■ Reception interrupt (Reception completion, reception error detection, LIN synch break detection) ■ Transmission interrupt (Transmission data empty) ■ Interrupt request to input capture (LIN synch field detection: LSYN)
Master/Slave Communication Function (Multi-processor Mode)	The "1-to-n" communication (between 1 master and multiple slave systems) can be performed. (both master and slave systems are supported)
Synchronous mode	Master or slave function
Pin access	The serial I/O pin state can be read and written directly.

Item	Function
LIN bus option	■ Master device operation ■ Slave device operation ■ LIN synch break generation ■ LIN synch break detection Detects the start/stop edge of LIN Synch field by input capture 0, 1, 2, 3, 4 or 5. (See the section "24.4.4 LIN Synch Field Switching Register: LSYNS" in the "Chapter: Input Capture".) Supports LIN protocol Revision 2.1
Synchronous serial clock	Continuous clock output to the SCK pin is allowed for synchronous communication using start/stop bits.
Clock delay option	Special synchronous clock mode for clock delay (effective for SPI)

41.2.2 Operation Mode

This section explains operation mode of the LIN-UART.

LIN-UART supports four operation modes, and the operation mode is determined using the MD0 and MD1 bits of the serial mode register (SMR). Mode 0 and 2 are used for bidirectional serial communication, and mode 1 is used for master/slave communication. Mode 3 is used for LIN master/slave communication.

Table 41-2. LIN-UART Operation Mode

Operation Mode		Data Length		Synchronous System	Stop Bit Length	Data Bit Format
		Parity No	Parity Yes			
0	Normal mode	7 or 8 bits		Asynchronous	1 bit or 2 bits	LSB First or MSBFirst
1	Multi-processor mode	7 bits or 8 bits + 1 bit [1]	-			
2	Normal mode	8 bits		Synchronous	No, 1 bit, 2 bits	
3	LIN Mode	8 bits	-	Asynchronous	1 bit or 2 bits	LSB First

-: Setting is prohibited

[1]: In the multi-processor mode, "+1" is used as a communication control address/data selection bit (AD).

Note:

Mode 1 (Multi-processor mode), when the master/slave are connected, supports the operation of both master and slave. In mode 3, communication format is fixed.
If the current mode is changed, LIN-UART stops the data transmission or reception and waits for the start of the next communication.

The following table shows the operation modes to be set by MD1 and MD0 bits of the serial mode register (SMR).

Table 41-3. Mode Bit Settings

MD1	MD0	Mode	Function
0	0	0	Asynchronous (normal mode)
0	1	1	Asynchronous (multi-processor mode)
1	0	2	Synchronous (normal mode)
1	1	3	Asynchronous (LIN mode)

41.3 Configuration

This section explains the configuration of the LIN-UART.

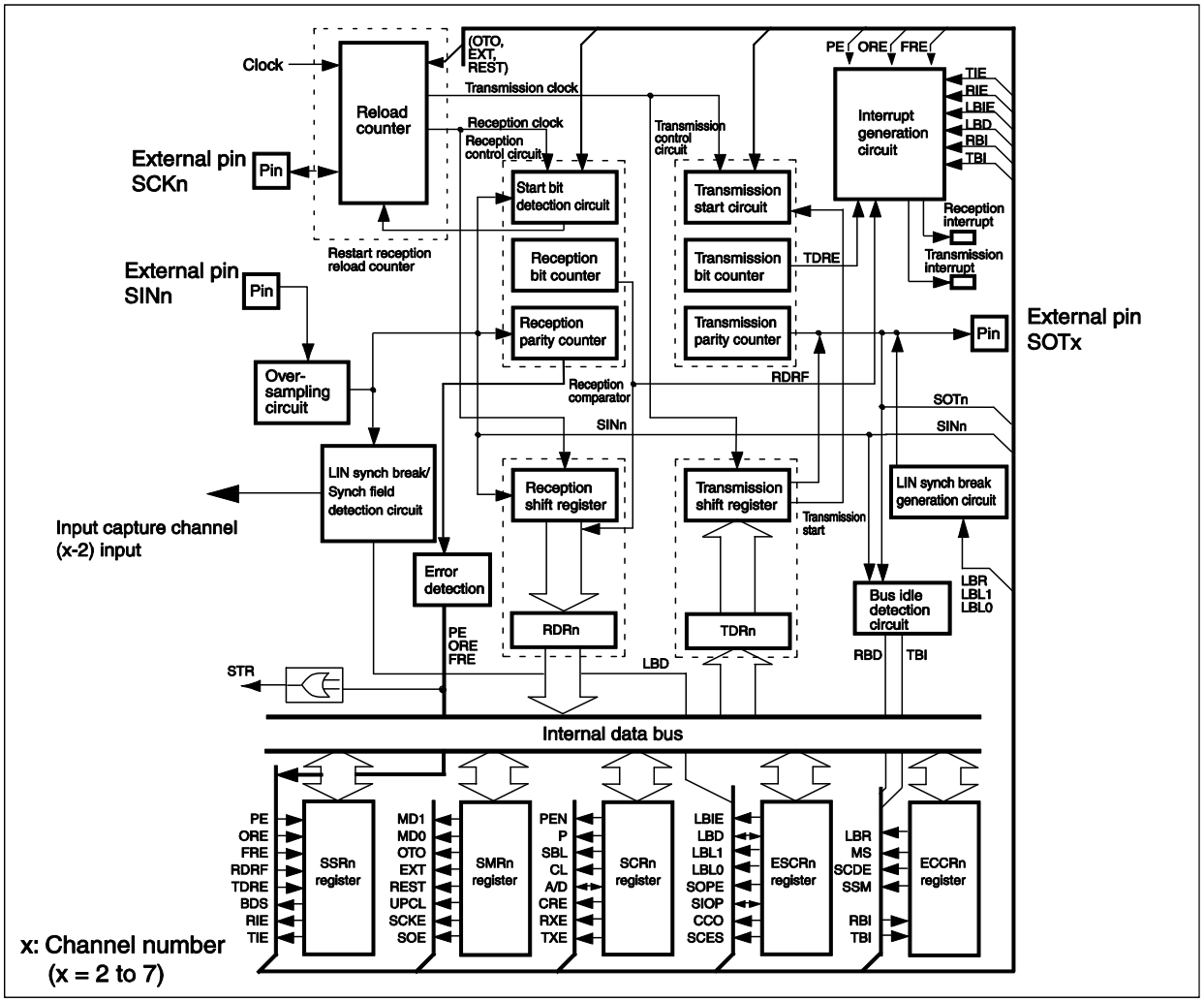
41.3.1 Block Diagram of the LIN-UART

This section explains the block diagram of the LIN-UART.

LIN-UART consists of the following functional blocks.

- Reload counter
- Reception control circuit
- Reception shift register
- Reception data register (RDR)
- Transmission control circuit
- Transmission shift register
- Transmission data register (TDR)
- Error detection circuit
- Over-sampling circuit
- Interrupt generation circuit
- LIN synch break or LIN synch field detection circuit
- Bus idle detection circuit
- Serial mode register (SMR)
- Serial control register (SCR)
- Serial status register (SSR)
- Extended communication control register (ECCR)
- Extended status control register (ESCR)

Figure 41-2. Block Diagram of the LIN-UART



41.3.2 Explanation of Each Block

This section explains each block of the LIN-UART.

41.3.2.1 Reload Counter
41.3.2.2 Reception Control Circuit
41.3.2.3 Reception Shift Register
41.3.2.4 Reception Data Register (RDR)
41.3.2.5 Transmission Control Circuit
41.3.2.6 Transmission Shift Register
41.3.2.7 Transmission Data Register (TDR)
41.3.2.8 Error Detection Circuit
41.3.2.9 Over-sampling Circuit
41.3.2.10 Interrupt Generation Circuit
41.3.2.11 LIN Synch Break/LIN Synch Field Detection Circuit
41.3.2.12 LIN Synch Break Generation Circuit
41.3.2.13 Bus Idle Detection Circuit
41.3.2.14 Serial Mode Register (SMR)
41.3.2.15 Serial Control Register (SCR)
41.3.2.16 Serial Status Register (SSR)
41.3.2.17 Extended Status Control Register (ESCR)
41.3.2.18 Extended Communication Control Register (ECCR)

41.3.2.1 Reload Counter

This section explains the reload counter of the LIN-UART.

The reload counter functions as the dedicated baud rate generator. The transmission/reception clocks are generated from either external or internal clocks. The reload counter has a 15-bit register as a reload value. The count value of the transmission reload counter can be read from the BGR value.

41.3.2.2 Reception Control Circuit

This section explains the reception control circuit of the LIN-UART.

The reception control circuit consists of the reception bit counter, the start bit detection circuit, and the reception parity counter.

The reception bit counter counts up the reception data. When a single data having the specified data length is received, the reception data full flag bit (SSR:RDRF) is set. If the reception interrupt is enabled (SSR:RIE=1) at this time, a reception interrupt request is generated.

The start bit detection circuit detects a start bit in the serial input signal, and sends a signal to the reload counter in synchronization with a falling edge of the start bit.

The reception parity counter calculates the parity of the reception data.

41.3.2.3 Reception Shift Register

This section explains reception shift register of the LIN-UART.

The reception shift register the retrieves reception data entered from the SIN pin by bit shifting. When the data reception is completed, the reception shift register transfers the reception data to the reception data register (RDR).

41.3.2.4 Reception Data Register (RDR)

This section explains the reception data register (RDR) of the LIN-UART.

The reception data register holds the reception data. The serial input data is converted and stored in the reception data register.

41.3.2.5 Transmission Control Circuit

This section explains the transmission control circuit of the LIN-UART.

The transmission control circuit consists of the transmission bit counter, the transmission start circuit, and the transmission parity counter.

The transmission bit counter counts up the transmission data bits, and sends a single data having the specified data length. When the transmission bit counter indicates a start of transmission of the written data, the flag of the serial status register is set. If the transmission interrupt is enabled at this time, a transmission interrupt request is generated. The transmission start circuit starts transmitting data when it is written in the TDR.

The transmission parity counter generates a parity bit of the transmission data if parity has been specified.

41.3.2.6 Transmission Shift Register

This section explains the transmission shift register of the LIN-UART.

The transmission shift register shifts the transmission data that has been written in the transmission data register (TDR), and outputs the data in bits to the SOT pin.

41.3.2.7 Transmission Data Register (TDR)

This section explains the transmission data register (TDR) of the LIN-UART.

The transmission data is set in the transmission data register. The data written in the transmission data register is converted into serial data and output.

41.3.2.8 Error Detection Circuit

This section explains the error detection circuit of the LIN-UART.

This circuit detects whether an error has occurred at the end of data reception. If an error has occurred, this circuit sets the corresponding error flag.

41.3.2.9 Over-sampling Circuit

This section explains the over-sampling circuit of the LIN-UART.

In the asynchronous mode, this circuit executes over-sampling five times with the machine clock and determines the reception value by the majority of the sampling value. This circuit does not operate in the synchronous mode.

41.3.2.10 Interrupt Generation Circuit

This section explains the interrupt generation circuit of the LIN-UART.

This circuit controls all interrupt factors. If a corresponding interrupt enable bit has been set, an interrupt occurs immediately.

41.3.2.11 LIN Synch Break/LIN Synch Field Detection Circuit

This section explains the LIN synch break/LIN synch field detection circuit of the LIN-UART.

When the LIN master node transmits a message header, a LIN synch break is detected. If a LIN synch break is detected, the LBD flag bit is set. An internal signal (LSYN) is output to the input capture in order to detect the 1st and 5th falling edges of the LIN synch field signal and to measure the actual serial clock synchronization signal that is transmitted by the master node.

41.3.2.12 LIN Synch Break Generation Circuit

This section explains the LIN synch break generation circuit of the LIN-UART.

This circuit generates a LIN synch break of length selected by the LIN synch break length select bit of the extended status control register.

41.3.2.13 Bus Idle Detection Circuit

This section explains the bus idle detection circuit of the LIN-UART.

This circuit detects that no transmission/reception is in progress, and sets the TBI or RBI flag bit.

41.3.2.14 Serial Mode Register (SMR)

This section explains the serial mode register.

The serial mode register is used to:

- Select a LIN-UART operation mode.
- Select a clock input.
- Select either "1-to-1" connection or reload counter connection for the external clock.
- Reactivate the dedicated reload timer.
- Reset the LIN-UART software (by keeping the register settings).
- Enable/disable output to the serial data pin (SOT).
- Enable/disable output to the serial clock pin (SCK).

41.3.2.15 Serial Control Register (SCR)

This section explains the serial control register.

The serial control register is used to:

- Select whether or not to use parity bits.
- Select a parity bit.
- Set the stop bit length.
- Set the data length.
- Select a frame data format in mode 1.
- Clear the error flag.
- Enable/disable data transmission.
- Enable/disable data reception.

41.3.2.16 Serial Status Register (SSR)

This section explains the serial status register.

The SSR is used to:

- Check the data transmission/reception and error state.
- Select the LSB First or MSB First data transfer direction.
- Enable/disable the transmission interrupt.
- Enable/disable the reception interrupt.

41.3.2.17 Extended Status Control Register (ESCR)

This section explains the extended status control register.

The ESCR is used to:

- Enable/disable the LIN synch break interrupt.
- Detect a LIN synch break
- Select the LIN synch break length.
- Directly access the SINn or SOTn pin.
- Set the continuous clock output in the LIN-UART synchronous clock mode.
- Select a sampling clock edge.

41.3.2.18 Extended Communication Control Register (ECCR)

This section explains the extended communication control register.

The ECCR is used to:

- Detect the bus idle status.
- Set t asynchronous clock.
- LIN synch break generation

41.4 Registers

This section explains the registers of the LIN-UART.

List of Base_addresses (Base_addr) and External Pins

Channel Number	Base_addr	External Pin Name		
		SIN	SOT	SCK
2	0x00D0	SIN2 / SIN2_1	SOT2 / SOT2_1	SCK2 / SCK2_1
3	0x00D8	SIN3 / SIN3_1	SOT3 / SOT3_1	SCK3 / SCK3_1
4	0x00E0	SIN4 / SIN4_1	SOT4 / SOT4_1	SCK4 / SCK4_1
5	0x00E8	SIN5 / SIN5_1	SOT5 / SOT5_1	SCK5 / SCK5_1
6	0x00F0	SIN6	SOT6	SCK6
7	0x00F8	SIN7_1	SOT7_1	SCK7_1

Select an external pin to be used for channels 2 to 5, using the IO relocation function.

Registers Map

Table 41-4. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x00D0	SCR2	SMR2	SSR2	RDR2/TDR2	Serial control register 2 Serial mode register 2 Serial status register 2 Transmission/reception data register 2
0x00D4	ESCR2	ECCR2	BGR2		Extended status control register 2 Extended communication control register 2 Baud rate generator register 2
0x00D8	SCR3	SMR3	SSR3	RDR3/TDR3	Serial control register 3 Serial mode register 3 Serial status register 3 Transmission/reception data register 3
0x00DC	ESCR3	ECCR3	BGR3		Extended status control register 3 Extended communication control register 3 Baud rate generator register 3
0x00E0	SCR4	SMR4	SSR4	RDR4/TDR4	Serial control register 4 Serial mode register 4 Serial status register 4 Transmission/reception data register 4
0x00E4	ESCR4	ECCR4	BGR4		Extended status control register 4 Extended communication control register 4 Baud rate generator register 4
0x00E8	SCR5	SMR5	SSR5	RDR5/TDR5	Serial control register 5 Serial mode register 5 Serial status register 5 Transmission/reception data register 5
0x00EC	ESCR5	ECCR5	BGR5		Extended status control register 5 Extended communication control register 5 Baud rate generator register 5
0x00F0	SCR6	SMR6	SSR6	RDR6/TDR6	Serial control register 6 Serial mode register 6 Serial status register 6 Transmission/reception data register 6

Address	Registers				Register Function
	+0	+1	+2	+3	
0x00F4	ESCR6	ECCR6	BGR6		Extended status control register 6 Extended communication control register 6 Baud rate generator register 6
0x00F8	SCR7	SMR7	SSR7	RDR7/TDR7	Serial control register 7 Serial mode register 7 Serial status register 7 Transmission/reception data register 7
0x00FC	ESCR7	ECCR7	BGR7		Extended status control register 7 Extended communication control register 7 Baud rate generator register 7

41.4.1 Serial Control Register: SCR

The bit configuration of the serial control register is shown below.

The serial control register (SCR) is used to set the parity bit, select the stop bit length and the data length, select the frame data format in mode 1, clear the reception error flag, and enable/disable data transmission and reception.

SCR: Address Base_addr + 00_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PEN	P	SBL	CL	AD	CRE	RXE	TXE
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R,W	R0,W	R/W	R/W

[bit7] PEN: Parity enable bit

PEN	Parity Enable
0	Without parity [initial value]
1	With parity

This bit configures whether to enable addition (transmission) and detection (reception) of the parity bit.

The parity bit is added only when the start/stop is set in mode 0 or mode 2 (ECCR:SSM="1"). This bit is fixed to "0" in modes 1 and 3.

[bit6] P: Parity selection bit

P	Parity Selection
0	Even parity [initial value]
1	Odd parity

If with parity (PEN=1), select either odd parity "1" or even parity "0".

[bit5] SBL: Stop bit length selection bit

SBL	Stop Bit Length
0	1 bit [initial value]
1	2 bits

This bit selects the length of the stop bit (the frame end mark of the transmission data) in the case where the start/stop bit is set to be used (ECCR:SSM=1) in operation mode 0, 1 or 3 (asynchronous) or in operation mode 2 (synchronous).

Note:

On reception, the framing error is detected only by one bit of the stop bit.

[bit4] CL: Data length selection bit

CL	Data Length Selection
0	7 bits [initial value]
1	8 bits

This bit specifies the data length of the transmission/reception data. This bit is fixed to "1" in mode 2 and mode 3.

[bit3] AD: Address/data format selection bit

AD	Address/Data Format Sselection
0	Data frame [initial value]
1	Address frame

This bit sets the data format to be used in the multi-processor mode (mode 1). The last received data format value is used for reading.

Note:

The AD bit read value is undefined in any mode other than the multi-processor mode (mode 1).
To use the AD bit, see "41.8 Notes on Usage".

[bit2] CRE: Reception error flag clear bit

CRE	Clearing of Reception Error	
	Write	Read
0	No effect [initial value]	The read value is always "0".
1	Clears all reception errors (PE, FRE and ORE).	

This bit clears the PE, FRE, and ORE flags of the serial status register (SSR).

Note:

Clear the reception error flag after the reception operation is disabled (RXE=0).

[bit1] RXE: Reception enable bit

RXE	Reception Enable
0	Reception disabled [initial value]
1	Reception enabled

This bit enables/disables the LIN-UART reception. If this bit is set to "0", the data frame reception is disabled. This bit does not affect the LIN synch break detection in mode 3.

Note:

If you disable reception (RXE=0) while a reception is in progress, the reception stops immediately. In this case, the data cannot be guaranteed.

When ECCR:MS=0 in mode 2, if you disable transmission (TXE=0) while a reception is in progress, also dsiable reception (RXE=0).

[bit0] TXE: Transmission enable bit

TXE	Transmission Enable
0	Transmission disabled [initial value]
1	Transmission enabled

This bit enables/disable the LIN-UART transmission.

Note:

If you disable transmission (TXE=0) while it is in progress, the transmission stops immediately. In this case, the data cannot be guaranteed.

41.4.2 Serial Mode Register: SMR

The bit configuration of the serial mode register (SMR) is shown below.

The serial mode register (SMR) is used to select the operation mode and baud rate clock. Also, this register is used to enable/disable output to the serial data and clock pin.

SMR: Address Base_addr + 01_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W	R/W	R/W	R/W	R0,W	R0,W	R/W	R/W

[bit7, bit6] MD1, MD0: Operation mode selection bits

MD1	MD0	Operation Mode Setting
0	0	Mode 0: Asynchronous normal mode [initial value]
0	1	Mode 1: Asynchronous multi-processor mode
1	0	Mode 2: Synchronous mode
1	1	Mode 3: Asynchronous LIN mode

These bits set the LIN-UART operation mode.

Note:

The communication mode must be changed while the LIN-UART operation is inactive. If the mode is changed while transmission or reception is in progress, the transmitted/received data cannot be guaranteed. If the mode is changed after writing data to the transmission data register (TDR), the data written to the TDR becomes invalid and the transmission data empty flag is set (SSR:TDRE=1).

[bit5] OTO: 1 to 1 external clock enable bit

OTO	External Clock Enabled
0	Uses the baud rate generator (reload counter). [initial value]
1	Uses an external clock directly.

If this bit is set to "1", an external clock will be allowed to be used directly as the LIN-UART serial clock. It is used during slave operation in mode 2 (synchronous) (ECCR:MS=1).

If EXT=0, the OTO bit is fixed to "0".

[bit4] EXT: External clock selection bit

EXT	External Serial Clock Enabled
0	Uses the baud rate generator (reload counter). [initial value]
1	Uses an external clock's serial clock source.

This bit can be used to select a clock for the reload counter.

[bit3] REST: Reload counter restart bit

REST	Reload Counter Restart	
	Write	Read
0	No effect [initial value]	The read value is always "0".
1	Counter restart	

[bit2] UPCL: LIN-UART programmable clear bit (software reset)

UPCL	LIN-UART Programmable Clear (Software Reset)	
	Write	Read
0	No effect [initial value]	The read value is always "0".
1	LIN-UART reset	

If this bit is set to "1", the LIN-UART is reset immediately. However, the register settings are maintained. The current transmission or reception is aborted.

All transmission/reception interrupt factors (TDRE, RDRF, LBD, PE, ORE, and FRE) and LIN synch break generation bit (LBR) are initialized. Reset LIN-UART with this bit after disabling the interrupt and the transmission. Also, the reception data register is cleared (RDR=00_H), and the reload counter is restarted.

Note:

Perform LIN-UART software reset (UPCL=1) when the TXE bit of the serial control register (SCR) is "0".

Excepting mode 2 , if the flaming error is detected, and the serial data input is detected and continues the state of "L", the start bit and LIN Break field are not detected until the serial data input becomes "H".

[bit1] SCKE: Serial clock output enabled

SCKE	Serial Clock Output Enabled
0	Clock input pin [initial value]
1	Serial clock output pin

This bit controls the I/O of the serial clock pin (SCK).

If this bit is set to "1", the clock is output in mode 2.

Notes:

- When using the SCK pin for serial clock input (SCKE="0"), set the external clock selection bit to the external clock state (EXT="1") at the same time.
- Set the SCK pin as a peripheral I/O pin. For information on the setting method, see "Chapter: I/O Ports".

[bit0] SOE: Serial data output enable bit

SOE	Serial Data Output Enabled
0	Output disabled [initial value]
1	Serial data output

To transmit data from LIN-UART set this bit to "1". The initial value of this bit is "0", and there is no case in which this bit must be set to "0".

41.4.3 Serial Status Register: SSR

The bit configuration of the serial status register (SSR) is shown below.

The serial status register (SSR) is used to check the current transmission/reception state and error occurrence. This register is also used to control transmission/reception interrupts.

SSR: Address Base_addr + 02_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE
Initial value	0	0	0	0	1	0	0	0
Attribute	R,WX	R,WX	R,WX	R,WX	R,WX	R/W	R/W	R/W

[bit7] PE: Parity error flag bit

PE	Parity Error
0	No parity error [initial value]
1	A parity error occurs during reception.

If a parity error occurs during data reception with PEN=1, this bit is set to "1". This flag bit is cleared to "0" if the CRE bit of the serial control register (SCR) is set to "1".

If both PE and RIE bits are set to "1", a reception interrupt request is generated.

If this flag is set, the data contained in the reception data register (RDR) becomes invalid.

[bit6] ORE: Overrun error flag bit

ORE	Overrun Error
0	No overrun error [initial value]
1	An overrun error occurs during reception.

If an overrun error occurs during data reception, this bit is set to "1". This flag bit is cleared to "0" if the CRE bit of the serial control register (SCR) is set to "1".

If both ORE and RIE bits are set to "1", a reception interrupt request is generated.

If this flag is set, the data contained in the reception data register (RDR) becomes invalid.

[bit5] FRE: Framing error flag bit

FRE	Framing Error
0	No framing error [initial value]
1	A framing error occurs during reception.

If a framing error occurs during data reception, this bit is set to "1". This flag bit is cleared to "0" if the CRE bit of the serial control register (SCR) is set to "1".

If both FRE and RIE bits are set to "1", a reception interrupt request is generated.

If this flag is set, the data contained in the reception data register (RDR) becomes invalid.

Note:

Only the first bit of the stop bit detects the framing error even if it sets it to SCR:SBL=1.

Excepting mode 2 , if the flaming error is detected , and the serial data input is detected and continues the state of "L", the start bit is not detected until the serial data input becomes "H".

[bit4] RDRF: Reception data full flag bit

RDRF	Reception Data Register Full
0	The reception data register has no data. [initial value]
1	The reception data register has data.

This bit indicates the state of the reception data register (RDR).

This bit is set to "1" when the reception data is stored in the RDR. This bit is cleared to "0" when data is read from the RDR.

If both RDRF and RIE bits are set to "1", a reception interrupt request is generated.

[bit3] TDRE: Transmission data empty flag bit

TDRE	Transmission Data Register Empty
0	The transmission data register has data.
1	The transmission data register has no data. [initial value]

This bit indicates the state of the transmission data register (TDR).

When the transmission data is written in the TDR, this bit is set to "0" indicating that the register has the transmission data. When data is stored in the transmission shift register and the transmission is started, this bit is set to "1".

If both TDRE and TIE bits are set to "1", a transmission interrupt request is generated.

If the TDRE bit is "1" and if the LBR bit of the extended communication control register (ECCR) is set to "1", the TDRE bit is switched to "0". If the TDR register does not have any valid data after the LIN synch break generation, this bit is set to "1".

Note:

The TDRE bit is initially set to "1".
If TDRE=0 is set by data writing in the transmission data register (TDR) and if the mode setting (SMR:MD[1:0]) is changed after that, the transmission data is made invalid and TDRE=1 is set.

[bit2] BDS: Transfer direction selection bit

BDS	Bit Direction Setting
0	LSB First (The least significant bit is transferred first.) [initial value]
1	MSB First (The most significant bit is transferred first.)

Either LSB first (BDS="0") or MSB first (BDS="1") can be selected for serial data transfer.

This bit is fixed to "0" in mode 3.

Note:

When data is written to or read from the reception data register, the high-order and low-order sides of the reception data are replaced. If the BDS bit value is changed after data has been written in the RDR, the data will become invalid.

[bit1] RIE: Reception interrupt request enable bit

RIE	Reception Interrupt Enabled
0	Reception interrupt disabled [initial value]
1	Reception interrupt enabled

This bit enables or disables output of a reception interrupt request to the CPU.

If the RIE bit and the reception data flag bit (RDRF) are set to "1" or if an error flag (PE, ORE, or FRE) is set to "1", a reception interrupt request is generated.

[bit0] TIE: Transmission interrupt request enable bit

TIE	Transmission Interrupt Enabled
0	Transmission interrupt disabled [initial value]
1	Transmission interrupt enabled

This bit enables or disables output of a transmission interrupt request to the CPU.

If both TIE and TDRE bits are set to "1", a transmission interrupt request is generated.

41.4.4 Reception Data Register/Transmission Data Register: RDR/TDR

This section explains the reception data register/transmission data register (RDR/TDR).

The reception data register (RDR) holds the reception data, and the transmission data register (TDR) holds the transmission data. RDR and TDR are placed in the same address.

41.4.4.1 Reception Data Register: RDR

The bit configuration of the reception data register (RDR) is shown below.

Reception data register (RDR):
RDR: Address Base_addr + 03_H (Access: Byte, Half-word, Word)

	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

[bit7 to bit0] D[7:0]: Data registers

Access	Data Register
Read	Reads data from the reception data register.

The reception data register (RDR) is the data buffer register for serial data reception.

Serial data signals sent to the serial input pin (SIN) are converted in the shift register and stored in the reception data register (RDR).

If the data length is 7 bits, the upper 1 bit (RDR:D7) is set to "0".

When the reception data is stored in the reception 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), a reception interrupt request is generated.

The reception 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 reception data register (RDR) is read out. If a reception interrupt is enabled but if no error has occurred, the reception interrupt is also cleared.

If a reception error occurs (SSR:PE, ORE or FRE "1"), the data in the reception data register (RDR) will become invalid.

41.4.4.2 Transmission Data Register: TDR

The bit configuration of the transmission data register (TDR) is shown below.

Transmission Data Register (TDR):
TDR: Address Base_addr + 03_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	-	-	-	-	-	-	-	-
Attribute	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W

[bit7 to bit0] D[7:0]: Data registers

Access	Data Register
Write	Writes data to the transmission data register.

When the transmission data is enabled to transmit and when it is written to the transmission data register, the data is transferred to the transmission shift register and it is converted into serial data and transmitted from the serial data output pin (SOT). If the data length is 7 bits, the most significant bit (D7) is not transmitted.

When the transmission data is written to the TDR register, the transmission data empty flag bit (TDRE bit of SSR) is cleared to "0". When data transfer to the transmission shift register is complete and when the transmission starts, the TDRE bit is set to "1". If the TDRE bit is "1", the next transmission data can be written to the TDR register. If a transmission interrupt request is enabled, a transmission interrupt is generated. If a transmission interrupt has occurred or if the TDRE bit is "1", write the next data.

Note:

The TDR is a write-only register, and the RDR is a read-only register. As these registers are located in the same address, the read value and the write value differ from each other. Therefore, none of read-modify-write instructions shall be used to access these registers.

41.4.5 Extended Status Control Register: ESCR

The bit configuration of the extended status control register is shown below.

Extended status control register can be used to set the LIN function. Also, it can be used to set the direct access to the SIN and SOT pins and the LIN-UART synchronous clock mode.

ESCR: Address Base_addr + 04_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES
Initial value	0	0	0	0	0	X	0	0
Attribute	R/W	R(RM1),W	R/W	R/W	R/W	R,W	R/W	R/W

[bit7] LBIE: LIN synch break detection interrupt enable bit

LBIE	LIN Synch Break Detection Interrupt Enabled
0	LIN synch break interrupt disabled [initial value]
1	LIN synch break interrupt enabled

An interrupt occurs when the LIN synch break detection flag (LBD) is set to "1" and interrupts are enabled (LBIE=1).

This bit is fixed to "0" in operation modes 1 and 2.

[bit6] LBD: LIN synch break detection flag bit

LBD	LIN Synch Break Detection	
	Write	Read
0	Clears the LIN synch break detection flag.	Does not detect the LIN synch break. [initial value]
1	No effect	Detects the LIN synch break.

When a LIN synch break is detected (if the serial input is "0" for more than 11-bit time), this bit is set to "1". If this bit is set to "0", the LBD flag bit is cleared. If the LIN synch break detection interrupt is enabled, the interrupt is also cleared.

When a read-modify-write instruction is issued, "1" is always returned. In such case, however, it does not signify a LIN synch break detection.

Note:

To detect a LIN synch break, enable LIN synch break detection interrupt (LBIE=1) and then disable reception (SCR:RXE=0).

[bit5, bit4] LBL1, LBL0: LIN synch break length selection bits

LBL1	LBL0	LIN Synch Break Length
0	0	13-bit length [initial value]
0	1	14-bit length
1	0	15-bit length
1	1	16-bit length

These bits define the serial bit length of the LIN synch break generated by the LIN-UART. The length is always fixed to 11 bits for LIN synch break reception.

[bit3] SOPE: Serial output pin direct access enable bit

SOPE	Serial Output Pin Direct Access
0	Serial output pin direct access disabled [initial value]
1	Serial output pin direct access enabled

When serial data output is enabled (sMR*SOE=1), if this bit is set to "1", SOT pin can be accessed directly.

See Table 41-5 for details.

Note:

SOT pin cannot be accessed directly regardless the value of this bit during transmission or the stop mode of the mode 2 with SCR:TXE="1".

[bit2] SIOP: Serial I/O pin direct access enable bit

SIOP	Serial I/O Pin Access	
	Write (If SOPE is "1")	Read
0	Outputs "0" at the SOT pin.	Reads value from SIN pin.
1	Outputs "1" at the SOT pin. [initial value]	

When a normal read instruction is issued, the SIN pin value is returned. Set the SOT pin value for data writing. When a read-modify-write instruction is issued, the SOT pin value is returned.

See the following table for details:

Table 41-5. SOPE and SIOP Functions

SOPE	SIOP	Writing to SIOP	Reading from SIOP
0	R/W	No effect (The write value is held.)	The SIN value is returned.
1	R/W	"0" or "1" is written in the SOT pin.	The SIN value is returned.
0	RMW	No effect (The write value is held.)	The SOT value is returned.
1	RMW	"0" or "1" is written in the SOT pin.	The SOT value is returned.

[bit1] CCO: Continuous clock output enable bit

CCO	Continuous Clock Output (Mode 2)
0	Continuous clock output disabled [initial value]
1	Continuous clock output enabled

If the LIN-UART is set as the master system (ECCR:MS=0) in mode 2 (synchronous mode) and if the SCK pin is set for clock output, the continuous serial clock output is enabled.

Note:

Set the SCK pin to the clock output (SMR:SCKE=1). When "1" is set to the CCO bit, it changes to start/stop bit addition setting (ECCR:SSM=1).
Set "0" to this bit at the slave setting of operation modes 0, 1, and 3 and operation mode 2.
Prescribed width of the clock might not be output to the serial clock output pin (SCK pin) immediately after the switch of the serial clock output when CCO and the SCES bit are set while enabling the serial clock output (SMR:SCKE="1") on the following conditions but, after this, it will be output normally.

- When the CCO bit changes the SCES bit in the state of "1"
- When the CCO bit and the SCES bit are changed at the same time
- When the CCO bit is changed from "1" to "0"

[bit0] SCES: Sampling serial clock edge selection bit

SCES	Sampling Serial Clock Edge Selection
0	Sample signals at a clock rising edge (normal) [initial value]
1	Sample signals at a clock falling edge (inverted clock)

If the LIN-UART is set as a slave system (ECCR:MS=1) in mode 2 (synchronous mode) and if the SCES is set to "1", the sampling edge is switched from the rising edge to the falling edge.

If it is set as the master system (ECCR:MS=0) in mode 2 (synchronous mode) and if the SCK pin is set for clock output, the internal serial clock and the output clock signal are inverted.

This bit must be set to "0" in modes 0, 1, and 3.

Note:

When "1" is set to the SCES bit, software reset is prohibited. Moreover, change the SCES bit when transmission/reception is prohibited.

41.4.6 Extended Communication Control Register: ECCR

The bit configuration of the extended communication control register (ECCR) is shown below.

The extended communication control register (ECCR) enables the bus idle detection setting, the synchronous clock setting, and the LIN synch break generation.

ECCR: Address Base_addr + 05H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Initial value	0	0	0	0	0	0	X	X
Attribute	R/W0	R0,W	R/W	R/W	R/W	R/W0	R,WX	R,WX

[bit7] Reserved bit

This bit must always be written to "0".

[bit6] LBR: LIN synch break generation bit

LBR	LIN Synch Break Generation	
	Write	Read
0	No effect [initial value]	The read value is always "0".
1	LIN synch break generation	

If the LBR bit is set to "1" in mode 3, a LIN synch break with a length specified in LBL1 and LBL0 in ESCR is generated. In operation mode 0, set this bit to "0".

[bit5] MS: Master/slave mode selection bit

MS	Master/Slave Function (Mode 2)
0	Master mode (Serial clock generated) [Initial value]
1	Slave mode (External serial clock received)

Select master or slave mode in mode 2 (synchronous). When master mode is selected, a synchronous clock is generated. When slave mode is selected, an external serial clock is received.

In operation modes 0, 1, and 3, this bit is fixed to "0".

Change the MS bit while transmission is disabled (SCR:TXE=0).

Note:

When slave mode is selected, set an external clock as the clock source and set it to one-to-one external clock input (SMR:SCKE="0"', EXT="1", OTO="1").

[bit4] SCDE: Serial clock delay enable bit

SCDE	Serial Clock Delay Enabled (Mode 2)
0	Clock delay disabled [initial value]
1	Clock delay enabled

In master mode of mode 2, set the SCDE bit to "1" to output a delayed serial clock as shown in "Figure 41-15".
In operation modes 0, 1, and 3, this bit is fixed to "0".

[bit3] SSM: Start/stop bit enable

SSM	Start/Stop Bit Enable (Mode 2)
0	Start/stop bit not available [initial value]
1	Start/stop bit available

In mode 2, start and stop bits are added to the synchronous data format.
In operation modes 0, 1, and 3, this bit is fixed to "0".

[bit2] Reserved bit

This bit must always be written to "0".

[bit1] RBI: Reception bus idle flag bit

RBI	Reception Bus Idle
0	Reception operation in progress
1	No reception operation

This bit is set to "1" when the SIN pin is "H" level and no reception operation is in progress.
Reception bus idle detection is not available in mode 2.

[bit0] TBI: Transmission bus idle flag bit

TBI	Transmission Bus Idle
0	Transmission operation is in progress.
1	No transmission operation is in progress.

This bit is set to "1" if no transmission operation is in progress in the SOT pin.

If slave mode is selected in mode2, transmission bus idle detection is not available.

41.4.7 Baud Rate Generator Register: BGR

The bit configuration of the baud rate generator register (BGR) is shown below.

The baud rate generator register (BGR) sets the division ratio of the serial clock. It can also read an accurate value of the transmission reload counter.

BGR: Address Base_addr + 06H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	-	B14	B13	B12	B11	B10	B09	B08
Initial value	-	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	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	B07	B06	B05	B04	B03	B02	B01	B00
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] - : Undefined bit

The read value is always "0". This does not affect the writing operation.

[bit14 to bit8] B[14:08]: Baud rate generator register 1

B14-08	Baud Rate Generator Register 1
Write	Writing of bit14 to bit8 of the reload value to the counter
Read	Reading of bit14 to bit8

[bit7 to bit0] B[07:00]: Baud rate generator register 0

B07-00	Baud Rate Generator Register 0
Write	Writing of bit7 to bit0 of the reload value to the counter
Read	Reading of bit7 to bit0

Baud rate generator register (BGR) sets the division ratio of the serial clock.

The reload value for counting can be written to, and the transmission reload counter value can be read from this register.

When a reload value is written to the baud rate generator register (BGR), the reload counter starts counting.

41.5 Interrupts

This section explains the interrupts.

41.5.1 Overview

41.5.2 Generation of Reception Interrupt and Flag Setting Timing

41.5.3 Occurrence of Transmission Interrupt and Flag Timing

41.5.1 Overview

Overview of the interrupts is shown below.

LIN-UART has reception and transmission interrupts. An interrupt request is generated in one of the following cases:

- Storage of reception data in the reception data register (RDR) or occurrence of a reception error
- Transfer of transmission data from the transmission data register (TDR) to the transmission shift register
- Detection of a LIN synch break

41.5.1.1 Interrupts of LIN-UART

Interrupts of LIN-UART are shown below.

The following table shows the interrupt control bits and the interrupt factors:

Table 41-6. Interrupt Control Bits and Interrupt Factors of LIN-UART

Reception/ Transmission/ Input Capture	Interrupt Request Flag bit	Flag Register	Operation Mode				Interrupt Factor	Interrupt Factor Enable Bit	Clearing of Interrupt Request
			0	1	2	3			
Reception	RDRF	SSR	○	○	○	○	Writing of reception data to RDR	SSR:RIE	Reading of reception data
	ORE	SSR	○	○	○	○	Overrun error		Writing of "1" to the reception error clear bit (SSR:CRE)
	FRE	SSR	○	○	△	○	Framing error		
	PE	SSR	○	×	△	×	Parity error		
	LBD	ESCR	×	×	×	○	Detection of a LIN-Synch-Break	ESCR:LBIE	Writing of "0" to the LBD bit of ESCR
Transmission	TDRE	SSR	○	○	○	○	Transmission register empty	SSR:TIE	Writing of transmission data, Writing of "1" to the LIN synch break generation bit (ECCR:LBR)
Input capture (ch.0 to ch.5)	ICP	ICS	×	×	×	○	1st falling edge of LIN synch field	ICS:ICE	Writing of "0" to the ICS:ICP bit
	ICP	ICS	×	×	×	○	5th falling edge of LIN synch field	ICS:ICE	

○: Available

△: Available if the ECCR:SSM bit is "1"

×: Not available

41.5.1.2 Reception Interrupt

Reception interrupt is shown below.

If one of the following occurs in reception mode, a corresponding flag bit in the serial status register (SSR) is set to "1".

- Data Reception Completion: RDRF

  Transfer of reception data from the reception shift register to the reception data register (RDR) and reading of the data.

- Overrun Error: ORE

  When RDRF="1", RDR is not read by the CPU and the next reception data is transferred from the reception shift register to the reception data register (RDR).(ORE=1)

- Framing Error: FRE

  When the serial data is detected as "L" in the first bit of the stop bit. (FRE=1)

- Parity Error: PE

  Parity detection error.(PE=1)

  When the reception interrupt is enabled (SSR:RIE = "1") and one or more of the above flags is set to "1", a reception interrupt request is generated.

  When the reception data register (RDR) is read, the RDRF flag is automatically cleared to "0". If the reception error flag clear bit (CRE) in the serial control register (SCR) is set to "1", all the error flags are cleared to "0".

  Note:

  The CRE bit is write-only and, when "1" is written, it retains "1" for one clock cycle.

41.5.1.3 Transmission Interrupt

Transmission interrupt is shown below.

When transmission data is transferred from the transmission data register (TDR) to the transmission shift register and transmission is started, the transmission data register empty flag bit (TDRE) in the serial status register (SSR) is set to "1". If the transmission interrupt enable bit (TIE) in SSR is set at this time, an interrupt request is generated.

Note:

After reset, the initial value of TDRE is "1". Therefore, a transmission interrupt occurs as soon as the TIE flag is set to "1". The TDRE flag is cleared only if data is written to the transmission data register (TDR) or "1" is written to the LIN synch break generation bit (ECCR:LBR).

41.5.1.4 LIN Synch Break Interrupt

LIN synch break interrupt is shown below.

This interrupt is enabled when LIN-UART works as a LIN slave in mode 3.

When the serial input bus is "0" (dominant) for 11-bit time or longer, the LIN synch break detection flag bit (LBD) in the extended status control register (ESCR) is set to "1". The LIN synch break interrupt and the LBD flag are cleared when the LBD flag is set to "0". Clear the LBD flag before a capture interrupt occurs in the LIN synch field.

Reception must be disabled (SCR:RXE=0) if LIN synch break is to be detected.

41.5.1.5 LIN Synch Field Edge Detection Interrupt

LIN synch field edge detection interrupt is shown below.

This interrupt is enabled when LIN-UART works as a LIN slave in mode 3.

After a LIN synch break is detected, the internal signal is set to "1" at the 1st falling edge of the LIN synch field and set to "0" after the 5th falling edge. When the input capture (ICP) is set to the LSYN input (LSYNS:LSYN=1), both edges are detected (ICS:EG1, EG0=11), and interrupt is enabled (ICS:ICE=1), an interrupt occurs at rising and falling edges of LSYN (internal signal).

The difference of count values detected by input capture is equivalent to 8 bits of the serial clock of the master and a new baud rate can be calculated. When a falling edge of the start bit is detected, the reload counter automatically restarts.

The figures below show the interrupt generation timings:

Figure 41-3. LIN Synch Break Detection and Flag Set Timing

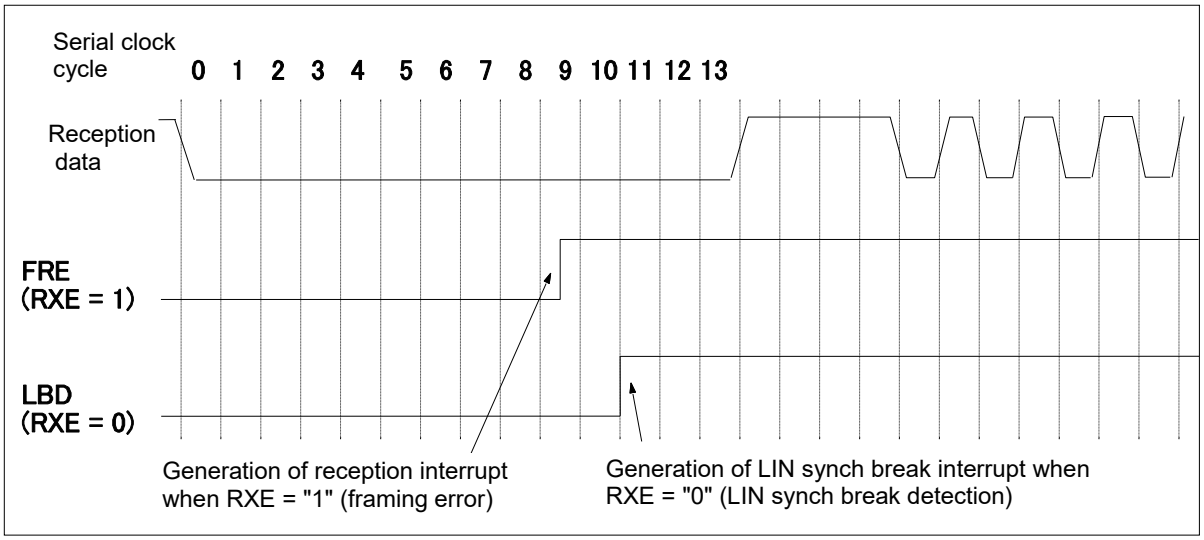
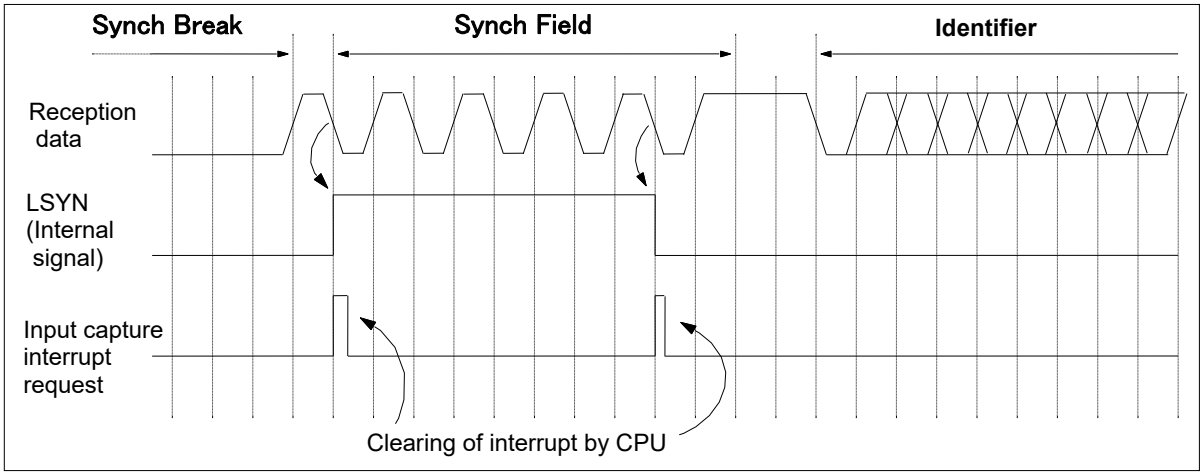


Figure 41-4. LIN Synch Field Edge Detection Interrupt Timing



41.5.2 Generation of Reception Interrupt and Flag Setting Timing

Generation of reception interrupt and flag setting timing is shown below.

This section explains the reception interrupt factors, reception completion (RDRF bit in SSR), and reception error occurrence (PE, ORE, and FRE bits in SSR).

41.5.2.1 Generation of Reception Interrupt and Flag Setting Timing

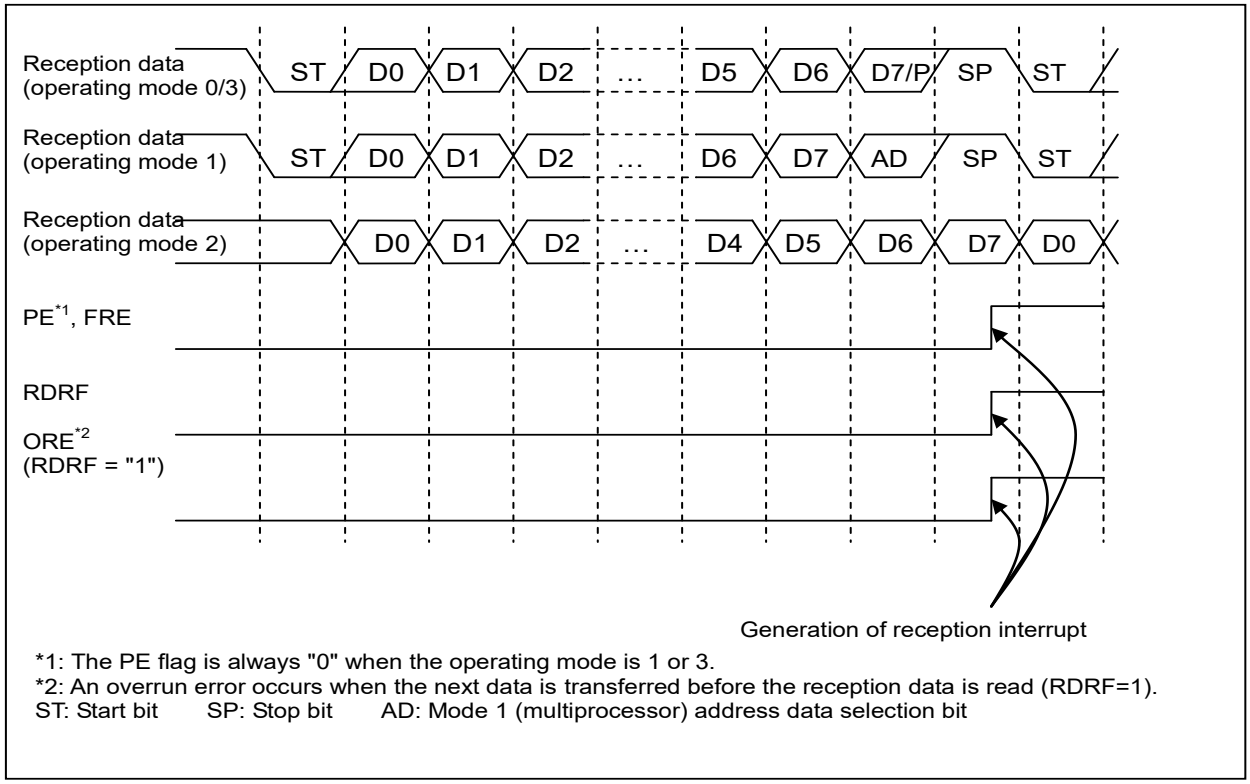
Generation of reception interrupt and flag setting timing is shown below.

When the first stop bit is detected in operation mode 0, 1, 2 (SSM=1), or 3 or the last data bit is detected in operation mode 2 (SSM=0), reception data is stored in the reception 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 interrupt is enabled at this time (SSR:RIE=1), a reception interrupt is generated.

Note:

When a reception error occurs, the data in the reception data register (RDR) becomes invalid in any mode.

Figure 41-5. Example of Reception Operation and Flag Set Timing

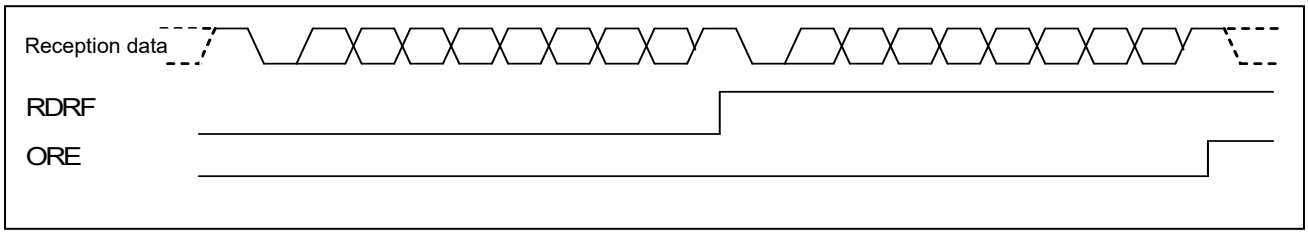


*1: The PE flag is always "0" when the operating mode is 1 or 3.
*2: An overrun error occurs when the next data is transferred before the reception data is read (RDRF=1).
ST: Start bit SP: Stop bit AD: Mode 1 (multiprocessor) address data selection bit

Note:

The above figure does not show all the reception options in mode 0 and mode 3.
In this example, they are "7p1" and "8N1" (p = "even parity" or "odd parity").

Figure 41-6. ORE Set Timing



41.5.3 Occurrence of Transmission Interrupt and Flag Timing

Occurrence of transmission interrupt and flag timing is shown below.

An interrupt occurs when transmission is started after transfer of transmission data from the transmission data register (TDR) to the transmission shift register.

41.5.3.1 Occurrence of Transmission Interrupt and Flag Timing

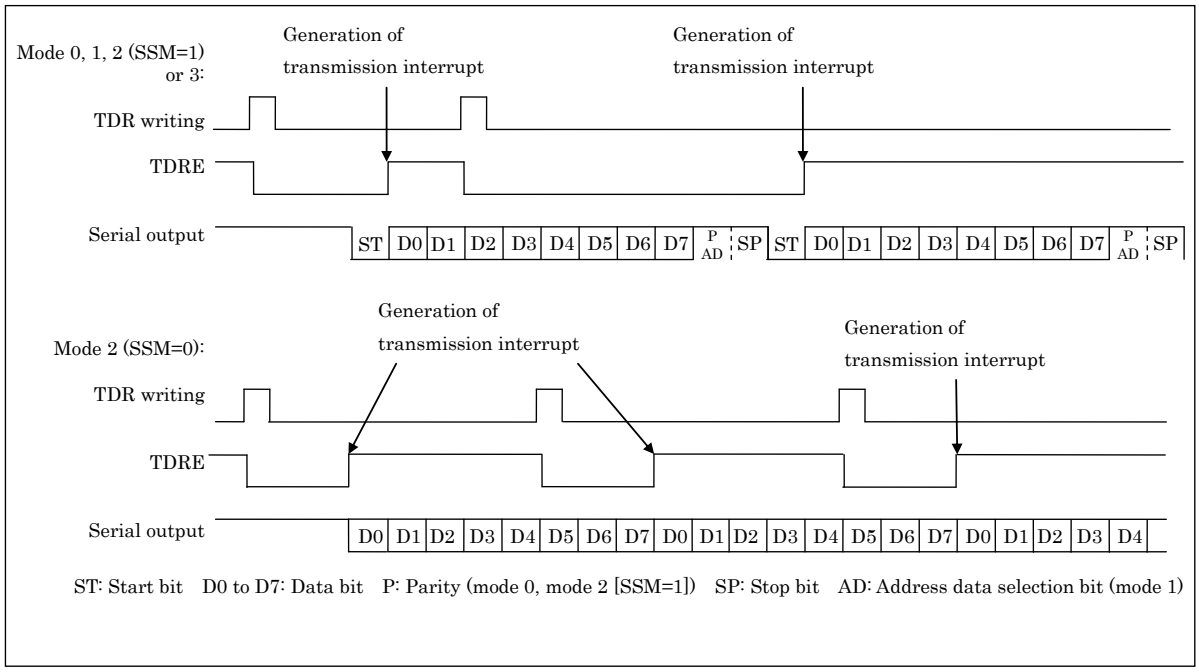
Occurrence of transmission interrupt and flag timing is shown below.

When data written to the transmission data register (TDR) is transferred to the transmission shift register and transmission is started, writing of the next data is enabled (SSR:TDRE=1). If the transmission interrupt is enabled (SSR:TIE=1) at this time, a transmission interrupt is generated.

The TDRE bit is cleared to "0" by the writing of data to the transmission data register (TDR).

The figure below shows an example of the LIN-UART transmission operation and flag set timing:

Figure 41-7. Example of Transmission Operation and Flag Set Timing



Note:

The example in the above figure does not show all the transmission options in mode 0.
In this example, they are "8p1" (p = "even parity" or "odd parity"). If the SSM bit is "0" in mode 3 or mode 2, the parity and the address/data selection bit are not added.

41.5.3.2 Transmission Interrupt Request Generation Timing

Transmission interrupt request generation timing is shown below.

A transmission interrupt request is generated when the TDRE flag is set to "1" while transmission interrupt is enabled (TIE bit in SSR is set to "1").

Note:

The initial value of TDRE is "1". Therefore, a transmission completion interrupt is set as soon as transmission interrupt is enabled (TIE="1"). TDRE is read-only. The TDRE flag is cleared only if data is written to the transmission data register (TDR) or "1" is written to the LIN synch break generation bit (ECCR:LBR).

41.6 Baud Rates

Baud rates are shown below.

The serial clock of LIN-UART can be one of the following:

- Dedicated baud rate generator (reload counter)
- Input of an external clock to the baud rate generator (reload counter)
- External clock (direct use of the SCK pin input clock)

41.6.1 Selection of Baud Rates

Selection of baud rates is shown below.

Figure 41-8 shows the baud rate selection circuit. One of the following 3 baud rates can be selected.

41.6.1.1 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 internal clock is shown below.

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 frequency of the internal clock by the specified value.

This baud rate can be used in asynchronous and synchronous modes (master).

To configure the clock source, select the internal clock and the baud rate generator clock (SMR:EXT=0, OTO=0).

41.6.1.2 Baud Rate Obtained when a Dedicated Baud Rate Generator (Reload Counter) Divides the Frequency of the External Clock

Baud rate obtained when a dedicated baud rate generator (reload counter) divides the frequency of the external clock is shown below.

Use the external clock for the clock source of the 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 frequency of the external clock by the specified value.

Use this baud rate in asynchronous mode.

To configure the clock source, select the external clock and the baud rate generator clock (SMR:EXT=1, OTO=0).

41.6.1.3 Baud Rate Due from External Clock (One-to-one Mode)

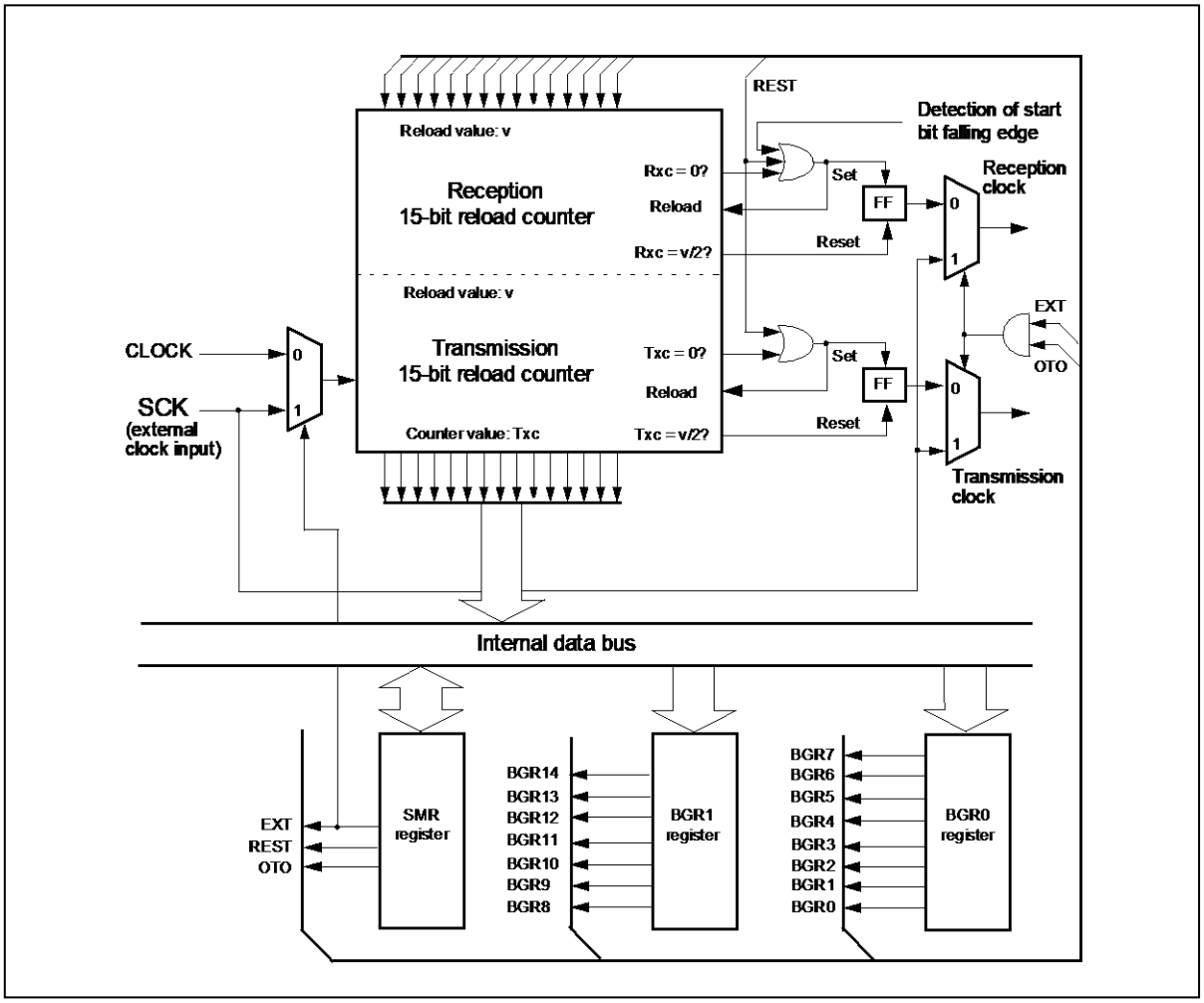
Baud rate due from external clock (one-to-one mode) is shown below.

The clock input from the LIN-UART clock input pin (SCK) is directly used as the baud rate (synchronous mode 2 slave operation (ECCR:MS=1)).

Use this baud rate in synchronous mode (slave).

To configure the clock source, select the external clock and the external clock direct use (SMR:EXT=1, OTO=1).

Figure 41-8. Baud Rate Selection Circuit (Reload Counter)



41.6.2 Baud Rate Setting

Baud rate setting is shown below.

This section indicates how to set the baud rate and the serial clock frequency calculations.

41.6.2.1 Baud Rate Calculations

Baud rate calculations are shown below.

The two 15-bit reload counters are set using the baud rate generator register (BGR).

The baud rate calculation formulas are as follows:

Reload value: v = (φ/b)-1

Baud rate value: b = φ/(v+1)

v: reload value b: baud rate φ: clock frequency

- Example of Calculation

  If the clock is 16 MHz and the target baud rate is 19200 bps, the reload value "v" is calculated in the following manner:

  Reload value:

$$v = \left(\frac{16 \times 10^6}{19200} \right) - 1 = 832$$

  Therefore, the real baud rate can be calculated based on the following:

$$b = \frac{\varphi}{(v + 1)} = \frac{16 \times 10^6}{833} = 19207.6831$$

Note:

Set the reload value to "0" to stop the reload counter. Therefore, the smallest division ratio becomes a division by 2. When sending or receiving is performed in asynchronous mode, it is necessary to set the reload value to 4 or more, because the reception value is determined by executing oversampling five times.

41.6.2.2 Baud Rate Setting Example for Each Clock Frequency

Baud rate setting example for each clock frequency is shown below.

The following table indicates the clock frequency and examples of the baud rate setting.

Table 41-7. Baud Rate Setting Example for Each Clock Frequency

Baud Rate	8 MHz		16 MHz		24 MHz		32 MHz		40 MHz	
	Value	dev.	Value	dev.	Value	dev.	Value	dev.	Value	dev.
4M	-	-	-	-	5	0	7	0	9	0
2M	-	-	7	0	11	0	15	0	19	0
1M	7	0	15	0	23	0	31	0	39	0
500000	15	0	31	0	47	0	63	0	79	0
460800	-	-	-	-	51	-0.16	68	-0.64	86	0.22
250000	31	0	63	0	95	0	127	0	159	0
230400	-	-	-	-	103	-0.16	138	0.08	173	0.22
153600	51	-0.16	103	-0.16	155	-0.16	207	-0.16	259	-0.16
125000	63	0	127	0	191	0	255	0	319	0
115200	68	-0.64	138	0.08	207	-0.16	277	0.08	346	-0.06
76800	103	-0.16	207	-0.16	311	-0.16	416	0.08	520	0.32
57600	138	0.08	277	0.08	416	0.08	555	0.08	693	-0.06
38400	207	-0.16	416	0.08	624	0	832	-0.04	1041	0.03
28800	277	0.08	554	-0.01	832	-0.03	1110	-0.01	1388	0.01
19200	416	0.08	832	-0.03	1249	0	1666	0.02	2082	-0.02
10417	767	0	1535	0	2303	0	3071	0	3839	0.003

Baud Rate	8 MHz		16 MHz		24 MHz		32 MHz		40 MHz	
	Value	dev.	Value	dev.	Value	dev.	Value	dev.	Value	dev.
9600	832	-0.04	1666	0.02	2499	0	3332	-0.01	4166	0.008
7200	1110	-0.01	2221	-0.01	3332	-0.01	4443	-0.01	5555	0.008
4800	1666	0.02	3332	-0.01	4999	0	6666	0	8332	0.004
2400	3332	-0.01	6666	0	9999	0	13332	0	16666	0.002
1200	6666	0	13332	0	19999	0	26666	0	-	-
600	13332	0	26666	0	-	-	-	-	-	-
300	26666	0	-	-	-	-	-	-	-	-

Note:

Unit of deviation is "%".
The greatest synchronous baud rate is the machine clock divided by 6.

41.6.2.3 Use of the External Clock

Use of the external clock is shown below.

When "1" is written in the EXT bit of the serial mode register (SMR), the external clock is chosen. In the baud rate generator, the external clock can be used in the same way as the internal clock.

When using the slave operation in synchronous mode 2, one to one external clock input mode (SMR:OTO=1) is chosen. Enter the external clock directly into the serial clock.

Note:

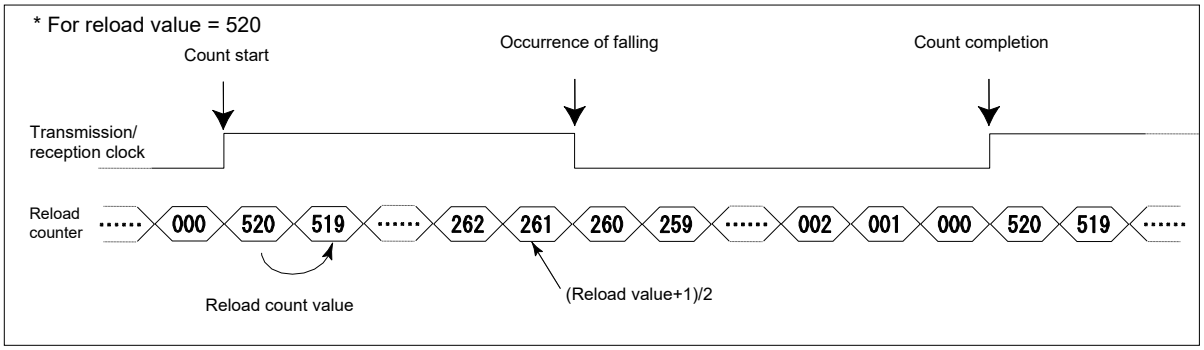
The external clock signals are synchronized with the internal clock by LIN-UART. Therefore, in the case where the external clock cannot be synchronized, LIN-UART will malfunction.

41.6.2.4 Reload Counter Operation

Reload counter operation is shown below.

The following indicates the operation examples of the transmission and reception reload counters.

Figure 41-9. Count Examples of the Reload Counter



Note:

A falling edge of the serial clock signal is generated after counting |(v + 1) / 2| that is the reload value divided by 2.

41.6.3 Reload Counter

Reload counter is shown below.

This is a 15-bit reload counter functioning as a dedicated baud rate generator.

The transmission/reception clock is generated from the external or internal clock.

In addition, the count value of the transmission reload counter can be read from the baud rate generator register (BGR).

41.6.3.1 Reload Counter Functions

Reload counter functions are shown below.

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. In addition, the count value of the transmission reload counter can be read via the baud rate generator register (BGR).

41.6.3.2 Count Start

Count start is shown below.

When a reload value is written to the baud rate generator register (BGR), the reload counter starts counting.

41.6.3.3 Restart

Restart is shown below.

Configure the LIN-UART reset (writing of "1" to SMR:UPCL) or restart (writing of "1" to SMR:REST) to restart both reload counters.

The reception reload counter is also restarted when a falling edge of the start bit is detected in the asynchronous mode, and the reception shift register is synchronized by the reception data.

41.6.3.4 Counter Clear

Counter clear is shown below.

By resetting, the reload value of the baud rate generator register (BGR) and the reload counter are cleared to "00_H", and the reload counter stops.

The counter value is temporarily cleared to "00_H" by LIN-UART reset (writing of "1" to SMR:UPCL), but the reload counter will restart because the reload value has been held. The counter value will not be cleared to "00_H" by the restart configuration (writing of "1" to the SMR:REST).

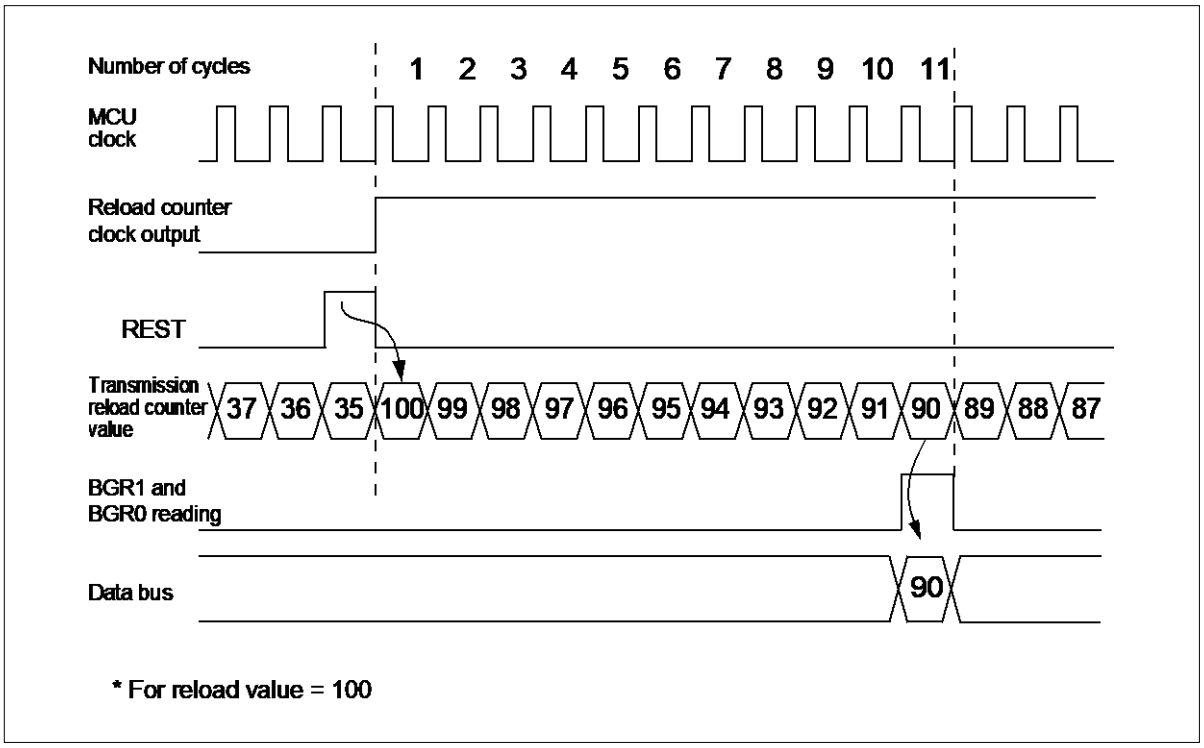
41.6.3.5 Simple Timer Use

Simple timer use is shown below.

The transmission reload counter can also be used as a simple timer.

The figure below indicates an example of usage as a simple timer:

Figure 41-10. Reload Counter Reactivation Example



In the example, the number of MCU clock cycles (cyc) after REST will be as follows:

cyc = v - c + 1 = 100 - 90 + 1 = 11

v: reload value, c: read counter value

41.7 Operation

The operation is shown below.

LIN-UART, in the operation mode 0, will usually operate as bidirectional serial communication. In mode 1, multi-processor communication as master/slave takes place. In mode 2 and mode 3, bidirectional communication as master/slave takes place.

41.7.1 Overview

The overview of the operation is shown.

41.7.1.1 Operation Mode

The operation mode is explained below.

LIN-UART has four operation modes from mode 0 to 3. The following table indicates the selectable operation mode based on the inter-CPU connection method and data transfer:

Table 41-8. LIN-UART Operation Mode

Operation mode		Data length		Synchronization system	Stop bit length	Data bit format
		Parity No	Parity Yes			
0	Normal mode	7 or 8 bits		Asynchronous	1 bit or 2 bits	LSB First or MSBFirst
1	Multi-processor mode	7 bits or 8 bits + 1 bit [1]	-			
2	Normal mode	8 bits		Synchronous	No, 1 bit, 2 bits	
3	LIN Mode	8 bits	-	Asynchronous	1 bit or 2 bits	LSB First

-: Setting is prohibited

[1]: In the multi-processor mode, "+1" is used as a communication control address/data selection bit (AD).

Note:

Mode 1 (Multi-processor mode) at the time of master/slave connection supports the operation of both master and slave. In mode 3, communication format is fixed.
Switch the mode after releasing all LIN-UART transmission/reception and standby operations and then reset (SMR:UPCL=1) LIN-UART.

41.7.1.2 Connection Method between CPUs

The connection method between CPUs is shown below.

Either the external clock one to one connection (normal mode) or the master/slave connection (multiprocessor mode) can be chosen. Whichever system is chosen, data length, parity, synchronization system, etc. must be consistent among all CPUs. The operation mode will be chosen in the following manner:

Choose the operation mode as shown below:

- In the case of one to one connection: It will be necessary to adopt the same system in either operation mode 0 or mode 2 in both CPUs. Choose operation mode 0 for an asynchronous system, and operation mode 2 for a synchronous system. Additionally, in operation mode 2, configure one CPU as master and the other as slave.
- In the case of master/slave connection: choose operation mode 1. Use as a master/slave system.

41.7.1.3 Synchronization System

The synchronization system is shown below.

In an asynchronous system, the receiving clock is synchronized to the falling edge of the reception start bit. In a synchronous system, it can be synchronized by the master clock signal or the clock signal when it operates as a master.

41.7.1.4 Signaling System

The signaling system is shown below.

The signaling system is NRZ (Non Return to Zero).

41.7.1.5 Transmission/Reception Start

Transmission/reception start is shown below.

When "1" is set to the transmission enable bit (SCR:TXE), the transmission will begin.

When "1" is set to the reception enable bit (SCR:RXE), the reception will begin.

41.7.1.6 Stopping of Transmission/Reception

Stopping of transmission/reception is shown below.

When "0" is set to the transmission enable bit (SCR:TXE), the transmission will stop.

When "0" is set to the reception enable bit (SCR:RXE), the reception will stop.

41.7.1.7 Stopping of Transmission/Reception in Progress

Stopping of transmission/reception in progress is shown below.

In the case when a transmission/reception operation is disabled (SCR2:TXE, RXE=0) while it is in progress, the transmission/reception operation will stop immediately. In this case, the data cannot be guaranteed.

41.7.2 Asynchronous Mode (Operation Modes 0 and 1)

Asynchronous mode (Operation modes 0 and 1) is shown below.

In the case of using the operation mode 0 (normal mode) or operation mode 1 (multi-processor mode), the transfer method becomes asynchronous.

41.7.2.1 Transmission/Reception Data Format

The transmission/reception data format is explained.

The transmission/reception data always starts from the start bit ("L" level) and, after the transmission/reception of data has taken place for the specified data bit length, ends at the stop bit ("H" level).

The direction of bit transfer (LSB First or MSB First) is determined by the BDS bit of the serial status register (SSR). If with parity, the parity bit will always be placed between the last data bit and the first stop bit.

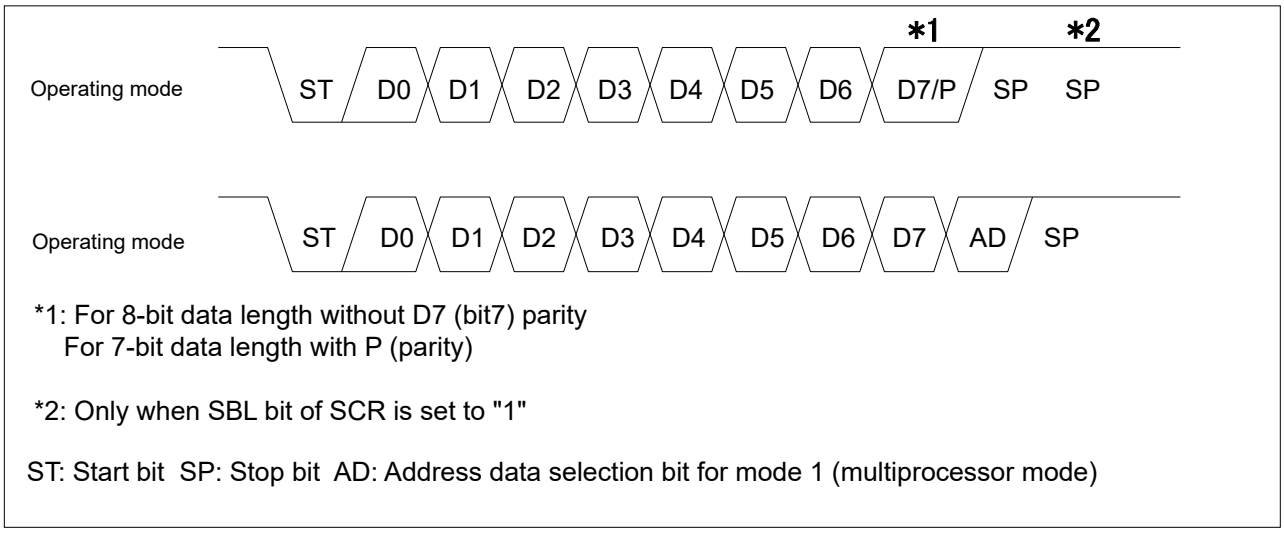
- In operation mode 0, data length of either 7 bits or 8 bits is chosen. Parity or no parity can be selected. A stop bit length (1 or 2) can be selected.
- In operation mode 1, data length is 7 or 8 bits, with no parity added, but with address/data bit added. A stop bit length (1 or 2) can be selected.

Calculation formula for the transfer frame bit length will be as follows:

Length = 1 + d + p + s

(d = number of data bit [7 or 8], p = parity [0 or 1], s = number of stop bit [1 or 2])

Figure 41-11. Example of Transfer Data Format (Operation Modes 0 and 1)



Note:

When the BDS bit of the serial status register (SSR) is set to "1", the bit stream is processed as D7, D6, ... , D1, D0, (P). In addition, in the case of data length of 7 bits, it is processed in the sequence of D6, . .., D1, D0, (P).

41.7.2.2 Transmission Operation

The transmission operation is shown below.

Transmission data is written to the transmission data register (TDR) when there is no transmission data in the transmission data register (TDR) (SSR:TDRE=1). Transmission will start when the transmission operation is subsequently enabled (SCR:TXE=1). The transmission data empty flag bit (SSR:TDRE) becomes "0" when transmission data is written to the transmission data register (TDR).

When transmission data is transferred from the transmission data register (TDR) to the transmission shift register, the transmission data empty flag bit (SSR:TDRE) is set to "1" again. If transmission interrupt is enabled (SSR:TIE=1) at this time, a transmission interrupt request is generated. The following transmission data can be written to the transmission data register (TDR) when processing the interrupt. If data length is configured to 7 bits (CL=0), MSB of the TDR becomes an unused bit, with no regard to the setting of the transfer direction selection bit (BDS) (LSB First or MSB First).

Note:

An interrupt occurs immediately after transmission interrupt is enabled (SSR:TIE) because the initial value of the transmission data empty flag bit (SSR:TDRE) is "1".

41.7.2.3 Reception Operation

The reception operation is shown below.

When the reception operation is enabled (SCR:RXE=1), the reception operation will start. When the start bit is detected, 1-frame data is received according to the data format set in the serial control register (SCR). When an error occurs, the error flag is set (SSR:PE, ORE, FRE). When 1 frame has been received, reception data is transferred from the reception shift register to the reception data register (RDR), setting the reception data register full flag bit (SSR:RDRF) to "1". At this point, if reception interrupt request is enabled (SSR:RIE=1), a reception interrupt request is output.

To read reception data, first check the error flag state after 1-frame data has been received. If reception has been completed successfully, read reception data from the reception data register (RDR). When a reception error has detected, correct the error.

After the reading of reception data, the reception data register full flag bit (SSR:RDRF) will be cleared to "0". If data length is configured to 7 bits (CL=0), MSB of the TDR becomes an unused bit, with no regard to the setting of the transfer direction selection bit (BDS) (LSB First or MSB First).

Note:

If the reception data register full flag bit (SSR:RDRF) is configured to "1" and no error occurs (SSR:PE, ORE, FRE=0), the data in the reception data register (RDR) will be effective.

41.7.2.4 Clock Usage

Clock usage is shown below.

An internal clock or external clock can be used. For the baud rate, select the baud rate generator (SMR:EXT=0 or 1, and OTO=0).

41.7.2.5 Stop Bit

The stop bit is shown below.

It is possible to select a 1-bit or 2-bit stop bit at the time of transmission. If a 2-bit stop bit is selected, only the first stop bit is detected at the time of reception.

If the first stop bit is detected, the reception data register full flag (SSR:RDRF) will be set to "1". If no start bit is detected after that, the reception bus idle flag (ECCR:RBI) will be set to "1" to indicate that the reception operation is not activated.

41.7.2.6 Error Detection

Error detection is shown below.

In operation mode 0, parity errors, overrun errors, framing errors can be detected.

In operation mode 1, overrun errors and framing errors can be detected. Parity errors cannot be detected.

41.7.2.7 Parity

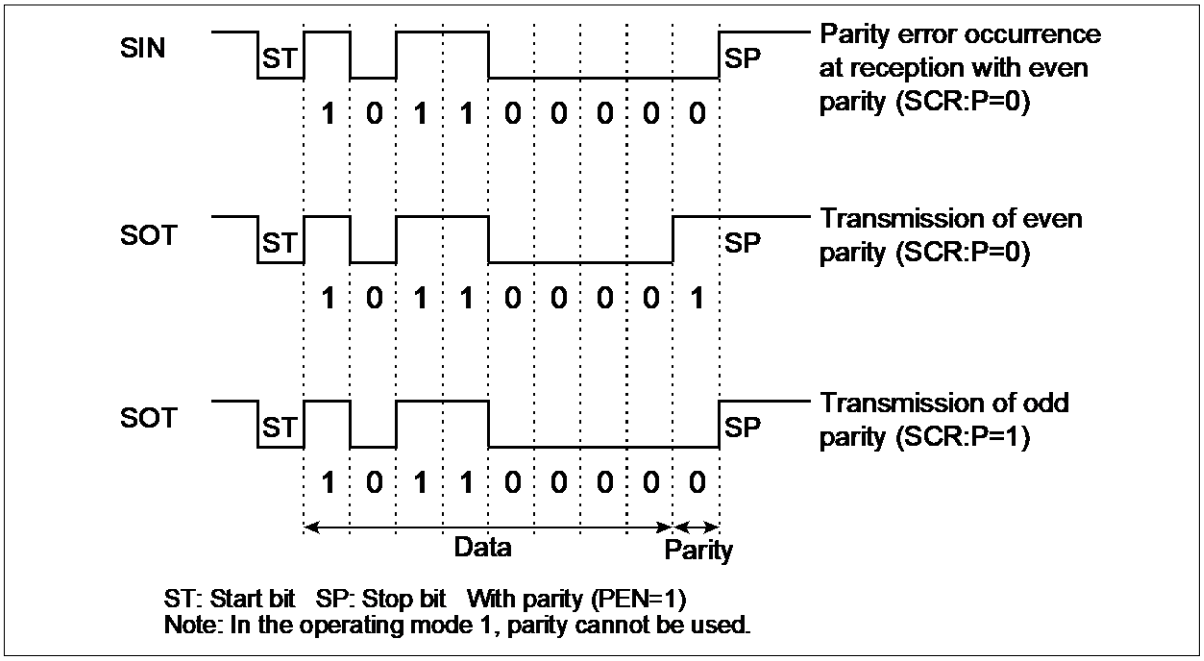
Parity is shown below.

It is possible to set the addition (at the time of transmission) and detection (at the time of reception) of a parity bit.

The parity enable bit (SCR:PEN) can specify whether to enable or disable the parity, and the parity selection bit (SCR:P) can specify whether to use even parity or odd parity.

Operation mode 1 does not use parity.

Figure 41-12. Transmission/Reception Data when Parity is Enabled



41.7.2.8 Data Signaling Method

The data signaling method is shown below.

This is based on the NRZ data format.

41.7.2.9 Data Transfer Method

The data transfer method is shown below.

It is possible to select LSB or MSB First as the data bit transfer method.

41.7.3 Synchronous Mode (Operation Mode 2)

Synchronous mode (Operation Mode 2) is shown below.

In the case of LIN-UART operation mode 2 (normal mode), the transfer method will be clock synchronous.

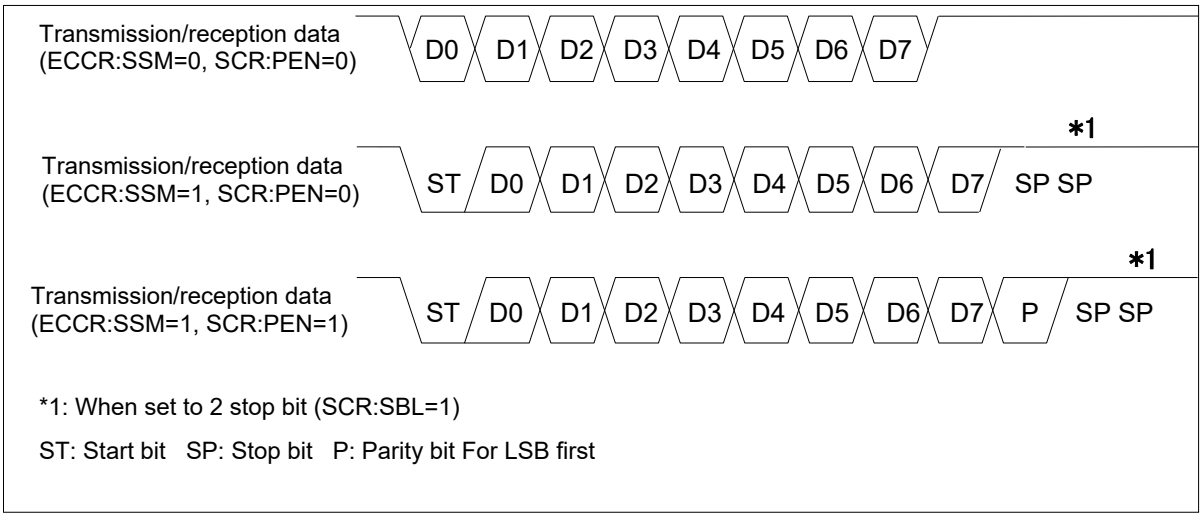
41.7.3.1 Transmission/Reception Data Format

The transmission/reception data format is shown below.

In the synchronous mode, it is possible to transmit and receive 8-bit data to select whether to include the start/stop bit or not (ECCR:SSM). In addition, if the start/stop bit is used (ECCR:SSM=1), you can choose whether to include the parity bit or not (SCR:PEN).

The figure below indicates the data format in the case when synchronous mode is used:

Figure 41-13. Transmission and Reception Data Format (Operation Mode 2)



41.7.3.2 Master/Slave Setting

Master/slave setting is shown below.

In mode 2, you can perform the settings for master and slave.

The master (ECCR:MS=0) generates the serial clock.

The slave (ECCR:MS=1) receives the external clock. Select the external clock and configure it to one-to-one external input (SMR:EXT, OTO=1).

41.7.3.3 Sampling Edge Selection

Sampling edge selection is shown below.

When sampling a data bit, it is possible to choose the sampling edge.

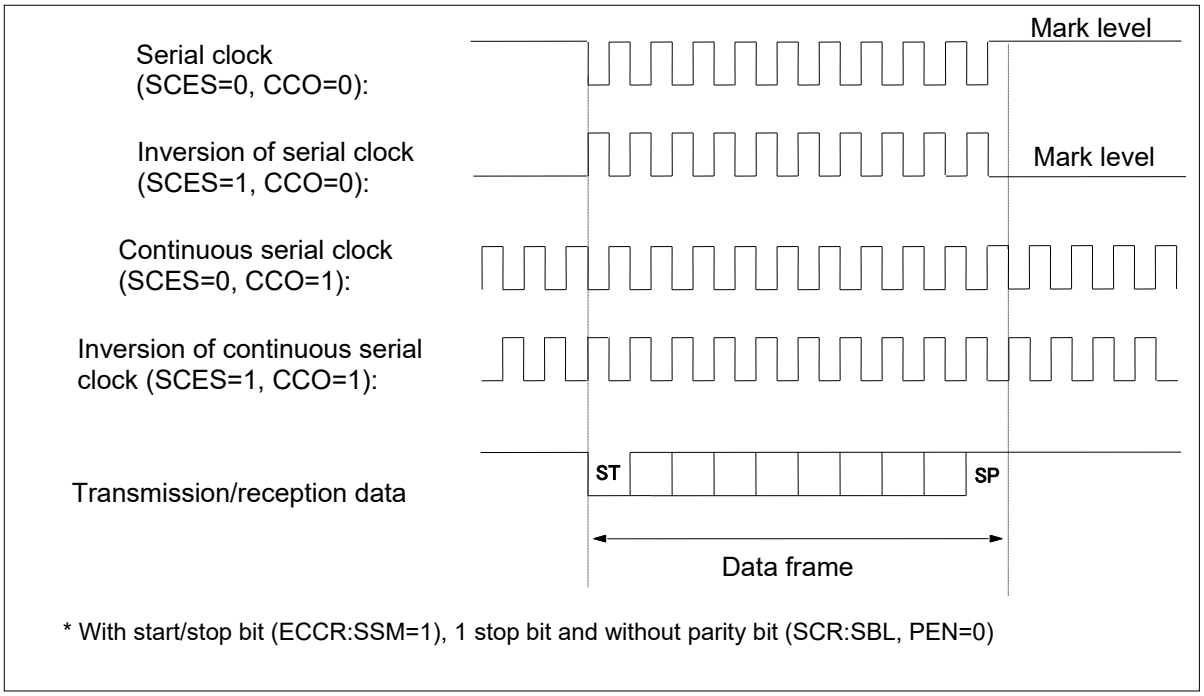
- Sampling in the rising edge (ESCR:SCES=0): Normal clock
- Sampling in the falling edge (ESCR:SCES=1): Clock inversion

As the transmission and reception clock, you can choose the serial clock (normal/delay) and sequential serial clock.

If the sequential serial clock is not used in the master mode (ESCR:CCO=0) and the clock is inverted (ESCR:SCES=1), the clock signal mark level becomes "L".

The figure below indicates a clock which is inverted by the selection of the sampling edge:

Figure 41-14. Inverted Clock by Sampling Edge Selection



* With start/stop bit (ECCR:SSM=1), 1 stop bit and without parity bit (SCR:SBL, PEN=0)

41.7.3.4 Clock Supply

Clock supply is shown below.

It is necessary to supply clocks equivalent to the number of transmission and reception bits in the clock synchronous mode.

Note:

When performing communication with start/stop bit, the number of clock cycles must match the number of added start/stop bits.

41.7.3.5 Clock Usage

Clock usage is shown below.

In the master mode, the internal clock is used. Sending of data automatically generates the synchronous clock for data reception.

As for the baud rate, select the baud rate generator (SMR:EXT=0, OTO=0).

In the slave mode, the external clock is used. It is necessary to supply a clock exactly equivalent to 1 byte from the external source after confirming that there is data in the transmission data register on the transmission side. In addition, it is necessary to ensure that the mark level ("H" if SCES=0, and "L" if SCES=1) is set before and after transmission.

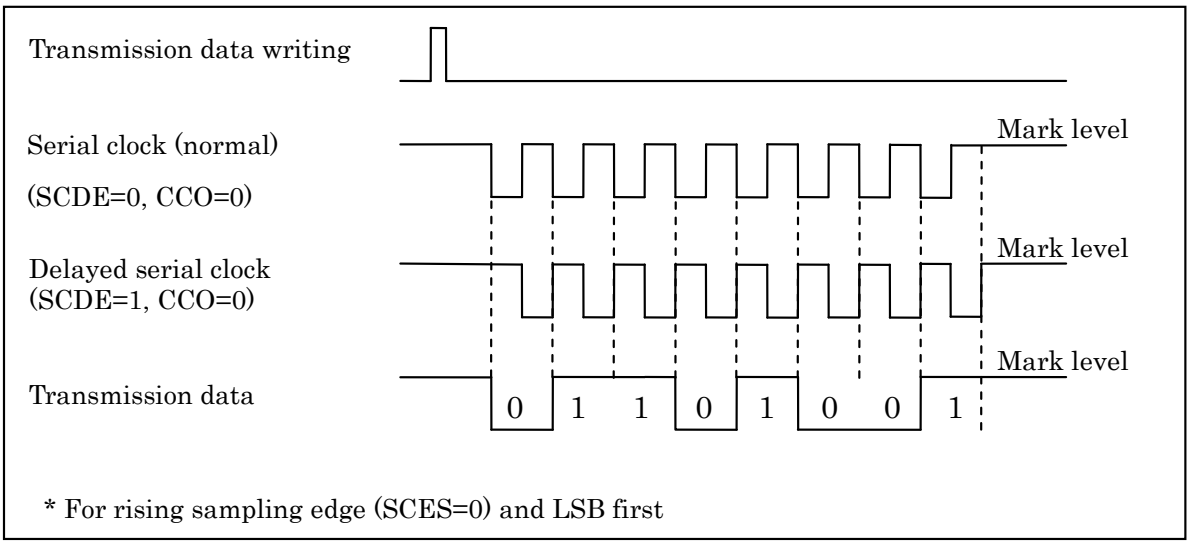
As for the baud rate, select the external clock (one to one) (SMR:EXT=1, OTO=1).

41.7.3.6 Delayed Serial Clock

The delayed serial clock is explained.

Setting the SCDE bit of the ECCR to "1" will output a delayed transmission clock as shown in the figure below. This function is required for the receiving device to sample data at the rising edge or falling edge of the clock.

Figure 41-15. Delayed Serial Clock Output by the Transmission Clock



Note:

If the sequential serial clock is selected for the transmission/reception clock (ESCR:CCO=1), setting the serial clock (normal/ delay) (ECCR:SCDE) will result in sequential serial clock and therefore will not delay.

Please set serial clock delay permission bit (ECCR:SCDE) to “0" when using slave mode (ECCR:MS=1) of the synchronous mode.

41.7.3.7 Sequential Serial Clock

The sequential serial clock is shown below.

If the sequential serial clock is selected, a serial clock is sequentially output from the SCK pin of the master. In addition, when using the sequential serial clock, ensure that the start/stop bit is added (ECCR:SSM=1) to indicate the start/end of transmission and reception.

41.7.3.8 Parity

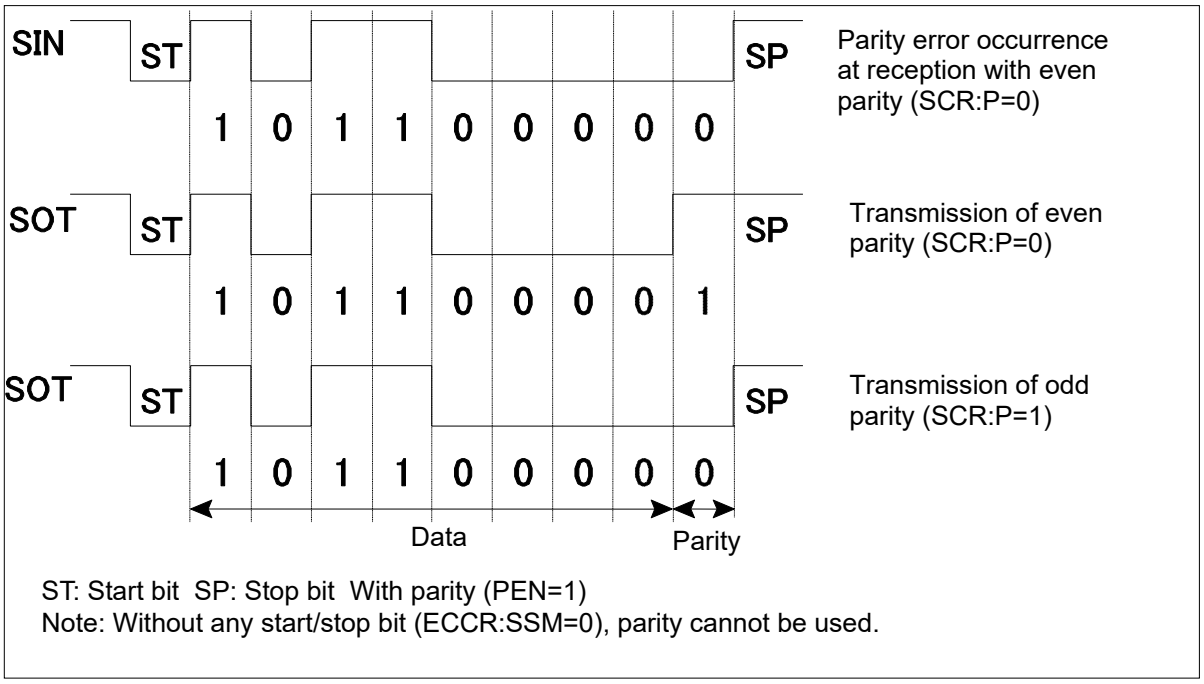
Parity is shown below.

It is possible to set the addition (at the time of transmission) and detection (at the time of reception) of a parity bit.

The parity enable bit (SCR:PEN) can specify whether to enable or disable the parity, and the parity selection bit (SCR:P) can specify whether to use even parity or odd parity.

It is not possible to use parity when there is no start/stop bit.

Figure 41-16. Transmission/Reception Data when Parity is Enabled



41.7.3.9 Data Signaling Method

The data signaling method is shown below.

This is based on the NRZ data format.

41.7.3.10 Stop Bit

The stop bit is shown below.

When transmitting, the stop bit in 1-bit or 2-bit can be selected. When only the first stop bit is received, it is detected when the stop bit in 2-bit is selected.

If the first stop bit is detected, the reception data register full flag (SSR:RDRF) will be set to "1". If no start bit is detected after that, the reception bus idle flag (ECCR:RBI) will be set to "1" to indicate that the reception operation is not activated.

41.7.3.11 Error Detection

Error detection is shown below.

If no start/stop bit exists (ECCR:SSM=0), only overrun error will be detected.

If the start/stop bit and parity bit exist, it is possible to detect parity error, overrun error, and framing error.

41.7.3.12 Communication Start

Communication start is shown below.

Communication starts when data is written to the transmission data register (TDR). Note that, in the case of data reception, it is also always necessary to first disable serial data output (SMR:SOE=0) and then write dummy data to the transmission data register (TDR) in order to start communication.

41.7.3.13 Communication End

Communication end is shown below.

When the transmission/reception of 1-frame data completes, the reception data register full flag bit (SSR:RDRF) will be set to "1". Check the error flag after reception to judge whether communication was performed successfully or not.

Note:

It is possible to configure a duplex communication system, such as asynchronous mode, by using the sequential clock and start/stop bit.

41.7.3.14 Data Transfer Method

The data transfer method is shown below.

It is possible to select LSB or MSB First as the data bit transfer method.

41.7.4 LIN Mode (Operation Mode 3)

The LIN mode (Operation mode 3) is shown below.

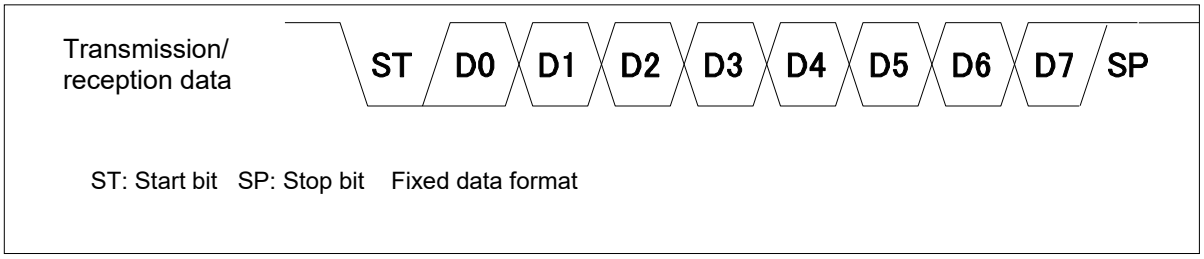
The LIN master/slave function is activated in LIN-UART operation mode 3. Asynchronous mode is adopted as the transfer method.

41.7.4.1 Transmission/Reception Data Format

The transmission/reception data format is shown below.

The data format is fixed in operation mode 3. 8-bit data is transmitted and received with additional start/stop bit, resulting in LSB First. No parity bit is added.

Figure 41-17. Transmission/Reception Data Format



41.7.4.2 LIN Master Operation

The LIN master operation is shown below.

In LIN master mode, all baud rates are determined to synchronize the slave with the master.

LIN communication starts when LIN synch break is transmitted from the master to the slave. LIN synch break generates 13-16 bits of "L" to the SOT pin. Select the length of LIN synch break (ESCR:LBL1/LBL0) to generate LIN synch break (ECCR:LBR=1).

LIN synch field (55_H) is transmitted after the LIN synch break. LIN synch break generation (writing of "1" to ECCR:LBR) changes the status to "transmission data exists" (SSR:TDRE=0). However, it is possible to write 55_H to the transmission data register (TDR). This prevents transmission interrupt from being generated after LIN synch break. Perform asynchronous communication after the LIN synch field. Perform asynchronous communication after the LIN synch field (55_H) has been transmitted.

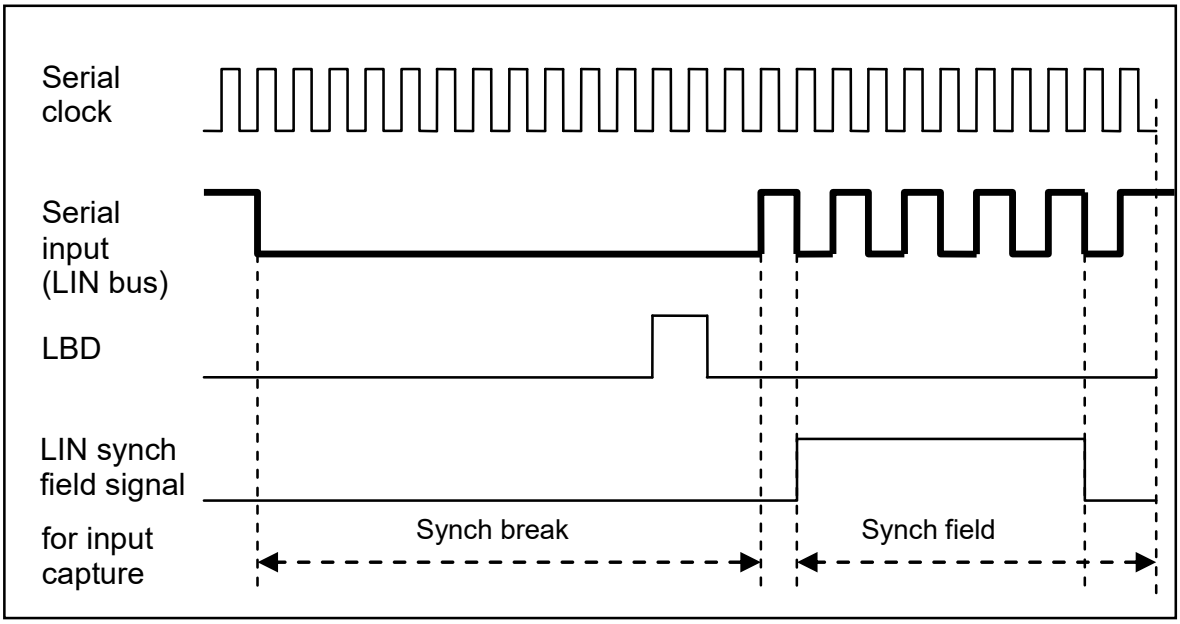
41.7.4.3 LIN Slave Operation

The LIN slave operation is shown below.

The LIN slave mode is synchronized with the master baud rate. When the bus (serial input) indicates "0" for 11-bit time or longer, LIN synch break of the LIN master will be detected (ESCR:LBD=1). To detect LIN synch break, it is necessary to either disable reception (SCR:RXE=0) or disable reception interrupt (SSR:RIE=0). If the LIN synch break interrupt is enabled (ESCR:LBIE=1) at this time, an interrupt is generated. Writing "0" to the LIN synch break detection flag bit (ESCR:LBD) will clear the interrupt. After a LIN synch break is detected, the internal signal is set to "1" at the 1st falling edge of the LIN synch field and set to "0" after the 5th falling edge. When both edges are detected and if input capture interrupt is enabled (ICS:ICE=1), an interrupt will be generated. When the LIN synch field is detected, the internal signal is equivalent to 8 bits of the master serial clock and is counted using input capture. After that, it will be possible to perform asynchronous communication. See "41.7.2 Asynchronous Mode (Operation Modes 0 and 1)".

The figure below shows a typical example of LIN communication frame start and LIN-UART operation.

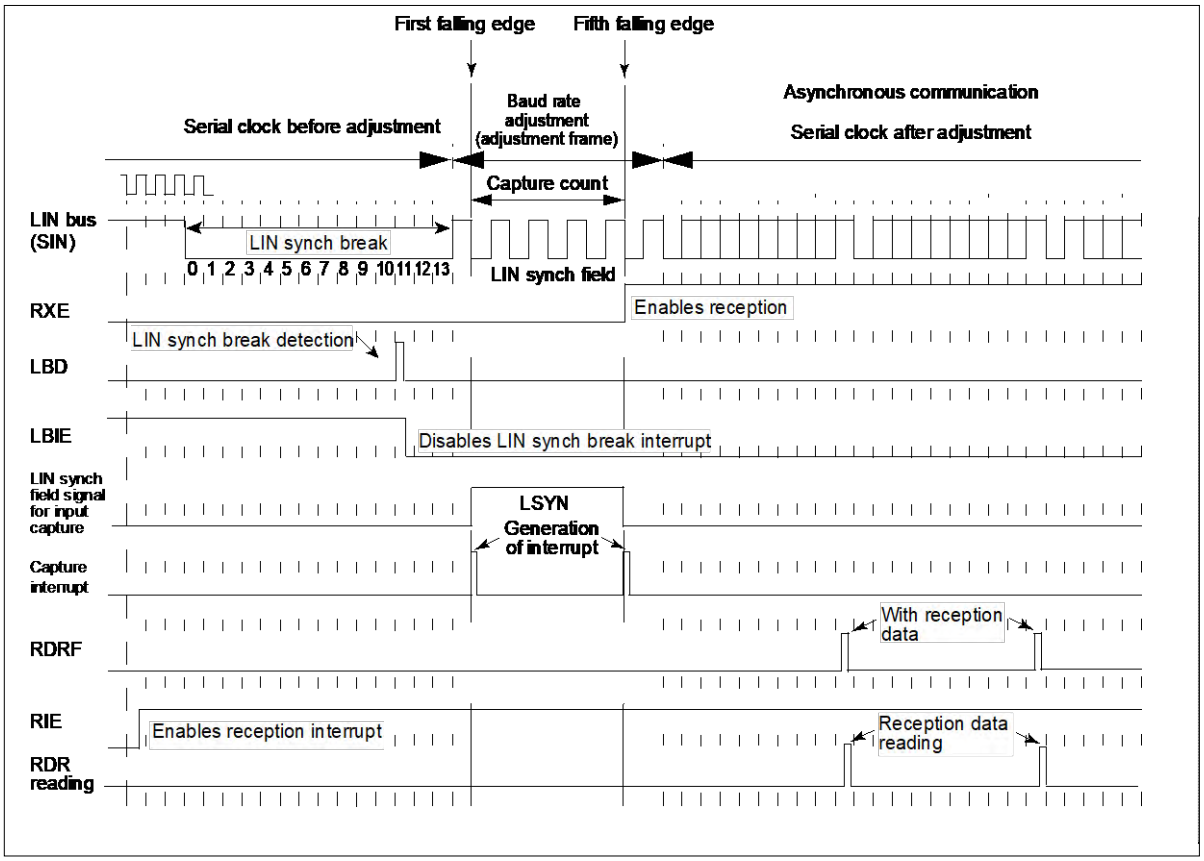
Figure 41-18. LIN Slave Operation



41.7.4.4 LIN Bus Timing

LIN bus timing is shown below.

Figure 41-19. LIN Bus Timing and LIN-UART Signal



41.7.4.5 Baud Rate Calculation

Baud rate calculation is explained.

As an example, the operation of LIN-UART ch.3 is described below. When the LIN-UART ch.3 detects the first falling edge of Synch Field, it sets the internal signal to be input to the input capture (ICU1) to "H" and starts the ICU1. This internal signal becomes "L" on the 5th falling edge. ICU1 needs to be set to LIN mode (LSYNS0:LSYN1). In addition, it is necessary to enable ICU1 interrupt (ICS01:ICE1) and detection of both edges (ICS01:EG11, ICS01:EG10). The time of when the input ICU1 signal is "1" will be equal to the baud rate multiplied by 8. The baud rate setting value can be calculated based on the following formula:

- If the Free-run Timer Has not Overflowed:

  BGR value = {(b - a) × Fe / (8 × φ) - 1

- If the Free-run Timer Has Overflowed:

  BGR value = {(max + 1 + b - a) × Fe / (8 × φ)} – 1

  max: Free-run timer maximum value
  a: ICU data register value after the first interrupt
  b: ICU data register value after the second interrupt
  φ: Machine clock frequency (MHz)
  Fe: External clock frequency (MHz)

  It is assumed that the internal baud rate generator is used (EXT=0) and Fe=φ.

Note:

As described above, do not set the baud rate if an error of baud rate ± 15% or greater occurs in the new BGR value calculated in the Synch field in LIN slave mode.

For the relationship between LIN-UART and ICU, see "21.5 Operation Description" in "Chapter: Free-run Timer" and "24.5 Operation Description" in "Chapter: Input Capture".

41.7.4.6 Clock Usage

Clock usage is shown below.

The internal clock is used. As for the baud rate, select the baud rate generator (SMR:EXT=0 or 1, and OTO=0).

41.7.4.7 Data Signaling Method

The data signaling method is shown below.

This is based on the NRZ data format.

41.7.4.8 Stop Bit

The stop bit is shown below.

When transmitting, the stop bit in 1-bit or 2-bit can be selected. When only the first stop bit is received, it is detected when the stop bit in 2-bit is selected.

If the first stop bit is detected, the reception data register full flag (SSR:RDRF) will be set to "1". If no start bit is detected after that, the reception bus idle flag (ECCR:RBI) will be set to "1" to indicate that the reception operation is not activated.

41.7.4.9 Error Detection

Error detection is shown below.

An overrun error and a framing error can be detected.

41.7.5 Direct Access to the Serial Pin

Direct access to the serial pin is shown below.

LIN-UART can directly access the transmission pin (SOT) and reception pin (SIN).

With LIN-UART, a programmer can directly access the serial I/O pin.

It is possible to read the status of the serial input pin (SIN) using the serial I/O pin direct access bit (ESCR:SIOP).

If serial output is enabled (SMR:SOE=1) after enabling direct writing to the serial output pin (SOT) (ESCR:SOPE=1) and writing "0" or "1" in the serial I/O pin direct access bit (ESCR:SIOP), it will be possible to set any value in the serial output pin (SOT).

In the LIN mode, it is possible to apply this procedure to read the transmitted data or handle errors when the physical LIN bus line signal was incorrect.

Note:

Access is allowed only when no transmission operation is in progress (i.e. when the transmission shift register is empty). Prior to accessing the output pin (SMR:SOE=1), write a value in the serial I/O pin direct access bit (ESCR:SIOP). This task prevents any unexpected level of signal from being output as the previous value is retained in the SIOP bit. In a read-modify-write operation, the value of the SOT pin in the read cycle is returned.

41.7.6 Bidirectional Communication Function (Normal Mode)

The bidirectional communication function (Normal mode) is shown below.

Normal serial bidirectional communication is allowed in operation mode 0 and mode 2. You can select asynchronous communication in operation mode 0, and synchronous communication in operation mode 2.

The figure below shows the LIN-UART settings in normal mode (operation mode 0 and mode 2).

Figure 41-20. LIN-UART Settings in Operation Mode 0 and Mode 2

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 0	◎	◎	◎	◎	×	◎	◎	◎	0	0	0	◎	◎	◎	0	◎
Mode 2 (MS=0)	□	□	□	+	×	◎	◎	◎	1	0	0	◎	◎	◎	1	◎
Mode 2 (MS=1)	□	□	□	+	×	◎	◎	◎	1	0	1	1	◎	◎	0	◎

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR, TDR/RDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Sets conversion data (at writing)/Retains reception data (at reading)							
Mode 0	◎	◎	◎	◎	◎	◎	◎	◎								
Mode 2 (MS=0)	□	◎	□	◎	◎	◎	◎	◎								
Mode 2 (MS=1)	□	◎	□	◎	◎	◎	◎	◎								

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 0	×	×	×	×	◎	◎	×	+	0	0	+	+	+	0	◎	◎
Mode 2 (MS=0)	+	×	×	×	◎	◎	◎	◎	0	+	0	◎	◎	0	×	◎
Mode 2 (MS=1)	+	×	×	×	◎	◎	0	◎	0	+	1	0	◎	0	×	×

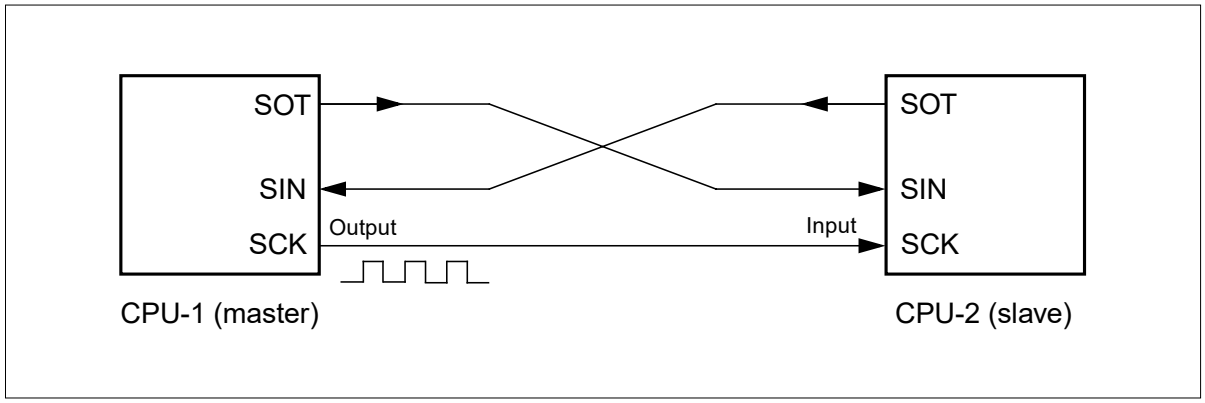
◎ : Available bit
× : Not-available bit
1 : Set to "1"
0 : Set to "0"
□ : Use when SSM="1"
+ : Bit with automatically setting correctly

41.7.6.1 Connection between CPUs

The connection between CPUs is shown below.

The connection between 2 CPUs in LIN-UART mode 2 is shown below:

Figure 41-21. Connection Example for Bidirectional Communication in LIN-UART Operation Mode 2

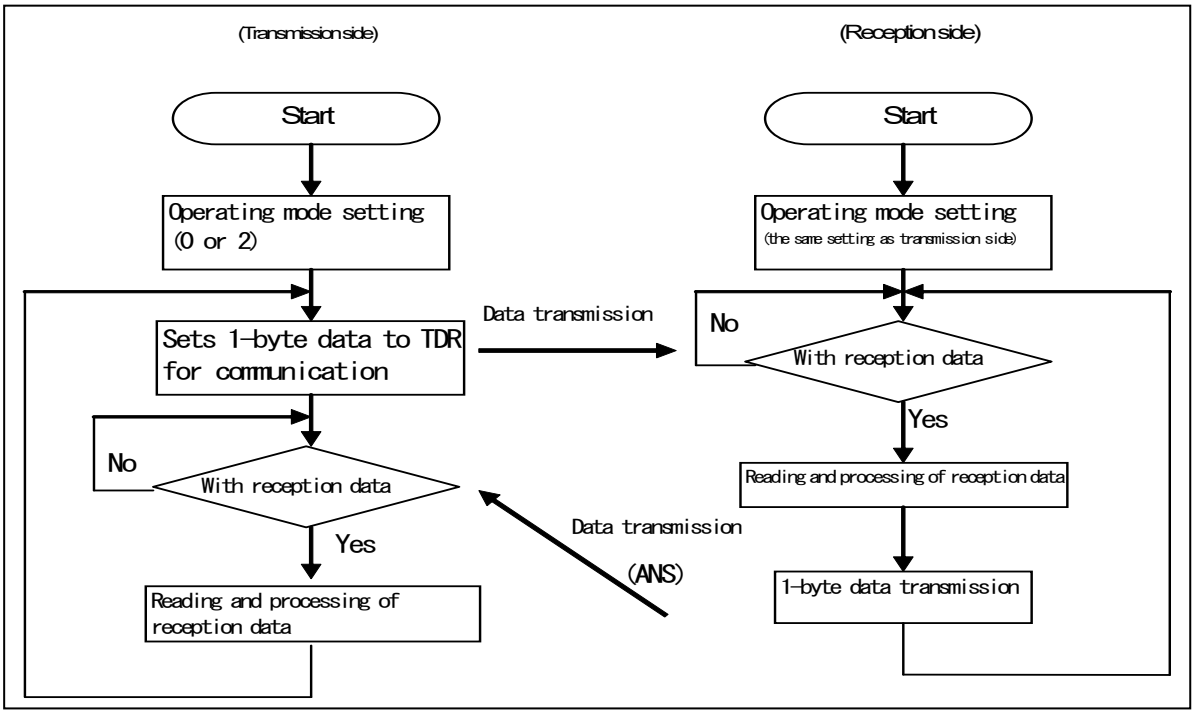


41.7.6.2 Communication Procedure

The communication procedure is shown below.

Communication is triggered by the transmitting side when transmission data becomes ready. When transmission data is received by the receiving side, ANS (1 byte in the example) is returned on a periodic basis. A flowchart example of bidirectional communication is shown below:

Figure 41-22. Example of Bidirectional Communication Flowchart



41.7.7 Master/Slave Mode Communication Function (Multi-processor Mode)

The master/slave mode communication function (Multi-processor mode) is shown below.

In operation mode 1, communications can be performed via master-slave connection between multiple CPUs. It can be used either as a master or slave.

The LIN-UART settings in multi-processor mode (operation mode 1) are shown below:

Figure 41-23. LIN-UART Settings in Operation Mode 1

	bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR, SMR		PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 1		+	×	◎	◎	◎	◎	◎	◎	0	1	0	◎	◎	◎	0	◎

	bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR, TDR/RDR		PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Sets conversion data (at writing)/Retains reception data (at reading)							
Mode 1		×	◎	◎	◎	◎	◎	◎	◎								

	bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESCR, ECCR		LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 1		+	×	×	×	◎	◎	×	+	0	×	+	+	+	0	◎	◎

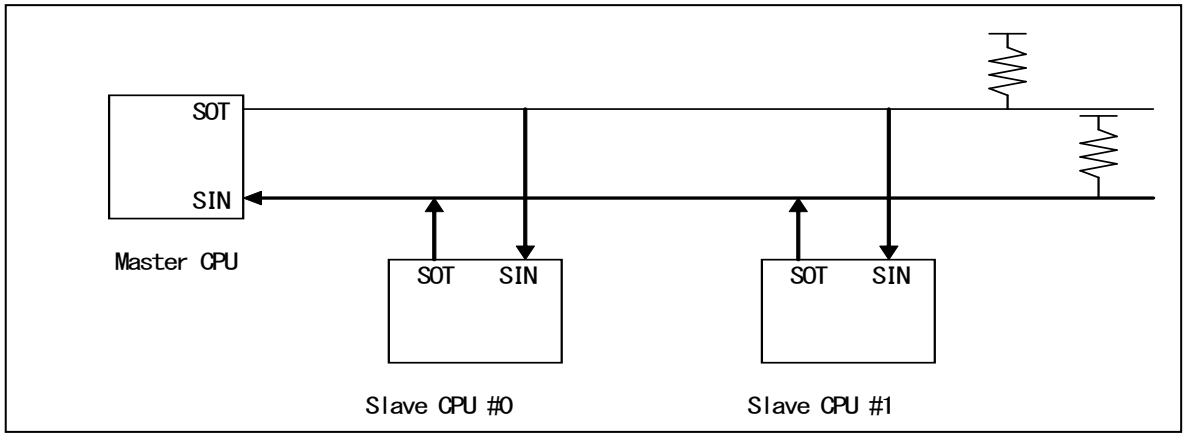
◎ : Available bit
× : Not-available bit
1 : Set to "1"
0 : Set to "0"
□ : Use when SSM="1"
+ : Bit with automatically setting correctly

41.7.7.1 Connection between CPUs

The connection between CPUs is shown below.

The figure below shows a communication system consisting of a master CPU and multiple slave CPUs that are connected with 2 communication lines. LIN-UART can be used either as a master or slave.

Figure 41-24. Example of LIN-UART Master-Slave Communication



41.7.7.2 Function Selection

The function selection is shown below.

For a master/slave communication, select an operation mode and data transfer method as shown in the table below:

Table 41-9. Setting of Master/Slave Communication Function

	Operation Mode		Data	Parity	Synchronization Method	Stop Bit	Bit Direction
	Master CPU	Slave CPU					
Address transmission and reception	Mode 1 (AD bit transmission and reception)		AD="1" + 7 or 8-bit address	No	Asynchronous	1 bit or 2 bits	LSB or MSB First
Data transmission and reception			AD="0" + 7 or 8-bit data				

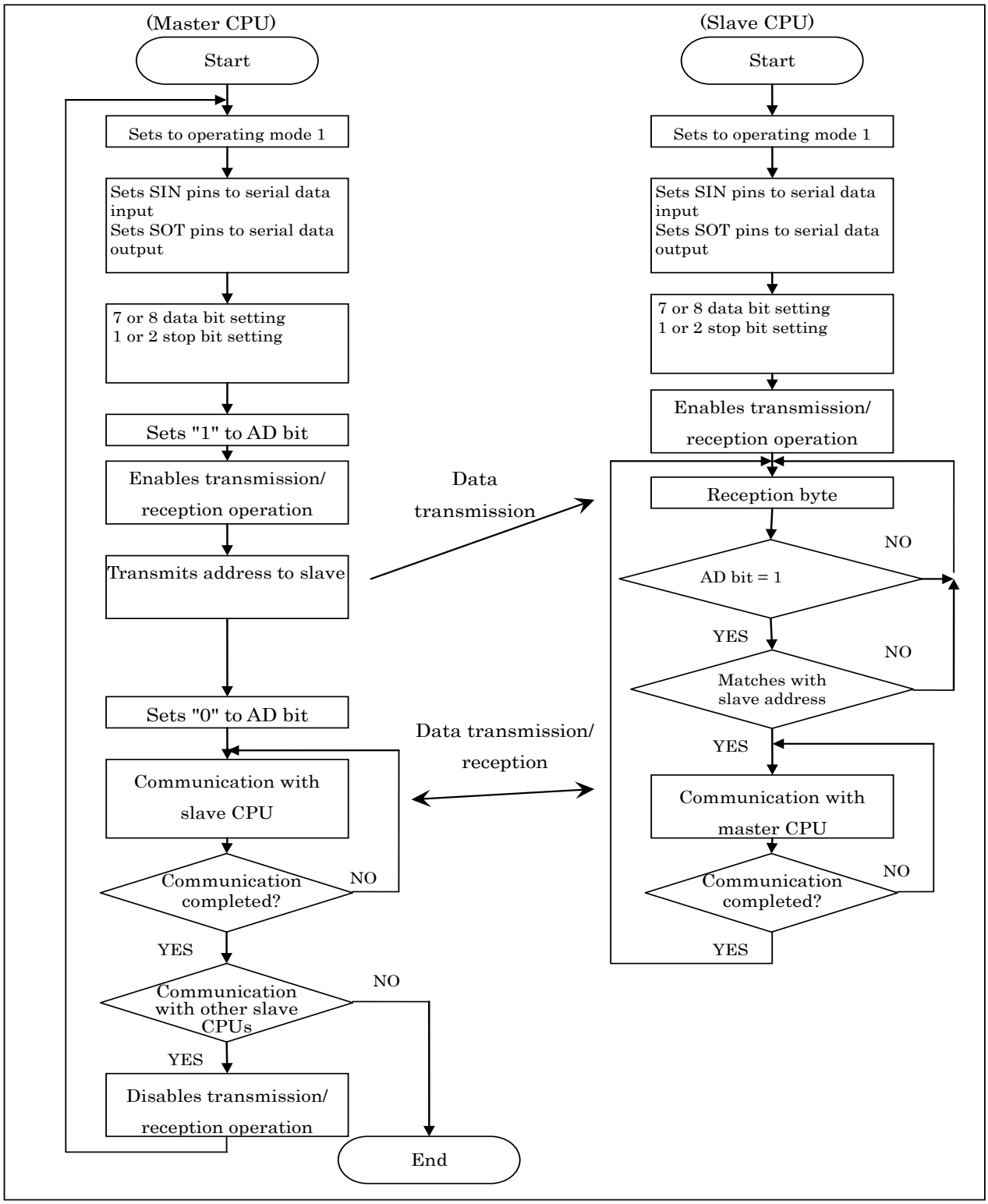
41.7.7.3 Communication Procedure

The communication procedure is shown below.

Communications start when the master CPU transmits address data. Address data refers to data with the AD 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 data with a matching address will communicate with the master CPU.

A flowchart example of a master-slave communication (multi-processor mode) is shown below:

Table 41-10. Master/Slave Communication Flowchart



41.7.8 LIN Communication Function

The LIN communication function is shown below.

LIN master/slave systems can be used in the LIN device during LIN-UART communication.

41.7.8.1 LIN Master/Slave Communication Function

The LIN master/slave communication function is shown below.

The LIN-UART settings in the LIN communication mode (operation mode 3) are shown below:

Figure 41-25. LIN-UART Setting in Operation Mode 3 (LIN)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 3	+	×	◎	+	×	◎	◎	◎	1	1	0	◎	◎	◎	0	◎

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR, TDR/RDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Sets conversion data (at writing)/Retains reception data (at reading)							
Mode 3	×	◎	◎	◎	◎	×	◎	◎								

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
Mode 3	◎	◎	◎	◎	◎	◎	×	+	0	◎	+	+	+	0	◎	◎

◎ : Available bit
× : Not-available bit
1 : Set to "1"
0 : Set to "0"
□ : Use when SSM="1"
+ : Bit with automatically setting correctly

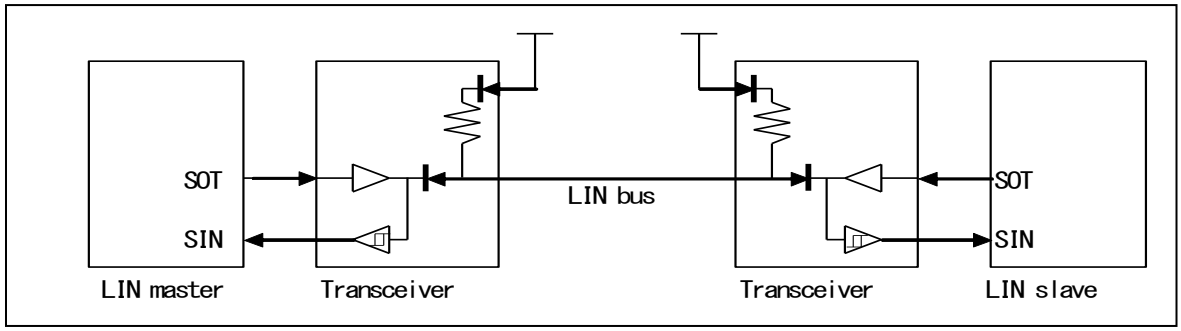
41.7.8.2 LIN Device Connection

The LIN device connection is shown below.

The connection of the LIN master and LIN slave devices is shown below.

LIN-UART can be set as LIN master or LIN slave.

Figure 41-26. Example of LIN Bus System Connection



41.7.9 LIN-UART Sample Flowchart in LIN Communication Mode (Operation Mode 3)

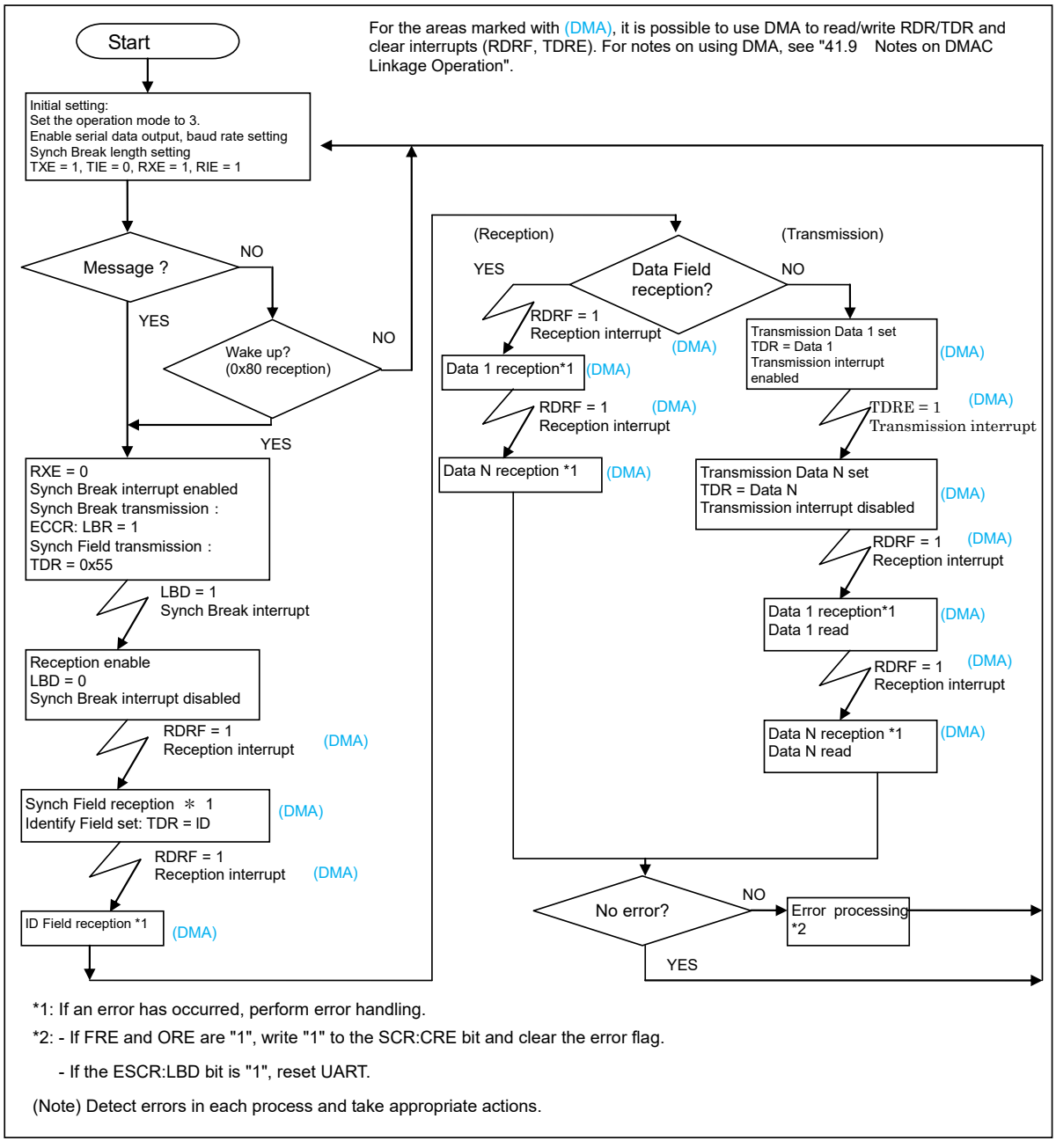
The LIN-UART sample flowchart in LIN communication mode (Operation mode 3) is shown below.

A LIN-UART flowchart example in the LIN communication mode is shown below.

41.7.9.1 LIN-UART as a Master Device

LIN-UART as a master device is shown below.

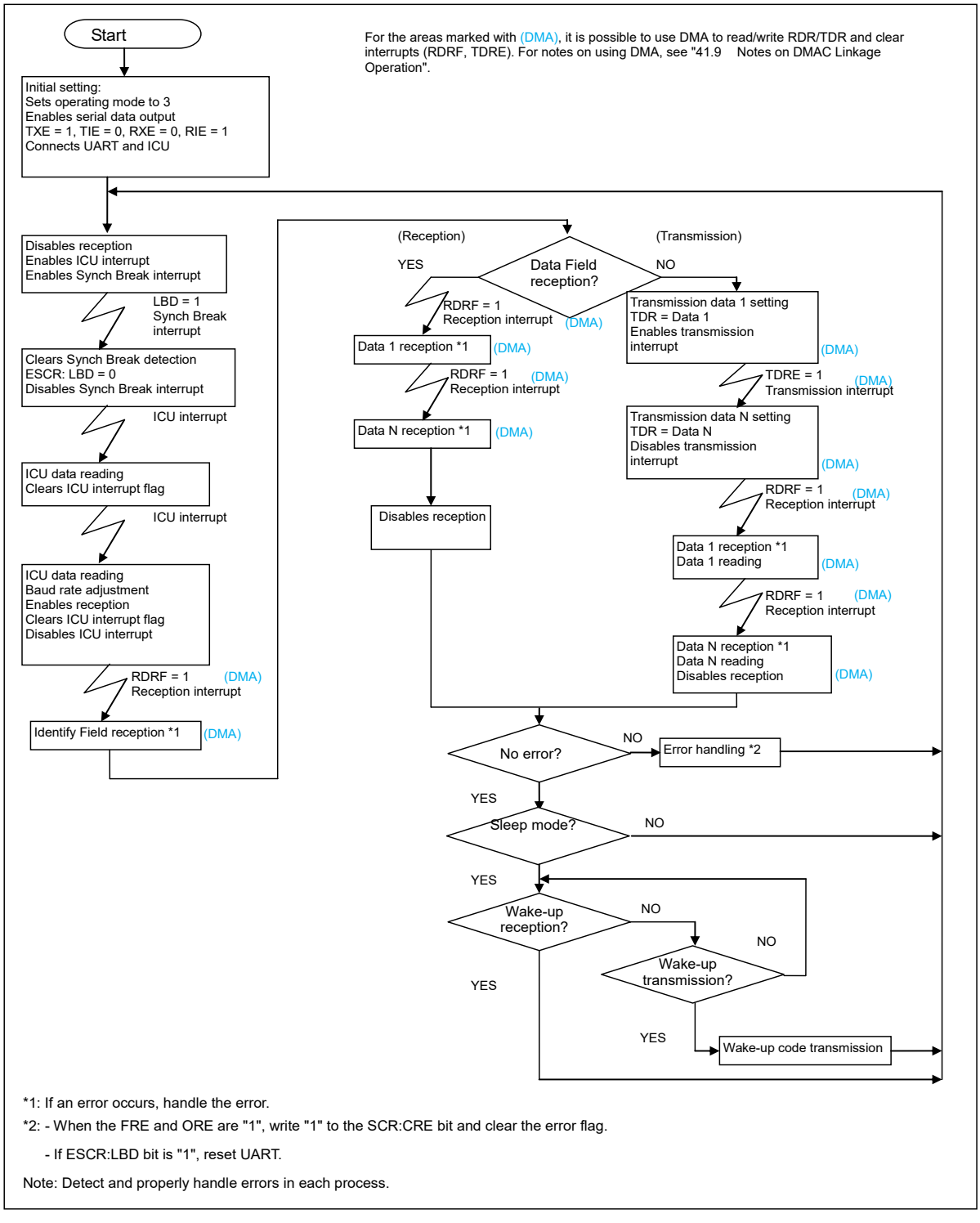
Figure 41-27. LIN-UART Flowchart in LIN Master Mode



41.7.9.2 LIN-UART as a Slave Device

LIN-UART as a slave device is shown below.

Figure 41-28. LIN-UART Flowchart in the LIN Slave Mode



*1: If an error occurs, handle the error.

*2: - When the FRE and ORE are "1", write "1" to the SCR:CRE bit and clear the error flag.

- If ESCR:LBD bit is "1", reset UART.

Note: Detect and properly handle errors in each process.

41.8 Notes on Usage

Notes on usage are shown below.

Notes on using LIN-UART are shown below.

41.8.1 Operation Enable

Operation enable is shown below.

LIN-UART has TXE (transmission) and RXE (reception) operation enable bits in the serial control register (SCR) for transmission and reception, respectively. It is necessary to enable the operation before data transfer because both transmission and reception are disabled in the default setting (initial value). It is also possible to disable operation and cancel data transfer as needed.

41.8.2 Communication Mode Setting

The communication mode setting is shown below.

The communication mode must be changed while the LIN-UART operation is inactive. If the mode is changed while transmission or reception is in progress, the transmitted/received data cannot be guaranteed. Please perform LIN-UART software reset by setting SMR:UPCL(LIN-UART programmable clear bit) after changing communication mode.

41.8.3 Timing of Enabling Transmission Interrupt

Timing of enabling transmission interrupt is shown below.

For the transmission data empty flag bit (SSR:TDRE), the default value (initial value) is set to "1" (no transmission data; writing of transmission data enabled). Therefore, a transmission interrupt request is generated as soon as transmission interrupt request is enabled (SSR:TIE=1). Always set the TIE flag to "1" after setting the transmission data.

41.8.4 Operation Setting Change

The operation setting change is shown below.

After changing any setting, such as adding a start/stop bit or changing the data format, it is recommended that you reset LIN-UART.

Setting the serial mode register (SMR) and resetting LIN-UART (SMR:UPCL=1) at the same time does not guarantee correct operation setting. Therefore, it is recommended that you reset LIN-UART (SMR:UPCL=1) again after setting the bit in the serial mode register (SMR).

41.8.5 Detection of a LIN Synch Break

Detection of a LIN synch break is shown below.

LIN synch break transmission time varies depending on the oscillation accuracy error between master and slave. The slave can detect LIN synch break with the length of 11 serial bits or longer.

If serial input is "0" for 11-bit width or more in mode 3 (LIN mode), LIN synch break is detected (ESCR:LBD=1) and LIN-UART will wait for synch field.

Therefore, if serial input is "0" for 11 bits or more at any point other than LIN synch break, LIN-UART recognizes it as synch break has been input (LBD=1) and will wait for synch field. In this case, reset LIN-UART (SMR:UPCL=1).

41.8.6 LIN Slave Setting

The LIN slave setting is shown below.

To ensure that the minimum 13-bit length of LIN synch break is detected, set the baud rate before receiving the first LIN synch break when activating the LIN slave.

41.8.7 Program Compatibility

The program compatibility is shown below.

LIN-UART is similar to the old FJ-UART. However, these two programs are not compatible. The programming type may be the same in some cases, but the register structure is different. Furthermore, at present, the baud rate setting is determined by the reload value, instead of selecting a predefined value.

41.8.8 Address/Data Format Selection Bit (SCR:AD)

The address/data format selection bit is shown below.

- The AD bit of the serial control register (SCR) performs the transmission address/data selection setting at the time of writing and returns the value of the last received AD bit at the time of reading. Internally, the transmission/reception AD bit values are stored in the individual registers.
- The read-modify-write instruction reads the value of the transmitted AD bit data.
- During the transmission operation (when the TDRE bit changes from "0" to "1"), the transmission AD bit is loaded to the transmission shift register together with data in the transmission data register (TDR). Hence, set the transmission AD bit before writing to the transmission data register (TDR).

41.8.9 LIN-UART Software Reset

LIN-UART software reset is shown below.

Perform LIN-UART software reset (SMR:UPCL=1) when the TXE bit of the serial control register (SCR) is "0".

41.8.10 Detection of LIN Synch Field in Input Capture

Detection of LIN synch field in input capture is shown below.

It is necessary to set the LSYNS0 register in input capture. See "Chapter: Input Capture".

41.8.11 Bus Idle Detection

Bus Idle Detection is shown below.

Reception Bus Idle Detection is not available in Operation mode 2. Transmission Bus Idle Detection is not available when Slave mode is selected.

41.9 Notes on DMAC Linkage Operation

Notes on the DMAC linkage operation are shown below.

LIN-UART transmission and reception interrupts are allocated to the DMAC transfer factor making it possible to write transmission data and read reception data using the DMA data transfer function.

41.9.1 Transmission Operation

The transmission operation is shown below.

Perform dummy writing (writing of any data) to the transmission data register (TDR) before starting the LIN-UART transmission operation (SCR:TXE=1) and before activating the transmission interrupt request enable bit (SSR:TIE=1). In addition, issue a LIN-UART software reset (SMR:UPCL=1) to discard TDR data.

Depending on a previously performed LIN-UART transfer operation (including the case in which DMAC was not used), there is a possibility that an interrupt request to DMAC might not be issued correctly. The purpose of the operation described above is to restore the state required to correctly issue an interrupt request to DMAC.

41.9.2 Reception Operation

The reception operation is shown below.

Perform the reception data register (RDR) read operation before starting the LIN-UART reception operation (SCR:RXE=1) and before activating the reception interrupt enable bit (SSR:RIE=1).

Occurrence of an error during LIN-UART reception and other factors may cause unnecessary reception data to stay in the reception data register (RDR). This will prevent an interrupt request to DMAC from being issued correctly.

It is possible to disable the reception data by issuing a LIN-UART software reset (SMR:UPCL=1). However, read the reception data register (RDR) to ensure that subsequent DMA transfers will be performed correctly.

42. CAN

This chapter explains the CAN.

42.1 Overview
42.2 Features
42.3 Configuration
42.4 Registers
42.5 Operation
42.6 Limitations

42.1 Overview

This section explains the overview of the CAN.

This series includes three CAN channels.

The 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, FA, and other industrial fields.

42.2 Features

This section explains the features of the CAN.

The CAN of this series has the following features:

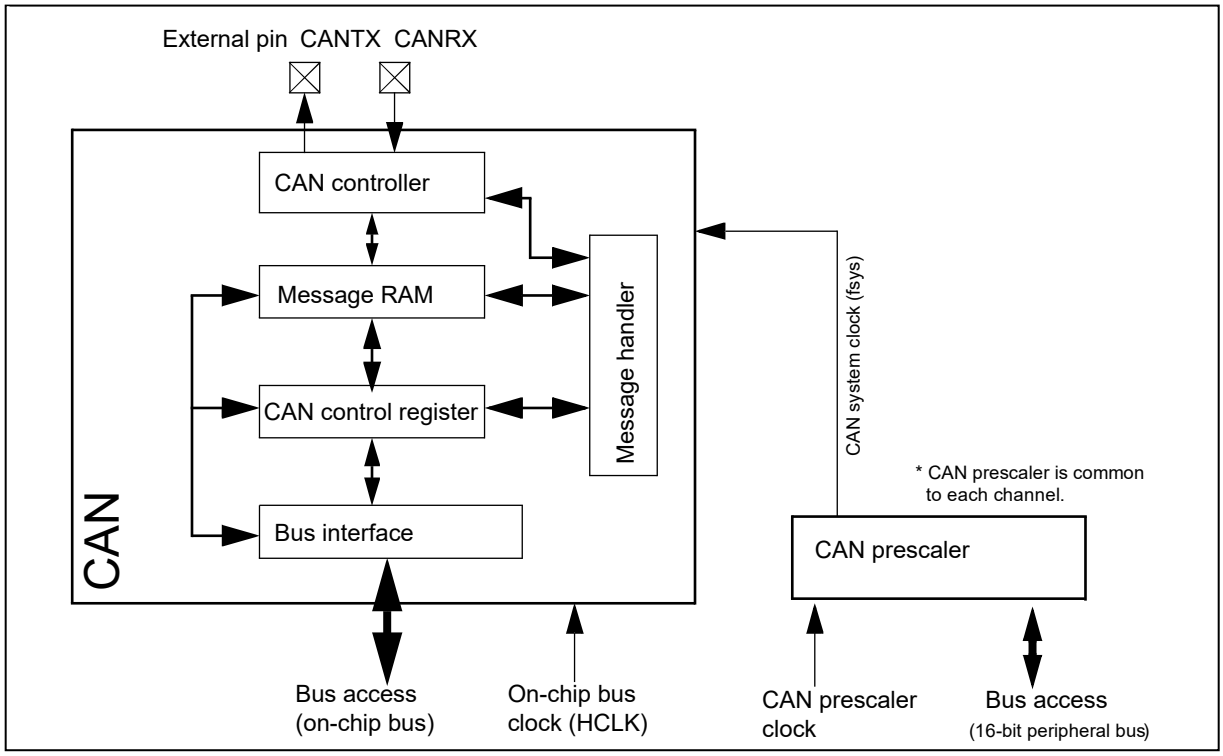
- CAN protocol ver. 2.0A/B is supported.
- Bit rates up to 1 Mbits/s are supported.
- An identification mask is applied to each message object.
- Programmable FIFO mode is supported.
- Maskable interrupts.
- Programmable loopback mode for self-test operation is supported.
- Data can be written to and read from a message buffer using an interface register.
- Support 32/64/128 message buffers. As the number depends on products and channels, see "Chapter: Overview".

42.3 Configuration

This section explains the configuration of the CAN.

A block diagram of the CAN is shown below:

Figure 42-1. Block Diagram of CAN (for one channel)



- CAN controller

  The CAN controller controls the CAN protocol and serial 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 interface with the FR internal bus.

- CAN prescaler

  Generates CAN system clocks (fsys).

42.4 Registers

The registers of the CAN are shown.

42.4.1 Overview
42.4.2 Overall Control Registers
42.4.3 Message Interface Register
42.4.4 Message Object
42.4.6 CAN Prescaler Register (CANPRE)

42.4.1 Overview

This section explains the overview of the registers of the CAN.

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)(IFxMCTR)
- IFx data registers A1, A2, B1, B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)
- CAN transmission request registers 1, 2 (TREQR1,TREQR2)
- CAN New Data registers 1, 2 (NEWDT1,NEWDT2)
- CAN interrupt pending registers 1, 2 (INTPND1,INTPND2)
- CAN message valid registers 1, 2 (MSGVAL1,MSGVAL2)
- CAN clock prescaler register (CANPRE)

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.

List of Base_addresses (Base_addr) and External Pins

Channel Number	Base_addr	External Pin Name	
		CANTX	CANRX
0	0x2000	TX0	RX0
1	0x2100	TX1	RX1
2	0x2200	TX2	RX2

List of Overall Control Register

Table 42-1. 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(CTLR) = 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 pin.
	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(CTLR) = "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	

List of Message Interface Register

Table 42-2. 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	

Address	Registers				Note
	+0	+1	+2	+3	
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	

List of Message Handler Register

Table 42-3. 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)		INTR1, 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 Base_addr + 88_H Base_addr + $8C_H$	Reservation area for supporting 32 or more message buffers (See CAN transmission request registers (TREQR1, TREQR2)) TREQ3 to TREQ4: 64 message buffers are supported TREQ3 to TREQ6: 96 message buffers are supported TREQ3 to TREQ8: 128 message buffers are supported				
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 Base_addr + 98_H Base_addr + $9C_H$	Reservation area for supporting 32 or more message buffers (See CAN data update registers (NEWDT1, NEWDT2)) NEWDT3 to NEWDT4: 64 message buffers are supported NEWDT3 to NEWDT6: 96 message buffers are supported NEWDT3 to NEWDT8: 128 message buffers are supported				
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 + $A4_H$ Base_addr + $A8_H$ Base_addr + AC_H	Reservation area for supporting 32 or more message buffers (See CAN interrupt pending registers (INTPND1, INTPND2)) INTPND3 to INTPND4: 64 message buffers are supported INTPND3 to INTPNT6: 96 message buffers are supported INTPND3 to INTPND8: 128 message buffers are supported				
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 + $B4_H$ Base_addr + $B8_H$ Base_addr + BC_H	Reservation area for supporting 32 or more message buffers (See CAN message valid registers (MSGVAL1, MSGVAL2)) MSGVAL3 to MSGVAL4: 64 message buffers are supported MSGVAL3 to MSGVAL6: 96 message buffers are supported MSGVAL3 to MSGVAL8: 128 message buffers are supported				

Clock Prescaler Register

Table 42-4. Clock Prescaler Register

Address	Registers				Note
	+0	+1	+2	+3	
00_04A4_H	CANPRE	-	-	-	CAN Prescaler
	bit[3:0]	-	-	-	
	CANPRE[3:0]	-	-	-	
	Reset: 00_H	-	-	-	

Overall Control 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)

Message Interface Register

- IFx command request register (IFxCREQ)
- IFx command mask register (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)

Message Handler Register

- CAN transmission request registers 1, 2 (TREQR1, TREQR2)
- CAN data update registers 1, 2 (NEWDT1, NEWDT2)
- CAN interrupt pending registers 1, 2 (INTPND1, INTPND2)
- CAN message valid registers 1, 2 (MSGVAL1, MSGVAL2)

Prescaler Register

- CAN clock prescaler register (CANPRE)

42.4.2 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)

42.4.2.1 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_addr+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	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

CAN Control Register (lower byte): Address Base_addr+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	R0,W0	R/W	R/W	R/W	R/W

[bit15 to bit8] Reserved

The read value is always "0". When writing to these bits, set "0".

[bit7] Test: 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] CCE: Bit timing register write enable bit

CCE	Function
0	Disables the writing to the CAN bit timing register (BTR) and the CAN prescaler extension register (BRPER). [Initial value]
1	Enables 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] DAR: Automatic retransmission disable bit

DAR	Function
0	Enables the automatic retransmission of the message when CAN loses the arbitration or when an error is detected. [Initial value]
1	Disables automatic retransmission.

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.

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 "42.4.4 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 "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 buffers are used are as follows:
- When "1" is set to TxRqst of *other* message buffers (when "1" is set to TxRqst of several message buffers) before CAN starts frame transmission or during transmission, all TxRqst set are reset to "0" and the data of the highest order message buffer is sent when frame transmission is started.
- When frame transmission is completed successfully, NewDat of sent message buffer is reset to "0", and IntPnd of the message object is set to "1" when the TxIE of the message buffer is "1".
- Other message buffers do not send frames at frame transmission start because TxRqst is reset to "0". After the message buffer sent by NewDat or IntPnd is checked, "1" needs to be set to TxRqst and NewDat again for the message buffer to be sent.

[bit4] Reserved

The read value is always "0". When writing to this bit, set "0".

[bit3] EIE: Error interrupt code enable bit

EIE	Function
0	Disables 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	Enables 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] SIE: Status interrupt code enable bit

SIE	Function
0	Disables 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	Enables 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] IE: Interrupt enable bit

IE	Function
0	Disables interrupt. [Initial value]
1	Enables interrupt.

[bit0] Init: Initialization bit

Init	Function
0	Operates 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 one time). The 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 one 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.
- Before transiting to the low-power consumption mode (Stop mode or Clock mode) or before changing the source clock supplied to CAN controller, CAN controller must be initialized by setting Init bit to "1".
- 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)
  - ☐ CAN prescaler register (CANPRE)
- Set "1" to the Init bit after transmission is completed. If "1" is set to the Init bit during transmission, clear the Init bit (Init="0") and request transmission (TxRqst="1") after taking 2 bit-times.

42.4.2.2 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_addr+02H (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

CAN Status Register (lower byte): Address Base_addr+03H (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,WX	R,WX	R,WX	R,W	R,W	R,WX	R,WX	R,WX

[bit15 to bit8]: Reserved

The read value is always "0". When writing to these bits, set "0".

[bit7] BOff: Bus-off bit

BOff	Function
0	Indicates the CAN controller is not in the bus-off state. [Initial value]
1	Indicates the CAN controller is in the bus-off state.

[bit6] EWarn: Warning bit

EWarn	Function
0	Indicates both the transmission and reception counters are below 96. [Initial value]
1	Indicates the transmission or reception counter is 96 or more.

[bit5] EPass: Error passive bit

EPass	Function
0	Indicates both the transmission and reception counters are below 128 (error active state). [Initial value]
1	Indicates the reception counter is the RP bit = "1" and the transmission counter is 128 or more (error passive state).

[bit4] RxOk: Successful message reception bit

RxOk	Function
0	Indicates successful message communication is not performed on the CAN bus or the bus is in the idle state. [Initial value]
1	Indicates successful message communication is performed on the CAN bus.

[bit3] TxOk: Successful message transmission bit

TxOk	Function
0	Indicates the bus is in the idle state or successful message transmission is not performed. [Initial value]
1	Indicates successful message transmission is performed.

Note:

The RxOk and TxOk bits are cleared only with "0" writing.

[bit2 to bit0] LEC[2:0]: Last error code bits

LEC[2:0]	State	Function
000	Normal	Indicates transmission or reception is performed successfully. [Initial value]
001	Stuff error	Indicates more than 6 bits of dominant or recessive is detected continuously in a message.
010	Form error	Indicates the fixed format segment of a received frame is detected as incorrect.
011	Ack error	Indicates the transmission message is not acknowledged by other nodes.
100	Bit1 error	Indicates dominant was detected even though recessive was sent with the message transmission data other than arbitration field.
101	Bit0 error	Indicates 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	Indicates that CRC data and CRC result calculated for a received message did not match.
111	Undetected	Indicates 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)

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.

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).

42.4.2.3 CAN Error Counter (ERRCNT)

The bit configuration of the CAN error counter is shown.

Indicates the reception error passive display, reception error counter, and transmission error counter.

CAN Error Counter Register (upper byte): Address Base_addr+04H (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Error Counter Register (lower byte): Base_addr+05H (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit15] RP: Reception error passive display

RP	Function
0	The reception error counter indicates that it is not the error passive state. [Initial value]
1	The reception error counter indicates that the error passive state that is defined in the CAN specification has been reached.

[bit14 to bit8] REC[6:0]: 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 REC[6:0]=127 is incremented by 8 for reception error, the result is RP=1 and REC[6:0]=127.
When REC[6:0]=126 is incremented by 8 for reception error, the result is RP=1 and REC[6:0]=126.
When REC[6:0]=119 is incremented by 8 for reception error, the result is RP=0 and REC[6:0]=127.

[bit7 to bit0] TEC[7:0]: 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 TEC[7:0]=255 is incremented by 8 for transmission error, the result is Init=1 and TEC[7:0]=255.
When TEC[7:0]=254 is incremented by 8 for transmission error, the result is Init=1 and TEC[7:0]=254.
When TEC[7:0]=247 is incremented by 8 for transmission error, the result is Init=0 and TEC[7:0]=255.

42.4.2.4 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_addr+06H (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	R0,W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CAN Bit Timing Register (lower byte): Address Base_addr+07H (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

The read value is always "0". When writing to this bit, set "0".

[bit14 to bit12] Tseg2: 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] Tseg1: 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] SJW: 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] BRP: 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).

42.4.2.5 CAN Interrupt Register (INTR)

The bit configuration of the CAN interrupt register is shown.

Checks the message interrupt and status interrupt codes.

CAN Interrupt Register (upper byte): Address Base_addr+08$_H$ (Access: Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntId15 to IntId8							
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

CAN Interrupt Register (lower byte): Address Base_addr+09$_H$ (Access: Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntId7 to IntId0							
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

IntId	Function
0000_H	No interrupt
(For 32msg) 0001_H to 0020_H (For 64msg) 0001_H to 0040_H (For 128msg) 0001_H to 0080_H	The message object number is indicated as an interrupt factor. (Message interrupt code)
(For 32msg) 0021_H to $7FFF_H$ (For 64msg) 0041_H to $7FFF_H$ (For 128msg) 0081_H to $7FFF_H$	Unused
8000_H	Indicates interrupts with the change of the CAN status register (STATR). (Status interrupt code)
8001_H to $FFFF_H$	Unused

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, ..., 0020_H) in descending order. (For 32msg. Same for 64 or 128msg.)

- When the IntId[15:0] 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[15:0] 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 "42.4.4 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.

42.4.2.6 CAN Test Register (TESTR)

The bit configuration of the CAN test register is shown.

Monitors the test mode setting and RX pins. For operation, see "42.5.7 Test Mode".

CAN Test Register (upper byte): Address Base_addr+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	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

CAN Test Register (lower byte): Address Base_addr+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,WX	R/W	R/W	R/W	R/W	R/W	R0,W0	R0,W0

The level on the CAN bus is displayed as the initial value (r) of Rx for bit7.

[bit15 to bit8] Reserved

The read value is always "0". When writing to these bits, set "0".

[bit7] Rx: Rx pin monitor bit

Rx	Function
0	Indicates the CAN bus is dominant.
1	Indicates the CAN bus is recessive.

[bit6, bit5] Tx1, Tx0: TX pin control bits

Tx1, Tx0	Function
00	Normal operation [Initial value]
01	Sampling points will be output to the TX pin.
10	Dominant will be output to the TX pin.
11	Recessive will be output to the TX pin.

[bit4] LBack: Loopback Mode

LBack	Function
0	Disables loopback mode. [Initial value]
1	Enables loopback mode.

[bit3] Silent: Silent Mode

Silent	Function
0	Disables silent mode. [Initial value]
1	Enables silent mode.

[bit2] Basic: Basic mode

Basic	Function
0	Disables basic mode. [Initial value]
1	Enables 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

The read value is always "0". When writing to these bits, set "0".

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".

42.4.2.7 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_addr+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	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

CAN Prescaler Extension Register (lower byte): Address Base_addr+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	R0,W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit15 to bit4] Reserved

The read value is always "0". When writing to these bits, set "0".

[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 bit), BRP (LSB:6 bit)} + 1 value is the prescaler of the CAN controller.

42.4.3 Message Interface Register

This section explains the message interface register.

Provides two pairs of message interface registers to control access from the CPU to the message RAM.

There are two pairs of message interface registers used to control access from the CPU to the message RAM. These two 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 "42.4.4 Message Object") transfers messages between the message interface register and the message RAM.

The functions for two pairs of message interface registers are the same expect the test basic mode, 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 42-5 shows two-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 42-5. IF1, IF2 Message Interface Registers

Address	IF1 Register Set	Address	IF2 Register Set
Base_addr+ 10_H	IF1 command request	Base_addr+ 40_H	IF2 command request
Base_addr+ 12_H	IF1 command mask	Base_addr+ 42_H	IF2 command mask
Base_addr+ 14_H	IF1 mask 2	Base_addr+ 44_H	IF2 mask 2
Base_addr+ 16_H	IF1 mask 1	Base_addr+ 46_H	IF2 mask 1
Base_addr+ 18_H	IF1 arbitration 2	Base_addr+ 48_H	IF2 arbitration 2
Base_addr+ $1A_H$	IF1 arbitration 1	Base_addr+ $4A_H$	IF2 arbitration 1
Base_addr+ $1C_H$	IF1 message control	Base_addr+ $4C_H$	IF2 message control
Base_addr+ 20_H	IF1 data A1	Base_addr+ 50_H	IF2 data A1
Base_addr+ 22_H	IF1 data A2	Base_addr+ 52_H	IF2 data A2
Base_addr+ 24_H	IF1 data B1	Base_addr+ 54_H	IF2 data B1
Base_addr+ 26_H	IF1 data B2	Base_addr+ 56_H	IF2 data B2

42.4.3.1 IFx Command Request Register (IFxCREQ)

The bit configuration of the IFx command request register is shown.

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_addr+10H & Base_addr+40H (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	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0	R0,W0

IFx Command Request Register (lower byte): Address Base_addr+11H & Base_addr+41H (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 clock 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

- Other than test basic mode

BUSY	Function
0	Indicates that data is not being transferred between the message interface register and the message RAM. [Initial value]
1	Indicates that data is being transferred between the message interface register and the message RAM.

- Test basic mode

  IF1 command request register

BUSY	Function
0	Disables the message transmission.
1	Enables the message transmission.

  IF2 command request register

BUSY	Function
0	Disables the message reception.
1	Enables the message reception.

[bit14 to bit8] Reserved

The read value is always "0". When writing to these bits, set "0".

[bit7 to bit0] Message Number: Message Number (For 32 message buffer CAN)

Message Number	Function
00_H	Setting prohibited. If this value is set, it is interpreted as 20_H and 20_H is read out.
01_H to 20_H	Sets the message number for processing.
21_H to $3F_H$	Setting prohibited. If this value is set, it is interpreted as 01_H to $1F_H$ and the value interpreted is read out.

[bit7 to bit0] Message Number: Message Number (For 64 message buffer CAN)

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	Sets 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.

[bit7 to bit0] Message Number: Message Number (For 128 message buffer CAN)

Message Number	Function
00_H	Setting prohibited. If this value is set, it is interpreted as 80_H and 80_H is read out.
01_H to 80_H	Sets the message number for processing.
81_H to FF_H	Setting prohibited. If this value is set, it is interpreted as 01_H to $7F_H$ and the value interpreted is read out.

Note:

The BUSY bit is readable/writable. Other than in the test basic mode, it does not affect the operation no matter which value is written to this bit. (See "42.5.7 Test Mode" for the details of the basic mode.)

42.4.3.2 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 test basic mode.

IFx Command Mask Register (upper byte): Address Base_addr+12H & Base_addr+42H (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

IFx Command Mask Register (lower byte): Address Base_addr+13H & Base_addr+43H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	WR/RD	Mask	Arb	Control	CIP	TxRqst/ NewDat	Data A	Data B
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

The read value is always "0". When writing to these bits, set "0".

[bit7] WR/RD: Write/read control bit

WR/RD	Function
0	Indicates 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, Data A, and Data B bits. [Initial value]
1	Indicates 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, Data A, and Data B 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).

- When the transfer direction is write (WR/RD="1")

[bit6] Mask: Mask data update bit

Mask	Function
0	Indicates not updating the mask data (ID mask + MDir + MXtd) of message object[1]. [Initial value]
1	Indicates updating the mask data (ID mask + MDir + MXtd) of message object[1].

[bit5] Arb: Arbitration data update bit

Arb	Function
0	Indicates not updating the arbitration data (ID + Dir + Xtd + MsgVal) of message object[1]. [Initial value]
1	Indicates updating the arbitration data (ID + Dir + Xtd + MsgVal) of message object[1].

[bit4] Control: Control data update bit

Control	Function
0	Indicates not updating the control data (IFx message control register (IFxMCTR)) of message object[1]. [Initial value]
1	Indicates updating the control data (IFx message control register (IFxMCTR)) of message object[1].

[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	Indicates not changing the TxRqst bit of message object[1] and CAN transmission request register (TREQR). [Initial value]
1	Indicates that "1" is set to the TxRqst bit of message object[1] and CAN transmission request register (TREQR) (transmission request) .

[bit1] Data A: Data 0 to 3 update bit

Data A	Function
0	Indicates not updating Data 0 to 3 of message object[1]. [Initial value]
1	Indicates updating Data 0 to 3 of message object[1].

[bit0] Data B: Data 4 to 7 update bit

Data B	Function
0	Indicates not updating Data 4 to 7 of message object[1]. [Initial value]
1	Indicates updating Data 4 to 7 of message object[1].

[1]: See "42.4.4 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 test basic mode.

- When the transfer direction is read (WR/RD="0")

[bit6] Mask: Mask data update bit

Mask	Function
0	Indicates not transferring data (ID mask +MDir + MXtd) from message object[1] to IFx mask registers 1, 2 (IFxMSK1, IFxMSK2). [Initial value]
1	Indicates 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	Indicates not transferring data (ID + Dir + Xtd + MsgVal) from message object[1] to IFx arbitration 1, 2 (IFxARB1, IFxARB2). [Initial value]
1	Indicates 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	Indicates not transferring data from message object*1 to IFx message control register (IFxMCTR). [Initial value]
1	Indicates transferring data from message object*1 to IFx message control register (IFxMCTR).

[bit3] CIP: Interrupt clear bit

CIP	Function
0	Indicates holding the IntPnd bit of message object*1 and CAN interrupt pending register (INTPND). [Initial value]
1	Indicates 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	Indicates holding the NewDat bit of message object[1] and CAN data update register. [Initial value]
1	Indicates clearing the NewDat bit of message object[1] and CAN data update register to "0".

[bit1] Data A: Data 0 to 3 update bit

Data A	Function
0	Indicates holding data of message object[1] and CAN data registers A1, A2. [Initial value]
1	Indicates updating data of message object[1] and CAN data registers A1, A2.

[bit0] Data B: Data 4 to 7 update bit

Data B	Function
0	Indicates holding data of message object[1] and CAN data registers B1, B2. [Initial value]
1	Indicates updating data of message object[1] and CAN data registers B1, B2.

[1]: See "42.4.4 Message Object".

Notes:

- It is possible to reset the IntPnd and NewDat bits to "0" by reading access to message object. However, for the IntPnd and NewDat bits of the IFx message control register (IFxMCTR), the IntPnd and NewDat bits prior to being reset by reading access will be stored.
- It becomes invalid in the test basic mode.

42.4.3.3 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 test basic mode, the configured mask data becomes invalid.

See "42.4.4 Message Object" for the functions of each bit.

IFx Mask Register 2 (upper byte): Address Base_addr+14H & Base_addr+44H (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	R1,W1	R/W	R/W	R/W	R/W	R/W

IFx Mask Register 2 (lower byte): Address Base_addr+15H & Base_addr+45H (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_addr+16H & Base_addr+46H (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_addr+17H & Base_addr+47H (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. When writing to this bit, set "1".

42.4.3.4 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 test basic mode.

See "42.4.4 Message Object" for the functions of each bit.

IFx Arbitration Register 2 (upper byte): Address Base_addr+18H & Base_addr+48H (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_addr+19H & Base_addr+49H (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_addr+1AH & Base_addr+4AH (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_addr+1BH & Base_addr+4BH (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.

42.4.3.5 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 test basic mode. NewDat and MsgLst of the IF2 message control register will operate normally and the DLC[3:0] bits will display the DLC of message received. Other control bits will operate as disabled ("0").

See "42.4.4 Message Object" for the functions of each bit.

IFx Message Control Register (upper byte): Address Base_addr+1C$_H$ & Base_addr+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_addr+1D$_H$ & Base_addr+4D$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	EoB	Reserved	Reserved	Reserved	DLC3-0			
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

Notes:

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".

42.4.3.6 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 Data A or Data B. It is therefore not possible to update only a part of the 4-byte data.

42.4.4 Message Object

The message object is explained.

The message RAM has 32 (up to 64 or 128 depending on the channel) 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.

42.4.4.1 Configuration of Message Object

The configuration of the message object is shown.

The configuration of the message object is shown below:

Table 42-6. Configuration of Message Object

UMask	Msk28-0	MXtd	MDir	EoB	NewDat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID28-0	Xtd	Dir	DLC3-0	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 bit of message RAM to "0".

42.4.4.2 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, Msk28 to Msk0, MXtd and MDir bits will be used 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 one or more message objects, it will be stored to the lowest matched message number. (See reception message acceptance filter in "42.5.3 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	Does not use Msk28 to 0, MXtd, and MDir.
1	Uses Msk28 to 0, 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.

ID28 to ID0: Message ID

ID	Function
ID28 to ID0	Instructs a 29-bit ID (extended frame).
ID28 to ID18	Instructs an 11-bit ID (standard frame).

Msk28 to Msk0: ID Mask

Msk	Function
0	Masks the bit corresponding to the message object ID.
1	Does 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	Does not compare the values between those set to the Xtd bit of the message object and those for the IDE bit in the received frame. The IDE bit in the received frame determines whether to compare it as a standard frame ID or an extended frame ID.
1	Compares the values between those set to the Xtd bit of the message object and those for the IDE bit 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	Indicates the reception direction. The remote frame will be transmitted when the TxRqst bit is set to "1", and the data frame that has passed through the acceptance filter will be received when it is set to "0".
1	Indicates the transmission direction. Data frame will be transmitted when the TxRqst bit is set to "1". If the TxRqst bit is "0" and the RmtEn is set to "1", the CAN controller itself sets its 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	Masks the message direction bit (Dir) in the acceptance filter.
1	Does 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 "42.5.4 FIFO Buffer Function" for details)

EoB	Function
0	Indicates that the massage object is used as FIFO buffer and is not the final message.
1	Indicates 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 32 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 bit remains unchanged after successful frame reception.
1	The IntPnd bit is set to "1" after successful frame reception.

TxIE: Transmission interrupt flag enable bit

TxIE	Function
0	The IntPnd bit remains unchanged after successful frame transmission.
1	The IntPnd bit 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 bit remains unchanged by remote frame reception.
1	The TxRqst bit will be set to "1" if a remote frame is received while the Dir bit is "1".

Notes:

- 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 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	Indicates the transmission idle state (neither transmission is in progress nor in the transmission wait state).
1	Indicates 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	First data byte of the CAN data frame
Data 1	Second data byte of the CAN data frame
Data 2	Third data byte of the CAN data frame
Data 3	Fourth data byte of the CAN data frame
Data 4	Fifth data byte of the CAN data frame
Data 5	Sixth data byte of the CAN data frame
Data 6	Seventh data byte of the CAN data frame
Data 7	Eighth 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 Data A or Data B. It is therefore not possible to update only a part of the 4-byte data.

42.4.5 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, 2 (TREQR1, TREQR2)
- CAN data update registers 1, 2 (NEWDT1, NEWDT2)
- CAN interrupt pending registers 1, 2 (INTPND1, INTPND2)
- CAN message valid registers 1, 2 (MSGVAL1, MSGVAL2)

42.4.5.1 CAN Transmission Request Registers (TREQR1, TREQR2)

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 2 (upper byte): Address Base_addr+ 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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Transmission Request Register 2 (lower byte): Address Base_addr+ 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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Transmission Request Register 1 (upper byte): Address Base_addr+ 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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Transmission Request Register 1 (lower byte): Address Base_addr+ 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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

TxRqst32 to TxRqst1: Transmission request bits

TxRqst32 to 1	Function
0	Indicates the transmission idle state (neither transmission is in progress nor in the transmission wait state).
1	Indicates that transmission is in progress or in the transmission wait state.

Set/reset conditions of the TxRqst bits are shown below.

Set condition

It is possible to set the TxRqst of a specific object by setting "1" to the WR/RD of the IFx command mask register (IFxCMSK) and "1" to the TxRqst while 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 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".

See the following table to confirm the transmission request bit for CAN macro equipped with 32 message buffers of higher.

Table 42-7. Transmission Request Bit for CAN Macro Equipped with 32 Message Buffers or Higher

		addr + 0	addr + 1	addr + 2	addr + 3
TREQR 4 & 3	TxRqst 64 to 33 (address 84_H)	TxRqst64 to 57	TxRqst56 to 49	TxRqst48 to 41	TxRqst40 to 33
TREQR 6 & 5	TxRqst 96 to 65 (address 88_H)	TxRqst96 to 89	TxRqst88 to 81	TxRqst80 to 73	TxRqst72 to 65
TREQR 8 & 7	TxRqst 128 to 97 (address $8C_H$)	TxRqst128 to 121	TxRqst120 to 113	TxRqst112 to 105	TxRqst104 to 97

Notes:

- When the message buffer 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 buffer.
  - ☐ 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 buffer with the lowest priority as the transmission message buffer 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".

42.4.5.2 CAN Data Update Registers (NEWDT1, NEWDT2)

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 2 (upper byte): Address Base_addr+ 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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Data Update Register 2 (lower byte): Address Base_addr+ 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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Data Update Register 1 (upper byte): Address Base_addr+ 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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Data Update Register 1 (lower byte): Address Base_addr+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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

NewDat32 to NewDat1: Data update bits

NewDat 32 to 1	Function
0	Indicates no valid data exists
1	Indicates 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).

See the following table to confirm the data update bit for CAN macro equipped with 32 message buffers or higher.

Table 42-8. Data Update Bit for CAN Macro Equipped with 32 Message Buffers or Higher

		addr + 0	addr + 1	addr + 2	addr + 3
NEWDT 4 & 3	NewDat 64 to 33 (address 94_H)	NewDat64 to 57	NewDat56 to 49	NewDat48 to 41	NewDat40 to 33
NEWDT 6 & 5	NewDat 96 to 65 (address 98_H)	NewDat96 to 89	NewDat88 to 81	NewDat80 to 73	NewDat72 to 65
NEWDT 8 & 7	NewDat 128 to 97 (address $9C_H$)	NewDat128 to 121	NewDat120-113	NewDat112 to 105	NewDat104 to 97

42.4.5.3 CAN Interrupt Pending Registers (INTPND1, INTPND2)

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 2 (upper byte): Address Base_addr+ $A0_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	IntPnd32 to IndPnd25							
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

CAN Interrupt Pending Register 2 (lower byte): Address Base_addr+ $A1_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	IntPnd24 to IndPnd17							
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

CAN Interrupt Pending Register 1 (upper byte): Address Base_addr+ $A2_H$ (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Interrupt Pending Register 1 (lower byte): Address Base_addr+ $A3_H$ (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

IntPnd32 to IntPnd1: Interrupt pending bits

IntPnd32 to 1	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 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 (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 of a specific object by writing data to the IFx command request register.

See the following table to confirm the interrupt pending bit for CAN macro equipped with 32 message buffers or higher.

Table 42-9. Interrupt Pending Bit for CAN Macro Equipped with 32 Message Buffers or Higher

		addr + 0	addr + 1	addr + 2	addr + 3
INTPND 4 & 3	IntPnd 64 to 33 (address $A4_H$)	IntPnd64 to 57	IntPnd56 to 49	IntPnd48 to 41	IntPnd40 to 33
INTPND 6 & 5	IntPnd 96 to 65 (address $A8_H$)	IntPnd96 to 89	IntPnd88 to 81	IntPnd80 to 73	IntPnd72 to 65
INTPND 8 & 7	IntPnd 128 to 97 (address AC_H)	IntPnd128 to 121	IntPnd120 to 113	IntPnd112 to 105	IntPnd104 to 97

42.4.5.4 CAN Message Valid Registers (MSGVAL1, MSGVAL2)

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 2 (upper byte): Address Base_addr+ $B0_H$ (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Message Valid Register 2 (lower byte): Address Base_addr+ $B1_H$ (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Message Valid Register 1 (upper byte): Address Base_addr+ $B2_H$ (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

CAN Message Valid Register 1 (lower byte): Address Base_addr+ $B3_H$ (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,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

MsgVal32 to MsgVal1: Message valid bits

MsgVal32 to 1	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).

See the following table to confirm the message valid bit for CAN macro equipped with 32 message buffers or higher.

Table 42-10. Message Valid Bit for CAN Macro Equipped with 32 Message Buffers or Higher

		addr + 0	addr + 1	addr + 2	addr + 3
MSGVAL 4 & 3	MsgVal 64 to 33 (address $B4_H$)	MsgVal64 to 57	MsgVal56 to 49	MsgVal48 to 41	MsgVal40 to 33
MSGVAL 6 & 5	MsgVal 96 to 65 (address $B8_H$)	MsgVal96 to 89	MsgVal88 to 81	MsgVal80 to 73	MsgVal72 to 65
MSGVAL 8 & 7	MsgVal 128 to 97 (address BC_H)	MsgVal128 to 121	MsgVal120 to 113	MsgVal112 to 105	MsgVal104 to 97

42.4.6 CAN Prescaler Register (CANPRE)

The bit configuration of the CAN prescaler register is shown.

This register sets the CAN system clock (fsys) generation prescaler. For details, see "42.5.6 Bit Timing and CAN System Clock (fsys) Generation". To change the value of this register, set the initialization bit (Init) in the CAN control register (CTRLR) to "1" 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			CANPRE3	CANPRE2	CANPRE1	CANPRE0
Initial value	0	0	0	0	0	0	0	0
Attribute	R/W0	R0,W0	R0,W0	R0,W0	R/W	R/W	R/W	R/W

[bit7] Reserved

Always write "0" to this bit.

[bit6 to bit4] Reserved

The read value is always "0". When writing to these bits, set "0".

[bit3 to bit0] CANPRE[3:0] :CAN prescaler setting bits

CANPRE[3:0]	Function	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

CANPRE[3:0]	Function	Input CAN Prescaler Clock: 80 MHz	Input CAN Prescaler Clock: 64 MHz	Input CAN Prescaler Clock: 48 MHz
1011	Selects 1/12 period of the system clock as the CAN clock.	6.7 MHz	5.4 MHz	4 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:

- Change the CAN prescaler setting bits after setting the initialization bit of the CAN control register (CTRLR) to "1" and stopping all the bus operations.
- The clock to be supplied to the CAN interface using the register setting must be 16 MHz or less.

42.5 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

42.5.1 Message Object

The message object is shown.

This section explains the message object and interface of message RAM.

42.5.1.1 Message Object

Message object is shown.

Message object settings (excluding MsgVal, NewDat, IntPnd and TxRqst bits) of message RAM will not be initialized by a hardware reset. Therefore, initialize message object by the CPU or disable the MsgVal bit (MsgVal="0"). Set CAN bit timing register (BTR) and CAN prescaler extension register (BRPER) while the Init bit of the CAN control register (CTRLR) is set to "1" and the CCE bit is set to "1".

Message object can be set by setting the data to the message interface register (IFx mask register, the IFx arbitration register, the IFx message control register (IFxMCTR) and IFx data register (IFxDTx)) and then writing the message number to the IFx command request register (IFxCREQ), as a result of which the data of the interface register will be transferred to the specified message object.

CAN controller starts operating when Init bit of the CAN control register (CTRLR) is cleared to "0". Reception message that has passed through the acceptance filter will be stored to message RAM. Messages with pending transmission request are transferred from message RAM to the shift register of the CAN controller and then transmitted to the CAN bus.

CPU reads the reception message via the message interface register and updates the transmission message. An interrupt is sent to the CPU according to the settings of the CAN control register (CTRLR) and IFx message control register (IFxMCTR) (message object).

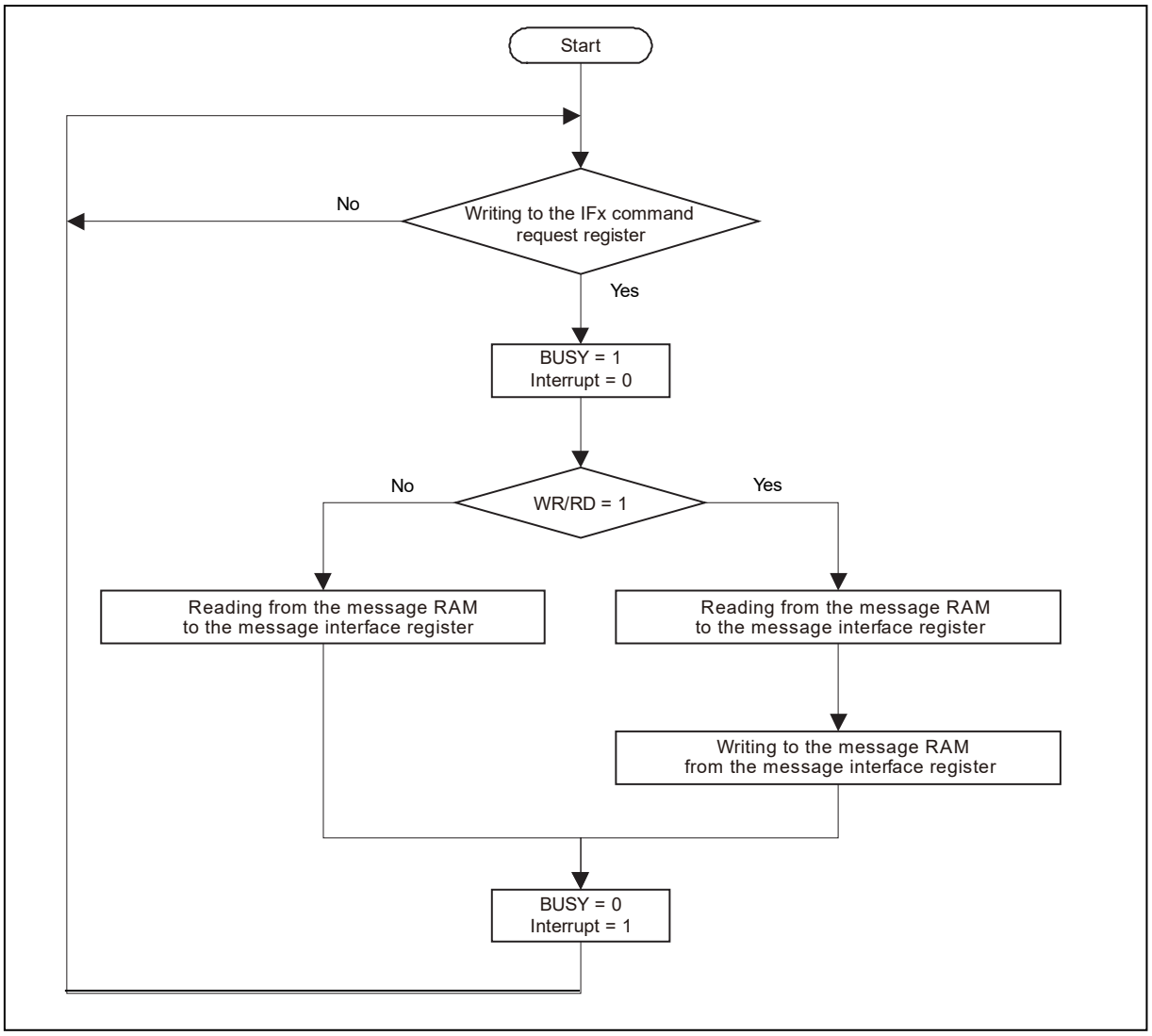
42.5.1.2 Data Transmission/Reception with Message RAM

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. The BUSY bit will be cleared to "0" after the transfer completion (see Figure 42-2).

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 42-2. Data Transfer between Message Interface Register and Message RAM



42.5.2 Message Transmission Operation

Message transmission operation is shown.

This section explains the setting method and transmission operation of the transmission message object.

42.5.2.1 Message Transmission

Message transmission is explained.

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 bit 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.

42.5.2.2 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 32 (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 buffer 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 buffer.
  - ☐ 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 buffer with the lowest priority as the transmission message buffer 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".

42.5.2.3 Transmission Message Object Setting

Transmission message object setting is explained.

The initialization method for the transmission object is shown below:

Table 42-11. 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 "42.5.3 Message Reception Operation" for details.

Note:

Mask is not allowed for the Dir bit of the IFx mask register.

42.5.2.4 Update of Transmission Message Object

Update of transmission message object is explained.

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".

42.5.3 Message Reception Operation

Message reception operation is shown.

This section explains the setting method and reception operation of the reception message object.

42.5.3.1 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).

42.5.3.2 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 32 (or the maximum equipped message object number) has the lowest priority. If 2 or more message objects match the acceptance filter, the one having the smaller message number will be the reception message object.

42.5.3.3 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 buffer 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.

42.5.3.4 Remote Frame

Remote frame is shown.

The following three 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"

   Remote frame will be disabled without receiving the message, even if the received remote frame matches the message object. (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.

42.5.3.5 Reception Message Object Setting

Reception message object setting is shown.

The initialization method for the reception message object is shown below:

Table 42-12. 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 the data frame reception in "42.5.3 Message Reception Operation" for details.

Note:

The Dir bit of the IFx mask register cannot be set as a mask.

42.5.3.6 Reception Message Processing

Reception message processing is explained.

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.

42.5.4 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.

42.5.4.1 Configuration of FIFO Buffer

The configuration of FIFO buffer is explained.

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 "42.5.3 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 block (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.

42.5.4.2 Message Reception by FIFO Buffer

Message reception by FIFO buffer is explained.

If the reception message matches the ID of the FIFO buffer, it will be stored to the reception message object in the FIFO buffer having the smallest message number.

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 the 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 = "1") is reached.

If the NewDat bit of the reception message object is not written to "0" (release of write protection) while valid data is stored up to the final FIFO buffer, the next reception message will be written to the final message object, overwriting the previous message.

42.5.4.3 Reading from FIFO Buffer

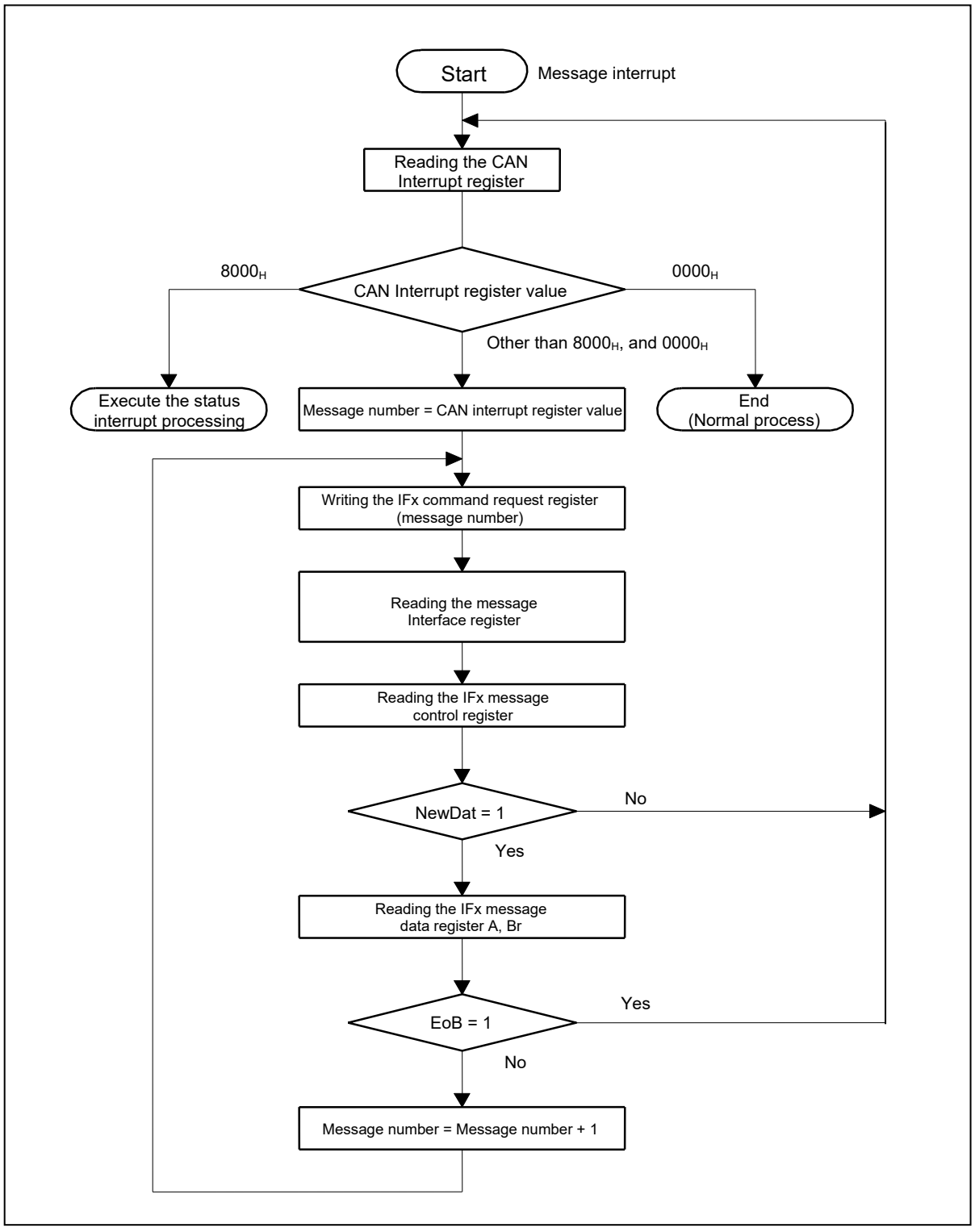
Reading from FIFO buffer is explained.

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 42-3. CPU Processing of FIFO Buffer



42.5.5 Interrupt Function

Interrupt function is explained.

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 IndPnd 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 "42.4.5 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.

42.5.6 Bit Timing and CAN System Clock (fsys) Generation

Bit timing and CAN system clock (fsys) generation is explained.

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 (fosc) 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 (df) of the oscillator.
The bit time is divided into the following four 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 and baud rate prescaler (BRP).

tq = BRP / fsys

CAN system clock (fsys) will be generated as shown in the figure below. 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 42-4. Schematic Diagram of CAN System Clock (fsys) Generation

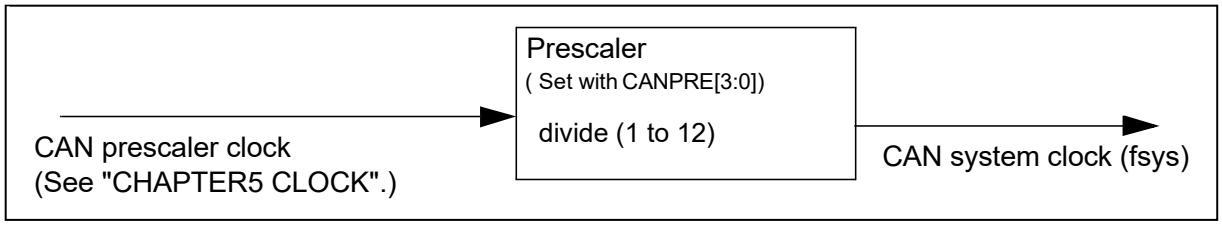


Figure 42-5. Bit Timing

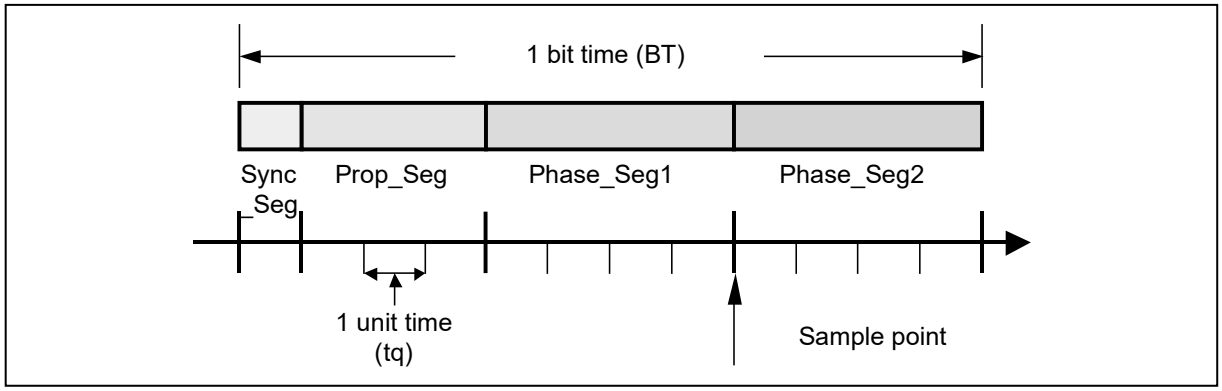


Table 42-13. CAN Bit Time Parameters

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 42-6. Bit Timing Effected by the CAN Controller

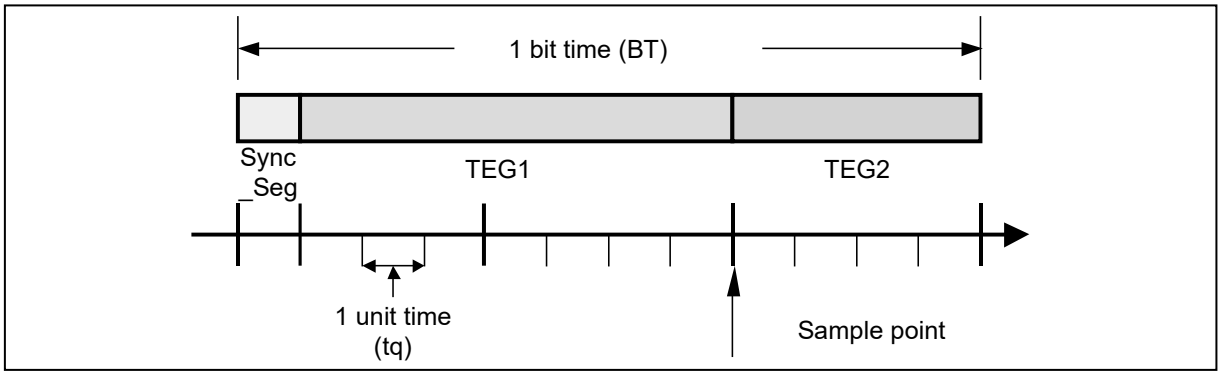


Table 42-14. CAN Controller Parameters

Parameter	Range	Function
BRPE,BRP	[0 to 1023]	Defines the time quantity tq. The prescaler can be extended up to 1024 using the bit timing and prescaler extension registers.
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)) × tq

= (3 + TSEG1 + TSEG2) × tq

42.5.7 Test Mode

Test mode is shown.

This section explains the test mode setting method and operation.

42.5.7.1 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.

All test register functions are invalidated when the Test bit of the CAN control register (CTRLR) is reset to "0".

42.5.7.2 Silent Mode

Silent mode is explained.

The CAN controller enters silent mode when the Silent bit of the CAN test register (TESTR) is set to "1".

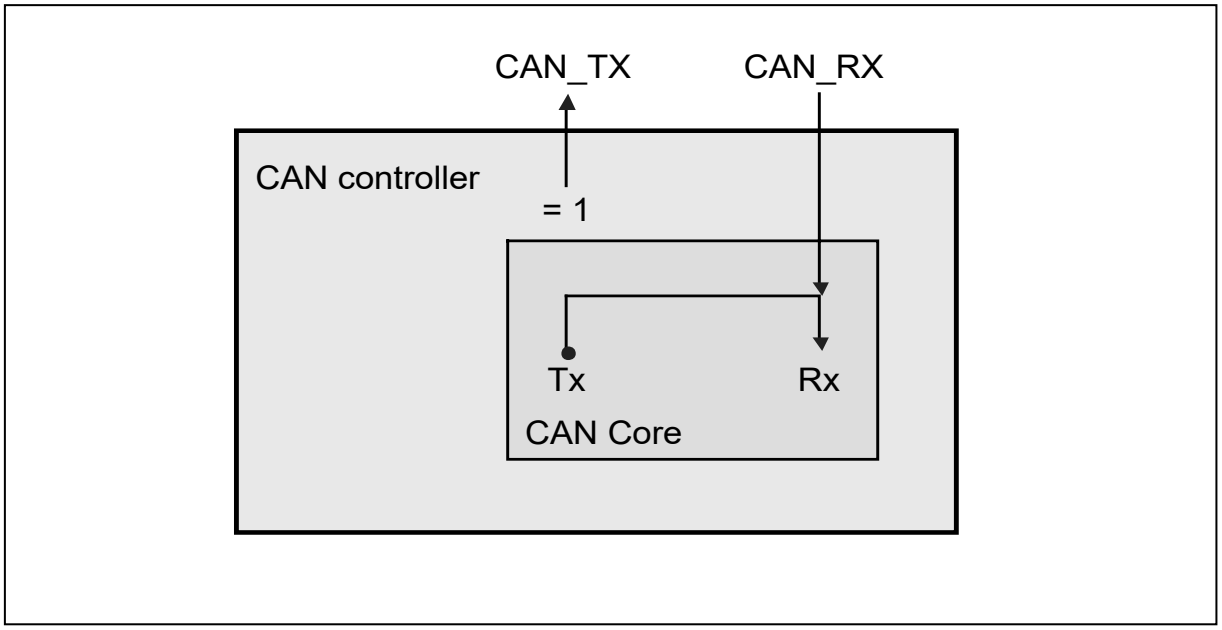
In silent mode, the CAN controller can receive data frames and remote frames, but only outputs a recessive level to the CAN bus and does not send messages or ACKs.

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 CAN_TX and CAN_RX are connected to the CAN controller in silent mode:

Table 42-15. CAN Controller in Silent Mode



42.5.7.3 Loopback Mode

Loopback mode is explained.

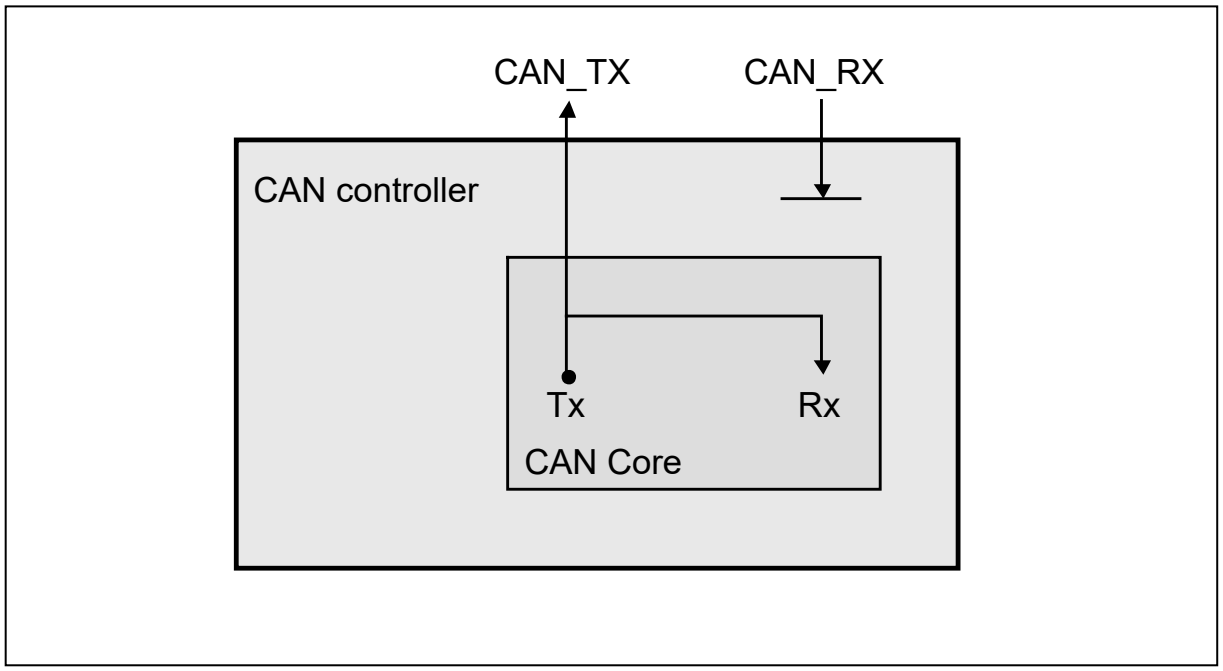
The CAN controller enters loopback mode when the LBack bit of the CAN test register (TESTR) is set to "1".

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 42-7. 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.

42.5.7.4 Combination of Silent and Loopback Modes

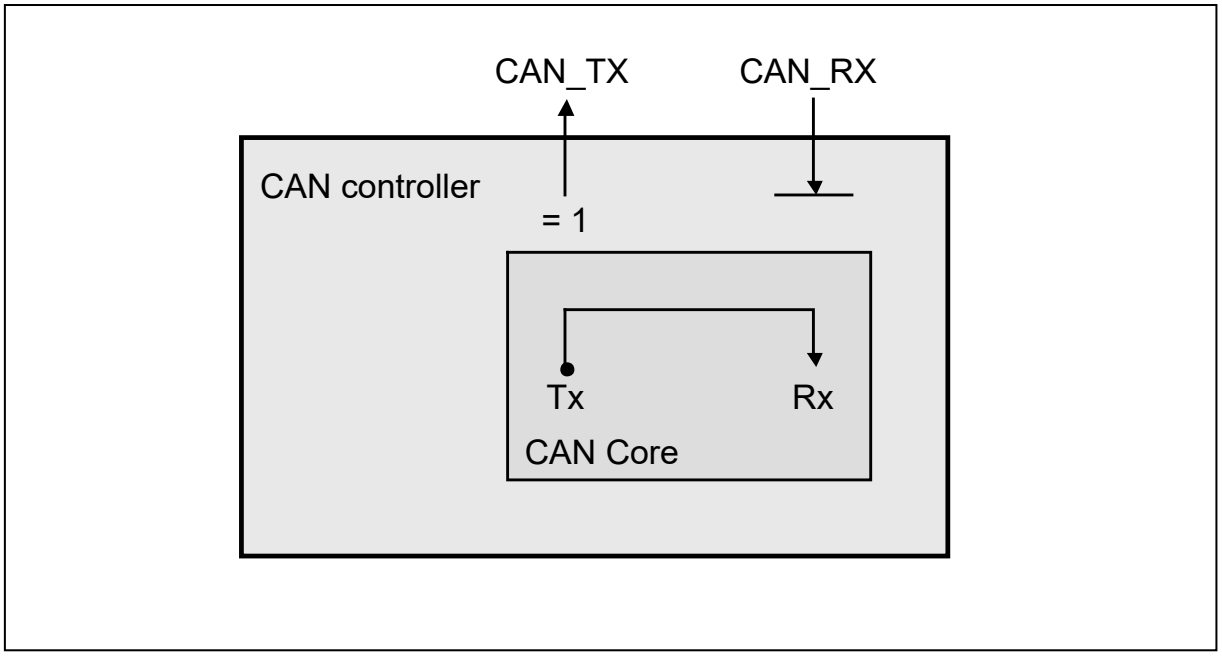
Combination of silent and loopback modes is explained.

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 pin and the input from the CAN_RX pin is invalid.

The figure below shows how signals CAN_TX and CAN_RX are connected to the CAN controller in the silent and loopback combined mode:

Figure 42-8. CAN Controller in the Silent and Loopback Combined Mode



42.5.7.5 Basic Mode

Basic mode is explained.

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.

All messages are received without using the acceptance filter. The received message can be read when the BUSY bit of the IF2 command request register is set to "1".

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".

42.5.7.6 Software Control of the CAN_TX Pin

Software control of the CAN_TX pin is explained.

The CAN_TX pin, which is the CAN transmission pin, has four 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 CAN_RX monitoring function of the CAN reception pin.

The CAN_TX pin 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 pin must be configured for serial data output.

42.5.8 Software Initialization

Software initialization is explained.

Software-controlled initialization is as follows:

The causes of software-controlled initialization are as follows:

- Hardware reset
- Setting of the Init bit of the CAN control register (CTRLR)
- Transition to bus-off state

A hardware reset initializes everything except the message RAM (excluding the MsgVal, NewDat, IntPnd, and TxRqst bits). After a hardware reset, initialize the message RAM by way of the CPU or reset the MsgVal bit of the message RAM to "0". If the bit timing register needs to be set, set it before clearing the Init bit of the CAN control register (CTRLR) to "0".

The Init bit of the CAN control register (CTRLR) is set to "1" on one of the following conditions:

- Write of "1" from the CPU
- Hardware reset
- Bus-off

When the Init bit is set to "1", all message transmission/reception over the CAN bus is suspended and the CAN_TX pin, which is for CAN bus output, is set to a recessive-level output state (except for CAN_TX test mode).

When the Init bit is set to "1", the error counter does not change and the registers do not change.

When the Init and CCE bits of the CAN control register (CTRLR) are set to "1", the baud rate control bit timing register and prescaler extension register can be configured.

Software initialization will terminate when the Init bit is reset to "0". The Init bit can only be reset to "0" through access from the CPU.

When the generation of 11 consecutive recessive bits (indicating a bus-idling state) are waited after the Init bit is reset to "0", the CAN controller can be synchronized with the data transfer over the CAN bus. This can be followed by message transfer.

If the message object Msk, ID, XTD, EoB, and/or RmtEn needs to be changed during ordinary operation, change it after invalidating the MsgVal bit.

42.6 Limitations

This section explains the limitations.

42.6.1 INIT Bit

42.6.1 INIT Bit

INIT bit is explained.

42.6.1.1 Limitations

42.6.1.2 Workaround

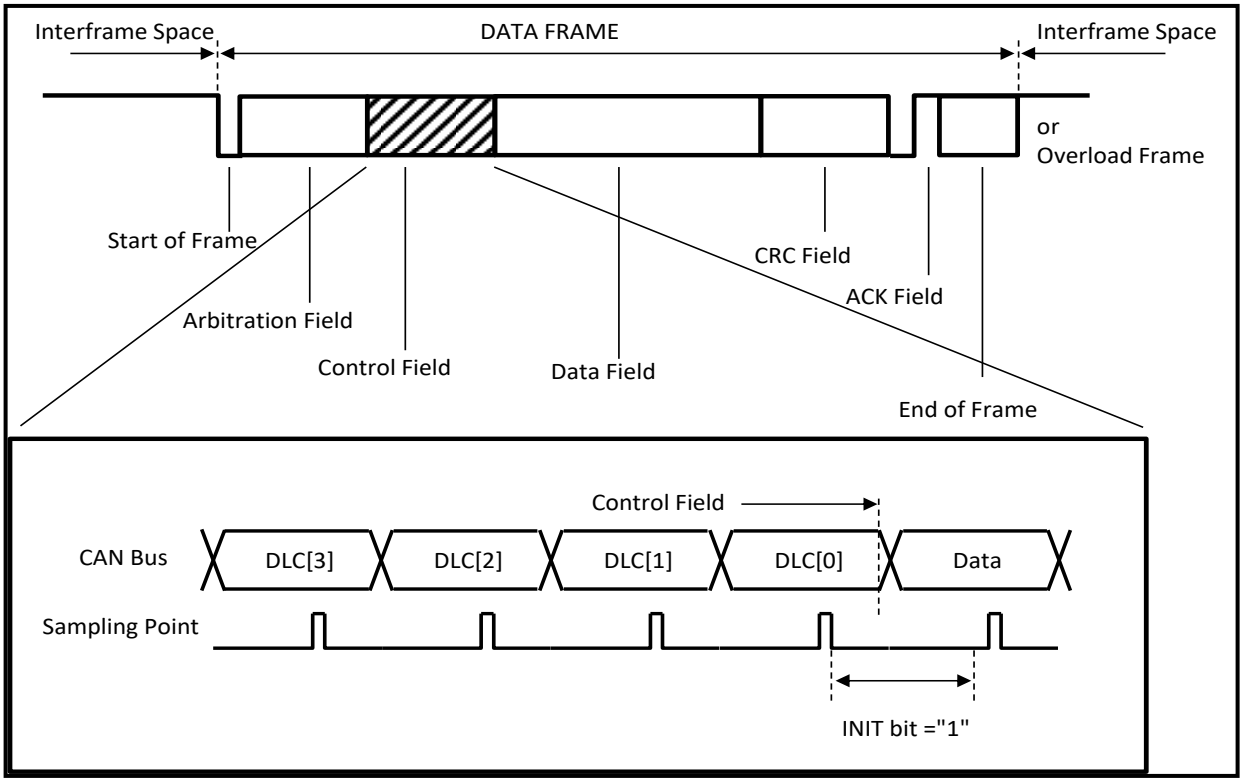
42.6.1.1 Limitations

The limitations are explained.

When the INIT bit of the CAN control register (CTRLR) is set to "1" while the final bit of the control field is transmitted (see below figure), a data field of the frame transmitted first results in left-shifted by 1bit.

After that, however, messages are transmitted correctly.

Furthermore, both the remote and data frames with 0 (zero) data length each have no effect even though the INIT bit is set in this timing.



42.6.1.2 Workaround

The workaround is explained.

Avoid these limitations by using either of the following workaround:

1. When setting the INIT bit of the CAN control register (CTRLR) to "1", set the INIT bit of the CAN control register (CTRLR) to "1" immediately after transmission is completed.
2. When setting the INIT bit of the CAN control register to "1" during transmission, and then setting the INIT bit to "0" to transmit:
    a. First set the INIT bit to "1".
    b. Second make the message buffer with the transmission request bit (TxRqst) = "1" cancel transmission (set the TxRqst bit to "0").
    c. Third set the INIT bit to "0".

After a 2-bit time passes, set the transmission request bit (TxRqst) of the message buffer to "1".

43. D/A Converter

This chapter explains the D/A converter.

43.1 Overview

43.2 Features

43.3 Configuration

43.4 Registers

43.5 Operation

43.6 Note

43.1 Overview

This section explains the overview of the D/A converter.

The D/A converter is a peripheral function to convert digital signals to analog signals. This product incorporates 2 channels of the 8-bit D/A converter.

43.2 Features

This section explains the features of the D/A converter.

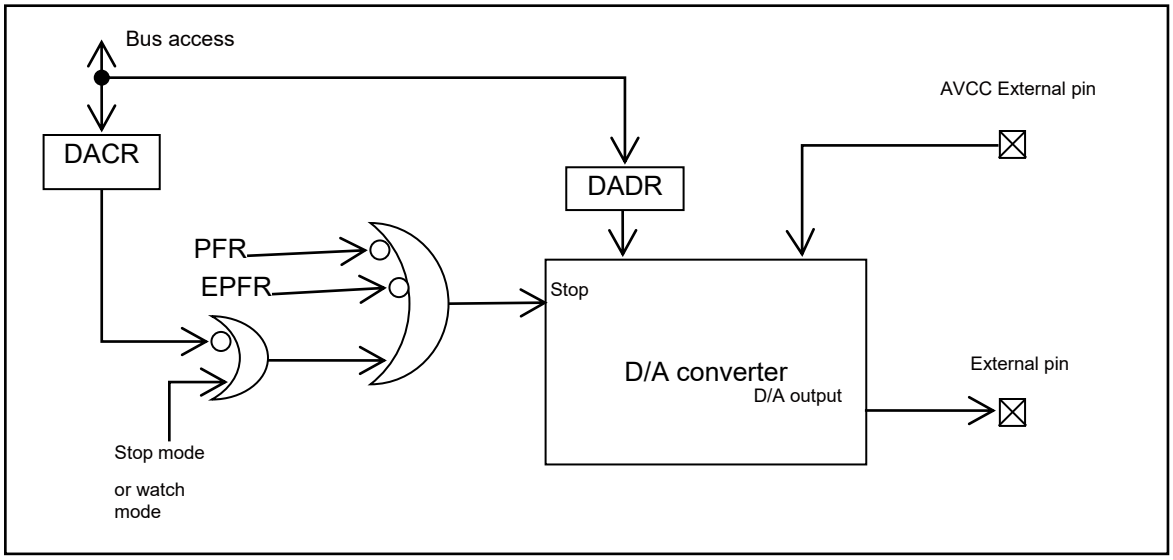
- Power Down Function

  The power down function which turns the power off when the outputs are disabled by the D/A converter is incorporated.

43.3 Configuration

This section explains the configuration of the D/A converter.

Figure 43-1 . Block Diagram



43.4 Registers

This section explains the registers of the D/A converter.

Base Address (Base_addr) and External Pin Table

Channel Number	Base_addr	External Pin
0	0x023C	DAO0
1	0x023E	DAO1

Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x023C	DACR0	DADR0	DACR1	DADR1	DA control register 0 DA data register 0 DA control register 1 DA data register 1

43.4.1 DA Control Register: DACR

The bit configuration of the DA control register is shown below.

This register enables the output from the DAO pin.

DACR: Address Base_addr (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved							DAE
Initial value	-	-	-	-	-	-	-	0
Attribute	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX	R/W

[bit0] DAE (DA Enable): DA output enable bit

0: Output disabled
1: Output enabled

Note:

When selecting the D/A converter as a port function, if setting this bit to "0", then the D/A converter as a port function will be disabled regardless of settings for the PFR/EPFR.In stop mode or watch mode, moreover, pin function of the D/A converter will be disabled forcibly.

As described in Section 11.5.2 of Chapter: I/O Ports, output function of the D/A converter is the highest priority. Notice that if such functions other than the D/A converter as the A/D, PPG, OCU, and SG are enabled when the D/A converter is disabled, those functions will be enabled. In PFR/EPFR/ADCH, when no other resource function than the D/A converter is selected, ports become general-purpose port function and their pin status can be selected with the PDR/DDR.

43.4.2 DA Data Register: DADR

The bit configuration of the DA data register is shown below.

This register is used to set the output voltage from the DAO pin. The output voltage from the D/A converter will be calculated based on the value stored in this register.

DADR: Address Base_addr+ 01_H (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	DA[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

[bit7 to bit0] DA[7:0] (DA): DA output value

DA[7:0]	Output Voltage
00000000	0/256 × AVCC
00000001	1/256 × AVCC
00000010	2/256 × AVCC
-	-
11111101	253/256 × AVCC
11111110	254/256 × AVCC
11111111	255/256 × AVCC

AVCC: Input voltage from AVCC external pin

Note:

This register will not be initialized by the reset.

43.5 Operation

The section explains the operation of the D/A converter.

The D/A converter outputs the analog voltage from the DAO pin by calculating the output voltage based on the values written in the D/A data register (DADR).

If values are written to the DA7 to DA0 bits of the D/A data register (DADR) and "1" is written to the DAE bit of the D/A control register (DACR0), analog signals will be output from the D/A converter.

When selecting the D/A converter as a port function, if setting this bit to "0", then the D/A converter as a port function will be disabled regardless of settings for the PFR/EPFR. In stop mode or watch mode, moreover, pin function of the D/A converter will be disabled forcibly.

As described in Section 11.5.2 of Chapter: I/O Ports, output function of the D/A converter is the highest priority. Notice that if such functions other than the D/A converter as the A/D, PPG, OCU, and SG are enabled when the D/A converter is disabled, those functions will be enabled.

In PFR/EPFR/ADCH, when no other resource function than the D/A converter is selected, ports become a general-purpose port function and their pin status can be selected with the PDR/DDR.

For output of the conversion result of the D/A converter to the external pin, a pin must be set to DA output by the DAE bit in the D/A control register (DACR), and PFR register and EPFR register.

43.6 Note

The section explains the notes about the D/A converter.

The DA output value cannot be read by the PDDR register.

44. A/D Converter

This chapter explains the A/D converter.

44.1 Overview

44.2 Features

44.3 Configuration

44.4 Registers

44.5 Operation

44.6 Setting

44.7 Q&A

44.8 Sample Program

44.9 Notes

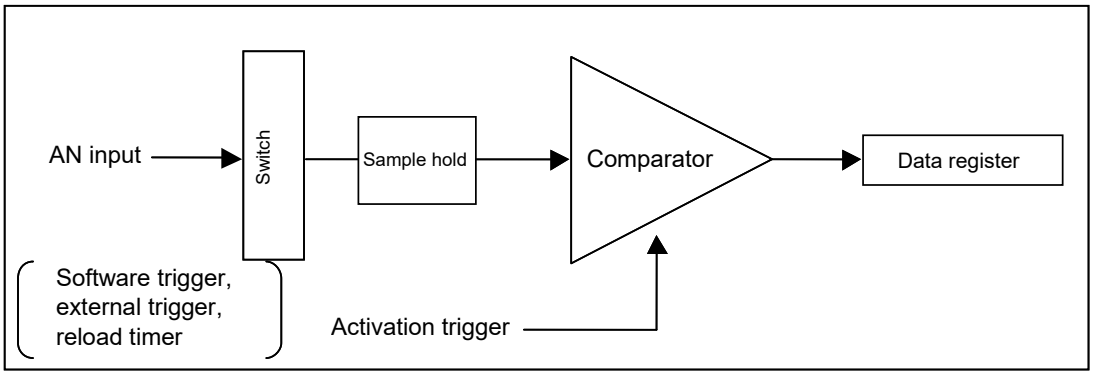
44.10 Term Definition for A/D Converter

44.1 Overview

This section explains the overview of the A/D converter.

A/D converter is a device which converts an analog input voltage to a digital value.

Conversion modes include the single conversion mode, continuous conversion mode, and stop conversion mode, and each mode has the single conversion operation and scan conversion operation as a conversion operation. An activation factor can be selected from various triggers (software trigger/external trigger/reload timer).



44.2 Features

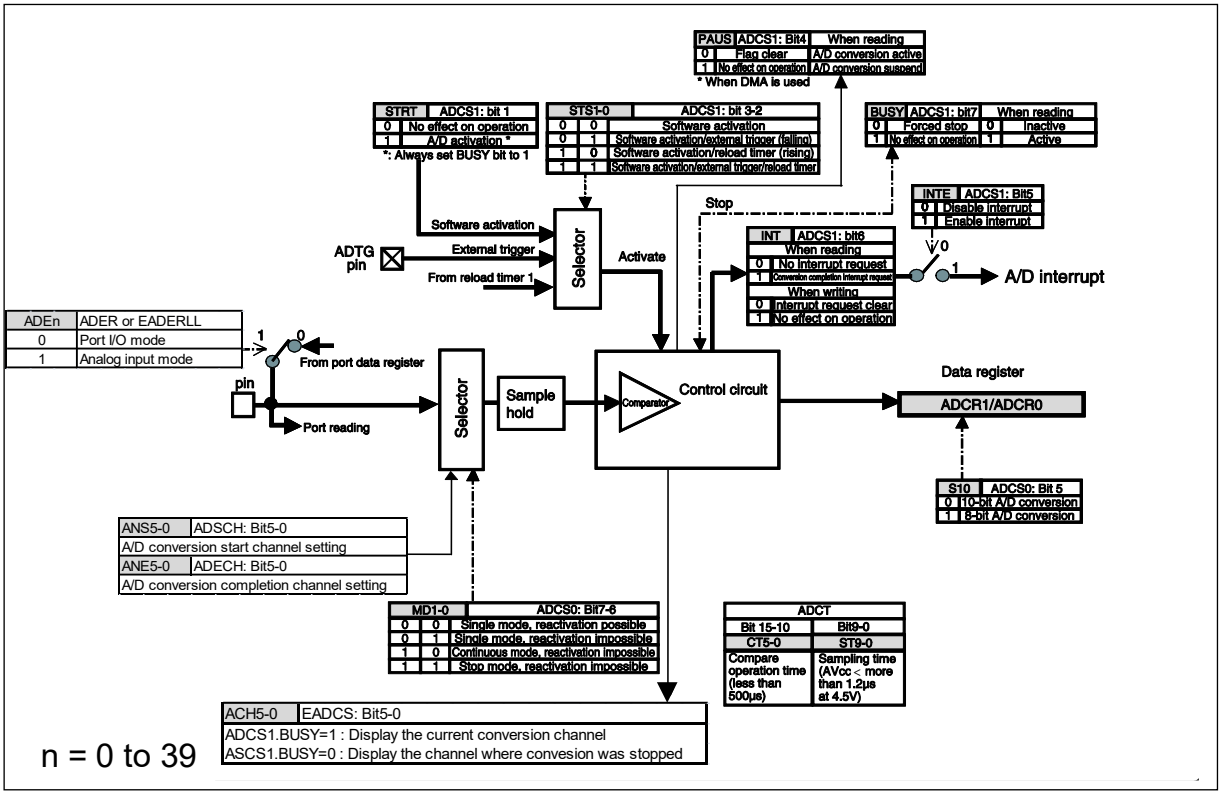
This section explains features of the A/D converter.

- Conversion method: RC type sequential comparison conversion method with sample hold circuit
- Number: 1 (A/D converter input - 40 channels: AN0 to AN39)
- Conversion time:
  Minimum 3µs (including sample hold time)
  Conversion time = Sampling + Conversion
- Resolution: 8/10-bit resolution
- Conversion mode:
  Single conversion mode : One cycle conversion of specified channel
  Continuous conversion mode: Repeated conversion of specified channel
  Stop conversion mode : Pause and wait until next activation after conversion of specified channel (conversion start can be synchronized)
- Conversion operation: The following are conversion operations for conversion modes above.
  Single conversion operation: One channel is selected and converted.
  Scan conversion operation: Continuous multiple channels are converted. Maximum 40 channels programs enabled.
- Activation factor:
  Soft trigger (ADCS1:STRT)
  External trigger, falling (ADTG pin)
  Reload timer, rising (Reload timer 1)
- Interrupt request:
  A/D conversion completion interrupt request is generated for CPU on A/D conversion completion.
- Interrupt: Conversion completion interrupt
- Function stop: Forced stop of A/D conversion operation is enabled.

44.3 Configuration

This section explains the configuration of the A/D converter.

Figure 44-1. Configuration Diagram



44.4 Registers

This section explains registers of the A/D converter.

Table 44-1. Registers Map

Address	Registers				Register Function
	+0	+1	+2	+3	
0x00A0	ADER				Analog input enable register
0x00A4	ADCS1	ADCS0	ADCR1	ADCR0	A/D control status register upper A/D control status register lower Data register upper Data register lower
0x00A8	ADCT		ADSCH	ADECH	Conversion time setting register A/D start channel setting register A/D completion channel setting register
0x00AC	Reserved	EADERLL	EADCS	Reserved	Extended analog input enable register(ch.39 to ch.32) Extended A/D control status register

44.4.1 Analog Input Enable Register: ADER

The analog input enable register is explained.

These registers set the appropriate pins to the analog inputs. (ch.31 to ch.0)

ADERH: Address 00A0H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADE31	ADE30	ADE29	ADE28	ADE27	ADE26	ADE25	ADE24
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
	ADE23	ADE22	ADE21	ADE20	ADE19	ADE18	ADE17	ADE16
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

ADERL: Address 00A2H (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	ADE15	ADE14	ADE13	ADE12	ADE11	ADE10	ADE9	ADE8
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
	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0
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] ADE[31:0]: Analog input enable

ADEn	Meaning
0	Port input/output mode
1	Analog input mode

n = 0 to 31

The analog input enable register (ADER) of the start channel and complete channel must be set with "1".

44.4.2 A/D Control Status Register (Upper): ADCS1

The A/D control status register (upper) is explained.

This register is for A/D converter control and status display.

ADCS1: Address 00A4$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	Reserved
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 (RM0),W	R0,W0

[bit7] BUSY: Forced stop directive bit/operation check bit

BUSY	Read	Write
0	The A/D converter is being stopped.	The A/D converter is stopped forcibly.
1	The A/D converter is being operated.	No effect on operations

- This bit is set by A/D conversion activation.
- It is cleared with conversion completion of the final channel in the single mode.
- It is not cleared until the A/D converter is stopped with "0" writing in the continuous/stop mode.
- Do not perform forced completion and software activation simultaneously (BUSY=0, STRT=1).
- When the soft trigger performs activation ("1" writing to the STRT bit), the forced stop directive bit must be written with "1". (If they are not set with "1" simultaneously, the activation is not started.)

[bit6] INT: A/D conversion completion flag/interrupt request

INT	Read	Write
0	Without interrupt request	Clear of flag
1	With interrupt request (A/D conversion completion, all scan conversion completion)	No effect on operations

Note:

Clear "0" writing during A/D stop.

[bit5] INTE: A/D interrupt request enable

INTE	Meaning
0	Interrupt request disable
1	Interrupt request enable

When the A/D interrupt request enable bit (INTE) and the A/D conversion interrupt request flag (INT) are set with "1", an interrupt is generated.

[bit4] PAUS: A/D pause flag

PAUS	Read	Write
0	A/D conversion is being operated	Clear of flag
1	A/D conversion operation paused	No effect on operations

Since the register to store the A/D conversion result is one, for continuous conversions, if the conversion result is not transferred, the previous data will break.
To protect this, if the content of the data register is not transferred, next conversion data will not be stored. During this, the A/D conversion operation is stopped. After transfer is completed, if the INT bit is cleared, the A/D conversion is restarted.

- Clearing is enabled only with "0" writing. (Clearing is not enabled with the transfer completion.)
- If next A/D conversion is executed in the state where the INT bit is set to "1", the PAUS bit will be set to "1". (To protect the data of the previous A/D conversion) after the A/D conversion of 1 channel, the INT bit must be cleared before next A/D conversion.
- For the protect function for conversion data, see "44.5.3 Conversion Mode".

[bit3, bit2] STS1, STS0: Selection of A/D conversion activation factor

STS1	STS0	Activation Factor
0	0	Software trigger
0	1	External trigger (falling) or software trigger
1	0	Reload timer output (rising) or software trigger
1	1	External trigger (falling) or reload timer output (rising), or software trigger

- If multiple activation factors are specified, the A/D conversion is activated by the factor that occurs first.
- The activation factor that occurs during A/D conversion is enabled when the conversion is restarted in the single conversion mode (ADCS0:MD[1:0]="00") and the stop conversion mode (ADCS0:MD[1:0]="11").
- The reactivation in the single conversion mode (ADCS0:MD[1:0]="01"), continuous conversion mode,or stop conversion mode must be performed after the A/D conversion operation is stopped once (BUSY="0").
- When you rewrite the activation factor setting during A/D conversion, pay attention to that the activation factor setting is changed right after rewriting.
- The external pin trigger detects the falling edge. If the external pin trigger activation is set by rewriting of the bit during the external trigger input level is "L", A/D might be activated.
- On the timer selection, 16-bit reload timer 1 is selected.

[bit1] STRT: A/D conversion software trigger

STRT	Function
0	No effect on operations
1	The A/D converter is activated (software trigger).

- When the software trigger activation is set, the forced stop directive bit (BUSY) also must be set to "1". If the forced stop directive bit (BUSY) is set to "0" simultaneously, the A/D is not activated.
- For reactivation, write "1" again after the forced stop by writing "0" to the BUSY bit.
- Reactivation is disabled in the continuous mode and stop mode operation functionally. Check the BUSY bit before "1" is written. (Activate after the BUSY bit is cleared.)
- Do not perform forced completion and software activation simultaneously (BUSY=0, STRT=1).

[bit0] Reserved

This bit must always be written to "0".

44.4.3 A/D Control Status Register (Lower): ADCS0

The A/D control status register (lower) is explained.

This register is for A/D converter control and status check.

Note:

Do not rewrite during the A/D conversion operation.

ADCS0: Address 00A5$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	MD1	MD0	S10	Reserved				
Initial value	0	0	0	-	-	-	-	-
Attribute	R/W	R/W	R/W	RX,WX	RX,WX	RX,WX	RX,WX	RX,WX

[bit7, bit6] MD1, MD0: Operation mode setting

MD1	MD0	Operating Mode
0	0	Single conversion mode. Any reactivation during operation is enabled.
0	1	Single conversion mode. Reactivation during operation is disabled.
1	0	Continuous conversion mode. Reactivation during operation is disabled.
1	1	Stop conversion mode. Reactivation during operation is disabled.

- Single conversion mode:

  A/D conversion is performed continuously for ADSCH:ANS[5:0] setting channel to ADECH:ANE[5:0] setting channel. When the conversion for all channels is completed, the A/D converter is stopped.

- Continuous conversion mode:

  A/D conversion is performed repeatedly for ADSCH:ANS[5:0] setting channel to ADECH:ANE[5:0] setting channel.

- Stop conversion mode:

  A/D conversion is performed and paused for each channel from ADSCH:ANS[5:0] setting channel to ADECH:ANE[5:0] setting channel. Conversion restart is performed with activation factor occurrence.

  1. When the A/D conversion is activated in the continuous conversion mode or stop conversion mode, the conversion operation is continued until the conversion is stopped forcibly with the BUSY bit.
  2. Forced stop is performed with "0" writing to the BUSY bit.
  3. On the activation after forced stop, the conversion is performed starting from the setting channel of ADSCH:ANS[5:0].
  4. The reactivation disable in the single, continuous, or stop conversion mode applies to any of timer, external trigger and software activation.

[bit5] S10: Resolution setting

S10	Configuration
0	10-bit A/D conversion
1	8-bit A/D conversion

The result of 8-bit A/D conversion is stored to ADCR0.

[bit4 to bit0] Reserved

Reading has no effects. Writing has no effect.

44.4.4 Data Register: ADCR0, ADCR1

The bit configuration for the data register is explained.

The data registers (ADCR0, ADCR1) are used for storage of digital values generated as the result of conversion. ADCR0 stores the lower 8-bit and ADCR1 stores the most significant 2-bit of the conversion result.

The data register value is updated for one conversion completion.

The data register normally stores the final conversion values.

ADCR1: Address 00A6$_H$ (Access: Byte, Half-word, Word)

ADCR0: Address 00A7$_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	Reserved						D9	D8
Initial value	-	-	-	-	-	-	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
	D7	D6	D5	D4	D3	D2	D1	D0
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

The conversion data protection function can be used. See "44.5.3 Conversion Mode".

Notes:

- Access ADCR0 after accessing ADCR1 when you access ADCR1(0x0000A6) and ADCR0(0x0000A7) using the byte. There is a possibility that the conversion result is overwritten before reading the value of the ADCR1 register when accessing it in order of ADCR0 and ADCR1 because the superscription of the conversion result waits for reading to ADCR0 and it is done.
- In the state that an interrupt clear is previously done, and the conversion result is not read, it waits until the conversion result is read without overwriting the conversion result when the following conversion ends. If the read is previously done, it suspends the overwrite of the result until the interrupt is cleared.

44.4.5 Conversion Time Setting Register: ADCT

The conversion time setting register is explained.

This register controls the sampling time and comparison time. It is for setting of A/D conversion time.

Note:

Do not rewrite during the A/D conversion operation.

Recommended setting value

To achieve the optimum conversion time, the following settings are recommended. (AVCC ≥ 4.5V)

Peripheral Clock (PCLK) (MHz)	Comparison Operation Time (CT5 to CT0)	Sampling Time (ST9 to ST0)	Conversion Time (µs)
16	000011 (03_H)	0000010110 (016_H)	2.125+1.375=3.500
24	000100 (04_H)	0000100001 (021_H)	1.833+1.375=3.208
32	000110 (06_H)	0000101100 ($02C_H$)	2.000+1.375=3.063

Conversion Time Setting Register: ADCT (ADCT0, ADCT1)

- ADCT1: Address $00A8_H$ (Access: Byte, Half-word, Word)
- ADCT0: Address $00A9_H$ (Access: Byte, Half-word, Word)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	CT5	CT4	CT3	CT2	CT1	CT0	ST9	ST8
Initial value	0	0	0	1	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
	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Initial value	0	0	1	0	1	1	0	0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[bit15 to bit10] CT5 to CT0: Comparison operation time clock division value setting

- ☐ If CT5 to CT0 are set to "000001" (01_H), no division = PCLK is set.
- ☐ Do not set CT5 to CT0 to "000000" (00_H).

Note:

The following restriction should be applied to the maximum spread frequency of the base clock, if the spread spectrum clock is used for the peripheral clock. And the restriction should be applied to the PLL clock frequency, if the non spread spectrum clock is used. See Figure 5-1 and Figure 5-6 in "Chapter: Clock" for the base clock and the PLL clock.

- If the frequency of the base clock or the PLL clock is faster than 32 MHz, the setting of division number of the peripheral clock must be larger than or equal to 2.
- If the frequency of the base clock or the PLL clock is faster than 80 MHz, CT5 to CT0 must be set larger than "000010"(02h) regardless of division setting of the peripheral clock (PCLK2).
- Set CT5 to CT0 so that the clock at the comparison operation time becomes 8 to 17 MHz.

[bit9 to bit0] ST9 to ST0: Analog input sampling time setting

They are initialized to "0000101100"($02C_H$) by reset.

Setting the following values to ST9 to ST0 is inhibited. Set the value larger than 3.

"00000010"(02_H), "00000001"(01_H), "00000000"(00_H)

44.4.6 A/D Start/Completion Channel Setting Register: ADSCH, ADECH

The bit configuration of the A/D start/completion channel setting register is explained.

This register is for setting of a start channel and a completion channel for A/D conversion.

Do not rewrite during the A/D conversion operation.

A/D Start Channel Setting Register: ADSCH

- ADSCH: Address $00AA_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
Initial value	-	-	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W

A/D Completion Channel Setting Register: ADECH

- ADECH: Address $00AB_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	ANE5	ANE4	ANE3	ANE2	ANE1	ANE0
Initial value	-	-	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] - : Undefined

The read value is always "0". Writing has no effect on operation.

[bit5 to bit0] ANS5 to ANS0/ANE5 to ANE0: Start/completion channel

ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	Start Channel
ANE5	ANE4	ANE3	ANE2	ANE1	ANE0	Completion Channel
0	0	0	0	0	0	AN0
0	0	0	0	0	1	AN1
0	0	0	0	1	0	AN2
0	0	0	0	1	1	AN3
0	0	0	1	0	0	AN4
0	0	0	1	0	1	AN5
0	0	0	1	1	0	AN6
0	0	0	1	1	1	AN7
0	0	1	0	0	0	AN8
0	0	1	0	0	1	AN9

ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	Start Channel
ANE5	ANE4	ANE3	ANE2	ANE1	ANE0	Completion Channel
0	0	1	0	1	0	AN10
0	0	1	0	1	1	AN11
0	0	1	1	0	0	AN12
0	0	1	1	0	1	AN13
0	0	1	1	1	0	AN14
0	0	1	1	1	1	AN15
0	1	0	0	0	0	AN16
0	1	0	0	0	1	AN17
0	1	0	0	1	0	AN18
0	1	0	0	1	1	AN19
0	1	0	1	0	0	AN20
0	1	0	1	0	1	AN21
0	1	0	1	1	0	AN22
0	1	0	1	1	1	AN23
0	1	1	0	0	0	AN24
0	1	1	0	0	1	AN25
0	1	1	0	1	0	AN26
0	1	1	0	1	1	AN27
0	1	1	1	0	0	AN28
0	1	1	1	0	1	AN29
0	1	1	1	1	0	AN30
0	1	1	1	1	1	AN31
1	0	0	0	0	0	AN32
1	0	0	0	0	1	AN33
1	0	0	0	1	0	AN34
1	0	0	0	1	1	AN35
1	0	0	1	0	0	AN36
1	0	0	1	0	1	AN37
1	0	0	1	1	0	AN38
1	0	0	1	1	1	AN39
The others						Forbidden

These bits are for setting of a start channel and a completion channel for A/D conversion.

- ☐ When the same one channel is written to ANS5 to ANS0 and ANE5 to ANE0, conversion is performed only for one channel (Single conversion).
- ☐ When the continuous mode or stop mode is set, after the conversion for channels set by these bits group is completed, it returns to the start channel set by ANS5 to ANS0.
- ☐ When the set channels are ANS > ANE, the conversion is performed from ANS until ch.39 and returning to ch.0 and then it is performed until ANE.
- ☐ These bits group is initialized to ANS="000000", ANE="000000" by reset.
  Example: When the channel setting is ANS=ch.6 ANE=ch.3 and in the single mode, the conversion is performed in the order below:

  ch.6 → ch.7 → ch.8 → ... → ch.39 → ch.0 → ch.1 → ch.2 → ch.3

44.4.7 Extended Analog Input Enable Register: EADERLL

The extended analog input enable register is explained.

These registers set the appropriate pins to the analog inputs. (ch.39 to ch.32)

EADERLL: Address 00AD$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ADE39	ADE38	ADE37	ADE36	ADE35	ADE34	ADE33	ADE32
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] ADE[39:32]: Analog input enable

ADEn	Meaning
0	Port input/output mode
1	Analog input mode

n = 32 to 39

The analog input enable register of the start channel and complete channel must be set with "1".

44.4.8 Extended A/D Control Status Register: EADCS

The extended A/D control status register is explained.

Displays current conversion channel number or the channel number whose convesion has been stopped.

EADCS: Address 00AE$_H$ (Access: Byte, Half-word, Word)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Reserved		ACH5	ACH4	ACH3	ACH2	ACH1	ACH0
Initial value	-	-	0	0	0	0	0	0
Attribute	R0,WX	R0,WX	R,WX	R,WX	R,WX	R,WX	R,WX	R,WX

[bit5 to bit0] ACH5 to ACH0: Analog conversion channels

ACH5	ACH4	ACH3	ACH2	ACH1	ACH0	Conversion Channel
0	0	0	0	0	0	AN0
0	0	0	0	0	1	AN1
0	0	0	0	1	0	AN2
0	0	0	0	1	1	AN3
0	0	0	1	0	0	AN4
0	0	0	1	0	1	AN5
0	0	0	1	1	0	AN6
0	0	0	1	1	1	AN7
0	0	1	0	0	0	AN8
0	0	1	0	0	1	AN9
0	0	1	0	1	0	AN10
0	0	1	0	1	1	AN11
0	0	1	1	0	0	AN12
0	0	1	1	0	1	AN13
0	0	1	1	1	0	AN14
0	0	1	1	1	1	AN15
0	1	0	0	0	0	AN16
0	1	0	0	0	1	AN17
0	1	0	0	1	0	AN18
0	1	0	0	1	1	AN19
0	1	0	1	0	0	AN20
0	1	0	1	0	1	AN21
0	1	0	1	1	0	AN22

ACH5	ACH4	ACH3	ACH2	ACH1	ACH0	Conversion Channel
0	1	0	1	1	1	AN23
0	1	1	0	0	0	AN24
0	1	1	0	0	1	AN25
0	1	1	0	1	0	AN26
0	1	1	0	1	1	AN27
0	1	1	1	0	0	AN28
0	1	1	1	0	1	AN29
0	1	1	1	1	0	AN30
0	1	1	1	1	1	AN31
1	0	0	0	0	0	AN32
1	0	0	0	0	1	AN33
1	0	0	0	1	0	AN34
1	0	0	0	1	1	AN35
1	0	0	1	0	0	AN36
1	0	0	1	0	1	AN37
1	0	0	1	1	0	AN38
1	0	0	1	1	1	AN39

ADCS1:BUSY	Channel Status on Read
1 (A/D conversion being performed)	Current conversion channel
0 (on forced completion)	Conversion stopped channel

44.5 Operation

This section explains the operation of the A/D converter.

The A/D operation modes are explained below.

44.5.1 Single Conversion Operation

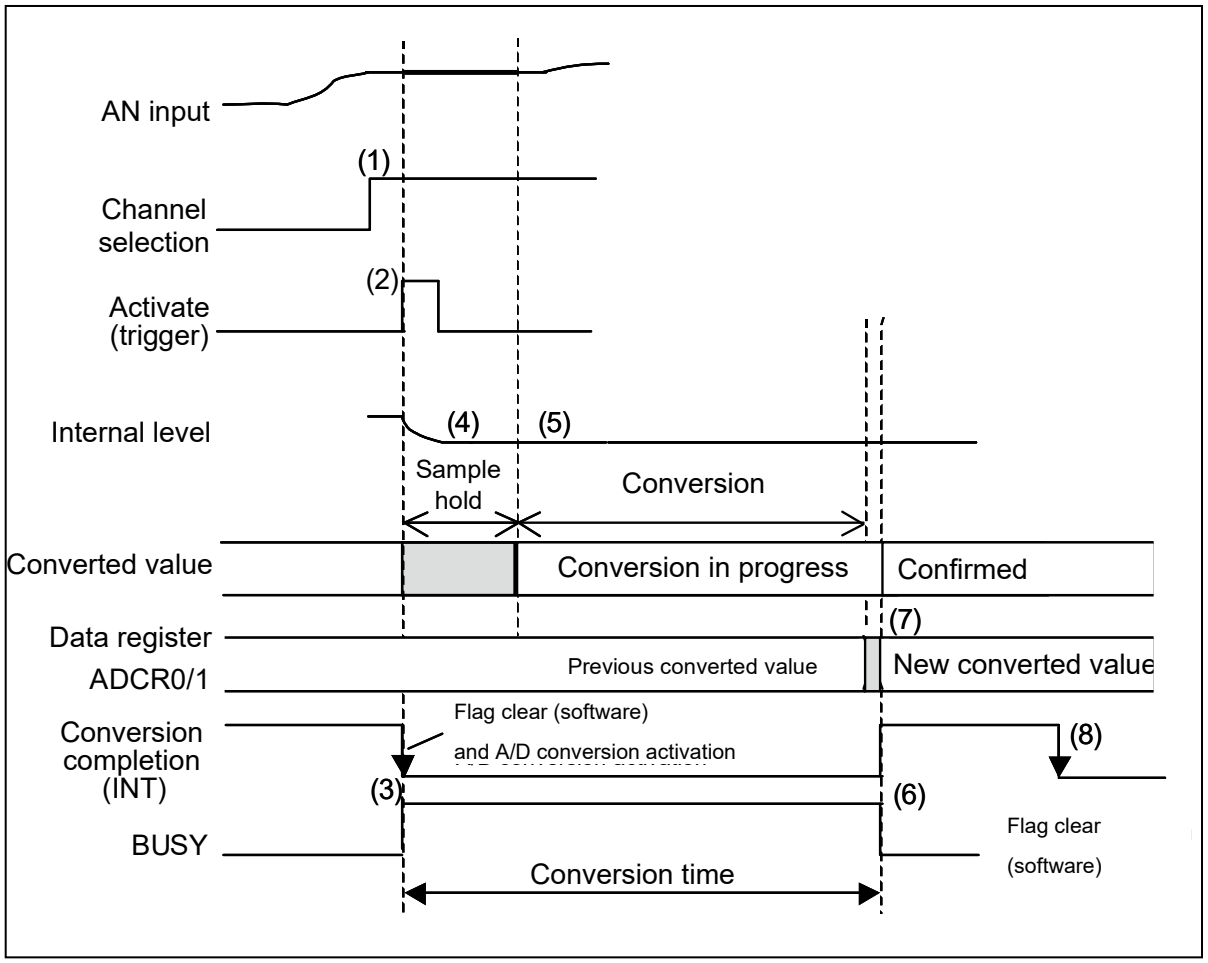
44.5.2 Scan Conversion Operation

44.5.3 Conversion Mode

44.5.1 Single Conversion Operation

The single conversion operation is explained.

Single Conversion Operation



(1) Channel selection

(2) A/D conversion activation (Trigger input: Software trigger/reload timer/external trigger)

(3) INT flag clear, BUSY flag set

(4) Sample hold

(5) Conversion

(6) Conversion completion, INT flag set, BUSY flag clear

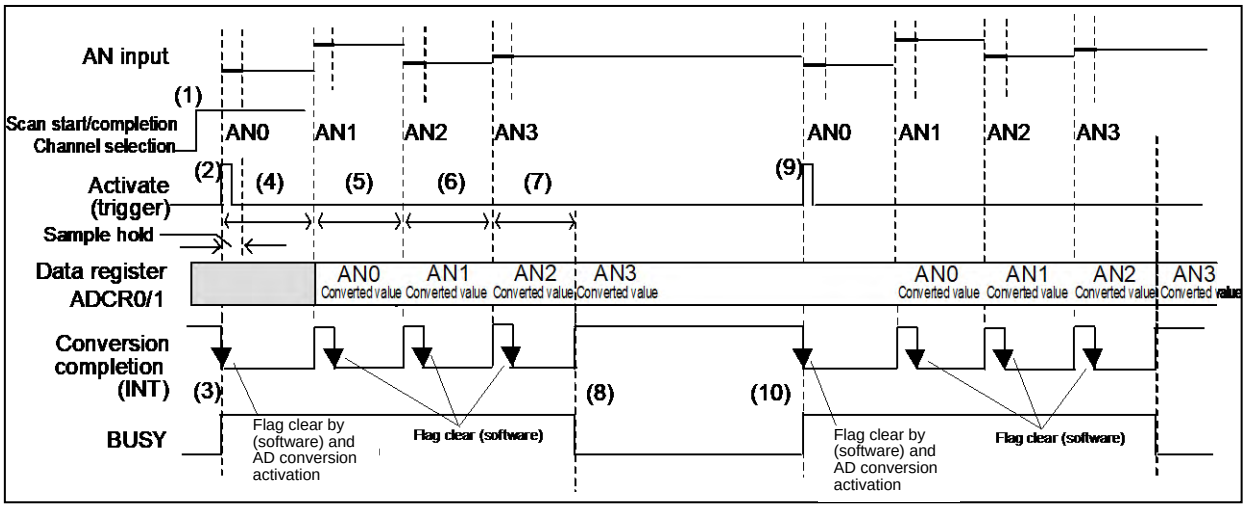
(7) Storage of conversion values to the data register

(8) INT flag clear by software

44.5.2 Scan Conversion Operation

The scan conversion operation is explained.

Scan Conversion Operation



(1) Activation channel selection

(2) A/D activation (Trigger: Software trigger/reload timer/external trigger)

(3) INT flag clear, BUSY flag set

(4) 1. AN0 conversion

2. INT flag set (Storage of AN0 conversion data)

(5) 1. AN1 conversion

2. AN0 conversion result transfer

3. INT flag clear

4. INT flag set (Storage of AN1 conversion data)

(6) 1. AN2 conversion

2. AN1 conversion result transfer

3. INT flag clear

4. INT flag set (Storage of AN2 conversion data)

(7) 1. AN3 conversion

2. AN2 conversion result transfer

3. INT flag clear

4. INT flag set (Storage of AN3 conversion data)

(8) INT flag set, BUSY flag clear

(9) Next A/D activation

(10) INT flag clear, BUSY flag set

44.5.3 Conversion Mode

The conversion mode is explained.

The A/D converter operates with a successive comparison method, and 10-bit or 8-bit resolution can be selected.

Since this A/D converter has one register (16-bit) for conversion result storage, the conversion data register (ADCR0 and ADCR1) is rewritten every time conversion is completed. Therefore, the A/D converter alone cannot perform continuous conversion process. It is recommended that conversion is performed while conversion data is transferred to a memory with the use of DMA. Operation modes are explained below.

Single Conversion Mode

In the single conversion mode, analog input set by ANS bits and ANE bits is converted in sequence. When the conversion is completed at the completion channel set by ANE bit, the operation of A/D converter is stopped. When the start channel and the completion channel is the same one (ANS = ANE), only one of them is converted.

[Example]

- ANS = 000000, ANE = 000011
  Start → AN0 → AN1 → AN2 → AN3 → Completion
- ANS = 000010, ANE = 000010
  Start → AN2 → Completion

Continuous Conversion Mode

In the continuous conversion mode, analog input set by ANS bits and ANE bits is converted in sequence. When the conversion is completed at the completion channel set by ANE bit, the conversion operation is continued returning to analog input of ANS bits. When the start channel and the completion channel is the same one (ANS = ANE), only conversion for one of them is continued.

[Example]

- ANS = 000000, ANE = 000011
  Start → AN0 → AN1 → AN2 → AN3 → AN0 - -→ Repeat
- ANS = 000010, ANE = 000010
  Start → AN2 → AN2 → AN2 - - → Repeat

In the continuous conversion mode the conversion is repeated until "0" is written to the BUSY bit ("0" writing to the BUSY bit → Forced completion). Note that when the forced completion is performed, the conversion that is being processed is stopped halfway (When the forced completion is performed, the previous converted data is stored in the conversion register.).

Stop Conversion Mode

In the stop conversion mode, analog input set by ANS bits and ANE bits is converted in sequence, and the conversion operation is paused after each conversion of 1ch. The pause can be released by performing activation again.

When the conversion is completed until the completion channel set by the ANE bit, the conversion operation is continued returning to the analog input by ANS. When the start channel and the completion channel is the same one (ANS = ANE), only one of them is converted.

[Example]

- ANS = 000000, ANE = 000011
  Start → AN0 → Stop → Activation → AN1 → Stop → Activation → AN2 → Stop → Activation → AN3 → Stop → Activation → AN0 - - → Repeat
- ANS = 000010, ANE = 000010
  Start → AN2 → Stop → Activation → AN2 → Stop → Activation → AN2 - - → Repeat

In this case the activation factors are only ones set by STS1, STS0.

With the use of this mode, the conversion start can be synchronized.

44.6 Setting

This section explains setting of the A/D converter.

Table 44-2. Necessary Configuration to Use A/D - Single Conversion Mode

Configuration	Register to be Configured	Method
Mode selection (Single conversion)	A/D control (ADCS0)	See 44.7.1
Bit length selection		See 44.7.2
Channel selection		See 44.7.3
Setting of conversion time	Conversion time setting (ADCT)	See 44.7.4
Setting of AN pin to input	Analog input enable (ADER) Extended analog input enable (EADERLL)	See 44.7.5
A/D activation trigger selection	A/D control (ADCS1)	See 44.7.6
A/D activation trigger occurrence Software trigger → Setting of software trigger bit		See 44.7.7
Reload timer → Reload timer rising output	See "Chapter: Reload Timer".	
External trigger → Input trigger to ADTG pin	Input from external	
Check of conversion completion flag	A/D control (ADCS1)	See 44.7.8
Conversion value read	Data register (ADCR0, ADCR1)	See 44.7.9

Table 44-3. Necessary Configuration to Use A/D - Continuous Conversion Mode

Configuration	Register to be Configured	Method
Mode selection (Continuous conversion)	A/D control (ADCS0, ADCS1)	See 44.7.1
Bit length selection		See 44.7.2
Start channel selection		See 44.7.3
Setting of conversion time	Conversion time setting (ADCT)	See 44.7.4
Setting of AN pin to input	Analog input enable (ADER) Extended analog input enable (EADERLL)	See 44.7.5
A/D activation trigger selection	A/D control (ADCS1)	See 44.7.6
A/D activation trigger occurrence Software trigger → Setting of software trigger bit		See 44.7.7
Reload timer → Reload timer rising output	See "Chapter: Reload Timer".	
External trigger → Input trigger to ADTG pin	Input from external	
Check of conversion completion flag	A/D control (ADCS1)	See 44.7.8
Conversion value read	Data register (ADCR0, ADCR1)	See 44.7.9

Table 44-4. Forced Stop of A/D Operation

Configuration	Register to be Configured	Method
Forced stop	A/D control (ADCS1)	See 44.7.10

Table 44-5. Items Necessary for A/D Interrupt

Configuration	Register to be Configured	Method
Setting of A/D interrupt vector and A/D interrupt level	See "Chapter: INterrupt Control (Interrupt Controller)".	See 44.7.11
A/D interrupt factor selection (A/D conversion completion)	A/D control register (ADCS1)	See 44.7.12
A/D interrupt setting Interrupt request clear Interrupt request enable		See 44.7.13

44.7 Q&A

This section explains Q&A of the A/D converter.

44.7.1 Conversion Mode Type and Setting Method
44.7.2 Specify the Bit Length
44.7.3 Select Channels
44.7.4 Set the Conversion Time
44.7.5 Enable the Analog Pin Input
44.7.6 Select the A/D Converter Activation Method
44.7.7 Activate the A/D Converter
44.7.8 Check the Conversion Completion
44.7.9 Read the Conversion Value
44.7.10 Stop the A/D Conversion Operation Forcibly
44.7.11 Interrupt-related Register
44.7.12 Interrupt Type
44.7.13 Enable/Disable/Clear the Interrupt

44.7.1 Conversion Mode Type and Setting Method

The conversion mode type and setting method are shown below.

The conversion includes the following four types:

- Single conversion mode where specified channels are converted for one cycle and terminated (Reactivation enabled during operation)
- Single conversion mode where specified channels are converted for one cycle and terminated (Reactivation disabled during operation)
- Continuous conversion mode where specified channels are converted repeatedly
- Stop conversion mode where conversion is performed for one channel and paused for specified channels

Set by the operation mode setting bits (ADCS0:MD[1:0]).

Operating Mode	Operation Mode Setting Bits (MD[1:0])
Single conversion mode (Reactivation enabled during operation)	Set "00".
Single conversion mode (Reactivation disabled during operation)	Set "01".
Continuous conversion mode	Set "10".
Stop conversion mode	Set "11".

44.7.2 Specify the Bit Length

This section explains how to specify the bit length.

Set the number of storage bits of conversion results (ADCS0:S10).

Operation	Number of Storage Bits of Conversion Results (S10)
To store with 10-bit to the data register	Set "0".
To store with 8-bit to the data register	Set "1".

44.7.3 Select Channels

This section explains how to select channels.

Specify channels to be converted by the A/D start channel setting bits (ADSCH:ANS[5:0]) and the A/D completion channel setting bits (ADECH:ANE[5:0]).

Specify the A/D conversion start channel.

A/D Conversion Start Channel	Channel Selection Bits (ANS[5:0])
To specify AN0	Set "000000".
To specify AN1	Set "000001".
To specify AN2	Set "000010".
To specify AN3	Set "000011".
To specify AN4	Set "000100".
To specify AN5	Set "000101".
To specify AN6	Set "000110".
To specify AN7	Set "000111".
To specify AN8	Set "001000".
To specify AN9	Set "001001".
To specify AN10	Set "001010".
To specify AN11	Set "001011".
To specify AN12	Set "001100".
To specify AN13	Set "001101".
To specify AN14	Set "001110".
To specify AN15	Set "001111".
To specify AN16	Set "010000".
To specify AN17	Set "010001".
To specify AN18	Set "010010".
To specify AN19	Set "010011".
To specify AN20	Set "010100".

A/D Conversion Start Channel	Channel Selection Bits (ANS[5:0])
To specify AN21	Set "010101".
To specify AN22	Set "010110".
To specify AN23	Set "010111".
To specify AN24	Set "011000".
To specify AN25	Set "011001".
To specify AN26	Set "011010".
To specify AN27	Set "011011".
To specify AN28	Set "011100".
To specify AN29	Set "011101".
To specify AN30	Set "011110".
To specify AN31	Set "011111".
To specify AN32	Set "100000"
To specify AN33	Set "100001"
To specify AN34	Set "100010"
To specify AN35	Set "100011"
To specify AN36	Set "100100"
To specify AN37	Set "100101"
To specify AN38	Set "100110"
To specify AN39	Set "100111"

Specify the A/D conversion completion channel.

A/D Conversion Completion Channel	Channel Selection Bits (ANE[5:0])
To specify AN0	Set "000000".
To specify AN1	Set "000001".
To specify AN2	Set "000010".
To specify AN3	Set "000011".
To specify AN4	Set "000100".
To specify AN5	Set "000101".
To specify AN6	Set "000110".
To specify AN7	Set "000111".
To specify AN8	Set "001000".
To specify AN9	Set "001001".
To specify AN10	Set "001010".
To specify AN11	Set "001011".
To specify AN12	Set "001100".
To specify AN13	Set "001101".
To specify AN14	Set "001110".
To specify AN15	Set "001111".
To specify AN16	Set "010000".
To specify AN17	Set "010001".
To specify AN18	Set "010010".
To specify AN19	Set "010011".
To specify AN20	Set "010100".
To specify AN21	Set "010101".
To specify AN22	Set "010110".
To specify AN23	Set "010111".

A/D Conversion Completion Channel	Channel Selection Bits (ANE[5:0])
To specify AN24	Set "011000".
To specify AN25	Set "011001".
To specify AN26	Set "011010".
To specify AN27	Set "011011".
To specify AN28	Set "011100".
To specify AN29	Set "011101".
To specify AN30	Set "011110".
To specify AN31	Set "011111".
To specify AN32	Set "100000"
To specify AN33	Set "100001"
To specify AN34	Set "100010"
To specify AN35	Set "100011"
To specify AN36	Set "100100"
To specify AN37	Set "100101"
To specify AN38	Set "100110"
To specify AN39	Set "100111"

44.7.4 Set the Conversion Time

This section explains how to set the conversion time.

Set the conversion time by the conversion time setting register (ADCT).

(Formula 1) Sampling time = ST Setting value × 1/F_{PCLK}

(Formula 2) Comparison operation time = CT Setting value ×1/F_{PCLK} ×10+4/F_{PCLK}

(Formula 3) Conversion time (total) = Sampling time + Comparison operation time

Setting Item	Control Bit	Recommended Value (F_{PCLK})			Remark
		At 16 MHz	At 24 MHz	At 32 MHz	
To set the sampling time	(ST[9:0])	"0000010110" (1.375 µs)	"0000100001" (1.375 µs)	"0000101100" (1.375 µs)	Set it to 1.2 µs or more. (For AV_{CC} < 4.5 V)
To set the comparison operation time	(CT[5:0])	000011 (2.125 µs)	000100 (1.833 µs)	000110 (2,000 µs)	Set it to 500 µs or less.

Note:

Set the ST[9:0] setting value for A/D sampling time to be the necessary sampling time or more.

See "44.4.5 Conversion Time Setting Register: ADCT" for the necessary sampling time.

44.7.5 Enable the Analog Pin Input

This section explains how to enable the analog pin input.

Enable it with the analog input enable register (ADER) or with the extended analog input enable register (EADERLL).

Operation	Control Bit	Configuration
To set AN0 pin to input	ADER:ADE0	Set "1".
To set AN1 pin to input	ADER:ADE1	Set "1".
To set AN2 pin to input	ADER:ADE2	Set "1".
To set AN3 pin to input	ADER:ADE3	Set "1".
To set AN4 pin to input	ADER:ADE4	Set "1".
To set AN5 pin to input	ADER:ADE5	Set "1".
To set AN6 pin to input	ADER:ADE6	Set "1".
To set AN7 pin to input	ADER:ADE7	Set "1".
To set AN8 pin to input	ADER:ADE8	Set "1".
To set AN9 pin to input	ADER:ADE9	Set "1".
To set AN10 pin to input	ADER:ADE10	Set "1".
To set AN11 pin to input	ADER:ADE11	Set "1".
To set AN12 pin to input	ADER:ADE12	Set "1".
To set AN13 pin to input	ADER:ADE13	Set "1".
To set AN14 pin to input	ADER:ADE14	Set "1".
To set AN15 pin to input	ADER:ADE15	Set "1".
To set AN16 pin to input	ADER:ADE16	Set "1".
To set AN17 pin to input	ADER:ADE17	Set "1".
To set AN18 pin to input	ADER:ADE18	Set "1".
To set AN19 pin to input	ADER:ADE19	Set "1".
To set AN20 pin to input	ADER:ADE20	Set "1".
To set AN21 pin to input	ADER:ADE21	Set "1".
To set AN22 pin to input	ADER:ADE22	Set "1".

Operation	Control Bit	Configuration
To set AN23 pin to input	ADER:ADE23	Set "1".
To set AN24 pin to input	ADER:ADE24	Set "1".
To set AN25 pin to input	ADER:ADE25	Set "1".
To set AN26 pin to input	ADER:ADE26	Set "1".
To set AN27 pin to input	ADER:ADE27	Set "1".
To set AN28 pin to input	ADER:ADE28	Set "1".
To set AN29 pin to input	ADER:ADE29	Set "1".
To set AN30 pin to input	ADER:ADE30	Set "1".
To set AN31 pin to input	ADER:ADE31	Set "1".
To set AN32 pin to input	EADERLL:ADE32	Set "1".
To set AN33 pin to input	EADERLL:ADE33	Set "1".
To set AN34 pin to input	EADERLL:ADE34	Set "1".
To set AN35 pin to input	EADERLL:ADE35	Set "1".
To set AN36 pin to input	EADERLL:ADE36	Set "1".
To set AN37 pin to input	EADERLL:ADE37	Set "1".
To set AN38 pin to input	EADERLL:ADE38	Set "1".
To set AN39 pin to input	EADERLL:ADE39	Set "1".

44.7.6 Select the A/D Converter Activation Method

This section explains how to select the A/D converter activation method.

The activation triggers have the following three types.

- Software trigger
- Reload timer rising signal
- External trigger input falling signal

Set the activation trigger with the activation factor selection bits (ADCS1:STS[1:0]).

A/D Activation Factor	Activation Factor Selection Bits (STS[1:0])
To specify the software trigger	Set "00".
To specify the external trigger/software trigger	Set "01".
To specify the reload timer/software trigger	Set "10".
To specify the external trigger/reload timer/software trigger	Set "11".

The A/D converter is activated by the factor that comes first among ones selected.

44.7.7 Activate the A/D Converter

This section explains how to activate the A/D converter.

- Software Trigger Generation Method

  Write into the A/D conversion software trigger bit (ADCS1:STRT) for the software trigger.

Operation	A/D Conversion Software Trigger Bit (STRT)
To generate the software trigger	Write "1".

- Activation Method with the Reload Timer 1

  The setting and activation of the reload timer are required. For details, see "Chapter: Reload Timer".

  When the reload timer output signal is rising by the reload timer underflow, the activation trigger is generated.
- Activation Method with the External Trigger

  Set the external trigger input pin ADTG for the external trigger.

  Set the ADTG pin to peripheral input. For setting method, see "Chapter: I/O Ports".

Operation	Configuration
To set ADTG pin to trigger input	Set the pin to peripheral input. See "Chapter: I/O Ports" for the setting method.

44.7.8 Check the Conversion Completion

This section explains how to check the conversion completion.

The methods to check the conversion completion include the following two methods.

- Check Method with the A/D Conversion Completion Interrupt Request Flag (ADCS1:INT)

(INT)	Meaning
When the read value is "0"	No A/D conversion completion interrupt request
When the read value is "1"	A/D conversion completion interrupt request exists

- Check Method with the Operation Check Bit (ADCS1:BUSY)

(BUSY)	Configuration
When the read value is "0"	A/D conversion completion (being stopped)
When the read value is "1"	A/D conversion is being operated

44.7.9 Read the Conversion Value

This section explains how to read the conversion value.

Conversion values can be read from the data register ADCR0, ADCR1.

ADCR0 stores the lower 8-bit and ADCR1 stores the most significant 2-bit of the conversion result.

Values of the data register are updated every time after one conversion is completed. Normally the final conversion value is stored.

Operation	Register
To read 10-bit conversion value	Read from ADCR1, ADCR0 registers
To read 8-bit conversion value	Read from the ADCR0 register

44.7.10 Stop the A/D Conversion Operation Forcibly

This section explains how to stop the A/D conversion operation forcibly.

Write to the forced stop bit (ADCS1:BUSY).

Operation	Forced Stop Bit (BUSY)
To stop the A/D conversion operation forcibly	Write "0".

Writing "1" to the forced stop bit (BUSY) has no effect on the A/D operation.

44.7.11 Interrupt-related Register

This section explains the interrupt-related register.

Setting of A/D interrupt vector and A/D interrupt level

The relation of the A/D number, interrupt level, and vector is as the table below:

For the interrupt level and interrupt vector, see "Chapter: Interrupt (Control Interrupt Controller)".

Interrupt Vector (Default)	Interrupt Level Setting Bits (ICR[4:0])
#48 Address: $0FFF3C_H$	Interrupt level register (ICR32) Address: 00460_H

44.7.12 Interrupt Type

This section explains the interrupt type.

The interrupt factor is A/D conversion completion only. There are no bits for selection.

44.7.13 Enable/Disable/Clear the Interrupt

This section explains how to enable/disable/clear the interrupt.

Interrupt request enable bit, interrupt request flag

Set the interrupt request enable bit (ADCS1:INTE) for the interrupt enable setting.

Operation	Interrupt Request Enable Bit (INTE)
To disable the interrupt request	Set "0".
To enable the interrupt request	Set "1".

Write into the interrupt request flag (ADCS1:INT) for the interrupt request clear.

Operation	Interrupt Request Flag (INT)
To clear the interrupt request	Write "0". (See "44.4.2 A/D Control Status Register (Upper): ADCS1".)

44.8 Sample Program

This section explains the sample program.

Configuration procedure example 1

The example of A/D conversion for the level input by AN0 (single conversion, software trigger) is described below.

<Initial setting>

-Port Register name.bit name

Port A/D input selection	ADERL. AN7-0

-A/D start/completion channel setting Register name.bit name

Conversion start channel setting	ADSCH .ANS5-0
Conversion completion channel setting	ADECH .ANE5-0

-A/D start/completion channel setting Register name.bit name

Conversion time setting	ADCT .CT5-0 .ST9-8 .ST7-0

-A/D control Register name.bit name

AN0 control	ADCS1 .BUSY
Clearing of interrupt request flag >> Interrupt disable>>	.INT .INTE
	.PAUS
Activation trigger selection>>	.STS .STRT
Conversion mode selection >> Bit length selection>>	.Reserved bit .MD[1:0] .S10
	.Reserved bit

-Interrupt relation Register name.bit name

A/D interrupt level setting	ICR32
I flag setting	(CCR)

<A/D activation>

-A/D control Register name.bit name

A/D interrupt enable	ADCS1 .INT .INTE
A/D0 software activation	ADCS1 .BUSY .STRT

<Interrupt>

-Conversion values read Register name.bit name

Interrupt disable, interrupt request flag clear	ADCS1 .INT .INTE
Conversion values read	D9 to D0
Interrupt enable	ADCS1 .INTE

<Interrupt vector>

Setting of the vector table

<Other>

(Note)

Clock-related setting and setting of "__set_il(value)" are required in advance. See “Chapter: Clock” and “Chapter: Interrupt Control (Interrupt Controller)”.

Program example 1

```
void AD_sample_1(void)
{
        AD_INITIAL();
        AD_ch0_start();
}

void AD_INITIAL(void)
{
        IO_ADERL = 0x01;          /* AN0 only A/D input */

        IO_ADSCH = 0x0000;        /* AN0 setting */
                                  /* 000000 */
        IO_ADECH = 0x0000;        /* AN0 setting */
                                  /* 000000 */


        IO_ADCT0 = 0x0816;        /* Value is the recommended value (at 16 MHz) */
                                  /* 000010 */
                                  /* 00 */
                                  /* 00010110 */


        IO_ADCS1.hword= 0x8000;  /* Setting value: 10000000 00000000 (bit)*/
                                  /* Bit15=1:     (No effect)*/
                                  /* Bit14=0:     Interrupt request clear */
                                  /* Bit13=0:     Interrupt disable */
                                  /* Bit12=0:     Flag clear*/
                                  /* Bit11-10=00:  Software trigger*/
                                  /* Bit9=0:      (No effect)*/
                                  /* Bit8=0:      */
                                  /* Bit7-6=00:    Single conversion */
                                  /* Bit5=0:      10 bits */
                                  /* Bit4-0=00000: */


        IO_ICR[32].bit.ICR =32;   /* Any value */
        __EI();                   /* Interrupt enable */
}

AD_ch0_start()
{
        IO_ADCS1.hword= 0x2000;
                                  /* Bit6=0: AD interrupt flag clear */
                                  /* Bit5=1: AD interrupt enable */
        IO_ADCS1.hword= 0xF200;
                                  /* Bit7=1: "1" writing is required.*/
                                  /* Bit1=1: Software activation */
}
__interrupt void AD_ch0_int()   /* */
{
        IO_ADCS1.hword = 0x8000;
                                          /* Bit6=0: AD interrupt flag clear */
                                          /* Bit5=0: AD interrupt disable */
        [Any storage location] = ADCR1,ADCR0;          /* Storage of conversion values */
        IO_ADCS1.hword= 0xA000;
                                          /* Bit5=1: AD interrupt enable */
}

Interrupt routine specification with the vector table is required.
#pragma intvect AD_ch0_int 48
```

Configuration procedure example 2

A/D conversion is performed for the level input by AN1 to AN3 (Scan conversion, external trigger).
(The external trigger (falling) input is required for ADTG.)

<Initial setting>

-Port Register name.bit name

Port A/D input selection	ADERL. AN7-0
External trigger port setting	See “Chapter: I/O Port”.

-A/D start/completion channel setting Register name.bit name

Conversion start channel setting	ADSCH .ANS5-0
Conversion completion channel setting	ADECH .ANE5-0

-A/D start/completion channel setting Register name.bit name

Conversion time setting	ADCT .CT5-0 .ST9-8 .ST7-0

-A/D control Register name.bit name

AN0 control	ADCS1 .BUSY
Clearing of interrupt request flag >>	.INT
Interrupt disable>>	.INTE
	.PAUS
Activation trigger selection>>	.STS
	.STRT
Conversion mode selection>>	.Reserved bit
Bit length selection>>	.MD[1:0]
	.S10
	.Reserved bit

-Interrupt relation Register name.bit name

A/D interrupt level setting	ICR32
I flag setting	(CCR)

<A/D activation>

-A/D control Register name.bit name

A/D interrupt enable	ADCS1 .INT .INTE

<Interrupt>

-Conversion values read Register name.bit name

Interrupt disable, interrupt request flag clear	ADCS1 .INT .INTE
Conversion values read	D9 to D0
Interrupt enable	ADCS1 .INTE

<Interrupt vector>
Setting of the vector table

<Other>
(Note)
Clock-related setting and setting of __set_il(numerical value) are required in advance. See “Chapter: Clock” and “Chapter: Interrupt Control (Interrupt Controller)”.

Program example 2 (Condition: PCLK = 16MHz)

```
void AD_sample_2(void)
{
        AD _1to3_INITIAL();
        AD_ch1to3_start();
}

void AD_1to3_INITIAL(void)
{
        IO_ADERL = 0x0E;        /* AN1 to AN3 only A/D input */
                                /* Set the ADTG pin to peripheral input. */
        PORT_SETTING_ADTG_IN();

        IO_ADSCH = 0x0001;      /* AN1 setting */
                                /* 000001 */
        IO_ADECH = 0x0003;      /* AN3 setting */
                                /* 000011 */

        IO_ADCT0 = 0x0816;      /* Value is the recommended value (at 16 MHz) */
                                /* 000010 */
                                /* 00 */
                                /* 00010110 */


        IO_ADCS1.hword= 0x8800; /* Setting value: 10001000 00000000 (bit)*/
                                /* Bit15=1:    (No effect)*/
                                /* Bit14=0:    Interrupt request clear */
                                /* Bit13=0:    Interrupt disable */
                                /* Bit12=0:    Flag clear*/
                                /* Bit11-10=01: External trigger*/
                                /* Bit9=0:     (No effect)*/
                                /* Bit8=0:     */
                                /* Bit7-6=00:  Single conversion*/
                                /* Bit5=0:     10 bits */
                                /* Bit4-0=00000: */


        IO_ICR[32].bit.ICR =32; /* Any value */
        __EI();                 /* Interrupt enable */
}

AD_ch01to3_start()
{
        IO_ADCS1.hword= 0xB400;
                                /* Bit6=0: AD interrupt flag clear */
                                /* Bit5=1: AD interrupt enable */
}

__interrupt void AD_ch01to3_int()   /* Interrupted after AN3 conversion */
{
        IO_ADCS1.hword = 0x9400;
                                /* Bit6=0: AD interrupt flag clear */
                                /* Bit5=0: AD interrupt disable */
        [Any storage location] = ADCR1,ADCR0; /* Storage of conversion values */
        IO_ADCS1.hword= 0xA400;
                                /* Bit5=1: AD interrupt enable */
}

Interrupt routine specification with the vector table is required.
#pragma intvect AD_ch01to3_int 48
```

Configuration procedure example 3

A/D conversion is performed for the level input by AN1 to AN3 (Scan conversion, external trigger, DMA use (Request by an interrupt. DMA channel 0)).
(The external trigger (falling) input is required for ADTG.)

<Initial setting>

-Port Register name.bit name

Port A/D input selection	ADERL. AN7-0
External trigger port setting	DDR7. P70

-A/D start/completion channel setting Register name.bit name

Conversion start channel setting	ADSCH .ANS5-0
Conversion completion channel setting	ADECH .ANE5-0

-A/D start/completion channel setting Register name.bit name

Conversion time setting	ADCT .CT5-0 .ST9-8 .ST7-0

-A/D control Register name.bit name

AN0 control	ADCS1 .BUSY
Clearing of interrupt request flag >>	.INT
Interrupt disable>>	.INTE
	.PAUS
Activation trigger selection>>	.STS
	.STRT
Conversion mode selection>>	.Reserved bit
Bit length selection>>	.MD[1:0]
	.S10
	.Reserved bit

-Interrupt relation Register name.bit name

A/D interrupt level setting	ICR32

-DMA relation Register name.bit name

Generation and clear of DMA transfer request (Setting of ICSEL is not required.)	IORR
DMA channel 0 Setting	DCCR
DMA channel 0 Transfer source	DSAR
DMA channel 0 Transfer destination	DDAR
DMA channel 0 Transfers number	DTCR
DMA channel 0 Status clear	DCSR
DMA enable	DMACR.DME

<A/D activation>

-A/D control Register name.bit name

A/D interrupt enable	ADCS1 .INT .INTE

Program example 3 (Condition: PCLK = 16MHz)

```
void AD_sample_3(void)
{
        AD _1to3_INITIAL();
        DMA_Setting();
        AD_ch1to3_start();
}

void AD_1to3_INITIAL(void)
{
        IO_ADERL = 0x0E;                    /* AN1 to AN3 only A/D input */
        IO_PORT1.IO_DDR7.bit. P70 = 0;  /* DDR7.P70 set to input */

        IO_ADSCH = 0x0001;          /* AN1 setting */
                                    /* 000001 */
        IO_ADECH = 0x0003;          /* AN3 setting */
                                    /* 000011 */

        IO_ADCT0 = 0x0816;          /* Value is the recommended value (at 16 MHz) */
                                    /* 000010 */
                                    /* 00 */
                                    /* 00010110 */


        IO_ADCS1.hword= 0x8400;  /* Setting value: 10001000 00000000 (bit)*/
                                    /* Bit15=1:    (No effect)*/
                                    /* Bit14=0:    Interrupt request clear */
                                    /* Bit13=0:    Interrupt disable */
                                    /* Bit12=0:    Flag clear*/
                                    /* Bit11-10=01: External trigger*/
                                    /* Bit9=0:     (No effect)*/
                                    /* Bit8=0:     */
                                    /* Bit7-6=00:   Single conversion */
                                    /* Bit5=0:     10 bits */
                                    /* Bit4-0=00000: */


        IO_ICR[32].bit.ICR =31;     /* Any value */
}

void DMA_Setting() {
        IO_DMA0_DCCR.word= 0x00000000     // Channel 0 disable
        IO_DMA0_DCSR.hword= 0x0000        // Channel 0 status clear
        IO_DMA0_DSAR.word=                //  Transferred from the ADC data register
            &IO_ADCR1.hword;              //
        IO_DMA0_DDAR.word= (any );        // Buffer in SRAM(FIFO with software)
        IO_DMA0_DTCR.hword= 0x0100;        // 256 maximum
        IO_DMAREQCLR_IORR0.byte=          // IO transfer request register
            (( 0x30 - 0x10 ) + 0x40);     // Vector number #0x30, IOE bit set
        IO_DMA0_DCCR.word= 0x8010B010;    // Channel 0 enable, request by an interrupt,
                                          // Block transfer, transfer source address fixed, ST = 1
                                          // Transfer destination address increment
                                          // 1 block = 2 bytes X once
        IO_DMACR.word= 0x80000000         // DMA enable
}

AD_ch01to3_start()
{
        IO_ADCS1.hword= 0xB400;
                                    /* Bit6=0: AD interrupt flag clear */
                                    /* Bit5=1: AD interrupt enable */
}
```

44.9 Notes

This section explains notes of the A/D converter.

Precautions when using the A/D converter are described.

Power-on Sequence

The A/D converter power (AV_{CC}, AVRH)-on and voltage application to analog input must be performed after MCU power (V_{CC}) is turned on.

$V_{CC} \geq A_{VCC} \geq$ AVRH
$A_{VCC} \geq$ AN (Analog applied voltage) $\geq V_{SS}$
Observe above.

For the input impedance of analog input pins, the A/D converter includes a sample hold circuit, and takes voltage of the analog input pin in a capacitor for sample hold after A/D conversion activation. Therefore, if the output impedance of the analog input external circuit is high, the analog input voltage might not be stabilized during the analog input sampling period. Thus, make the output impedance of the external circuit low enough.

If the output impedance of the external circuit cannot be made low, extend the sampling time enough.

As | AVRH-AV_{SS} | gets smaller, an error gets larger relatively.

44.10 Term Definition for A/D Converter

This section explains the term definition for the A/D converter.

- Resolution

  Analog change identifiable by an A/D converter

- Linearity Error

  Deviation between the line that connect the following transition points and the actual conversion characteristics:

  The zero transition point (00 0000 0000 ↔ 00 0000 0001) and;

  The full scale transition point (11 1111 1110 ↔ 11 1111 1111)

- Differential Linearity Error

  Deviation from the ideal value of input voltage necessary for 1LSB change of output code

$$1LSB = \frac{V_{FST} - V_{OT}}{1022} \quad [V]$$

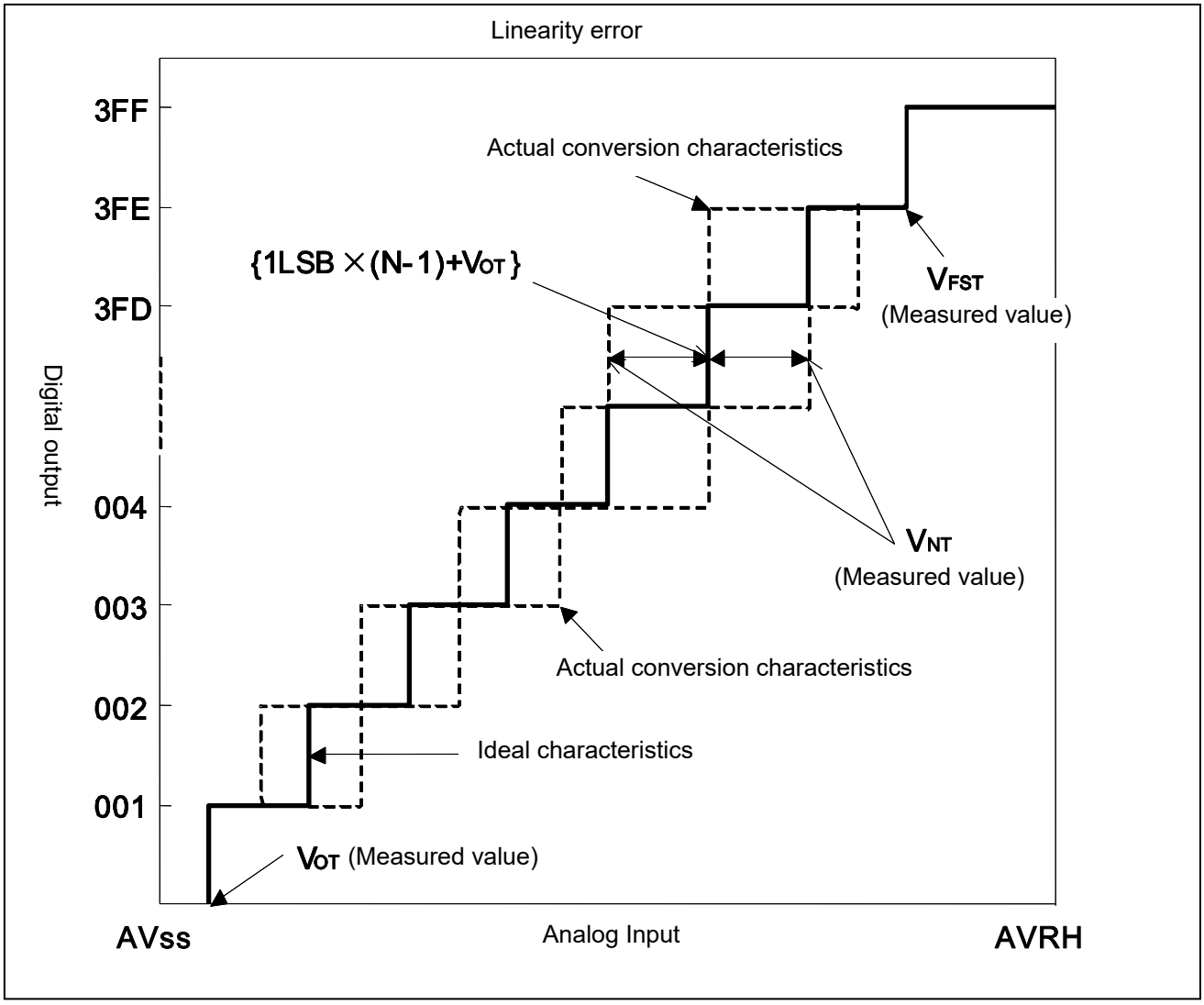
V_{OT} : Voltage for digital output to transit from $(000)_H$ to $(001)_H$

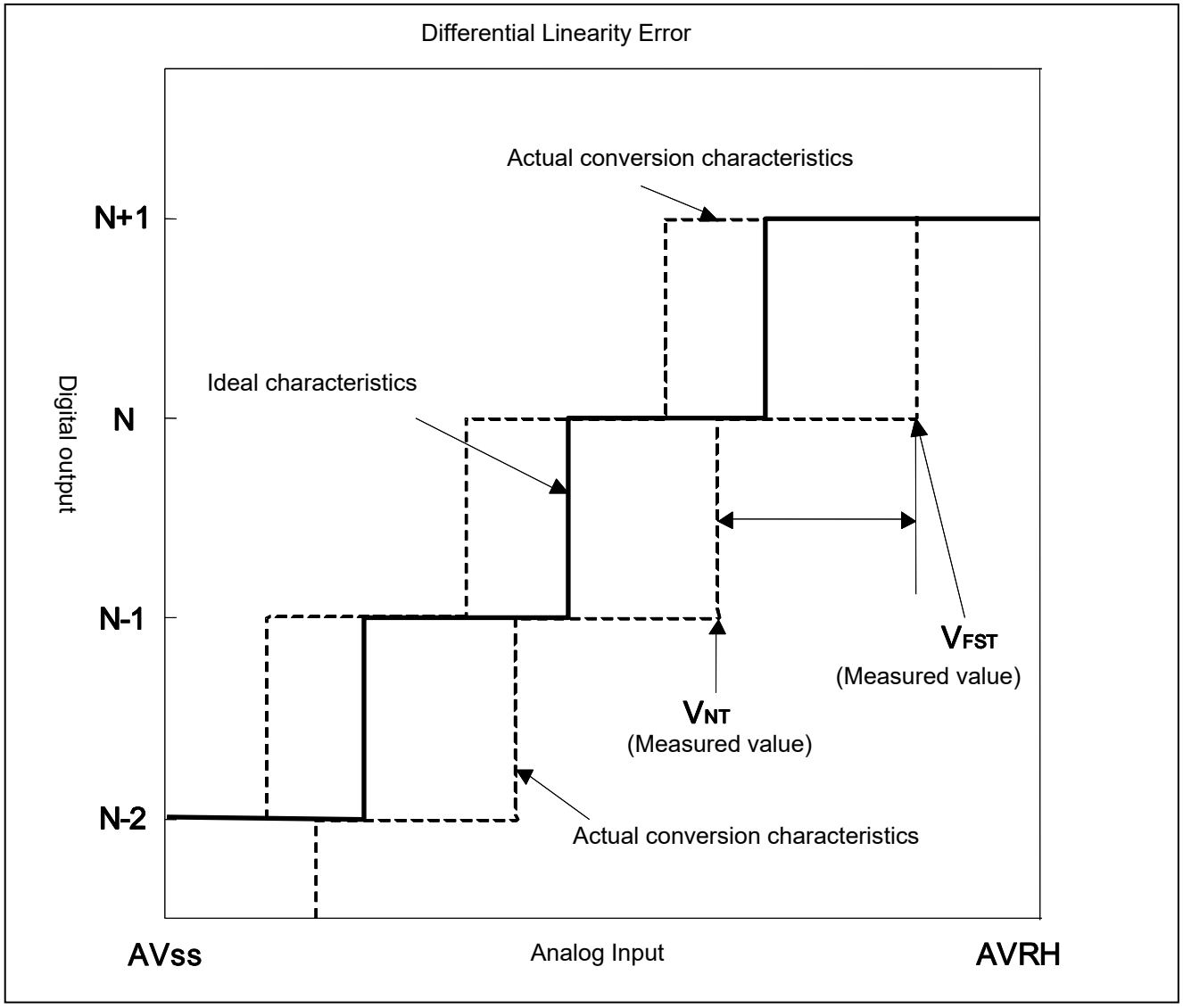
V_{FST}: Voltage for digital output to transit from $(3FE)_H$ to $(3FF)_H$

$$\text{Linearity error of digital output N} = \frac{V_{NT} - \{1LSB \times (N-1) + V_{OT}\}}{1LSB} \quad [LSB]$$

$$\text{Differential linearity error of digital output N} = \frac{V_{(N+1)T} - V_{NT}}{1LSB} - 1 \quad [LSB]$$

V_{NT}: Voltage for digital output to transit from (N+1) to N


Linearity error
3FF
3FE
3FD
004
003
002
001
Digital output
Actual conversion characteristics
{1LSB × (N-1)+VOT}
VFST
(Measured value)
VNT
(Measured value)
Actual conversion characteristics
Ideal characteristics
VOT (Measured value)
AVss
Analog Input
AVRH


Differential Linearity Error
Actual conversion characteristics
N+1
Ideal characteristics
Digital output
N
N-1
N-2
VFST
(Measured value)
VNT
(Measured value)
Actual conversion characteristics
AVss
Analog Input
AVRH

- Total Error

  Difference between an actual value and theoretical value, and an error including a zero transition error/full scale transition error/linearity error.

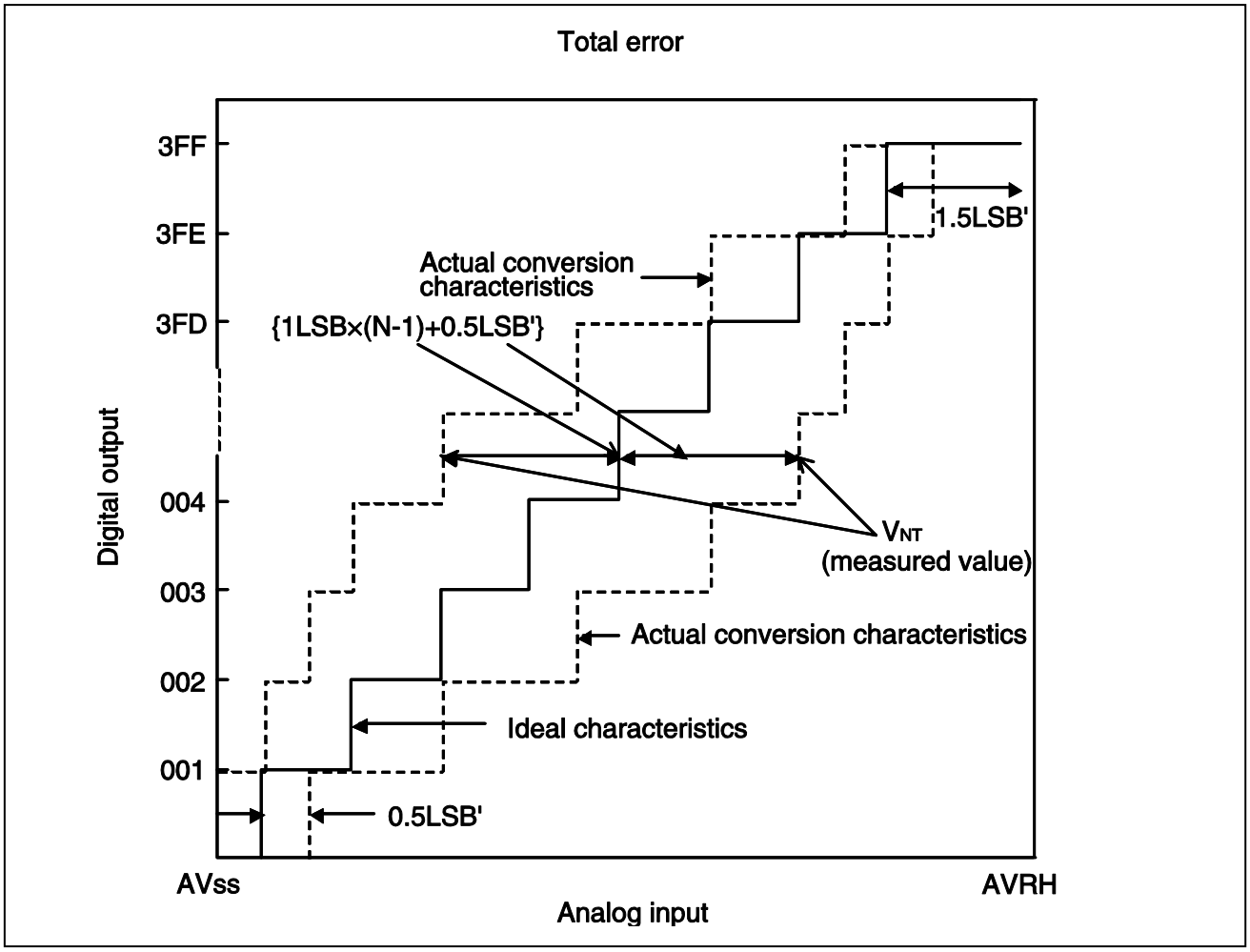
$$\text{1LSB' (Ideal value)} = \frac{AVRH - AV_{SS}}{1024} \quad [V]$$

$$V_{OT}\text{' (Ideal value)} = AV_{SS} + 0.5LSB' \quad [V]$$

$$V_{FST}\text{' (Ideal value)} = AVRH - 1.5LSB' \quad [V]$$

$$\text{Total error of digital output N} = \frac{V_{NT} - \{1LSB' \times (N - 1) + 0.5LSB'\}}{1LSB'}$$

V_{NT}: Voltage for digital output to transit from (N + 1) to N



45. Flash Memory

This chapter explains the flash memory.

45.1 Overview
45.2 Features
45.3 Configuration
45.4 Registers
45.5 Operation

45.1 Overview

This section explains an overview of the flash memory.

The size of the flash memory built in this series is

576KB (512K+64KB): CY91F575

1088KB (1024K+64KB): CY91F577

Error correction codes (ECC) are attached.

45.2 Features

This section explains features of the flash memory.

- Usable Capacity:
  - ☐ CY91F575: 576KB (large sector group 128K × 4 bytes + small sector group 16K × 4 bytes)
  - ☐ CY91F577: 1088KB (large sector group 128K × 8 bytes + small sector group 16K × 4 bytes)

  Since this series has ECC code storage, there are 6 bits of built-in flash memory for every 4 bytes described above.

  Note: The sector itself has 64KB (large sector)/8KB (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 128KB (large sectors)/16KB (small sectors) in real use.
- High Speed Operation:

  Reading at the word (32-bit) unit can be performed in 1 cycle at 80 MHz.
- 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[1]) by CPU
- Can be Read, Written, or Erased (Automatic Algorithm[1]) by ROM Writer
- Security Function
  - ☐ In order to prevent the content of flash memory from being read by a third party, when security is on, operation from external source after instruction fetch and writing/erasing other than chip erase are suppressed.
  - ☐ After password authentication when using the on-chip debugger (OCD), this can be read externally using the OCD even when security is on.
- Error Correction Code (ECC) Security 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 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.
  - ☐ Errors are detected when data are read during chip erase/sector erase. When you need to read data in a erase state (FFFF) correctly, read it after writing "FFFF" without fail.

[1]: Automatic algorithm = Embedded Algorithm

45.3 Configuration

This section explains the configuration of the flash memory.

45.3.1 Block Diagram

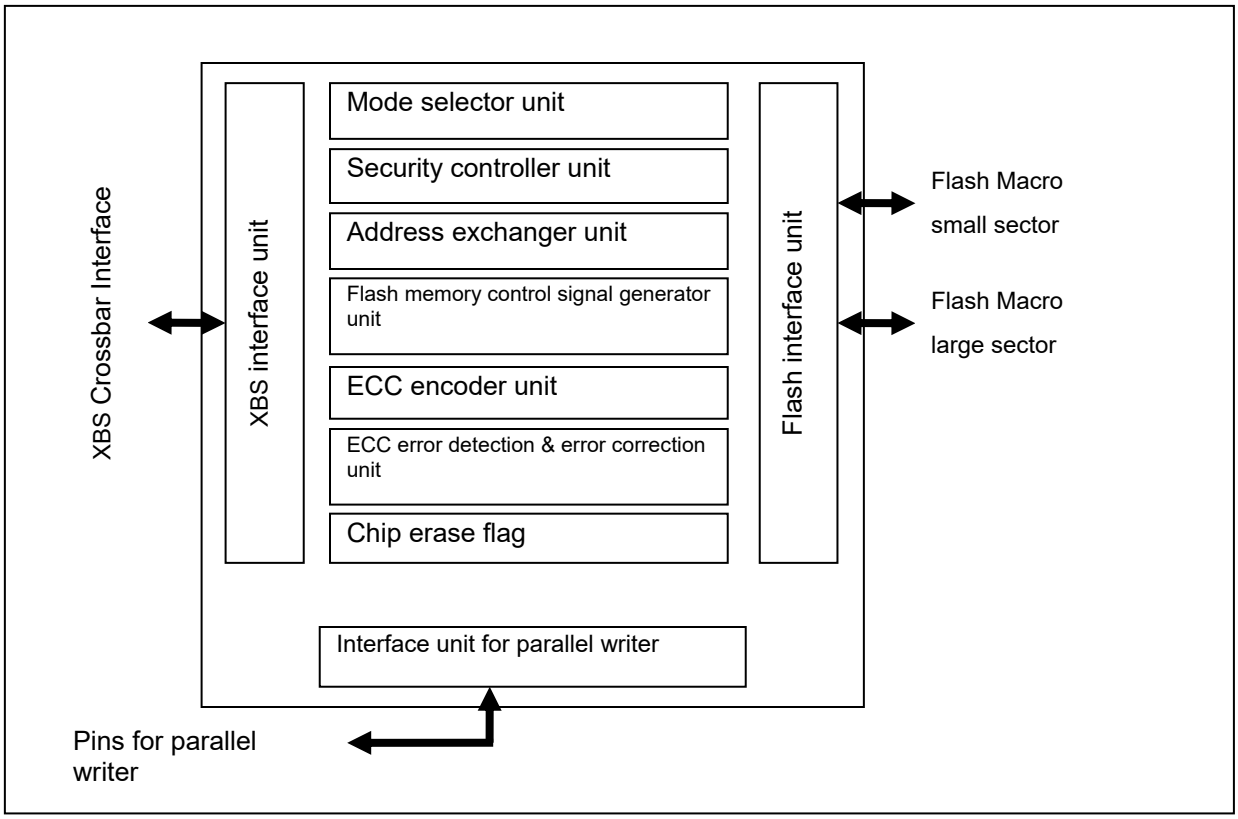
45.3.2 Sector Configuration Diagram

45.3.3 Sector Number and Flash Macro Number Correspondence Chart

45.3.1 Block Diagram

The block diagram of the flash memory is shown below.

Figure 45-1. Block Diagram (1024KB/512KB+64KB products)



45.3.2 Sector Configuration Diagram

The sector configuration diagram of the flash memory is shown below.

Figure 45-2. Sector Configuration Diagram (1024KB+64KB)

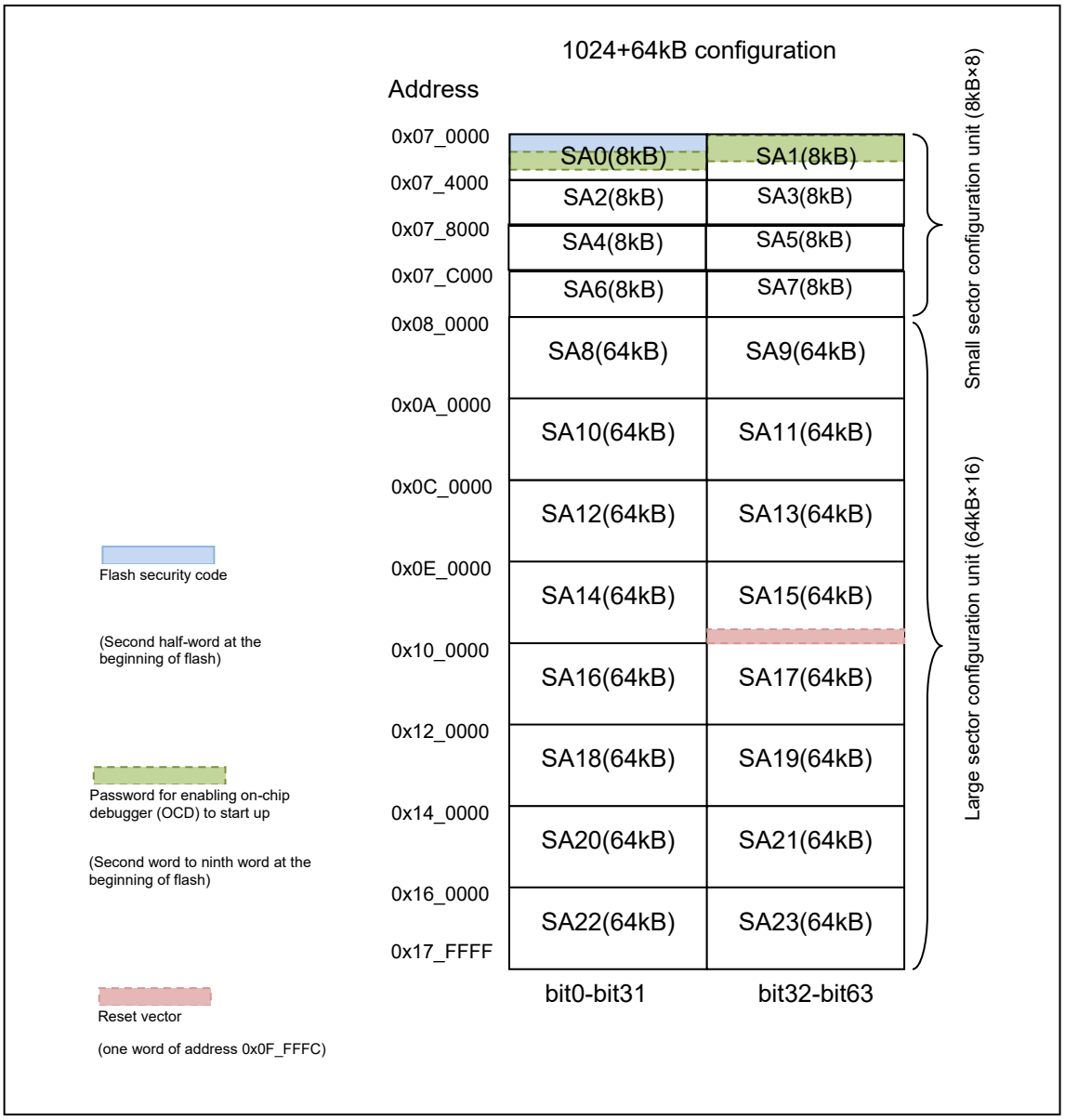
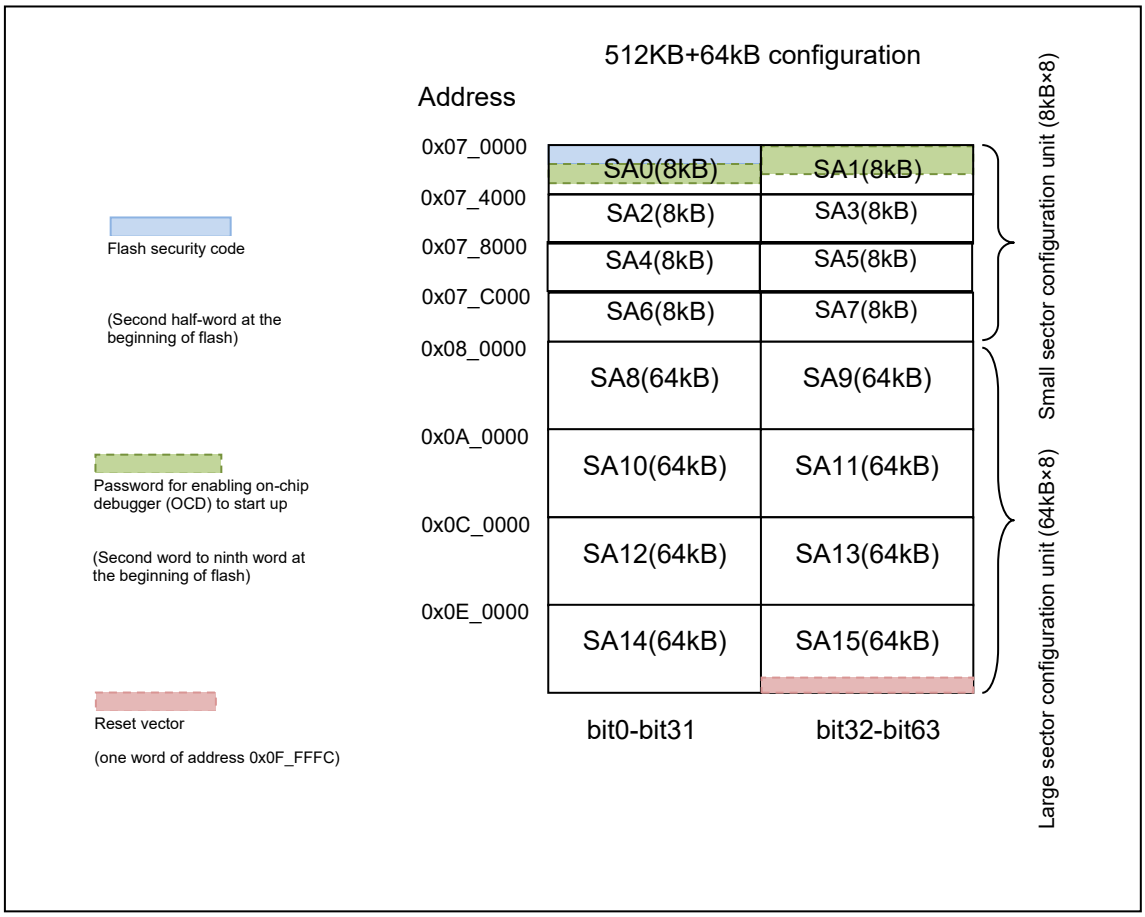


Figure 45-3. Sector Configuration Diagram (512KB+64KB)



Notes:

- 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 default state just after the flash erase (all bits=1).

45.3.3 Sector Number and Flash Macro Number Correspondence Chart

A sector configuration of the flash memory is shown below.

Table 45-1. Sector Number Table (1024KB/512KB+64KB)

Sector Number	Address	Sector Size	Remark
SA0	0x07_0000 to 0x07_3FFB (Lower 32 bits)	8KB	Flash security code area (0x07_0002 to 0x07_0003) Password area for enabling on-chip debugger (OCD) startup (0x07_0008 to 0x07_000B, 0x07_0010 to 0x07_0013, 0x07_0018 to 0x07_001B, 0x07_0020 to 0x07_0023)
SA1	0x07_0004 to 0x07_3FFF (Upper 32 bits)	8KB	Password area for enabling on-chip debugger (OCD) startup (0x07_0004 to 0x07_0007, 0x07_000C to 0x07_000F, 0x07_0014 to 0x07_0017, 0x07_001C to 0x07_001F)
SA2	0x07_4000 to 0x07_7FFB (Lower 32 bits)	8KB	
SA3	0x07_4004 to 0x07_7FFF (Upper 32 bits)	8KB	
SA4	0x07_8000 to 0x07_BFFB (Lower 32 bits)	8KB	
SA5	0x07_8004 to 0x07_BFFF (Upper 32 bits)	8KB	
SA6	0x07_C000 to 0x07_FFFB (Lower 32 bits)	8KB	
SA7	0x07_C004 to 0x07_FFFF (Upper 32 bits)	8KB	
SA8	0x08_0000 to 0x09_FFFB (Lower 32 bits)	64KB	
SA9	0x08_0004 to 0x09_FFFF (Upper 32 bits)	64KB	
SA10	0x0A_0000 to 0x0B_FFFB (Lower 32 bits)	64KB	
SA11	0x0A_0004 to 0x0B_FFFF (Upper 32 bits)	64KB	
SA12	0x0C_0000 to 0x0D_FFFB (Lower 32 bits)	64KB	
SA13	0x0C_0004 to 0x0D_FFFF (Upper 32 bits)	64KB	

Sector Number	Address	Sector Size	Remark
SA14	0x0E_0000 to 0x0F_FFFB (Lower 32 bits)	64KB	
SA15	0x0E_0004 to 0x0F_FFFF (Upper 32 bits)	64KB	Reset vector position (0x0F_FFFC to 0x0F_FFFF)
SA16	0x10_0000 to 0x11_FFFB (Lower 32 bits)	64KB	
SA17	0x10_0004 to 0x11_FFFF (Upper 32 bits)	64KB	
SA18	0x12_0000 to 0x13_FFFB (Lower 32 bits)	64KB	
SA19	0x12_0004 to 0x13_FFFF (Upper 32 bits)	64KB	
SA20	0x14_0000 to 0x15_FFFB (Lower 32 bits)	64KB	
SA21	0x14_0004 to 0x15_FFFF (Upper 32 bits)	64KB	
SA22	0x16_0000 to 0x17_FFFB (Lower 32 bits)	64KB	
SA23	0x16_0004 to 0x17_FFFF (Upper 32 bits)	64KB	

45.4 Registers

This section explains registers of the flash memory.

Table 45.4-1. 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

45.4.1 Flash Control Register: FCTLR

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)

It is the write enable bit to flash. Setting this bit configures CPU programming mode. Use the FSTR:FRDY bit to check whether or not 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	Flash write disabled (Initial value)
1	Flash write enabled

Note:

When writing to FLASH, instruction fetch from FLASH is disabled.

[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 two bits, bit11 and bit12, 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 read access time of the flash memory is 12.5ns, it is prohibited to access the flash memory at 80MHz or more.
It is indispensable to insert wait with these bits. Please set it to FAW=1(1wait) when you access it at 80MHz or more.

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	1cycle
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 80MHz or less, and set value of the FAW bits to 2'b00(0cycle) 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	Flash access Enable (Initial value)
1	Flash access Disable

[bit6, bit5] Reserved

Reserved bits. The read value is undefined. When writing, always write "0" to these bits.

[bit4] RDYF (Ready Flag): RDY negating instruction when branch is accessed

The wait cycle insertion when the branch is access is directed. When the branch is accessed, 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 FLASH I/F 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	Wait cycle insert

[bit3 to bit0] Reserved

These bits are reserved. The read value is undefined. When writing, always write "0" to these bits.

45.4.2 Flash Status Register: FSTR (Flash Status Register)

The bit configuration of the flash status register is shown below.

This register indicates the flash state.

FSTR: Address 0843H (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)	Clears this bit
1	ECC error correction occurred during data read	No effect

If there are errors in 2 bits 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 Elapsed Flag Bit)), the flash memory will go into the HANG state. If this bit becomes "1", issue a reset command (See "45.5.3.1 Command Sequence").

The correct value might not be read out immediately after a command of automatic algorithm has been issued. In that case, ignore the first 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 data cannot be written or erased while the operation is in progress.

FRDY	Description
0	Operation in progress (write/erase disabled, read status enabled)
1	Operation finished (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 first read value of this bit after the command issuance.

45.4.3 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 flash I/F. This register is shared among program flash and WorkFlash.

FLIFCTLR: Address 2308H (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 disable

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 cycle time.

DFWDSBL	Description
0	Wait cycle enabled (Initial value)
1	Wait cycle disabled

Note:

When you change the FLIFCTLR.DFWDSBL bit from "1" to "0", be sure to do so after you write FCTR.FAW="00".

[bit3] Reserved

This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] Reserved

This bit is reserved. When writing, always write "0" to this bit.

[bit1] ECCDSBL1 (ECC Disable1): ECC function disable 1

This bit sets the ECC function enabled/disabled while the CPU is accessing the WorkFlash memory in order to write or fetch data.

ECCDSBL1	Description
0	ECC function enabled (Initial value)
1	ECC function disabled

[bit0] ECCDSBL0 (ECC Disable0): ECC function disable 0

This bit sets the ECC function enabled/disabled while the CPU is accessing the program flash memory in order to write or fetch data.

ECCDSBL0	Description
0	ECC function enabled (Initial value)
1	ECC function disabled

45.4.4 Flash I/F Feature Extension Register 1: FLIFFER1

The bit configuration of the flash I/F 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.

45.4.5 Flash I/F Feature Extension Register 2: FLIFFER2

The bit configuration of the flash I/F 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.

45.5 Operation

This section explains operations of the flash memory.

45.5.1 Access Mode Setting
45.5.2 Programming Flash Memory by CPU
45.5.3 Automatic Algorithm
45.5.4 Reset Command
45.5.5 Write Command
45.5.6 Chip Erase Command
45.5.7 Sector Erase Command
45.5.8 Sector Erase Suspend Command
45.5.9 Security Function
45.5.10 Notes on Using Flash Memory

45.5.1 Access Mode Setting

The access mode setting is shown below.

The flash memory in this series has the following three modes. The methods for configuring modes (1) and (2) are explained in this section. See the instruction manual of the ROM writer you are using for details on (3).

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 reading and writing, only Half-word access)
3. Flash memory mode
   (Access to flash memory from external is enabled.)

45.5.1.1 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. When the FRDY bit of the flash status register (FSTR) is "1", read from the flash memory is enabled in this mode. In the mode, write to the flash memory is disabled. After released reset, the mode will be the CPU-ROM mode.

45.5.1.2 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.

45.5.2 Programming Flash Memory by CPU

This section explains programming flash memory by CPU.

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 two operations to write one half-word. If this procedure is not followed, the written values will not be read correctly because the values will be written to flash memory without calculating the ECC.

1. Set the flash access size to 16 bits. (FCTLR:FSZ[1:0]=01)
2. Issue the write command. Write address = PA, write data = PD[31:16] See "45.5.5 Write Command" for details on the write command.
3. Read the hardware sequence flag until the write has finished. See "45.5.3.2 Automatic Algorithm Execution State" for details on reading the hardware sequence flag.
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 from erasing the flash memory.

PA: Write target address (word aligned)
PD[31:0]: Write data
PD[31:16]: Write data upper 16 bits
PD[15:0]: Write data lower 16 bits

45.5.3 Automatic Algorithm

This section explains the automatic algorithm.

When using CPU programming mode, writes and erasures of flash memory are performed by starting the automatic algorithm. This section explains the automatic algorithm.

45.5.3.1 Command Sequence

The command sequence is shown below.

The automatic algorithm starts when half-word (16-bit) data is written to the flash memory once to six times in a row. This is called a command. The command sequences are shown below.

Table 45-2. Command Sequence

Command	Number of Writes	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 bits. The upper 8 bits can be any value. The commands are written as half-words or bytes.

The addresses written in the table only show the lower 16 bits. Set the upper 16 bits 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 aligned)
PD: Write data (Write as half-word.)
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:

- Do as follows to LSB 2-bit of the sector address (SA) to input when the command address and the sector erase command are issued.
  - ☐ When half-word access: 2'b00
  - ☐ When byte access: 2’b01 or 2’b11

  Example 1:

  When byte access and command address = (LSB 2-bit of the standard command address is changed to 2’b01.)

  $yAA8_H \rightarrow yAA9_H$, $x554_H \rightarrow x555_H$, and SA → { SA[31:2] and 2'b01 }

  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector)

  Example 2:

  When byte access and command address = (LSB 2-bit of the standard command address is changed to 2’b11.)

  $yAA8_H \rightarrow yAAB_H$, $x554_H \rightarrow x577_H$, and SA → { SA[31:2] and 2'b11 }

  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector)
- If an incorrect address value or data value is written, or an incorrect sequence is written, the commands written till then will be cleared.

Reset Command

Each command input shown in Table 45-2 input to send the reset command to the object flash memory till then can try to be canceled, and to input the command from the first time again.

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 the reset command is input.

Read Command

The flash memory can be read by sending the reading command to the target sector. If the read command is issued, the flash memory stays in read state until another command is issued.

Program (Write) Command

If a write command is sent to the target sector four 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 CPU programming mode, data is written in half-words or bytes. Once the fourth 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 "45.5.5 Write Command" for details on the actual operation.

Notes:

- If the fourth write command (write data cycle) is written to an odd address when writing in half-word, the write is not performed correctly. Always write to an even address.
- In the first write command sequence, only a single half-word data can be written. If you want to write multiple data, issue one write command sequence for each data.
- When security is ON, there are restrictions in the procedure for writing the flash. See "45.5.9.4 Flash Access Restrictions When Security is ON" for details.

Chip Erase Command

If the chip erase command is sent to the target sector six times in a row, all sectors of the flash memory can be erased in one step. Once the sixth 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 flash memory before the chip erases to verify the margins (preprogramming). Furthermore, while verifying the margin, there is no need to control the flash memory externally.

See "45.5.6 Chip Erase Command" for details on the actual operation.

Sector Erase Command

If the sector erase command is sent to the target sector six times in a row, the sector of the flash memory is erased. When 40 µs elapses (timeout period) after the sixth 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 40µs (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 flash memory before erasing the sector, and there is no need to write to 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 "45.5.7 Sector Erase Command" for details on the actual operation.

Note:

When security is ON, there are restrictions in the procedure for erasing the sector of the flash. See "45.5.9.4 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 erase resume command is accepted, the state comes back to the sector erase condition, resuming the erase operation.
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 erase deletion, and the sector erase operation is restarted at once.

See "45.5.8 Sector Erase Suspend Command" for actual operation.

Notes:

16.7µs 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.

45.5.3.2 Automatic Algorithm Execution State

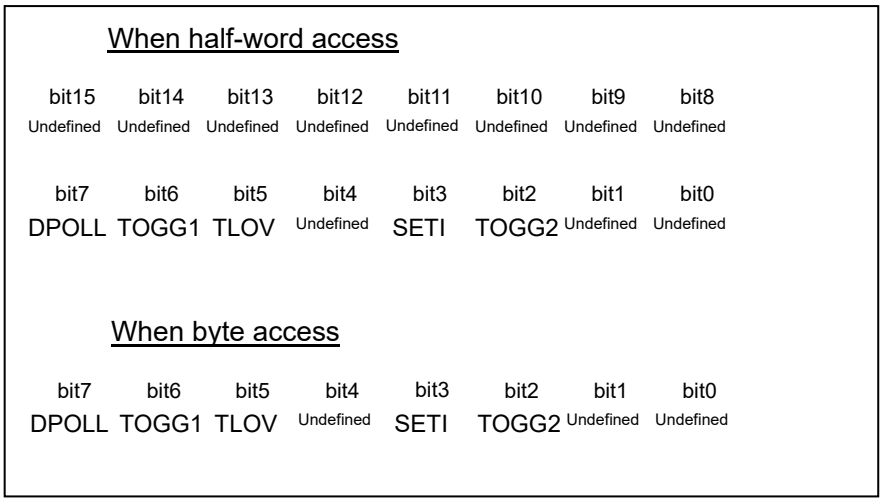
This section explains the automatic algorithm execution state.

Because writing and erasing flash memory is performed by an automatic algorithm, the operating state can be checked by the hardware sequence flag using the FRDY bit of the FLASH status register (FSTR) to determine whether or not the automatic algorithm is executing.

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 following shows the bit configuration of the hardware sequence flag.

Figure 45-4. Bit Configuration of Hardware Sequence Flag



Notes:

- It is impossible to read by word access. Always read using half-word or byte access in CPU programming mode.
- In CPU ROM mode, the hardware sequence flag cannot be read no matter which address is read.

- Each bit and Flash Memory Status

  Table 5-2 shows the correspondence between the state of each bit of the hardware sequence flag and the flash memory status.

Table 45-3. Correspondence between Flags and Flash Memory Status

Status		DPOLL	TOGG1	TLOV	SETI	TOGG2
Run	Writing	Inverted data[1]	Toggle	0	0	-
	Sector/Chip erasing	0	Toggle	0	1	-
Time limit exceed	Write command	Inverted data[1]	Toggle	1	0	-
	Sector erase/Chip erase command	0	Toggle	1	1	-
Sector erase suspend	Erase target sector	-	-	-	-	Toggle

[1]: 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 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 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)	"0" is read from the sector erase suspend sector.
Sector erase operation completion	"1" is read 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 status.

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 exceed 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 state.

During write / sector erase / chip erase
The next values are read.

"0"	Within the rated time
"1"	Exceeds 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 40µs 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 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. Reads out the next value 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"	When exceeding 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 (read), but the target sector for erase cannot be read. This flag indicates that output data is toggled and the 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

45.5.4 Reset Command

The reset command is explained.

The flash memory can be reset by sending reset commands sequentially 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 data read 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.

45.5.5 Write Command

The write command is shown below

Writes are performed in the following order.

1. Send write commands sequentially to the target sector

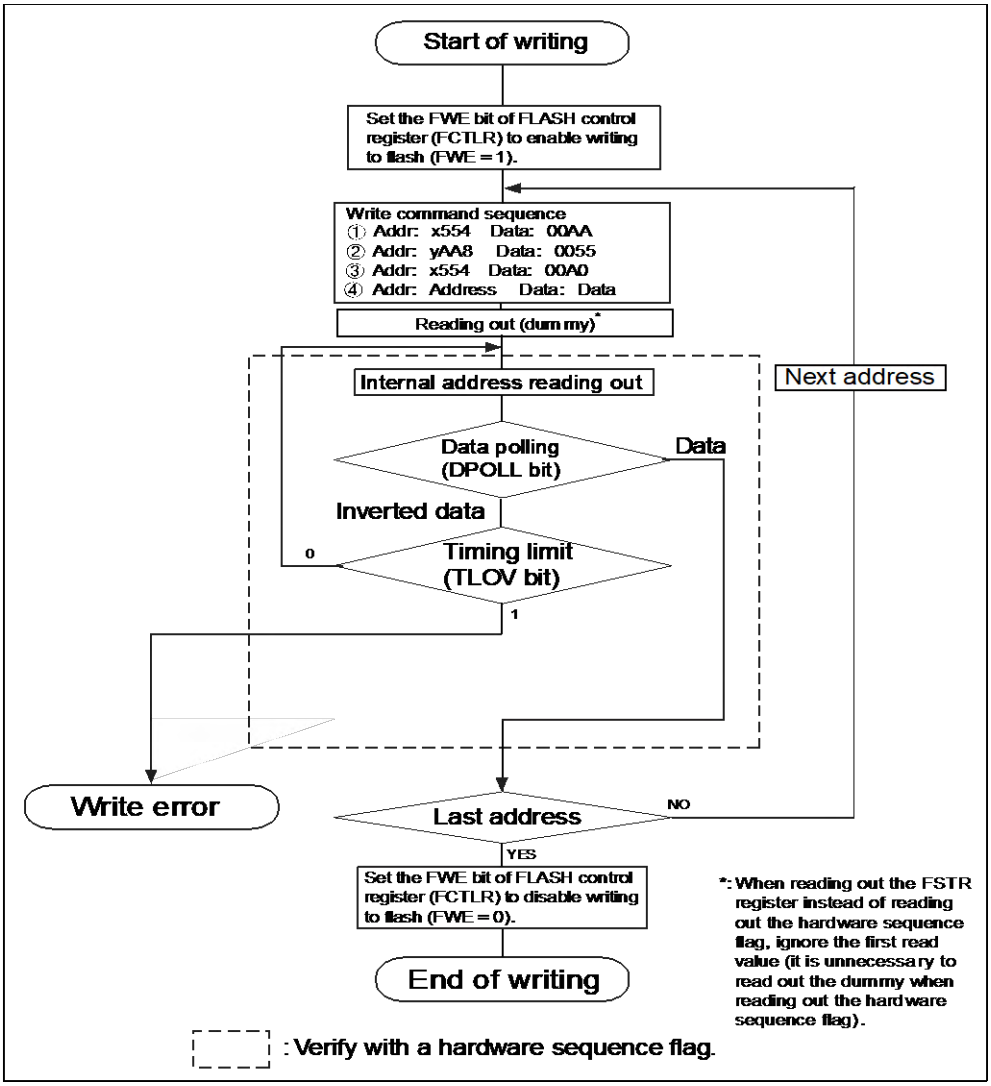
   The automatic algorithm is started and data is written to the flash memory. After issuing a write command, there is no need to control the flash memory externally.

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 (inverse data) of the value of bit7 of the last written data is read out.

The following shows an example of a write operation to the flash memory.

Figure 45-5. Example of Write Procedure



Notes:

- Once the write has finished, because the flash memory returns to read mode, write addresses are no longer accepted.
- See "45.5.3 Automatic Algorithm" for details on write commands.
- Because the DPOLL bit and the TLOV bit of the hardware sequence flag change at the same time, even when the TLOV bit is "1", it is necessary to confirm again.
- When the TOGG1 bit and the TLOV bit of the hardware sequence flag change to "1", the toggle operation stops at the same time. Therefore, it is necessary to confirm the TOGG1 bit again even the TLOV bit is "1".
- 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 one 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 series has the ECC bit added, data always needs to be written as 32-bit by two 16-bit writes. See "45.5.2 Programming Flash Memory by CPU" for the procedure.

45.5.6 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 "45.5.3 Automatic Algorithm" for details on the chip erase command.

1. Send chip erase commands sequentially to a sector in the flash 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.

Notes:

- 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 needs to write to 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 "45.5.9.3 Unlocking Flash Security" for details.

45.5.7 Sector Erase Command

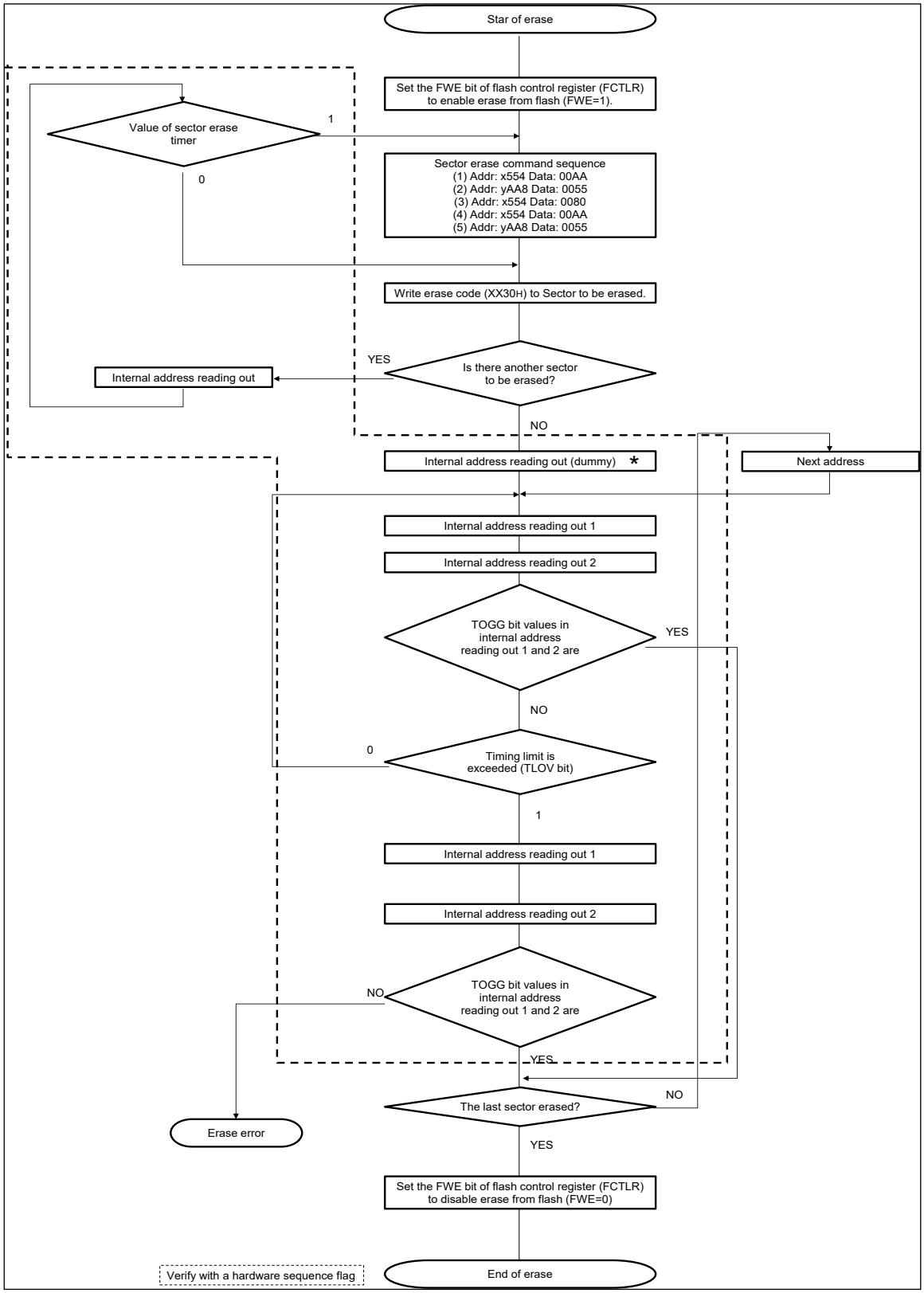
The sector erase command is shown below.

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 40µs (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 45-6. Example of Sector Erase Procedure



Notes:

- The time required to erase the sector is [(sector erase time + sector write time (pre- program)) × number of sectors].
- When the sector erase operation finishes, the flash memory returns to the read/reset state.
- See "45.5.3 Automatic Algorithm" for details on the sector erase command.
- Because the DPOLL bit and the TLOV bit of the hardware sequence flag change at the same time, even when TLOV bit is "1", it is necessary to confirm again.
- When the TOGG1 bit and the TLOV bit of the hardware sequence flag change to "1", toggle operation stops at the same time. Therefore, it is necessary to confirm the TOGG1 bit again even the TLOV bit is "1".
- If a command other than the sector erase command is issued during sector erase including the timeout period, the flash memory returns to the read/reset state. In this case, because the flash memory is reset, the previously issued command or multiple sector erase commands become invalid. To erase the sector, reissue the sector erase commands from the beginning.
- When the automatic erase algorithm is started, "0" is written to the cells to erase in flash memory before erasing the sector, and there is no need to write to 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.

45.5.8 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.

The reading operation of the memory cell of the sector that is not the erase target becomes possible in the suspend condition of the sector erase. 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 of 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.

- The TOGG1 bit that does the toggle while erasing the sector does not toggle in the sector erase suspend condition.
- FRDY of the flash status register becomes "1".

The thing that entered the sector erase suspend condition can be confirmed by using these.

Note:

16.7 µs 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.

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 45-2 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.

Accepting the erase restart command, the flash memory comes back to the sector erase condition, resuming the erase operation.

45.5.9 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, the operation after an instruction fetches from the external bus, and writes and erases other than chip erase are suppressed. See "45.5.9.4 Flash Access Restrictions When Security is ON" for details on the restrictions.

45.5.9.1 Flash Security On/Off Determination When Reset Released

Flash security on/off determination when reset released is shown below.

The flash interface of this series reads two 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.

45.5.9.2 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 45-3), security is turned on. Once security has been turned on, the security is not turned off unless the entire flash memory area is erased.

45.5.9.3 Unlocking Flash Security

Unlocking flash security is shown below.

The chip erase command can be performed on all flash macros using the following procedure.

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 order of the chip erase to each flash macro is recommended to be executed from the viewpoint of the data protection stored in the flash macro as shown in the above-mentioned.

45.5.9.4 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 45-4. Access Restrictions when Security is ON

Operating Mode	Access Restriction
User/External bus	In normal mode (the state where there are no access restrictions due to the following flash security violations), writing in the security information area (first nine 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 source. Accesses to the flash memory are not accepted thereafter. The flash memory returns to the normal state by reset.
Other than above (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 "45.5.9.3 Unlocking Flash Security".

Furthermore, while the security is ON, when a data read is performed to the security information storage area (9 words at the start 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" 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.

45.5.10 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 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 bits by two 16-bit writes. See "45.5.2 Programming 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.
- Once authentication is complete using the debugger (OCD) password, OCD can be used to read the content of flash memory externally even if security is on. If you want to prevent a third party from reading, always set a password for enabling on-chip debugger (OCD) startup.
- Changing to the standby state is a prohibition during FLASH program/erase.
- Because of build-in ECC in this flash memory, the data superscription to the address where some values have already been written cannot be done.

46. WorkFlash Memory

This chapter explains the WorkFlash memory.

46.1 Overview
46.2 Features
46.3 Configuration
46.4 Registers
46.5 Operation

46.1 Overview

This section explains the overview of the WorkFlash memory.

The size of the WorkFlash in this series is 64KB. Error correction codes (ECC) are attached.

46.2 Features

This section explains features of the WorkFlash memory.

- Usable capacity:
  CY91F575/CY91F577 : 64KB (8KB × 8 sectors)
  For ECC code storage, there are 6 bits of built-in flash memory for every 4 bytes.
- High-speed operation:
  read on a word-by -word basis (32 bit) is possible by 80 MHz × 2 cycle.
- Write from external: Possible from ROM writer
- Operation mode:
  1. CPU-ROM mode
     (CPU/DMA accesses flash memory. Only read)
     Only data access is enabled. Instruction fetch is not enabled.
  2. CPU programming mode
     (CPU accesses flash memory. Read/Write/Erase)
  3. Flash memory mode
     (Access flash memory from the external is enabled.)
- Security function
  - ☐ Operations after instruction fetch from external and write/erase except for chip erase at security on are inhibited to avoid reading out flash memory data by an outsider.
  - ☐ The use of on-chip debugger (OCD) enables read from external by using OCD, even if security is on after password authentication.
- 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.

Automatic Algorithm = Embedded Algorithm

46.3 Configuration

This section explains the configuration of the WorkFlash memory.

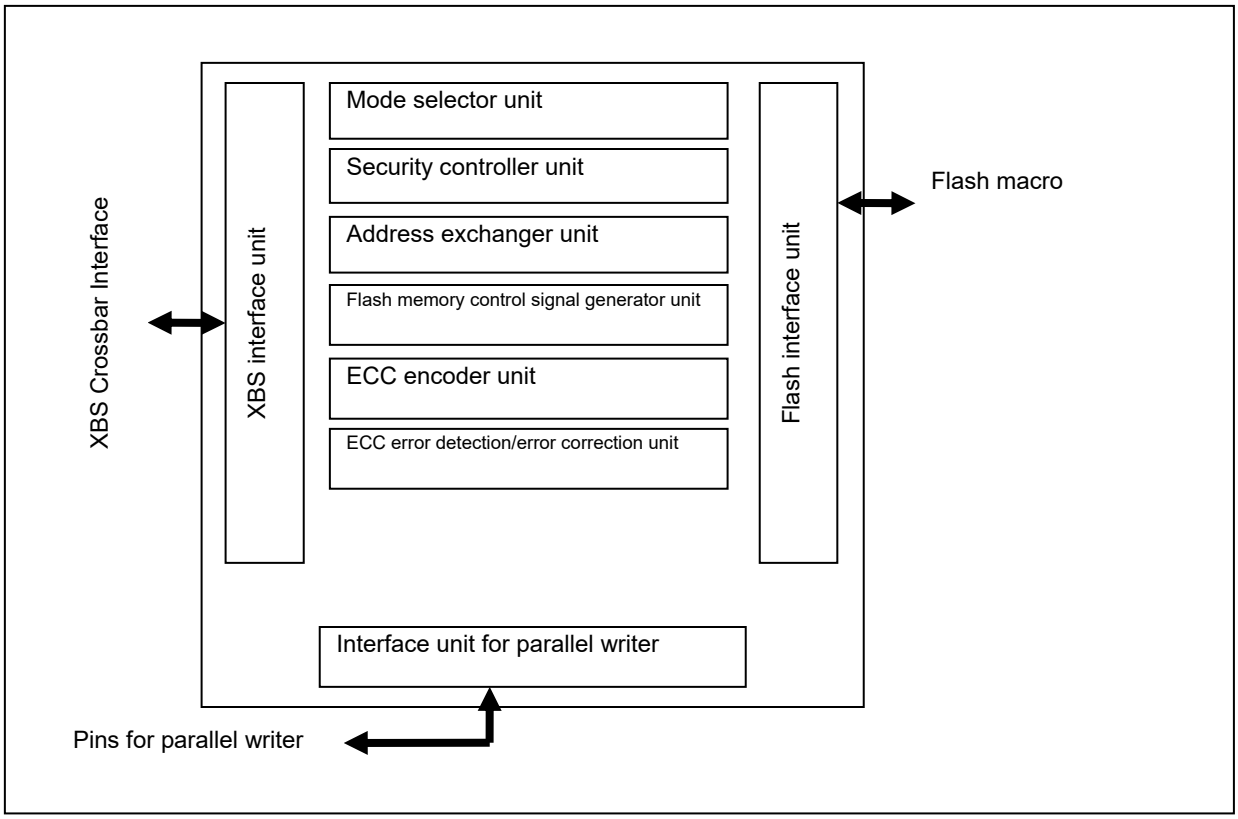
46.3.1 Block Diagram

46.3.2 Sector Configuration Diagram

46.3.1 Block Diagram

This section shows the block diagram of the WorkFlash memory.

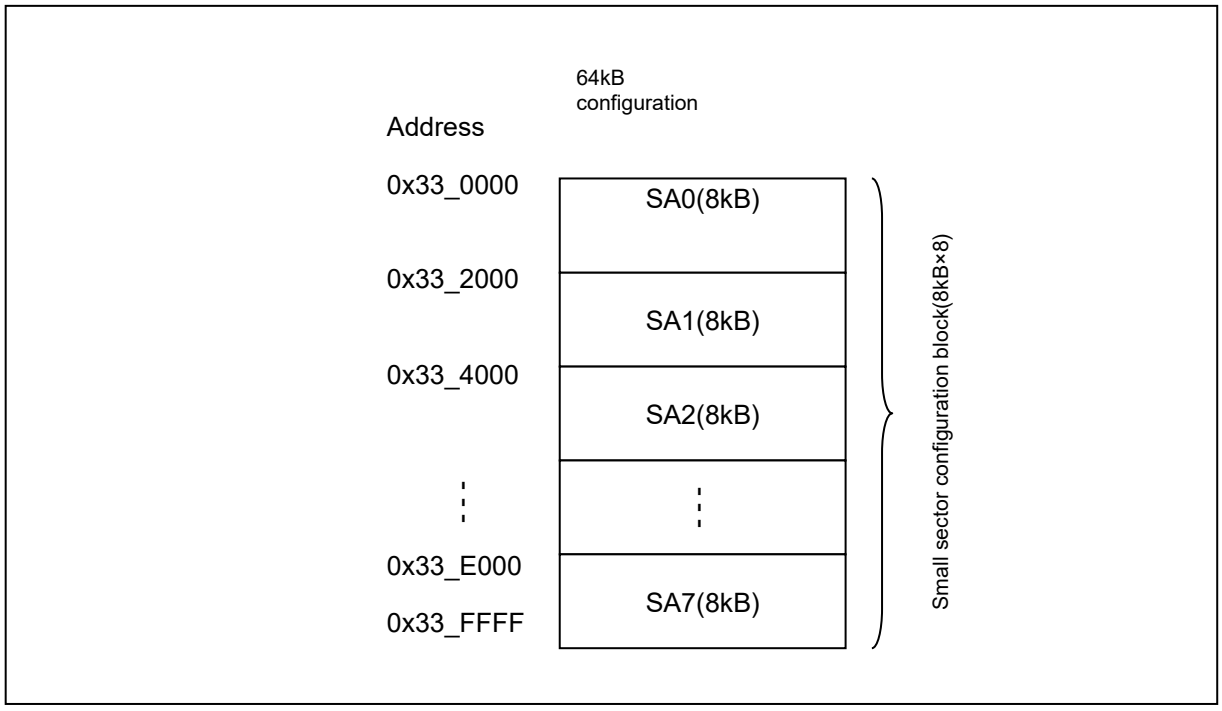
Figure 46-1. Block Diagram (64KB Products)



46.3.2 Sector Configuration Diagram

The sector configuration diagram of the WorkFlash memory is shown below.

Figure 46-2. Sector Configuration Diagram



46.4 Registers

This section explains registers of the WorkFlash memory.

Table 46-1. 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

46.4.1 WorkFlash Control Register: DFCTLR

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)

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	Flash write disabled (Initial value)
1	Flash write enabled

[bit13 to bit0] Reserved

These bits are reserved. The read value is undefined. Writing has no effect on the operation

46.4.2 WorkFlash Status Register: DFSTR

The bit configuration of the WorkFlash status register is shown below.

This register indicates the WorkFlash status.

DFSTR: Address 2303H (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)	Clears 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 Elapsed Flag Bit) "), the flash memory will go into the HANG status. If this bit becomes "1", issue the Reset command (See "46.5.3.1 Command Sequence").

The correct value might not be read out immediately after a command of automatic algorithm has been issued. Therefore, ignore the first 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 flash memory write/erase operation by automatic algorithm is currently running or finished. Flash memory data 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 first read value of this bit after the command issuance.

46.4.3 Flash Interface Control Register: FLIFCTLR

The bit configuration of the flash interface control register is shown below.

This register controls Flash I/F. This register is shared among program flash and WorkFlash.

FLIFCTLR: Address 2308H (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	Wait cycle enabled (Initial value)
1	Wait cycle disabled

[bit3] Reserved

This bit is reserved. The read value is undefined. Writing has no effect on the operation.

[bit2] Reserved

This bit is reserved. When writing, 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 to WorkFlash memory in the CPU mode.

ECCDSBL1	Description
0	ECC function enabled (Initial value)
1	ECC function disabled

[bit0] ECCDSBL0 (ECC Disable0): ECC function disable 0

This bit configures enable/disable for the ECC function when write access and data fetch to program flash memory in the CPU mode.

ECCDSBL0	Description
0	ECC function enabled (Initial value)
1	ECC function disabled

46.5 Operation

The section explains the operation of the WorkFlash memory.

This section explains the method for accessing the flash area.

46.5.1 Access Mode Setting
46.5.2 Writing Flash Memory by CPU
46.5.3 Automatic Algorithm
46.5.4 Reset Command
46.5.5 Write Command
46.5.6 Chip Erase Command
46.5.7 Sector Erase Command
46.5.8 Sector Erase Suspend Command
46.5.9 Security Function
46.5.10 Notes on Using Flash Memory

46.5.1 Access Mode Setting

Access mode setting is shown below.

The flash memory in this series has the following three 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 reading and writing, only Half-word access)
3. Flash memory mode
   (Access to flash memory from external is enabled.)

46.5.1.1 Configuring CPU-ROM Mode Below

Configuring CPU-ROM mode is shown below.

When the FWE bit of the WorkFlash control register (DFCTLR) is "0", it is CPU-ROM mode. When the DFRDY bit of the WorkFlash status register (DFSTR) is "1", read from the flash memory is enabled in this mode. In the mode, write to the flash memory is disabled. After released reset, the mode will be the CPU-ROM mode.

46.5.1.2 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 the flash memory is enabled in this mode.

46.5.2 Writing Flash Memory by CPU

Writing the flash memory by 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 two operations to write one 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 calculating the ECC.

1. Set the flash access size to 16-bit. (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 "46.5.5 Write Command" for details on the write command.

3. Read the hardware sequence flag until the write has finished.

   See "46.5.3.2 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 that was written and check that the correct value is read. Even if the correct value could be read, check the DFSTR:DFECCERR bit to confirm that there was no ECC correction. If ECC correction occurred, write again starting from 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

46.5.3 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.

46.5.3.1 Command Sequence

The command sequence is shown below.

The automatic algorithm starts when half-word (16-bit) data is written to the flash memory once to six times in a row. This is called a command. The command sequences are shown below.

Table 46-2. 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:

- Do as follows to LSB 2-bit of the sector address (SA) to input when the command address and the sector erase command are issued.
  - ☐ When half-word access: 2'b00
  - ☐ When byte access : 2’b01 or 2’b11

  Example 1:

  When byte access and command address = (LSB 2-bit of the standard command address is changed to 2’b01.)

  $AA8_H \rightarrow AA9_H$, $554_H \rightarrow 555_H$, and SA → { SA[31:2] and 2'b01 }

  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector)

  Example 2:

  When byte access and command address = (LSB 2-bit of the standard command address is changed to 2’b11.)

  $AA8_H \rightarrow AAB_H$, $554_H \rightarrow 577_H$, and SA → { SA[31:2] and 2'b11 }

  (SA: When the sector erase command is issued, it is an arbitrary address in the input erase target sector)
- 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.

Reset Command

When the reset command is given to the target flash memory, a sequential input of each command shown in Table 5-1 is cancelled, and another sequential input can be done again from the first 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 four times in a row, the automatic program 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. Once the fourth 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 "46.5.5 Write Command" for details on the actual operation.

Notes:

- When writing in half-word, if the forth command (write data cycle) is written in the odd address, writing is not performed correctly. Always write in even address.
- In the first write command sequence, a single half-word data can be written. If you want to write multiple data, issue one write command sequence for each data.
- While security is ON, writing of flash is limited. See "46.5.9.4 Flash Access Restrictions When Security is ON" for details.

Chip Erase Command

If the chip erase command is sent to the target sector six times in a row, all sectors of the flash memory can be erased in one step. Once the sixth write has finished, the automatic program 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 needs 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 "46.5.6 Chip Erase Command" for details on the actual operation.

Sector Erase Command

If the sector erase command is sent to the target sector six times in a row, the sector of the flash memory is erased. When 40µs elapses (timeout period) after the sixth 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 40µs (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 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 "46.5.7 Sector Erase Command" for details on the actual operation.

Note:

While security is ON, the sector erase procedure of flash is limited. See "46.5.9.4 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 "46.5.8 Sector Erase Suspend Command" for actual operation.

Notes:

- 16.7 µs 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.

46.5.3.2 Automatic Algorithm Execution State

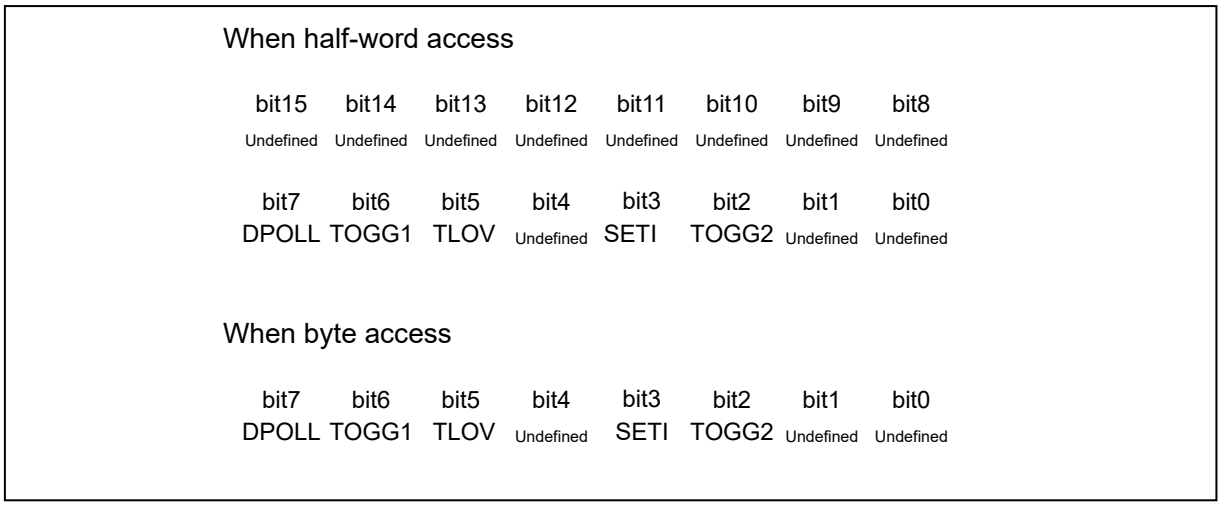
The automatic algorithm execution state is shown below.

Because writing and erasing flash memory is performed by an automatic algorithm, the operating state can be checked by the hardware sequence flag using the DFRDY bit of the WorkFlash status register (DFSTR) to determine whether or not the automatic algorithm is executing.

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 flash memory. Following figure shows the bit configuration of the hardware sequence flag.

Figure 46-3. Bit Configuration of Hardware Sequence



Notes:

- It is impossible to read by word access. Always read using half-word or byte access in CPU programming mode.
- In CPU ROM mode, the hardware sequence flag cannot be read no matter which address is read.

- Each Bit and Flash Memory Status

  Following table shows the correspondence between the status of each bit of the hardware sequence flag and the flash memory status.

Table 46-3. Correspondence between Flags and Flash Memory Status

State		DPOLL	TOGG1	TLOV	SETI	TOGG2
Run	Writing	Inverted data[1]	Toggle	0	0	-
	Sector/Chip erasing	0	Toggle	0	1	-
Time limit exceed	Write command	Inverted data[1]	Toggle	1	0	-
	Sector erase/Chip erase command	0	Toggle	1	1	-
Sector erase suspend	Erase target sector	-	-	-	-	Toggle

[1]: 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 state.

  ☐ 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.

  ☐ 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".

☐ 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".

☐ During sector erase suspend

State of suspend (incomplete end)	"0" is read from the sector erase suspend sector.
Sector erase operation completion	"1" is read 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 status.

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 state.

During write / sector erase / chip erase

The next values are read.

"0"	Within the rated time
"1"	Exceeds 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 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. The next value 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"	When exceeding 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 (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

46.5.4 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 data read 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 set command when reading data.

46.5.5 Write Command

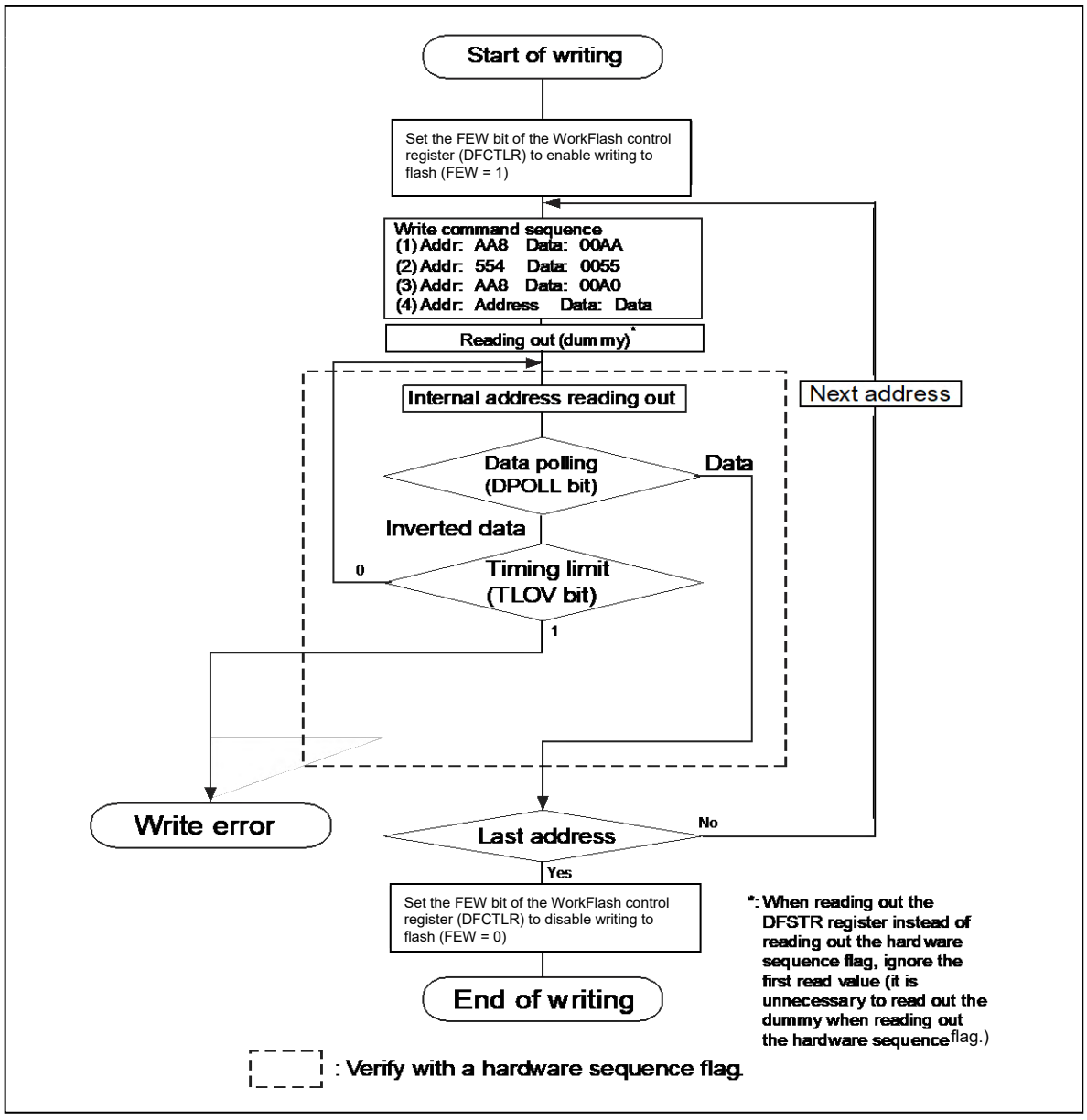
The write command is shown below.

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 bit7's value of the last written data is read out.

The following figure shows an example of write operation to the flash memory.

Figure 46-4. Example of Write Procedure



Notes:

- If write completes, the write address is not accepted because the flash memory returns to the read mode.
- See "46.5.3 Automatic Algorithm" for details on the write 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".
- 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 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 one 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 flash memory are ignored.
- If this series is reset during a write, the data that was written cannot be guaranteed.
- Because this series has the ECC bit added, data always needs to be written as 32-bit by two 16-bit writes. See "46.5.2 Writing Flash Memory" for procedure.

46.5.6 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 sector sequentially, the automatic algorithm starts and all sectors of the flash memory can be erased in one step. See "46.5.3 Automatic Algorithm" for details on the chip erase command.

1. Send chip erase commands sequentially to a sector in the flash 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 × total no. of sectors + chip write time (preprogram)]. When the chip erase operation finishes, the flash 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 flash memory chip before erasing the entire chip, and there is no needs 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 "46.5.9.3 Unlocking Flash Security" for details.

46.5.7 Sector Erase Command

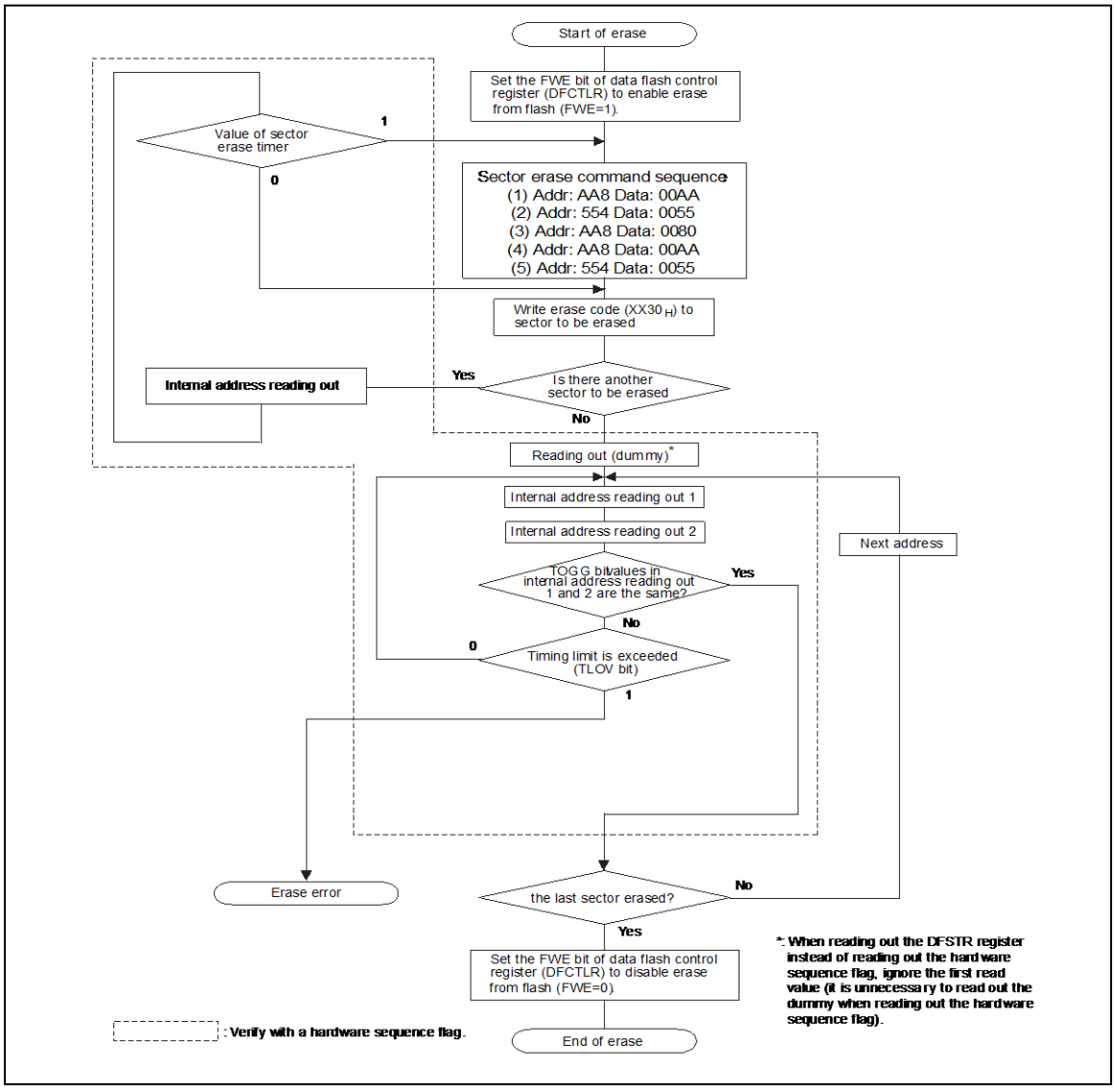
The sector erase command is shown below.

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 (30_H) to the address of the sector to erase within the 40µs (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 46-5. Example of Sector Erase Procedure



Notes:

- The time required to erase the sector is [(sector erase time + sector write time (preprogram)) × no. of sectors].
- When the sector erase operation finishes, the flash memory returns to the read/reset state.
- See "46.5.3 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".
- The moment when the TOGG1 bit of the hardware sequence flag and the 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.
- If commands other than sector erase command are issued while erasing a sector including the time out period, the flash memory becomes read/reset state. In this case, because the flash memory is reset, the sector erase command one or multiple prior to command that is issued is invalid. When sector erase is performed, reissue the sector erase command from scratch.
- 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.

46.5.8 Sector Erase Suspend Command

The sector erase suspend command is explained below

The sector erase can temporarily be stopped in the command time-out or while executing the sector erase.

The reading operation of the memory cell of the sector that is not the erase target becomes possible in the suspend condition of the sector erase. 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 of 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:

16.7 µs 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.

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 46-2 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.

Accepting the erase restart command, the flash memory comes back to the sector erase condition, resuming the erase operation.

46.5.9 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, the operation after an instruction fetches from the external bus, and writes and erases other than chip erase are suppressed. See "46.5.9.4 Flash Access Restrictions When Security is ON" for details of the restrictions.

46.5.9.1 Flash Security On/Off Determination When Reset Released

Flash security on/off determination when reset released is shown below.

For flash interface of this series, 2 bytes in the area of flash security code are read after releasing reset. When the value is 0x0001, security is ON and from then on access limitation to flash memory occurs. When the value is other than that, security becomes OFF.

46.5.9.2 Flash Security Setting Method

The flash security setting method is shown below.

If the input and the release of reset are done after 0x0001 is written in the flash security code area (see " Figure 45-2, Figure 45-3" in "Chapter: Flash Memory"), it becomes security ON. If security is ON once, security does not become OFF without erasing all flash memory area.

46.5.9.3 Unlocking Flash Security

Unlocking flash security is shown below.

The chip erase command can be performed on all flash macros using the following procedure.

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 order of the chip erase to each flash macro is recommended to be executed from the viewpoint of the data protection stored in the flash macro as shown in the above-mentioned.

46.5.9.4 Flash Access Restrictions When Security is ON

Flash access restrictions when security is ON is shown below.

When security is on, the restrictions shown below are created by the start mode.

Table 46-4. Access Restrictions when Security is ON

Operating Mode	Access Restriction
User /External bus	In normal mode (the state where there are no access restrictions due to the following flash security violations), there are no restrictions on access to FLASH memory. If an instruction fetch is performed to the on-chip bus area, a reset request is issued by the flash security violation reset source. Accesses to the flash memory are not accepted thereafter. The flash memory returns to 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 "46.5.9.3 Unlocking Flash Security".

46.5.10 Notes on Using Flash Memory

Notes on using the 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 WorkFlash control register (DFCTLR), do not perform the program in 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 WorkFlash control register (DFCTLR) 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 two 16-bit writes. See "46.5.2 Writing Flash Memory" for procedure.
- Concurrent commands (parallel) to multiple macros must not be issued. After checking the completion of command by the hardware sequence flag or DFRDY bit, command for the next macro must be input.
- 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 stop reading by an outsider, password for on-chip debugger (OCD) activation approval must be configured.
- Changing to the state of the standby is a prohibition during FLASH 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.

47. HS-SPI

This function is not available to use. It is reserved for future expansion.

48. On chip Debugger (OCD)

This chapter explains the on chip debugger (OCD).

48.1 Overview

48.2 Features

48.3 Configuration

48.4 Registers

48.5 Operation

48.1 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.

48.2 Features

This section explains features of the on chip debugger (OCD).

- One-wire debug tool I/F
- 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 (8bytes=4 instructions)
  - CPU register save register (PC/PS)
  - PC monitor function
  - Reset function
    1. Chip reset (INT)
    2. CPU reset (RST)
- 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)

- Trace function
  - Special state trace
  - Branch trace
  - Data trace
  - Trace delay
  - Number of trace frames: 512

48.3 Configuration

This section shows the configuration of the on chip debugger (OCD).

Figure 48-1. Block Diagram of OCDU

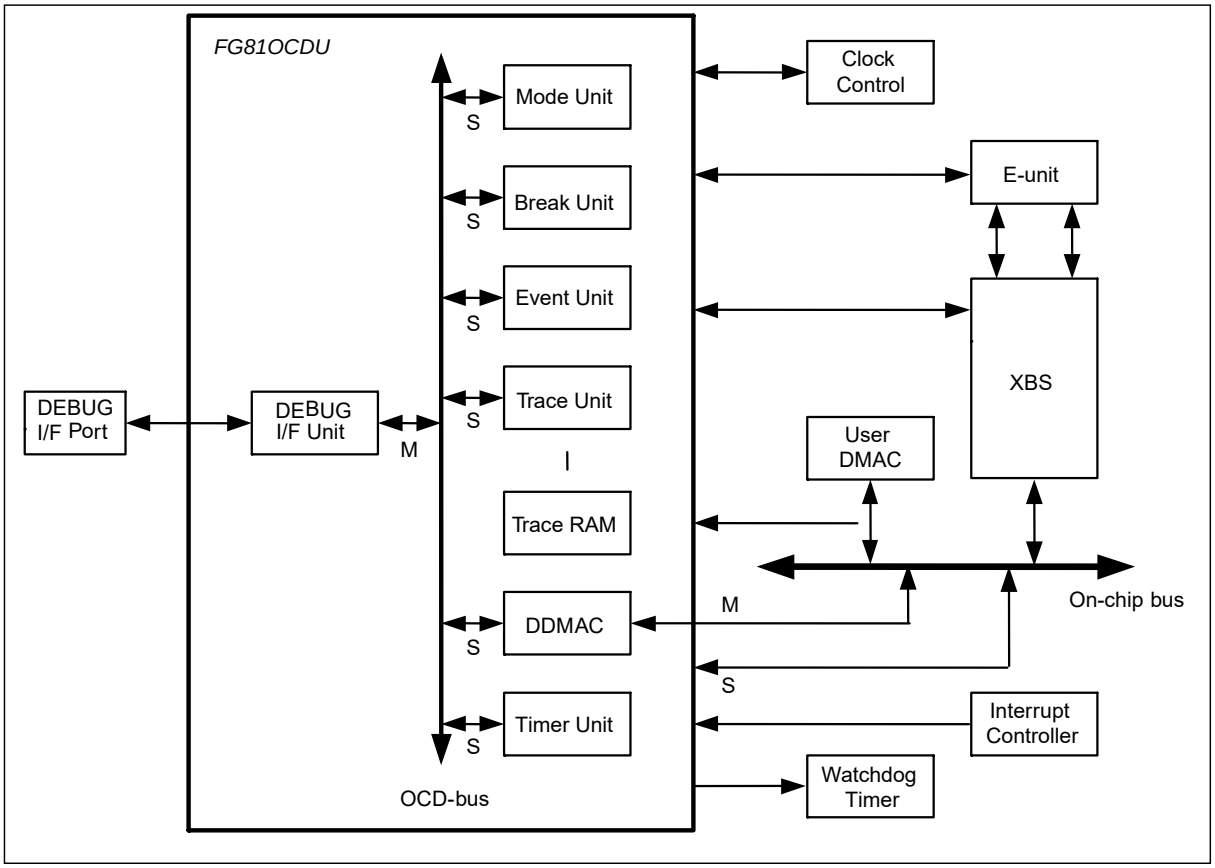
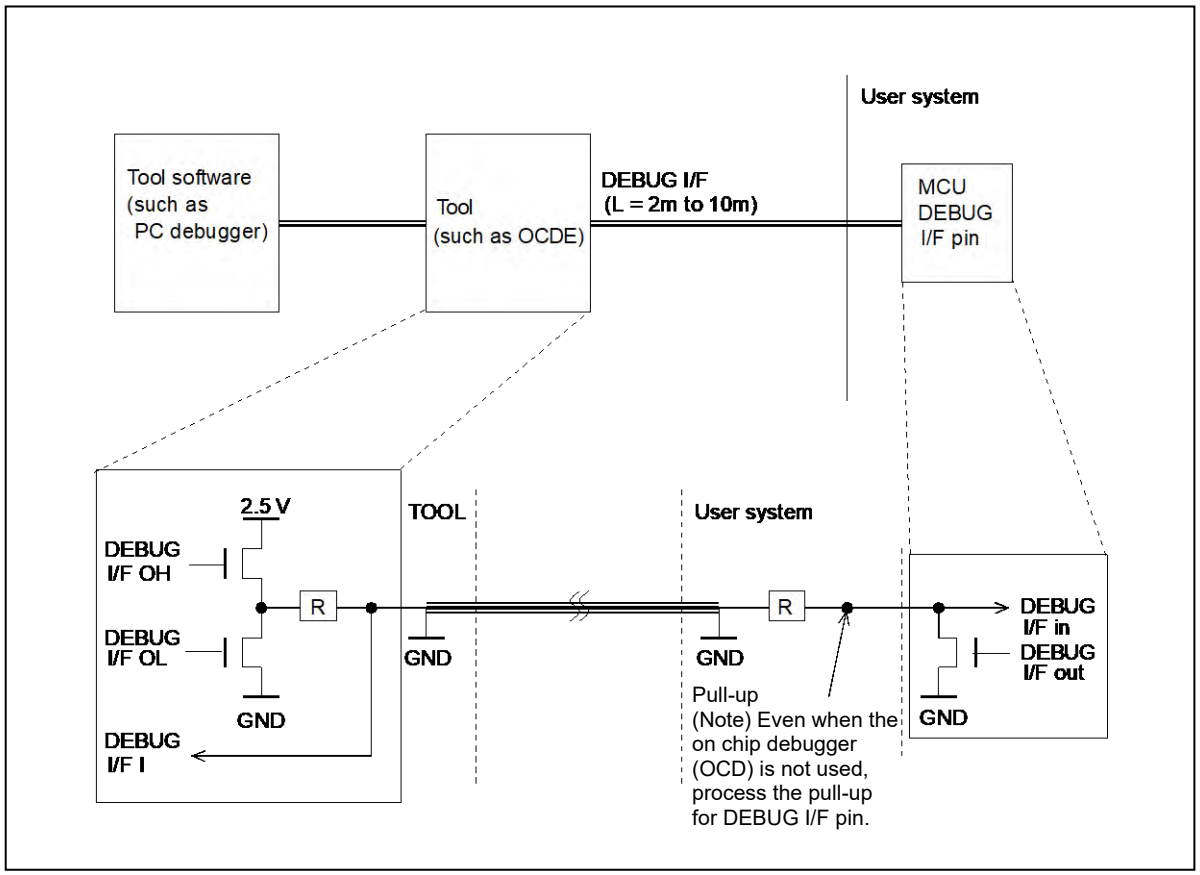


Figure 48-2. OCD Connection Diagram



48.3.1 DEBUG I/F Clock

DEBUG I/F clock is shown.

See "Chapter: Clock" for the clock connection configuration of the DEBUG I/F clock.

48.3.1.1 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.

48.3.1.2 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.

48.4 Registers

This section explains the registers of the on-chip debugger (OCD).

48.4.1 DBG Register

48.4.2 User IO Register

48.4.1 DBG Register

The bit configuration of the DBG register is shown.

Table 48-1. Register Map (DBG Register)

Address	Register				Register Function
	+0	+1	+2	+3	
0xFF00	DSUCR		Reserved		DSU control register

DSU Control Register (DSUCR)

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

48.4.2 User IO Register

The bit configuration of the User IO register is shown.

Table 48-2. Register Map (User IO Register)

Address	Register				Register Function
	+0	+1	+2	+3	
0x0BFC	Reserved		UER		User event register

User Event Register (UER)

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

48.5 Operation

This section explains the operation of the on-chip debugger (OCD).

48.5.1 OCDU Operating Mode
48.5.2 Overview of DEBUG I/F
48.5.3 Specification Restrictions at Connection to OCD Tool of this Series
48.5.4 OCD-DSU ID Code and Mount Type Information on this Series

48.5.1 OCDU Operating Mode

OCDU operating mode is shown.

48.5.1.1 Operating Mode

48.5.1.2 Operating Mode Status Transition

48.5.1.1 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

48.5.1.2 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 occurring 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.

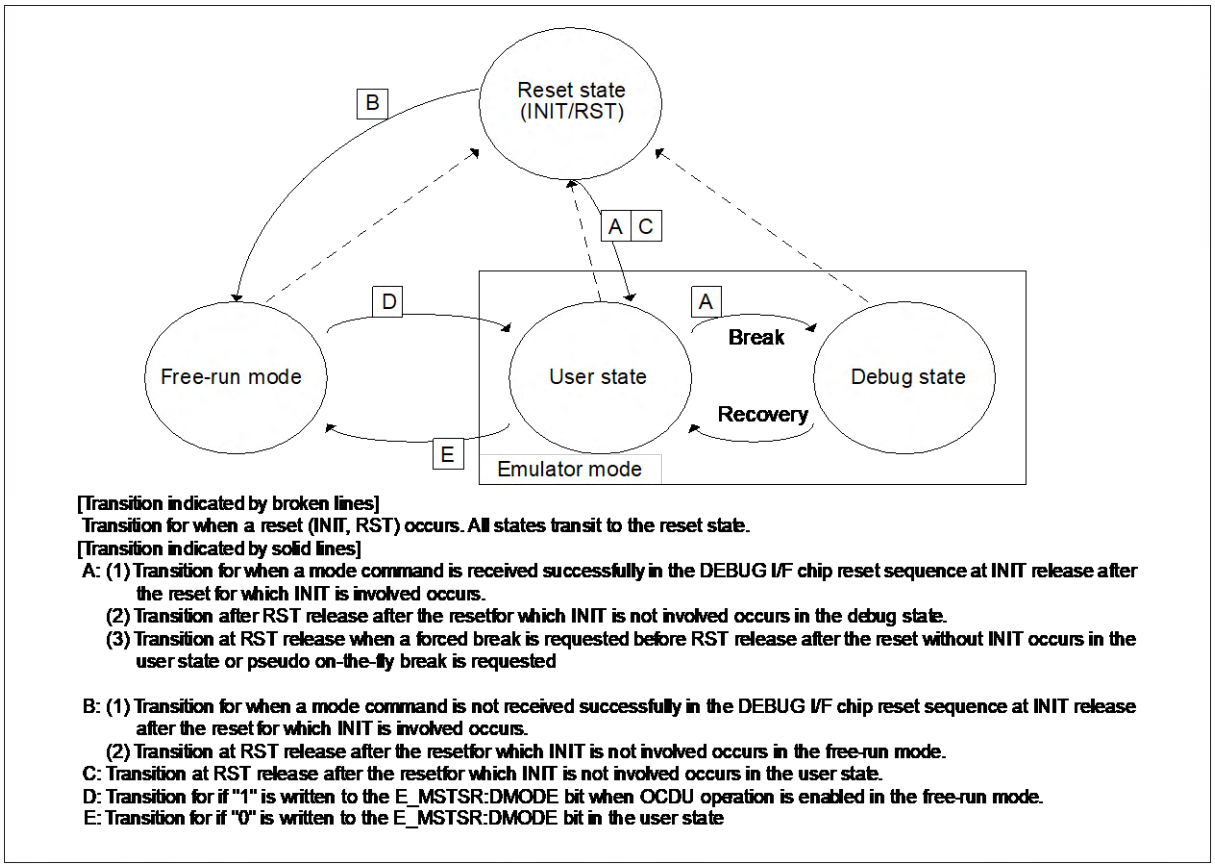
Moreover, transition between the free-run mode and user state of emulator mode is enabled by OCD register control.

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 48-3. 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 DEBUG I/F chip reset sequence at INIT release after the reset for which INIT is involved occurs.
(2) Transition after RST release after the resetfor 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 DEBUG I/F chip reset sequence at INIT release after the reset for which INIT is involved occurs.
(2) Transition at RST release after the resetfor which INIT is not involved occurs in the free-run mode.
C: Transition at RST release after the resetfor 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

48.5.2 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 one 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)

The two-way pin of DEBUG I/F is accomplished by N-ch 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 48-2.

48.5.2.1 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 480 sampling clock cycles of the normal UART is counted. OCDU outputs L to DEBUG I/F and notifies the generation of INIT to the tool in this phase.

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 level sense 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 one cycle or more for inputting H to DEBUG I/F 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.

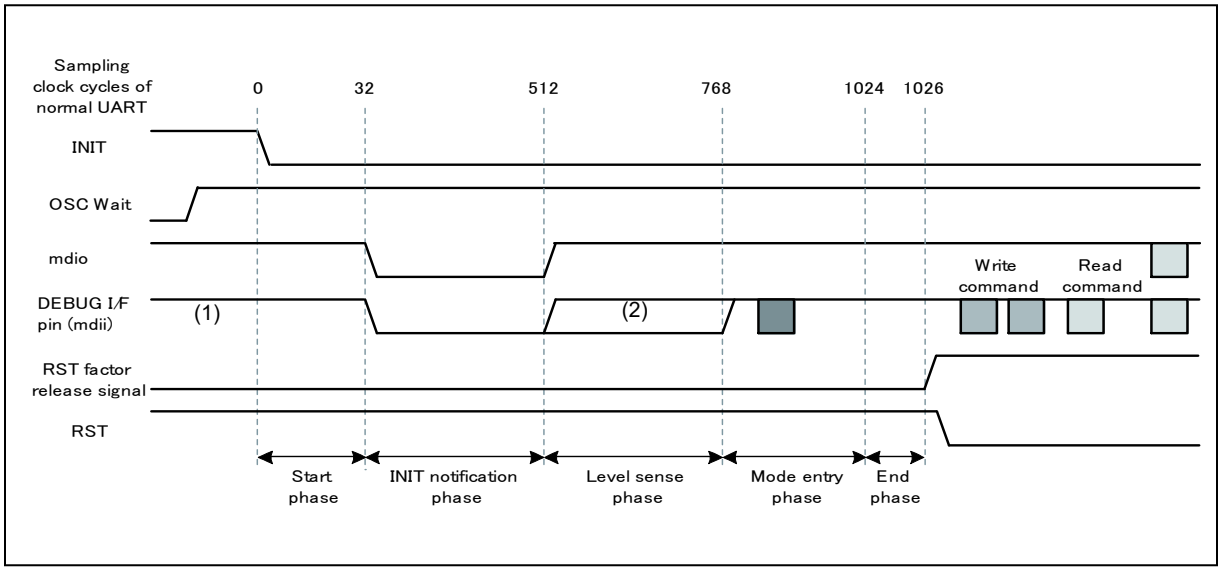
End Phase

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 "7.5.4.3 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	Level Sense Phase	Mode Entry Phase	End Phase
Sampling clock cycles of the normal UART from INIT release	1 - 32	33 - 512	513 - 768	769 - 1024	1025, 1026

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): DEBUG I/F is set to H level by pull-up processing of the tool.

(2): DEBUG I/F becomes the level of pull-up processing on the user system.

48.5.2.2 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 number of specified length and the E_SLPR register.

Security Information

The debug security area is allocated at 30 bytes of built-in flash start address+4 to +33. For OCDU, see this area using the security sequence.

The following security information is available for 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. 14 areas that writes data are provided. 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.

48.5.3 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.

48.5.3.1 Clock Setting

Clock setting is shown.

- PLL oscillation continues while OCD high-speed UART and also phase modulation UART are communicating. Accordingly, change in the settings of the following PLL setting registers will not be effective.
- Write and read, however, are enabled as well as the case that the OCD tool is unconnected.
- PLLCR.ODS
- PLLCR.PMS
- PLLCR.PDS
- CCPSDIVR.PODS
- CCPLLFBR.IDIV

48.5.3.2 Standby Mode

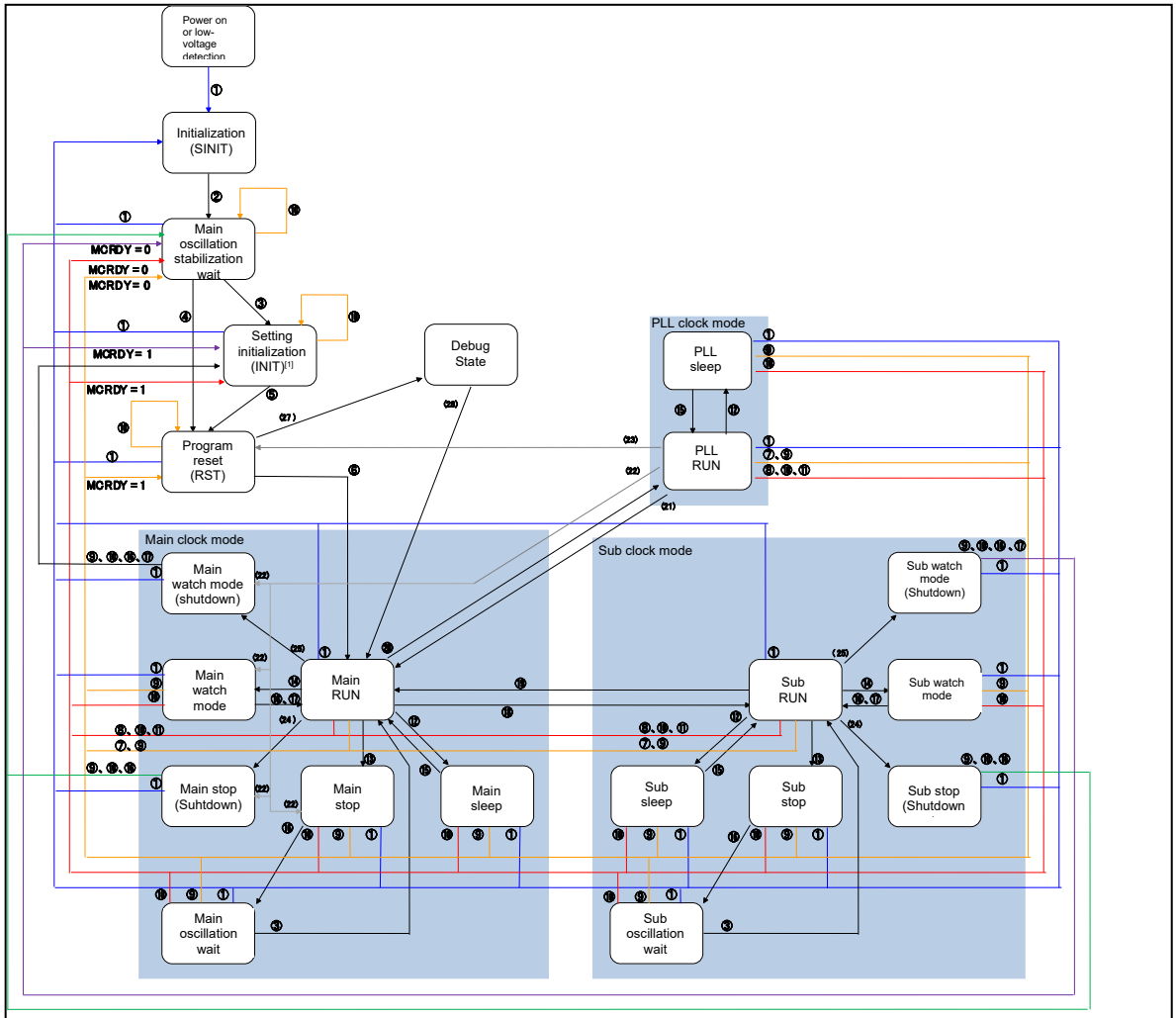
Standby mode is shown.

- Even if the watch mode is entered, PLL oscillation does not stop when OCD high-speed UART and phase modulation UART communication are enabled.
- The main clock oscillation does not stop even if the stop mode is entered. PLL oscillation does not stop when OCD high-speed UART and phase modulation UART communication are enabled. During sub clock oscillation, the sub clock does not stop and oscillation continues.
- The following shows the functions that differ in operation when the OCD tool is not connected, according to the above restrictions:
  - ☐ CAN operation continues in the watch mode and stop mode when PLL is stopped down (CSELR:PCEN=0) or OCD high-speed UART and phase modulation UART communication are enabled. (This operation is performed within the range in which no CPU processing occurs.)
  - ☐ In the stop mode, the LCD controller operates in the same way as in the watch mode.
  - ☐ The real-time clock continues operating even during stop mode.
  - ☐ The counter operation for the RTC/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 and sub timer do not run in the stop mode because they are 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, main clock, and sub clock oscillation continue.

48.5.3.3 Clock Reset State Transitions

Clock reset state transitions is shown.

Figure 48-4. Device State



(1) Power-on reset or internal low-voltage detection or simultaneously assert of external reset and NMI
② Power-on reset release and internal low-voltage detection release and release of simultaneously assert of external reset and NMI
③ End of oscillation stabilization wait
④ End of oscillation stabilization wait (if the reset factor is ⑦ or ⑨)
⑤ INIT release
⑥ RST release
⑦ Software reset
⑧ Software watchdog reset (including irregular) or software reset (irregular)
⑨ External reset input (NMI disabled) or external low-voltage detection
⑩ External reset input (NMI disabled + irregular) or external low-voltage detection (irregular)
⑪ Hardware watchdog reset (including irregular)
⑫ Sleep mode (write instruction)
⑬ Stop mode (write instruction)
⑭ Watch mode (write instruction)
⑮ Interrupt (including ⑯ and ⑰)
⑯ Interrupt (clock not required)/NMI
⑰ Main timer interrupt/Sub timer interrupt/RTC interrupt
⑱ Switch from main to sub (write instruction)
⑲ Switch from sub to main (write instruction)
⑳ Switch from main to PLL (write instruction)
(21) Switch from PLL to main (write instruction)
(22) Illegal standby mode transition
(23) Illegal standby mode transition detection reset
(24) Stop mode and shutdown (write instruction)
(25) Watch mode and shutdown (write instruction)
(27) RST release (When setting initialization (INIT) is accompanied and the mode command is received normally in the chip reset sequence or when there is a forced break request)
(28) Break return
* : (26) is a missing number.

[1]: There is a register not reset when returning from the watch mode (Shutdown) and returning from the stop mode (Shutdown). See " Limitations of power shutdown and normal standby control" of "Chapter: Power Consumption Control" for details.

Note: As single clock products do not have sub clock input, they do not make a transition to the sub clock mode.

48.5.3.4 Summary of Specification Restrictions

Summary of specification restrictions is shown.

- Communication mode[1]: Normal UART

  Note: Debug the shut-down standby mode (stop mode) when the OCD tool is not connected.

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/sub stop mode or main/sub 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 low voltage detection reset (irregular)			
External low voltage detection reset			No voltage step-down circuit switch stabilization wait time[2] **Note:** Only recovery from main/sub stop mode or main/sub watch mode
Illegal standby mode transition detection reset (irregular)			Causes a transition to the emulator mode (debug state) after reset is released
Illegal standby mode transition detection reset		No	
Internal 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	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/sub stop mode or main/sub 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		
Sub RUN/sub sleep mode		
Main/sub stop mode	Yes	Voltage step-down circuit is fixed Main oscillation continues Sub oscillation continues Operation continues (LCD controller, real-time clock, RTC/WDT1 calibration counter operation)
Main/sub watch mode		Voltage step-down circuit is fixed Main oscillation continues **Note:** Sub watch mode

- 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/sub stop mode or main/sub 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 low voltage detection reset (irregular)			
External low voltage detection reset			No voltage step-down circuit switch stabilization wait time[2] **Note:** Only recovery from main/sub stop mode or main/sub watch mode
Illegal standby mode transition detection reset (irregular)	No	Yes	Causes a transition to the emulator mode (debug state) after reset is released
Illegal standby mode transition detection reset		No	
Internal low voltage detection reset	No	Yes	Causes a transition to the emulator mode (debug state) after reset is released
Flash security violation reset (irregular)		Yes	
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/sub stop mode or main/sub 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	
Sub RUN/sub sleep mode	Yes	Main oscillation continues PLL oscillation continues
Main/sub stop mode		Voltage step-down circuit is fixed Main oscillation continues Sub oscillation continues PLL oscillation continues (illegal standby mode transition detection is disabled) Operation continues (CAN, LCD controller, real-time clock, RTC/WDT1 calibration counter operation)
Main/sub watch mode		Voltage step-down circuit is fixed Main oscillation continues **Note:** Sub watch mode Operation continues (CAN)

[1]: For communication mode settings, see "SOFTUNE Workbench Operating Manual".

[2]: Voltage step-down circuit stabilization wait time: about 6 µs

48.5.4 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 48-3. 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	0x06	
DSU version ID	4	E_IDVCR	0x003	0x1	
Device ID	16	E_IDDCR	0x002	*	* CY91F577: 0x0018 CY91F575: 0x0018
Device version ID	4	E_IDVCR	0x003	0x1	

Table 48-4. Mount Type Information of This Product Type

Product Name	Number of Code Events	Number of Data Events	Data Event (Compare)	Sequencer Event	Trace
CY91F577 CY91F575	8	8	○	○	512 frames

A. Appendix

Appendix is shown.

A.1 Memory Map

A.2 I/O Map

A.3 Lists of Interrupt Vector

A.4 Pins Statuses in State of CPU

A.1 Memory Map

Memory map is shown.

Figure A-1. Memory Map CY91F577

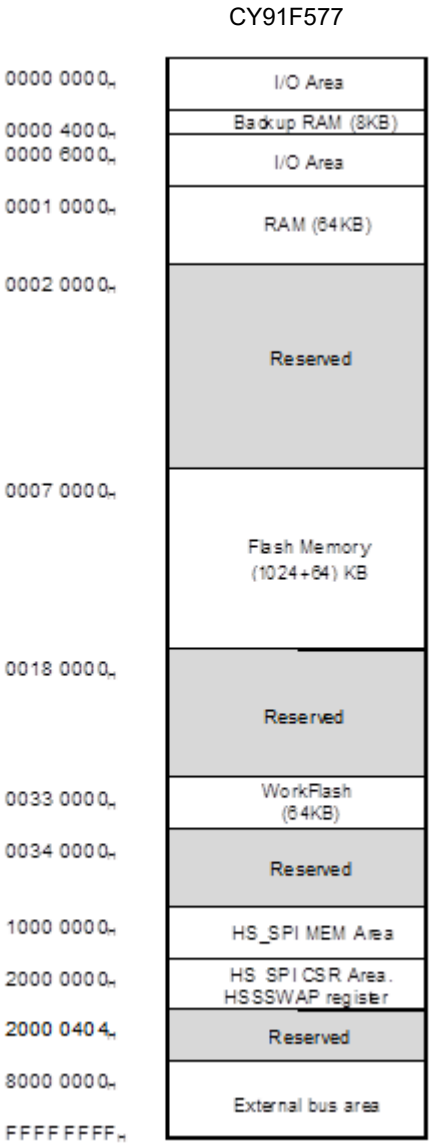
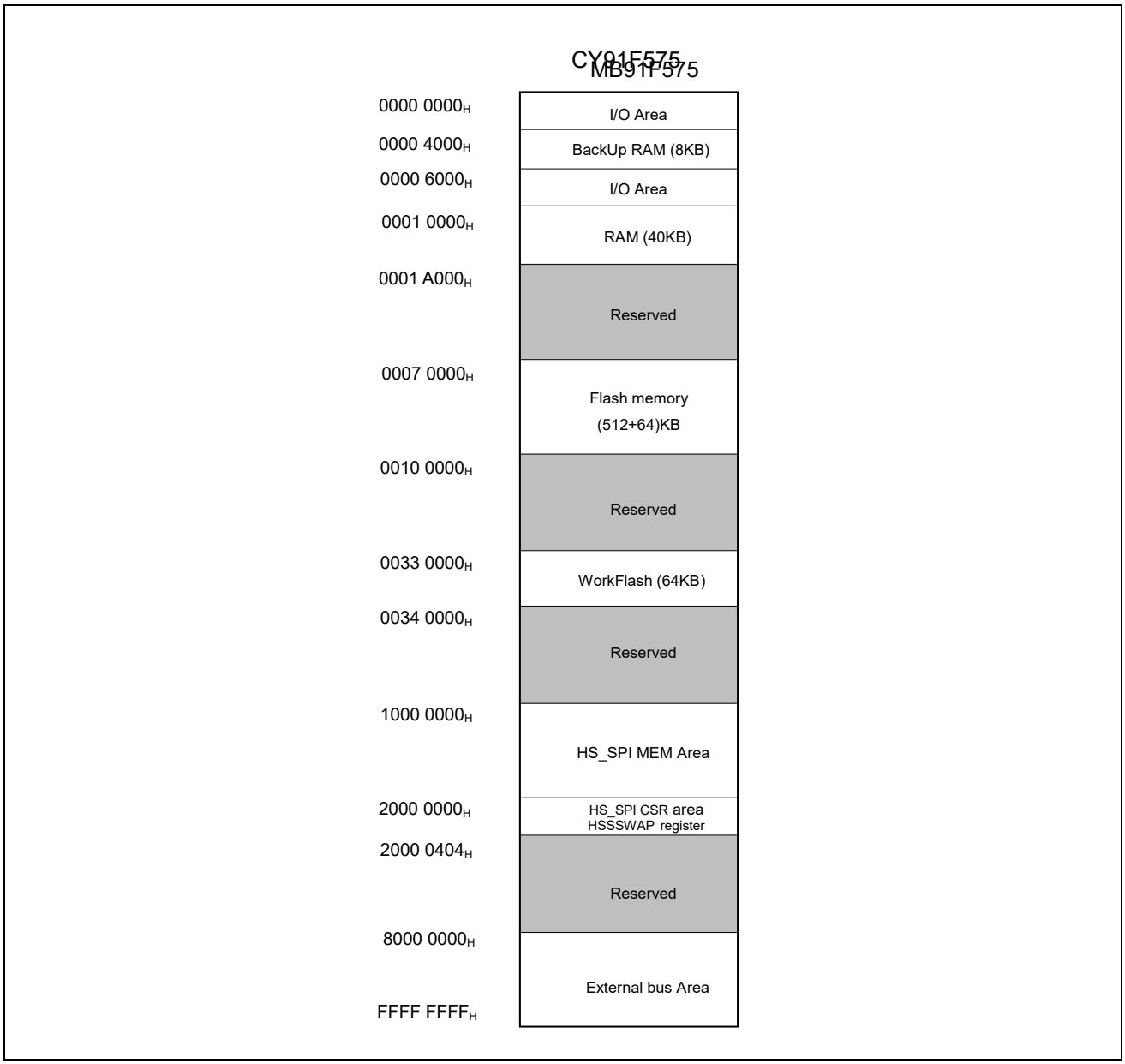


Figure A-2. Memory Map CY91F575

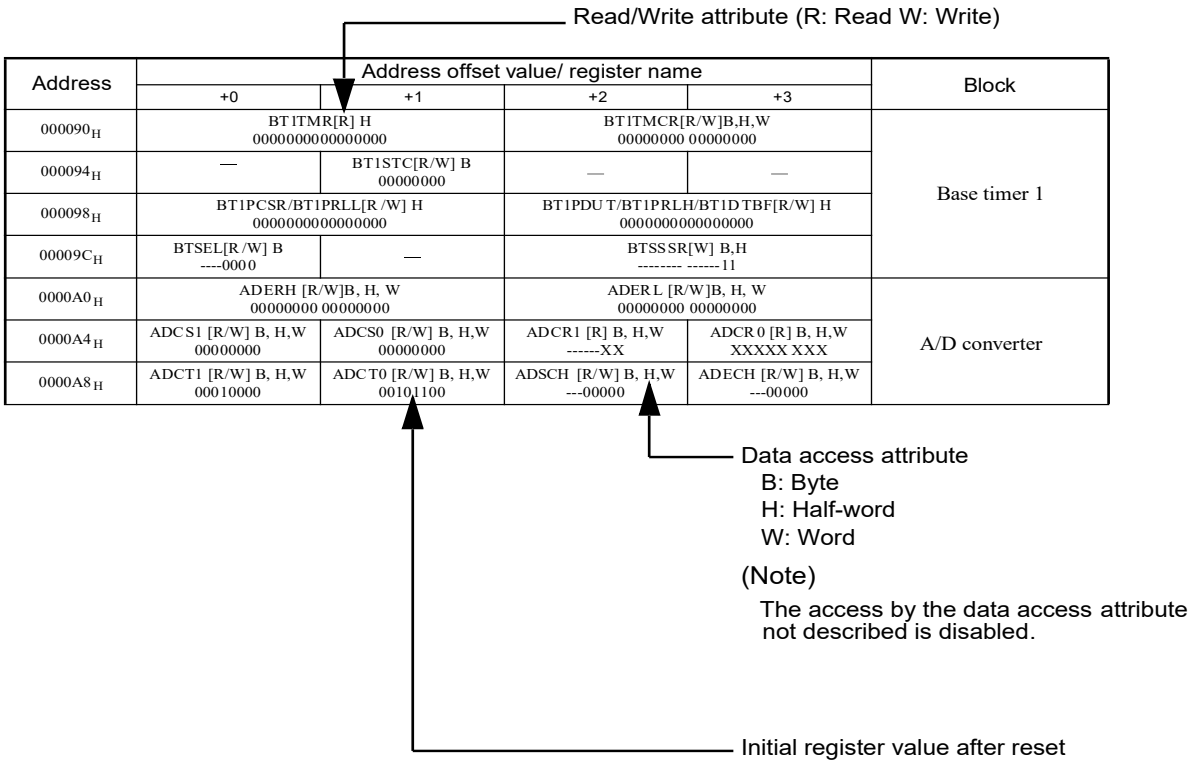


A.2 I/O Map

IO map is shown.

The following I/O map shows the relationship between memory space and registers for peripheral resources.

Figure A-3. 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:

It is prohibited to access addresses not described here.

Table A-1. I/O Map

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000000_H	PDR00[R/W] B,H,W XXXXXXXX	PDR01[R/W] B,H,W XXXXXXXX	PDR02[R/W] B,H,W XXXXXXXX	PDR03[R/W] B,H,W XXXXXXXX	Port data register
000004_H	PDR04[R/W] B,H,W XXXXXXXX	PDR05[R/W] B,H,W XXXXXXXX	PDR06[R/W] B,H,W XXXXXXXX	PDR07[R/W] B,H,W XXXXXXXX	
000008_H	PDR08[R/W] B,H,W XXXXXXXX	PDR09[R/W] B,H,W XXXXXXXX	PDR10[R/W] B,H,W XXXXXXXX	PDR11[R/W] B,H,W XXXXXXXX	
$00000C_H$	PDR12[R/W] B,H,W XXXXXXXX	PDR13[R/W] B,H,W XX-XXXXX	-	-	
000010_H to 000038_H	-	-	-	-	Reserved
$00003C_H$	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	Watchdog timer [S]
000040_H	-	-	-	-	Reserved
000044_H	DICR [R/W] B -------0	-	-	-	Delayed interrupt
000048_H	TMRLRA4 [R/W] H XXXXXXXX XXXXXXXX		TMR4 [R] H XXXXXXXX XXXXXXXX		Reload timer 4
$00004C_H$	TMRLRB4 [R/W] H XXXXXXXX XXXXXXXX		TMCSR4 [R/W] B, H,W 00000000 0-000000		
000050_H	TMRLRA5 [R/W] H XXXXXXXX XXXXXXXX		TMR5 [R] H XXXXXXXX XXXXXXXX		Reload timer 5
000054_H	TMRLRB5 [R/W] H XXXXXXXX XXXXXXXX		TMCSR5 [R/W] B, H,W 00000000 0-000000		
000058_H	TMRLRA6 [R/W] H XXXXXXXX XXXXXXXX		TMR6 [R] H XXXXXXXX XXXXXXXX		Reload timer 6
$00005C_H$	TMRLRB6 [R/W] H XXXXXXXX XXXXXXXX		TMCSR6 [R/W] B, H,W 00000000 0-000000		
000060_H	TMRLRA0 [R/W] H XXXXXXXX XXXXXXXX		TMR0 [R] H XXXXXXXX XXXXXXXX		Reload timer 0
000064_H	TMRLRB0 [R/W] H XXXXXXXX XXXXXXXX		TMCSR0 [R/W] B, H,W 00000000 0-000000		
000068_H to $00007C_H$	-	-	-	-	Reserved

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000080_H	BT0TMR[R] H 00000000 00000000		BT0TMCR[R/W]H -0000000 00000000		Base timer 0
000084_H	-	BT0STC[R/W] B 0000-000	-	-	
000088_H	BT0PCSR/BT0PRLL[R/W] H XXXXXXXX XXXXXXXX		BT0PDUT/BT0PRLH/BT0DTBF[R/W] H XXXXXXXX XXXXXXXX		
$00008C_H$	-	-	-	-	Reserved
000090_H	BT1TMR[R] H 00000000 00000000		BT1TMCR[R/W]H -0000000 00000000		Base timer 1
000094_H	-	BT1STC[R/W] B 0000-000	-	-	
000098_H	BT1PCSR/BT1PRLL[R/W] H 00000000 00000000		BT1PDUT/BT1PRLH/BT1DTBF[R/W] H 00000000 00000000		
$00009C_H$	BTSEL01[R/W] B ----0000	-	BTSSSR[W] B,H -------- ------11		Base timer 0,1
$0000A0_H$	ADERH [R/W]B, H, W 00000000 00000000		ADERL [R/W]B, H, W 00000000 00000000		A/D converter
$0000A4_H$	ADCS1 [R/W] B, H,W 0000000-	ADCS0 [R/W] B, H,W 000-----	ADCR1 [R] B, H,W ------XX	ADCR0 [R] B, H,W XXXXXXXX	
$0000A8_H$	ADCT1 [R/W] B, H,W 00010000	ADCT0 [R/W] B, H,W 00101100	ADSCH [R/W] B, H,W --000000	ADECH [R/W] B, H,W --000000	
$0000AC_H$	-	EADERLL [R/W] B, H,W 00000000	EADCS [R] B, H,W --000000	-	
$0000B0_H$	SCR0/(IBCR0) [R/W] B,H,W 0--00000	SMR0 [R/W] B,H,W 000-0000	SSR0 [R/W] B,H,W 0-000011	ESCR0/(IBSR0) [R/W] B,H,W -0000000	Multi-UART0 [1]: Byte access is permitted only for access to lower 8 bits [2]: Reserved because I^2C mode is not set immediately after reset.
$0000B4_H$	RDR0/(TDR0)[R/W] B,H,W [1] -------0 00000000		BGR0 [R/W] H,W 00000000 00000000		
$0000B8_H$	- / (ISMK0) [R/W] B,H,W -------- [2]	- / (ISBA0) [R/W] B,H,W -------- [2]	-	-	
$0000BC_H$	FCR10 [R/W] B,H,W ---00100	FCR00 [R/W] B,H,W -0000000	FBYTE20 [R/W] B,H,W 00000000	FBYTE10 [R/W] B,H,W 00000000	

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0000C0_H$	SCR1/(IBCR1) [R/W] B,H,W 0--00000	SMR1 [R/W] B,H,W 000-0000	SSR1 [R/W] B,H,W 0-000011	ESCR1/(IBSR1) [R/W] B,H,W -0000000	Multi-UART1 [1]: Byte access is permitted only for access to lower 8 bits [2]: Reserved because I^2C mode is not set immediately after reset.
$0000C4_H$	RDR1/(TDR1)[R/W] B,H,W [1] -------0 00000000		BGR1 [R/W] H,W 00000000 00000000		
$0000C8_H$	- / (ISMK1) [R/W] B,H,W -------- [2]	- / (ISBA1) [R/W] B,H,W -------- [2]	-	-	
$0000CC_H$	FCR11 [R/W] B,H,W ---00100	FCR01 [R/W] B,H,W -0000000	FBYTE21 [R/W] B,H,W 00000000	FBYTE11 [R/W] B,H,W 00000000	
$0000D0_H$	SCR2 [R/W] B, H, W 00000000	SMR2 [R/W] B, H, W 00000000	SSR2 [R/W] B, H, W 00001000	RDR2 /TDR2 [R/W] B, H, W 00000000	LIN-UART2
$0000D4_H$	ESCR2 [R/W] B, H, W 00000X00	ECCR2 [R/W] B, H, W -0000-XX	BGR2 [R/W] B, H, W -0000000 00000000		
$0000D8_H$	SCR3 [R/W] B, H, W 00000000	SMR3 [R/W] B, H, W 00000000	SSR3 [R/W] B, H, W 00001000	RDR3 /TDR3 [R/W] B, H, W 00000000	LIN-UART3
$0000DC_H$	ESCR3 [R/W] B, H, W 00000X00	ECCR3 [R/W] B, H, W -0000-XX	BGR3 [R/W] B, H, W -0000000 00000000		
$0000E0_H$	SCR4 [R/W] B, H, W 00000000	SMR4 [R/W] B, H, W 00000000	SSR4 [R/W] B, H, W 00001000	RDR4 /TDR4 [R/W] B, H, W 00000000	LIN-UART4
$0000E4_H$	ESCR4 [R/W] B, H, W 00000X00	ECCR4 [R/W] B, H, W -0000-XX	BGR4 [R/W] B, H, W -0000000 00000000		
$0000E8_H$	SCR5 [R/W] B, H, W 00000000	SMR5 [R/W] B, H, W 00000000	SSR5 [R/W] B, H, W 00001000	RDR5 /TDR5 [R/W] B, H, W 00000000	LIN-UART5
$0000EC_H$	ESCR5 [R/W] B, H, W 00000X00	ECCR5 [R/W] B, H, W -0000-XX	BGR5 [R/W] B, H, W -0000000 00000000		
$0000F0_H$	SCR6 [R/W] B, H, W 00000000	SMR6 [R/W] B, H, W 00000000	SSR6 [R/W] B, H, W 00001000	RDR6 /TDR6 [R/W] B, H, W 00000000	LIN-UART6
$0000F4_H$	ESCR6 [R/W] B, H, W 00000X00	ECCR6 [R/W] B, H, W -0000-XX	BGR6 [R/W] B, H, W -0000000 00000000		
$0000F8_H$	SCR7 [R/W] B, H, W 00000000	SMR7 [R/W] B, H, W 00000000	SSR7 [R/W] B, H, W 00001000	RDR7 /TDR7 [R/W] B, H, W 00000000	LIN-UART7
$0000FC_H$	ESCR7 [R/W] B, H, W 00000X00	ECCR7 [R/W] B, H, W -0000-XX	BGR7 [R/W] B, H, W -0000000 00000000		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000100_H	TMRLRA1 [R/W] H XXXXXXXX XXXXXXXX		TMR1 [R] H XXXXXXXX XXXXXXXX		Reload timer 1
000104_H	TMRLRB1 [R/W] H XXXXXXXX XXXXXXXX		TMCSR1 [R/W] B, H,W 00000000 0-000000		
000108_H	TMRLRA2 [R/W] H XXXXXXXX XXXXXXXX		TMR2 [R] H XXXXXXXX XXXXXXXX		Reload timer 2
$00010C_H$	TMRLRB2 [R/W] H XXXXXXXX XXXXXXXX		TMCSR2 [R/W] B, H,W 00000000 0-000000		
000110_H	TMRLRA3 [R/W] H XXXXXXXX XXXXXXXX		TMR3 [R] H XXXXXXXX XXXXXXXX		Reload timer 3
000114_H	TMRLRB3 [R/W] H XXXXXXXX XXXXXXXX		TMCSR3 [R/W] B, H,W 00000000 0-000000		
000118_H to $00011C_H$	-	-	-	-	Reserved
000120_H	OCCP6 [R/W] W 00000000 00000000 00000000 00000000				Output compare 6,7
000124_H	OCCP7 [R/W] W 00000000 00000000 00000000 00000000				
000128_H	OCFS67 [R/W] B, H, W ------11	-	OCSH67[R/W] B, H, W ---0--00	OCSL67[R/W] B, H, W 0000--00	
$00012C_H$	OCCP8 [R/W] W 00000000 00000000 00000000 00000000				Output compare 8,9
000130_H	OCCP9 [R/W] W 00000000 00000000 00000000 00000000				
000134_H	OCFS89 [R/W] B, H, W ------11	-	OCSH89[R/W] B, H, W ---0--00	OCSL89[R/W] B, H, W 0000--00	
000138_H	OCCP10 [R/W] W 00000000 00000000 00000000 00000000				Output compare 10,11
$00013C_H$	OCCP11 [R/W] W 00000000 00000000 00000000 00000000				
000140_H	OCFS1011 [R/W] B, H, W ------11	-	OCSH1011[R/W] B, H, W ---0--00	OCSL1011[R/W] B, H, W 0000--00	
000144_H	GCN13 [R/W] H 00110010 00010000		-	GCN23 [R/W] B ----0000	PPG12, 13, 14, 15 control
000148_H	GCN14 [R/W] H 00110010 00010000		-	GCN24 [R/W] B ----0000	PPG16, 17, 18, 19 control
$00014C_H$	GCN15 [R/W] H 00110010 00010000		-	GCN25 [R/W] B ----0000	PPG20, 21, 22, 23 control

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000150_H	PTMR11 [R] H,W 11111111 11111111		PCSR11 [W] H,W XXXXXXXX XXXXXXXX		PPG11
000154_H	PDUT11 [W] H,W XXXXXXXX XXXXXXXX		PCN11 [R/W] B,H,W 0000000- 000000-0		
000158_H	PTMR12 [R] H,W 11111111 11111111		PCSR12 [W] H,W XXXXXXXX XXXXXXXX		PPG12
$00015C_H$	PDUT12 [W] H,W XXXXXXXX XXXXXXXX		PCN12 [R/W] B,H,W 0000000- 000000-0		
000160_H	PTMR13 [R] H,W 11111111 11111111		PCSR13 [W] H,W XXXXXXXX XXXXXXXX		PPG13
000164_H	PDUT13 [W] H,W XXXXXXXX XXXXXXXX		PCN13 [R/W] B,H,W 0000000- 000000-0		
000168_H	PTMR14 [R] H,W 11111111 11111111		PCSR14 [W] H,W XXXXXXXX XXXXXXXX		PPG14
$00016C_H$	PDUT14 [W] H,W XXXXXXXX XXXXXXXX		PCN14 [R/W] B,H,W 0000000- 000000-0		
000170_H	PTMR15 [R] H,W 11111111 11111111		PCSR15 [W] H,W XXXXXXXX XXXXXXXX		PPG15
000174_H	PDUT15 [W] H,W XXXXXXXX XXXXXXXX		PCN15 [R/W] B,H,W 0000000- 000000-0		
000178_H	PTMR16 [R] H,W 11111111 11111111		PCSR16 [W] H,W XXXXXXXX XXXXXXXX		PPG16
$00017C_H$	PDUT16 [W] H,W XXXXXXXX XXXXXXXX		PCN16 [R/W] B,H,W 0000000- 000000-0		
000180_H	PTMR17 [R] H,W 11111111 11111111		PCSR17 [W] H,W XXXXXXXX XXXXXXXX		PPG17
000184_H	PDUT17 [W] H,W XXXXXXXX XXXXXXXX		PCN17 [R/W] B,H,W 0000000- 000000-0		
000188_H	PTMR18 [R] H,W 11111111 11111111		PCSR18 [W] H,W XXXXXXXX XXXXXXXX		PPG18
$00018C_H$	PDUT18 [W] H,W XXXXXXXX XXXXXXXX		PCN18 [R/W] B,H,W 0000000- 000000-0		
000190_H	PTMR19 [R] H,W 11111111 11111111		PCSR19 [W] H,W XXXXXXXX XXXXXXXX		PPG19
000194_H	PDUT19 [W] H,W XXXXXXXX XXXXXXXX		PCN19 [R/W] B,H,W 0000000- 000000-0		
000198_H	PTMR20 [R] H,W 11111111 11111111		PCSR20 [W] H,W XXXXXXXX XXXXXXXX		PPG20
$00019C_H$	PDUT20 [W] H,W XXXXXXXX XXXXXXXX		PCN20 [R/W] B,H,W 0000000- 000000-0		
$0001A0_H$	PTMR21 [R] H,W 11111111 11111111		PCSR21 [W] H,W XXXXXXXX XXXXXXXX		PPG21
$0001A4_H$	PDUT21 [W] H,W XXXXXXXX XXXXXXXX		PCN21 [R/W] B,H,W 0000000- 000000-0		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0001A8_H$	PTMR22 [R] H,W 11111111 11111111		PCSR22 [W] H,W XXXXXXXX XXXXXXXX		PPG22
$0001AC_H$	PDUT22 [W] H,W XXXXXXXX XXXXXXXX		PCN22 [R/W] B,H,W 0000000- 000000-0		
$0001B0_H$	PTMR23 [R] H,W 11111111 11111111		PCSR23 [W] H,W XXXXXXXX XXXXXXXX		PPG23
$0001B4_H$	PDUT23 [W] H,W XXXXXXXX XXXXXXXX		PCN23 [R/W] B,H,W 0000000- 000000-0		
$0001B8_H$ to $0001FC_H$	-	-	-	-	Reserved
000200_H	PWC20 [R/W] H,W ------XX XXXXXXXX		PWC10 [R/W] H,W ------XX XXXXXXXX		Stepping motor controller 0
000204_H	-	PWC0 [R/W] B -00000--	PWS20 [R/W] B,H,W -0000000	PWS10 [R/W] B,H,W --000000	
000208_H	PWC21 [R/W] H,W ------XX XXXXXXXX		PWC11 [R/W] H,W ------XX XXXXXXXX		Stepping motor controller 1
$00020C_H$	-	PWC1 [R/W] B -00000--	PWS21 [R/W] B,H,W -0000000	PWS11 [R/W] B,H,W --000000	
000210_H	PWC22 [R/W] H,W ------XX XXXXXXXX		PWC12 [R/W] H,W ------XX XXXXXXXX		Stepping motor controller 2
000214_H	-	PWC2 [R/W] B -00000--	PWS22 [R/W] B,H,W -0000000	PWS12 [R/W] B,H,W --000000	
000218_H	PWC23 [R/W] H,W ------XX XXXXXXXX		PWC13 [R/W] H,W ------XX XXXXXXXX		Stepping motor controller3
$00021C_H$	-	PWC3 [R/W] B -00000--	PWS23 [R/W] B,H,W -0000000	PWS13 [R/W] B,H,W --000000	
000220_H	PWC24 [R/W] H,W ------XX XXXXXXXX		PWC14 [R/W] H,W ------XX XXXXXXXX		Stepping motor controller 4
000224_H	-	PWC4 [R/W] B -00000--	PWS24 [R/W] B,H,W -0000000	PWS14 [R/W] B,H,W --000000	
000228_H	PWC25 [R/W] H,W ------XX XXXXXXXX		PWC15 [R/W] H,W ------XX XXXXXXXX		Stepping motor controller 5
$00022C_H$	-	PWC5 [R/W] B -00000--	PWS25 [R/W] B,H,W -0000000	PWS15 [R/W] B,H,W --000000	
000230_H to 000238_H	-	-	-	-	Reserved
$00023C_H$	DACR0 [R/W] B,H,W -------0	DADR0 [R/W] B,H,W XXXXXXXX	DACR1 [R/W] B,H,W -------0	DADR1 [R/W] B,H,W XXXXXXXX	DA converter

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000240_H	CPCLR0 [R/W] W 11111111 11111111 11111111 11111111				Free-run timer 0
000244_H	TCDT0 [R/W] W 00000000 00000000 00000000 00000000				
000248_H	TCCSH0 [R/W]B,H,W 0-----00	TCCSL0 [R/W]B,H,W -1-00000	-		
$00024C_H$	CPCLR1 [R/W] W 11111111 11111111 11111111 11111111				Free-run timer 1
000250_H	TCDT1 [R/W] W 00000000 00000000 00000000 00000000				
000254_H	TCCSH1 [R/W]B,H,W 0-----00	TCCSL1 [R/W]B,H,W -1-00000	-		
000258_H	-	-	-	-	Reserved
$00025C_H$	GCN10 [R/W] H 00110010 00010000		-	GCN20 [R/W] B ----0000	PPG0, 1, 2, 3 control
000260_H	GCN11 [R/W] H 00110010 00010000		-	GCN21 [R/W] B ----0000	PPG4, 5, 6, 7 control
000264_H	GCN12 [R/W] H 00110010 00010000		-	GCN22 [R/W] B ----0000	PPG8, 9, 10, 11 control
000268_H	-	-	-	PPGDIV [R/W] B ------00	PPG0
$00026C_H$	PTMR0 [R] H,W 11111111 11111111		PCSR0 [W] H,W XXXXXXXX XXXXXXXX		
000270_H	PDUT0 [W] H,W XXXXXXXX XXXXXXXX		PCN0 [R/W] B, H,W 0000000- 000000-0		
000274_H	PTMR1 [R] H,W 11111111 11111111		PCSR1 [W] H,W XXXXXXXX XXXXXXXX		PPG1
000278_H	PDUT1 [W] H,W XXXXXXXX XXXXXXXX		PCN1 [R/W] B,H,W 0000000- 000000-0		
$00027C_H$	PTMR2 [R] H,W 11111111 11111111		PCSR2 [W] H,W XXXXXXXX XXXXXXXX		PPG2
000280_H	PDUT2 [W] H,W XXXXXXXX XXXXXXXX		PCN2 [R/W] B,H,W 0000000- 000000-0		
000284_H	PTMR3 [R] H,W 11111111 11111111		PCSR3 [W] H,W XXXXXXXX XXXXXXXX		PPG3
000288_H	PDUT3 [W] H,W XXXXXXXX XXXXXXXX		PCN3 [R/W] B,H,W 0000000- 000000-0		
$00028C_H$	PTMR4 [R] H,W 11111111 11111111		PCSR4 [W] H,W XXXXXXXX XXXXXXXX		PPG4
000290_H	PDUT4 [W] H,W XXXXXXXX XXXXXXXX		PCN4 [R/W] B,H,W 0000000- 000000-0		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000294_H	PTMR5 [R] H,W 11111111 11111111		PCSR5 [W] H,W XXXXXXXX XXXXXXXX		PPG5
000298_H	PDUT5 [W] H,W XXXXXXXX XXXXXXXX		PCN5 [R/W] B,H,W 0000000- 000000-0		
$00029C_H$	PTMR6 [R] H,W 11111111 11111111		PCSR6 [W] H,W XXXXXXXX XXXXXXXX		PPG6
$0002A0_H$	PDUT6 [W] H,W XXXXXXXX XXXXXXXX		PCN6 [R/W] B,H,W 0000000- 000000-0		
$0002A4_H$	PTMR7 [R] H,W 11111111 11111111		PCSR7 [W] H,W XXXXXXXX XXXXXXXX		PPG7
$0002A8_H$	PDUT7 [W] H,W XXXXXXXX XXXXXXXX		PCN7 [R/W] B,H,W 0000000- 000000-0		
$0002AC_H$	PTMR8 [R] H,W 11111111 11111111		PCSR8 [W] H,W XXXXXXXX XXXXXXXX		PPG8
$0002B0_H$	PDUT8 [W] H,W XXXXXXXX XXXXXXXX		PCN8 [R/W] B,H,W 0000000- 000000-0		
$0002B4_H$	PTMR9 [R] H,W 11111111 11111111		PCSR9 [W] H,W XXXXXXXX XXXXXXXX		PPG9
$0002B8_H$	PDUT9 [W] H,W XXXXXXXX XXXXXXXX		PCN9 [R/W] B,H,W 0000000- 000000-0		
$0002BC_H$	PTMR10 [R] H,W 11111111 11111111		PCSR10 [W] H,W XXXXXXXX XXXXXXXX		PPG10
$0002C0_H$	PDUT10 [W] H,W XXXXXXXX XXXXXXXX		PCN10 [R/W] B,H,W 0000000- 000000-0		
$0002C4_H$	IPCP0 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input capture 0,1
$0002C8_H$	IPCP1 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0002CC_H$	ICFS01 [R/W] B, H, W ------00	-	LSYNS0 [R/W] B,H,W --000000	ICS01 [R/W] B, H, W 00000000	
$0002D0_H$	IPCP2 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input capture 2,3
$0002D4_H$	IPCP3 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0002D8_H$	ICFS23 [R/W] B, H, W ------00	-	-	ICS23 [R/W] B, H, W 00000000	
$0002DC_H$	IPCP4 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input capture 4,5
$0002E0_H$	IPCP5 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0002E4_H$	ICFS45 [R/W] B, H, W ------00	-	-	ICS45 [R/W] B, H, W 00000000	

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0002E8_H$	OCCP0 [R/W] W 00000000 00000000 00000000 00000000				Output compare 0,1
$0002EC_H$	OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
$0002F0_H$	OCFS01 [R/W] B, H, W ------11	-	OCSH01[R/W] B, H, W ---0--00	OCSL01[R/W] B, H, W 0000--00	
$0002F4_H$	OCCP2 [R/W] W 00000000 00000000 00000000 00000000				Output compare 2,3
$0002F8_H$	OCCP3 [R/W] W 00000000 00000000 00000000 00000000				
$0002FC_H$	OCFS23 [R/W] B, H, W ------11	-	OCSH23[R/W] B, H, W ---0--00	OCSL23[R/W] B, H, W 0000--00	
000300_H to $00030C_H$	-	-	-	-	Reserved
000310_H	-	-	MPUCR [R/W] H 000000-0 ----0100		MPU [S] (Only the CPU can access this area)
000314_H	-	-	-	-	
000318_H	-				
$00031C_H$	-	-	-		
000320_H	DPVAR [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000324_H	-	-	DPVSR [R/W] H -------- 00000--0		
000328_H	DEAR [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$00032C_H$	-	-	DESR [R/W] H -------- 00000--0		
000330_H	PABR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000334_H	-	-	PACR0 [R/W] H 000000-0 00000--0		
000338_H	PABR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00033C_H$	-	-	PACR1 [R/W] H 000000-0 00000--0		
000340_H	PABR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000344_H	-	-	PACR2 [R/W] H 000000-0 00000--0		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000348_H	PABR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				MPU [S] (Only the CPU can access this area)
$00034C_H$	-	-	PACR3 [R/W] H 000000-0 00000--0		
000350_H	PABR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000354_H	-	-	PACR4 [R/W] H 000000-0 00000--0		
000358_H	PABR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00035C_H$	-	-	PACR5 [R/W] H 000000-0 00000--0		
000360_H	PABR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000364_H	-	-	PACR6 [R/W] H 000000-0 00000--0		
000368_H	PABR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00036C_H$	-	-	PACR7 [R/W] H 000000-0 00000--0		
000370_H	PABR8 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				MPU [S] (Only product supporting MPU 12 channels or 16 channels) (Only the CPU can access this area)
000374_H	-	-	PACR8 [R/W] H 000000-0 00000--0		
000378_H	PABR9[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00037C_H$	-	-	PACR9 [R/W] H 000000-0 00000--0		
000380_H	PABR10 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
000384_H	-	-	PACR10 [R/W] H 000000-0 00000--0		
000388_H	PABR11 [R/W] ,W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00038C_H$	-	-	PACR11 [R/W] H 000000-0 00000--0		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000390_H	PABR12 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				MPU [S] (Only product supporting MPU 16 channels) (Only the CPU can access this area)
000394_H	-	-	PACR12 [R/W] H 000000-0 00000--0		
000398_H	PABR13 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$00039C_H$	-	-	PACR13 [R/W] H 000000-0 00000--0		
$0003A0_H$	PABR14 [R/W]W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$0003A4_H$	-	-	PACR14 [R/W] H 000000-0 00000--0		MPU [S] (Only product supporting MPU 16 channels) (Only the CPU can access this area)
$0003A8_H$	PABR15 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXX0000				
$0003AC_H$	-	-	PACR15 [R/W] H 000000-0 00000--0		
$0003B0_H$ to $0003FC_H$	-	-	-	-	Reserved [S]
000400_H	ICSEL0[R/W] B, H, W -----000	ICSEL1[R/W] B, H, W -----000	ICSEL2[R/W] B, H, W -------0	ICSEL3[R/W] B, H, W -------0	Generation and clear of DMA transfer request
000404_H	ICSEL4[R/W] B, H, W -------0	ICSEL5[R/W] B, H, W -------0	ICSEL6[R/W] B, H, W -----000	ICSEL7[R/W] B, H, W -----000	
000408_H	ICSEL8[R/W] B, H, W ------00	ICSEL9[R/W] B, H, W ------00	ICSEL10 [R/W]B, H, W ------00	ICSEL11[R/W] B, H, W ------00	
$00040C_H$	ICSEL12[R/W] B, H, W ------00	ICSEL13[R/W] B, H, W -------0	ICSEL14 [R/W]B, H, W -------0	ICSEL15[R/W] B, H, W -------0	
000410_H	ICSEL16[R/W] B, H, W -------0	ICSEL17[R/W] B, H, W -------0	ICSEL18 [R/W]B, H, W -------0	ICSEL19[R/W] B, H, W -----000	
000414_H	ICSEL20[R/W] B, H, W -----000	ICSEL21[R/W] B, H, W ------00	ICSEL22 [R/W]B, H, W ------00	-	

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000418_H	IRPR0H[R] B, H, W 00------	IRPR0L[R] B, H, W 00------	IRPR1H[R] B, H, W 00------	IRPR1L[R] B, H, W 00------	Interrupt request batch read register
$00041C_H$	IRPR2H[R] B, H, W 00------	IRPR2L[R] B, H, W 00------	IRPR3H[R] B, H, W 000000--	IRPR3L[R] B, H, W 000000--	
000420_H	IRPR4H[R] B, H, W 0000----	IRPR4L[R] B, H, W 0000----	IRPR5H[R] B, H, W 0000----	IRPR5L[R] B, H, W 000-----	
000424_H	IRPR6H[R] B, H, W --000---	IRPR6L[R] B, H, W 00000---	IRPR7H[R] B, H, W -0000---	IRPR7L[R] B, H, W ------00	
000428_H	IRPR8H[R] B, H, W 000-----	IRPR8L[R] B, H, W 000-----	IRPR9H[R] B, H, W 00------	IRPR9L[R] B, H, W 00------	
$00042C_H$	IRPR10H[R] B, H, W 00------	IRPR10L[R] B, H, W 00------	IRPR11H[R] B, H, W 00------	IRPR11L[R] B, H, W 00------	
000430_H	IRPR12H[R] B, H, W 000000--	IRPR12L[R] B, H, W 000000--	IRPR13H[R] B, H, W 000-----	IRPR13L[R] B, H, W 00000---	
000434_H	IRPR14H[R] B, H, W 00000000	IRPR14L[R] B, H, W 00000000	IRPR15H[R] B, H, W 000-----	-	
000438_H to $00043C_H$	-	-	-	-	Reserved
000440_H	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]
000444_H	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	
000448_H	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	
$00044C_H$	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	
000450_H	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	
000454_H	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	
000458_H	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	
$00045C_H$	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	
000460_H	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	

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000464_H	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	Interrupt controller [S]
000468_H	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	
$00046C_H$	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	
000470_H to $00047C_H$	-	-	-	-	Reserved [S]
000480_H	RSTRR [R] B,H,W XXXX--XX	RSTCR [R/W] B,H,W 111----0	STBCR [R/W] B,H,W[3] 000---11	-	Reset control [S] Power consumption control [S] [3]: Writing to STBCR by DMA is not permitted
000484_H	-	-	-	-	Reserved [S]
000488_H	DIVR0 [R/W] B,H,W 000-----	DIVR1 [R/W] B,H,W 0001----	DIVR2 [R/W] B,H,W 0011----	-	Clock control [S]
$00048C_H$	-	-	-	-	Reserved [S]
000490_H	IORR0[R/W] B, H, W -0000000	IORR1[R/W] B, H, W -0000000	IORR2[R/W] B, H, W -0000000	IORR3[R/W] B, H, W -0000000	DMA transfer request from a peripheral [S]
000494_H	IORR4[R/W] B, H, W -0000000	IORR5[R/W] B, H, W -0000000	IORR6[R/W] B, H, W -0000000	IORR7[R/W] B, H, W -0000000	
000498_H	IORR8[R/W] B, H, W -0000000	IORR9[R/W] B, H, W -0000000	IORR10[R/W] B, H, W -0000000	IORR11[R/W] B, H, W -0000000	
$00049C_H$	IORR12[R/W] B, H, W -0000000	IORR13[R/W] B, H, W -0000000	IORR14[R/W] B, H, W -0000000	IORR15[R/W] B, H, W -0000000	
$0004A0_H$	-	-	-	-	Reserved
$0004A4_H$	CANPRE [R/W] B,H,W ----0000	-	-	-	CAN prescaler
$0004A8_H$ to $0004B4_H$	-	-	-	-	Reserved

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0004B8_H$	CUCR0 [R/W] B,H,W -------- ---0--00		CUTD0 [R/W] B,H,W 10000000 00000000		RTC/WDT1 calibration (Calibration)
$0004BC_H$	CUTR0 [R] B,H,W -------- 00000000 00000000 00000000				
$0004C0_H$	-	-	-	-	
$0004C4_H$	CUCR1 [R/W] B,H,W -------- ---0--00		CUTD1[R/W] B,H,W 11000011 01010000		
$0004C8_H$	CUTR1 [R] B,H,W -------- 00000000 00000000 00000000				
$0004CC_H$	CRTR [R/W] B,H,W 01111111	-	-	-	
$0004D0_H$ to $0004DC_H$	-	-	-	-	Reserved
$0004E0_H$	SCR8/(IBCR8) [R/W] B,H,W 0--00000	SMR8 [R/W] B,H,W 000-0000	SSR8 [R/W] B,H,W 0-000011	ESCR8/(IBSR8) [R/W] B,H,W -0000000	Multi-UART8 [1]: Byte access is permitted only for access to lower 8 bits
$0004E4_H$	RDR8/(TDR8)[R/W] B,H,W[1] -------0 00000000		BGR8 [R/W] H,W 00000000 00000000		
$0004E8_H$	-	-	-	-	
$0004EC_H$	FCR18 [R/W] B,H,W ---00100	FCR08 [R/W] B,H,W -0000000	FBYTE28 [R/W] B,H,W 00000000	FBYTE18 [R/W] B,H,W 00000000	
$0004F0_H$	SCR9/(IBCR9) [R/W] B,H,W 0--00000	SMR9 [R/W] B,H,W 000-0000	SSR9 [R/W] B,H,W 0-000011	ESCR9/(IBSR9) [R/W] B,H,W -0000000	Multi-UART9 [1]: Byte access is permitted only for access to lower 8 bits
$0004F4_H$	RDR9/(TDR9)[R/W] B,H,W[1] -------0 00000000		BGR9 [R/W] H,W 00000000 00000000		
$0004F8_H$	-	-	-	-	
$0004FC_H$	FCR19 [R/W] B,H,W ---00100	FCR09 [R/W] B,H,W -0000000	FBYTE29 [R/W] B,H,W 00000000	FBYTE19 [R/W] B,H,W 00000000	
000500_H to $00050C_H$	-	-	-	-	Reserved
000510_H	CSELR [R/W] B,H,W 001---00	CMONR [R] B,H,W 001---00	MTMCR [R/W] B,H,W 00001111	STMCR [R/W] B,H,W 0000-111	Clock control [S]
000514_H	PLLCR [R/W] B,H,W -------- 11110000		CSTBR [R/W] B,H,W -0000000	PTMCR [R/W] B,H,W 00------	
000518_H	-	-	CPUAR [R/W] B,H,W 0----XXX	-	Reset [S]
$00051C_H$	-	-	-	-	Reserved [S]

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000520_H	CCPSSELR [R/W] B,H,W -------0	-	-	CCPSDIVR [R/W] B,H,W -000-000	Clock control 2
000524_H	-	CCPLLFBR [R/W] B,H,W -0000000	CCSSFBR0 [R/W] B,H,W --000000	CCSSFBR1 [R/W] B,H,W ---00000	
000528_H	-	CCSSCCR0 [R/W] B,H,W ----0000	CCSSCCR1[R/W] H,W 000----- --------		
$00052C_H$	-	CCCGRCR0 [R/W] B,H,W 00----00	CCCGRCR1 [R/W] B,H,W 00000000	CCCGRCR2 [R/W] B,H,W 00000000	
000530_H	CCRTSELR [R/W] B,H,W 0------0	-	CCPMUCR0 [R/W] B,H,W 0-----00	CCPMUCR1 [R/W] B,H,W 0--00000	
000534_H	-	-	-	-	
000538_H	-	-	-	-	Clock control 2
$00053C_H$	-	-	-	-	
000540_H to $00054C_H$	-	-	-	-	Reserved
000550_H	EIRR0[R/W] B,H,W XXXXXXXX	ENIR0[R/W] B,H,W 00000000	ELVR0[R/W] B,H,W 00000000 00000000		External interrupt (INT0 to INT7)
000554_H	EIRR1[R/W] B,H,W XXXXXXXX	ENIR1[R/W] B,H,W 00000000	ELVR1[R/W] B,H,W 00000000 00000000		External interrupt (INT8 to INT15)
000558_H	-	-	-	-	Reserved
$00055C_H$	-	-	WTDR[R/W] H 00000000 00000000		Real-time clock
000560_H	-	WTCRH [R/W] B ------00	WTCRM [R/W] B,H 00000000	WTCRL [R/W] B,H ----00-0	
000564_H	-	WTBRH [R/W] B --XXXXXX	WTBRM [R/W] B XXXXXXXX	WTBRL [R/W] B XXXXXXXX	
000568_H	WTHR [R/W] B,H ---00000	WTMR [R/W] B,H --000000	WTSR [R/W] B --000000	-	
$00056C_H$	-	CSVCR[R/W]B -001110- -001010-[4]	-	-	Clock supervisor
000570_H to $00057C_H$	-	-	-	-	Reserved
000580_H	REGSEL [R/W] B,H,W 0110011-	-	-	-	Regulator control
000584_H	LVD5R [R/W] B,H,W -------1	LVD5F [R/W] B,H,W 0-100--1	LVD [R/W] B,H,W 01000--0	-	Low-voltage detection

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000588_H to $00058C_H$	-	-	-	-	Reserved
000590_H	PMUSTR [R/W] B,H,W 0-----1X	PMUCTLR [R/W] B,H,W 0-00----	PWRTMCTL [R/W] B,H,W -----011	-	PMU
000594_H	PMUINTF0 [R/W] B,H,W 00000000	PMUINTF1 [R/W] B,H,W 00000000	PMUINTF2 [R/W] B,H,W 0000----	-	
000598_H	-	-	-	-	
$00059C_H$ to $0005A4_H$	-	-	-	-	Reserved
$0005A8_H$	LCDCMR [R/W] B,H,W 0-------	LCRS [R/W] B,H,W 00000000	LCR0 [R/W] B,H,W 00010000	LCR1 [R/W] B,H,W --------	LCD controller
$0005AC_H$	VRAM0[R/W] B,H,W 00000000	VRAM1[R/W] B,H,W 00000000	VRAM2[R/W] B,H,W 00000000	VRAM3[R/W] B,H,W 00000000	
$0005B0_H$	VRAM4[R/W] B,H,W 00000000	VRAM5[R/W] B,H,W 00000000	VRAM6[R/W] B,H,W 00000000	VRAM7[R/W] B,H,W 00000000	
$0005B4_H$	VRAM8[R/W] B,H,W 00000000	VRAM9[R/W] B,H,W 00000000	VRAM10[R/W] B,H,W 00000000	VRAM11[R/W] B,H,W 00000000	LCD controller
$0005B8_H$	VRAM12[R/W] B,H,W 00000000	VRAM13[R/W] B,H,W 00000000	VRAM14[R/W] B,H,W 00000000	VRAM15[R/W] B,H,W 00000000	
$0005BC_H$	LDR0[R/W] B,H,W -------0	LDR1[R/W] B,H,W 00000000	-	-	
$0005C0_H$ to $0005FC_H$	-	-	-	-	Reserved
000600_H	ASR0 [R/W] W 00000000 00000000 -------- 1111-001				External bus Interface [S]
000604_H	ASR1 [R/W] W XXXXXXXX XXXXXXXX -------- XXXX-XX0				
000608_H	ASR2 [R/W] W XXXXXXXX XXXXXXXX -------- XXXX-XX0				
$00060C_H$	ASR3 [R/W] W XXXXXXXX XXXXXXXX -------- XXXX-XX0				
000610_H to $00063C_H$	-	-	-	-	Reserved [S]

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
000640_H	ACR0 [R/W] W -------- -------- -------- 01--00--				External bus Interface [S]
000644_H	ACR1 [R/W] W -------- -------- -------- XX--XX--				
000648_H	ACR2 [R/W] W -------- -------- -------- XX--XX--				
$00064C_H$	ACR3 [R/W] W -------- -------- -------- XX--XX--				
000650_H to $00067C_H$	-	-	-	-	Reserved [S]
000680_H	AWR0 [R/W] W ----1111 00000000 11110000 00000-0-				External bus Interface [S]
000684_H	AWR1 [R/W] W ----XXXX XXXXXXXX XXXXXXXX XXXXX-X-				
000688_H	AWR2 [R/W] W ----XXXX XXXXXXXX XXXXXXXX XXXXX-X-				
$00068C_H$	AWR3 [R/W] W ----XXXX XXXXXXXX XXXXXXXX XXXXX-X-				
000690_H to $00070C_H$	-	-	-	-	Reserved (to $0006FF_H$[S])
000710_H	BPCCRA[R/W] B 00000000	BPCCRB[R/W] B 00000000	BPCCRC[R/W] B 00000000	-	Bus performance counter
000714_H	BPCTRA [R/W] W 00000000 00000000 00000000 00000000				
000718_H	BPCTRB [R/W] W 00000000 00000000 00000000 00000000				
$00071C_H$	BPCTRC [R/W] W 00000000 00000000 00000000 00000000				
000720_H to $0007F8_H$	-	-	-	-	Reserved
$0007FC_H$	BMODR[R] B, H, W XXXXXXXX	-	-	-	Operation mode
000800_H to $00083C_H$	-	-	-	-	Reserved [S]
000840_H	FCTLR[R/W] H -0--1000 0--0----		-	FSTR[R/W] B -----001	Flash memory register [S]
000844_H	-	-	-	-	Reserved [S]
000848_H	-	-	-	-	Reserved [S]
$00084C_H$	-	-	-	-	
000850_H	-	-	-	-	
000854_H	-	-	-	-	
000858_H	-	-	WREN[R/W] H 00000000 00000000		Wild register [S]

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$00085C_H$	-	-	-	-	Reserved [S]
000860_H	-	-	-	-	
000864_H	-	-	-	-	
000868_H	-	-	-	-	
$00086C_H$	-	-	-	-	
000870_H	-	-	-	-	
000874_H	-	-	-	-	
000878_H	-	-	-	-	
$00087C_H$	-	-	-	-	
000880_H	WRAR00 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild register [S]
000884_H	WRDR00 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000888_H	WRAR01 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$00088C_H$	WRDR01 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000890_H	WRAR02 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
000894_H	WRDR02 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000898_H	WRAR03 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$00089C_H$	WRDR03 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008A0_H$	WRAR04 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008A4_H$	WRDR04 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008A8_H$	WRAR05 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008AC_H$	WRDR05 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008B0_H$	WRAR06 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008B4_H$	WRDR06 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008B8_H$	WRAR07 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008BC_H$	WRDR07 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0008C0_H$	WRAR08 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				Wild register [S]
$0008C4_H$	WRDR08 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008C8_H$	WRAR09 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008CC_H$	WRDR09 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008D0_H$	WRAR10 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008D4_H$	WRDR10 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008D8_H$	WRAR11 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008DC_H$	WRDR11 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008E0_H$	WRAR12 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008E4_H$	WRDR12 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008E8_H$	WRAR13 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008EC_H$	WRDR13 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008F0_H$	WRAR14 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008F4_H$	WRDR14 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0008F8_H$	WRAR15 [R/W] W -------- --XXXXXX XXXXXXXX XXXXXX--				
$0008FC_H$	WRDR15 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000900_H to $000BF8_H$	-	-	-	-	Reserved
$000BFC_H$	-	-	UER [W] B,H,W -------- -------X		OCDU

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000C00_H$	DCCR0[R/W] W 0----000 --00--00 00000000 0-000000				DMA controller [S]
$000C04_H$	DCSR0[R/W] H 0------- -----000		DTCR0[R/W] H 00000000 00000000		
$000C08_H$	DSAR0[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C0C_H$	DDAR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C10_H$	DCCR1 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C14_H$	DCSR1 [R/W] H 0------- -----000		DTCR1 [R/W] H 00000000 00000000		
$000C18_H$	DSAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C1C_H$	DDAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C20_H$	DCCR2 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C24_H$	DCSR2 [R/W] H 0------- -----000		DTCR2 [R/W] H 00000000 00000000		
$000C28_H$	DSAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C2C_H$	DDAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C30_H$	DCCR3[R/W] W 0----000 --00--00 00000000 0-000000				
$000C34_H$	DCSR3 [R/W] H 0------- -----000		DTCR3 [R/W] H 00000000 00000000		
$000C38_H$	DSAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C3C_H$	DDAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C40_H$	DCCR4 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C44_H$	DCSR4 [R/W] H 0------- -----000		DTCR4 [R/W] H 00000000 00000000		
$000C48_H$	DSAR4[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000C4C_H$	DDAR4[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMA controller [S]
$000C50_H$	DCCR5 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C54_H$	DCSR5 [R/W] H 0------- -----000		DTCR5 [R/W] H 00000000 00000000		
$000C58_H$	DSAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C5C_H$	DDAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C60_H$	DCCR6 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C64_H$	DCSR6 [R/W] H 0------- -----000		DTCR6 [R/W] H 00000000 00000000		
$000C68_H$	DSAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C6C_H$	DDAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C70_H$	DCCR7 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C74_H$	DCSR7 [R/W] H 0------- -----000		DTCR7 [R/W] H 00000000 00000000		
$000C78_H$	DSAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C7C_H$	DDAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C80_H$	DCCR8 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C84_H$	DCSR8 [R/W] H 0------- -----000		DTCR8 [R/W] H 00000000 00000000		
$000C88_H$	DSAR8 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C8C_H$	DDAR8 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000C90_H$	DCCR9 [R/W] W 0----000 --00--00 00000000 0-000000				
$000C94_H$	DCSR9 [R/W] H 0------- -----000		DTCR9 [R/W] H 00000000 00000000		
$000C98_H$	DSAR9 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000C9C_H$	DDAR9 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMA controller [S]
$000CA0_H$	DCCR10 [R/W] W 0----000 --00--00 00000000 0-000000				
$000CA4_H$	DCSR10[R/W] H 0------- -----000		DTCR10[R/W] H 00000000 00000000		
$000CA8_H$	DSAR10 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CAC_H$	DDAR10 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CB0_H$	DCCR11[R/W] W 0----000 --00--00 00000000 0-000000				
$000CB4_H$	DCSR11 [R/W] H 0------- -----000		DTCR11 [R/W] H 00000000 00000000		
$000CB8_H$	DSAR11 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CBC_H$	DDAR11 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CC0_H$	DCCR12 [R/W] W 0----000 --00--00 00000000 0-000000				
$000CC4_H$	DCSR12 [R/W] H 0------- -----000		DTCR12 [R/W] H 00000000 00000000		
$000CC8_H$	DSAR12 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CCC_H$	DDAR12 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CD0_H$	DCCR13 [R/W] W 0----000 --00--00 00000000 0-000000				
$000CD4_H$	DCSR13[R/W] H 0------- -----000		DTCR13[R/W] H 00000000 00000000		
$000CD8_H$	DSAR13[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CDC_H$	DDAR13[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CE0_H$	DCCR14[R/W] W 0----000 --00--00 00000000 0-000000				
$000CE4_H$	DCSR14[R/W] H 0------- -----000		DTCR14[R/W] H 00000000 00000000		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000CE8_H$	DSAR14[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMA controller [S]
$000CEC_H$	DDAR14[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CF0_H$	DCCR15[R/W] W 0----000 --00--00 00000000 0-000000				
$000CF4_H$	DCSR15[R/W] H 0------- -----000		DTCR15[R/W] H 00000000 00000000		
$000CF8_H$	DSAR15[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000CFC_H$	DDAR15[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000D00_H$ to $000DF0_H$	-	-	-	-	Reserved [S]
$000DF4_H$	-	-	DNMIR[R/W] B 0------0	DILVR[R/W] B ---11111	DMA controller [S]
$000DF8_H$	DMACR[R/W] W 0------- -------- 0------- --------				
$000DFC_H$	-	-	-	-	Reserved [S]
$000E00_H$	DDR00[R/W] B,H,W 00000000	DDR01[R/W] B,H,W 00000000	DDR02[R/W] B,H,W 00000000	DDR03[R/W] B,H,W 00000000	Data direction register
$000E04_H$	DDR04[R/W] B,H,W 00000000	DDR05[R/W] B,H,W -0000000	DDR06[R/W] B,H,W 00000000	DDR07[R/W] B,H,W 00000000	
$000E08_H$	DDR08[R/W] B,H,W 00000000	DDR09[R/W] B,H,W 00000000	DDR10[R/W] B,H,W 00000000	DDR11[R/W] B,H,W 00000000	
$000E0C_H$	DDR12[R/W] B,H,W 00000000	DDR13[R/W] B,H,W 00-00000	-	-	
$000E10_H$ to $000E1C_H$	-	-	-	-	Reserved
$000E20_H$	PFR00[R/W] B,H,W 00000000	PFR01[R/W] B,H,W 00000000	PFR02[R/W] B,H,W 00000000	PFR03[R/W] B,H,W 10000000	Port function register
$000E24_H$	PFR04[R/W] B,H,W 11111111	PFR05[R/W] B,H,W 11111111	PFR06[R/W] B,H,W 00000000	PFR07[R/W] B,H,W 00000000	
$000E28_H$	PFR08[R/W] B,H,W 00000000	PFR09[R/W] B,H,W 0-000000	PFR10[R/W] B,H,W 00000000	PFR11[R/W] B,H,W 00000000	
$000E2C_H$	PFR12[R/W] B,H,W 00000000	PFR13[R/W] B,H,W 00-00000	-	-	

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000E30_H$ to $000E3C_H$	-	-	-	-	Reserved
$000E40_H$	PDDR00[R] B,H,W XXXXXXXX	PDDR01[R] B,H,W XXXXXXXX	PDDR02[R] B,H,W XXXXXXXX	PDDR03[R] B,H,W XXXXXXXX	Input data direct read register
$000E44_H$	PDDR04[R] B,H,W XXXXXXXX	PDDR05[R] B,H,W XXXXXXXX	PDDR06[R] B,H,W XXXXXXXX	PDDR07[R] B,H,W XXXXXXXX	
$000E48_H$	PDDR08[R] B,H,W XXXXXXXX	PDDR09[R] B,H,W XXXXXXXX	PDDR10[R] B,H,W XXXXXXXX	PDDR11[R] B,H,W XXXXXXXX	
$000E4C_H$	PDDR12[R] B,H,W XXXXXXXX	PDDR13[R] B,H,W XX-XXXXX	-	-	
$000E50_H$ to $000E5C_H$	-	-	-	-	Reserved
$000E60_H$	EPFR00[R/W] B,H,W 00000000	EPFR01[R/W] B,H,W 00000000	EPFR02[R/W] B,H,W ---00000	EPFR03[R/W] B,H,W ---00000	Extended port function register
$000E64_H$	EPFR04[R/W] B,H,W ---00000	EPFR05[R/W] B,H,W ---00000	EPFR06[R/W] B,H,W ---00000	EPFR07[R/W] B,H,W ---00000	
$000E68_H$	EPFR08[R/W] B,H,W ---00000	EPFR09[R/W] B,H,W ---00000	EPFR10[R/W] B,H,W -0000000	EPFR11[R/W] B,H,W --000000	
$000E6C_H$	EPFR12[R/W] B,H,W --000000	EPFR13[R/W] B,H,W --000000	EPFR14[R/W] B,H,W --000000	EPFR15[R/W] B,H,W -0000000	
$000E70_H$	EPFR16[R/W] B,H,W 00000000	EPFR17[R/W] B,H,W 00000000	EPFR18[R/W] B,H,W 10000000	EPFR19[R/W] B,H,W 11111111	
$000E74_H$	EPFR20[R/W] B,H,W 11111111	EPFR21[R/W] B,H,W 00000000	EPFR22[R/W] B,H,W 00000000	EPFR23[R/W] B,H,W 00000000	
$000E78_H$	EPFR24[R/W] B,H,W -----000	EPFR25[R/W] B,H,W -----000	EPFR26[R/W] B,H,W ----0000	EPFR27[R/W] B,H,W ---00000	
$000E7C_H$	EPFR28[R/W] B,H,W ----0000	EPFR29[R/W] B,H,W 00000000	EPFR30[R/W] B,H,W 00000000	EPFR31[R/W] B,H,W 00000000	
$000E80_H$	EPFR32[R/W] B,H,W 00000000	EPFR33[R/W] B,H,W ---00000	EPFR34[R/W] B,H,W ---00000	EPFR35[R/W] B,H,W ---00000	
$000E84_H$	EPFR36[R/W] B,H,W ---00000	EPFR37[R/W] B,H,W 00000000	EPFR38[R/W] B,H,W ---00000	EPFR39[R/W] B,H,W 00000000	

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000E88_H$	EPFR40[R/W] B,H,W --000000	EPFR41[R/W] B,H,W -----000	EPFR42[R/W] B,H,W ------00	EPFR43[R/W] B,H,W 00000000	Extended port function register
$000E8C_H$	EPFR44[R/W] B,H,W 00000000	EPFR45[R/W] B,H,W 00000000	EPFR46[R/W] B,H,W --000000	EPFR47[R/W] B,H,W -------0	
$000E90_H$	-	-	-	-	
$000E94_H$	EPFR52[R/W] B,H,W -------0	EPFR53[R/W] B,H,W ---00000	EPFR54[R/W] B,H,W ----0000	-	
$000E98_H$ to $000E9C_H$	-	-	-	-	Reserved
$000EA0_H$	PPCR00[R/W] B,H,W 11111111	PPCR01[R/W] B,H,W 11111111	PPCR02[R/W] B,H,W 11111111	PPCR03[R/W] B,H,W 11111111	Port pull-up/down control register
$000EA4_H$	PPCR04[R/W] B,H,W 11111111	PPCR05[R/W] B,H,W 11111111	PPCR06[R/W] B,H,W 11111111	PPCR07[R/W] B,H,W 11111111	
$000EA8_H$	PPCR08[R/W] B,H,W 11111111	PPCR09[R/W] B,H,W 11111111	PPCR10[R/W] B,H,W 11111111	PPCR11[R/W] B,H,W 11111111	
$000EAC_H$	PPCR12[R/W] B,H,W 11111111	PPCR13[R/W] B,H,W 11-11111	-	-	Port pull-up/down control register
$000EB0_H$ to $000EBC_H$	-	-	-	-	Reserved
$000EC0_H$	PPER00[R/W] B,H,W 00000000	PPER01[R/W] B,H,W 00000000	PPER02[R/W] B,H,W 00000000	PPER03[R/W] B,H,W 00000000	Port pull-up/down enable register
$000EC4_H$	PPER04[R/W] B,H,W 00000000	PPER05[R/W] B,H,W 00000000	PPER06[R/W] B,H,W 00000000	PPER07[R/W] B,H,W 00000000	
$000EC8_H$	PPER08[R/W] B,H,W 00000000	PPER09[R/W] B,H,W 00000000	PPER10[R/W] B,H,W 00000000	PPER11[R/W] B,H,W 00000000	
$000ECC_H$	PPER12[R/W] B,H,W 00000000	PPER13[R/W] B,H,W 00-00000	-	-	
$000ED0_H$ to $000EDC_H$	-	-	-	-	Reserved

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000EE0_H$	PILR00[R/W] B,H,W 11111111	PILR01[R/W] B,H,W 11111111	PILR02[R/W] B,H,W 11111111	PILR03[R/W] B,H,W 11111111	Port input level selection register
$000EE4_H$	PILR04[R/W] B,H,W 11111111	PILR05[R/W] B,H,W 11111111	PILR06[R/W] B,H,W 11111111	PILR07[R/W] B,H,W 11111111	
$000EE8_H$	PILR08[R/W] B,H,W 11111111	PILR09[R/W] B,H,W 11111111	PILR10[R/W] B,H,W 11111111	PILR11[R/W] B,H,W 11111111	
$000EEC_H$	PILR12[R/W] B,H,W 11111111	PILR13[R/W] B,H,W 11-11111	-	-	
$000EF0_H$ to $000EFC_H$	-	-	-	-	Reserved
$000F00_H$	EPILR00[R/W] B,H,W 00000000	EPILR01[R/W] B,H,W 00000000	EPILR02[R/W] B,H,W 00000000	EPILR03[R/W] B,H,W 00000000	Extended Port input level selection register
$000F04_H$	EPILR04[R/W] B,H,W 00000000	EPILR05[R/W] B,H,W 00000000	EPILR06[R/W] B,H,W 00000000	EPILR07[R/W] B,H,W 00000000	
$000F08_H$	EPILR08[R/W] B,H,W 00000000	EPILR09[R/W] B,H,W 00000000	EPILR10[R/W] B,H,W 00000000	EPILR11[R/W] B,H,W 00000000	
$000F0C_H$	EPILR12[R/W] B,H,W 00000000	EPILR13[R/W] B,H,W 00-00000	-	-	
$000F10_H$ to $000F1C_H$	-	-	-	-	Reserved
$000F20_H$	PODR00[R/W] B,H,W 00000000	PODR01[R/W] B,H,W 00000000	PODR02[R/W] B,H,W 00000000	PODR03[R/W] B,H,W 00000000	Port output drive register
$000F24_H$	PODR04[R/W] B,H,W 00000000	PODR05[R/W] B,H,W 00000000	PODR06[R/W] B,H,W 00000000	PODR07[R/W] B,H,W 00000000	Port output drive register
$000F28_H$	PODR08[R/W] B,H,W 00000000	PODR09[R/W] B,H,W 00000000	PODR10[R/W] B,H,W 00000000	PODR11[R/W] B,H,W 00000000	
$000F2C_H$	PODR12[R/W] B,H,W 00000000	PODR13[R/W] B,H,W 00-00000	-	-	
$000F30_H$	-	-	-	-	Reserved
$000F34_H$	-	EPODR01 [R/W] B,H,W 00000000	EPODR02 [R/W] B,H,W 00000000	EPODR03 [R/W] B,H,W -0000000	Extended Port output drive register
$000F38_H$	EPODR06 [R/W] B,H,W 00000000	EPODR07 [R/W] B,H,W 00000000	EPODR08 [R/W] B,H,W 00000000	-	
$000F3C_H$	-	-	-	-	Reserved

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000F40_H$	PORTEN [R/W] B,H,W -------0	-	-	-	Port input enable register
$000F44_H$ to $000F6C_H$	-	-	-	-	Reserved
$000F70_H$	RCRH0[W] H,W XXXXXXXX	RCRL0[W] B,H,W XXXXXXXX	UDCRH0[R] H,W 00000000	UDCRL0[R] B,H,W 00000000	Up/down counter 0
$000F74_H$	CCR0[R/W] B,H 00000000 -0001000		-	CSR0[R/W] B 00000000	
$000F78_H$ to $000F7C_H$	-	-	-	-	Reserved
$000F80_H$	RCRH1[W] H,W XXXXXXXX	RCRL1[W] B,H,W XXXXXXXX	UDCRH1 [R] H,W 00000000	UDCRL1[R] B,H,W 00000000	Up/down counter 1
$000F84_H$	CCR1[R/W] B,H 00000000 -0001000		-	CSR1[R/W] B 00000000	
$000F88_H$ to $000F8C_H$	-	-	-	-	Reserved
$000F90_H$	OCCP4 [R/W] W 00000000 00000000 00000000 00000000				Output compare 4,5
$000F94_H$	OCCP5 [R/W] W 00000000 00000000 00000000 00000000				
$000F98_H$	OCFS45 [R/W] B, H, W ------11	-	OCSH45 [R/W] B, H, W ---0--00	OCSL45[R/W] B, H, W 0000--00	
$000F9C_H$	-	-	-	-	Reserved
$000FA0_H$	CPCLR2 [R/W] W 11111111 11111111 11111111 11111111				Free-run timer 2
$000FA4_H$	TCDT2 [R/W] W 00000000 00000000 00000000 00000000				
$000FA8_H$	TCCSH2 [R/W] B,H,W 0-----00	TCCSL2 [R/W] B,H,W -1-00000	-		
$000FAC_H$	CPCLR3 [R/W] W 11111111 11111111 11111111 11111111				Free-run timer 3
$000FB0_H$	TCDT3 [R/W] W 00000000 00000000 00000000 00000000				
$000FB4_H$	TCCSH3 [R/W] B,H,W 0-----00	TCCSL3 [R/W] B,H,W -1-00000	-		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$000FB8_H$	CPCLR4 [R/W] W 11111111 11111111 11111111 11111111				Free-run timer 4
$000FBC_H$	TCDT4 [R/W] W 00000000 00000000 00000000 00000000				
$000FC0_H$	TCCSH4 [R/W] B,H,W 0-----00	TCCSL4 [R/W] B,H,W -1-00000	-		
$000FC4_H$	CPCLR5 [R/W] W 11111111 11111111 11111111 11111111				Free-run timer 5
$000FC8_H$	TCDT5 [R/W] W 00000000 00000000 00000000 00000000				
$000FCC_H$	TCCSH5 [R/W]B,H,W 0-----00	TCCSL5 [R/W]B,H,W -1-00000	-		
$000FD0_H$	IPCP6 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input capture 6,7
$000FD4_H$	IPCP7 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000FD8_H$	ICFS67 [R/W] B, H, W ------00	-	LSYNS1 [R/W] B,H,W ----0000	ICS67 [R/W] B, H, W 00000000	
$000FDC_H$	IPCP8 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input capture 8,9
$000FE0_H$	IPCP9 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000FE4_H$	ICFS89 [R/W] B, H, W ------00	-	-	ICS89 [R/W] B, H, W 00000000	
$000FE8_H$	IPCP10 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Input capture 10,11
$000FEC_H$	IPCP11 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000FF0_H$	ICFS1011 [R/W] B, H, W ------00	-	-	ICS1011 [R/W] B, H, W 00000000	
$000FF4_H$ to $000FFC_H$	-	-	-	-	Reserved
001000_H	SACR [R/W] B,H,W -------0	PICD [R/W] B,H,W ----0011	-	-	Clock control
001004_H to $00103C_H$	-	-	-	-	Reserved

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
001040_H	-	SGDER0 [R/W] B,H,W 00000000	SGCR0[R/W] B,H,W -0000-0- 000--000		Sound generator 0
001044_H	SGAR0[R/W] B,H,W 00000000 00000000		SGFR0[R/W] B,H,W 00000000	SGNR0[R/W] B,H,W 00000000	
001048_H	SGTCR0[R/W] B,H,W 00000000	SGIDR0[R/W] B,H,W 00000000	SGPCR0[R/W] B,H,W 00000000 11111111		
$00104C_H$	SGDMAR0[W] B,H,W 00000000 00000000 00000000 00000000				
001050_H to $00105C_H$	-	-	-	-	Reserved
001060_H	-	SGDER1[R/W] B,H,W 00000000	SGCR1[R/W] B,H,W -0000-0- 000--000		Sound generator 1
001064_H	SGAR1[R/W] B,H,W 00000000 00000000		SGFR1[R/W] B,H,W 00000000	SGNR1[R/W] B,H,W 00000000	
001068_H	SGTCR1[R/W] B,H,W 00000000	SGIDR1[R/W] B,H,W 00000000	SGPCR1[R/W] B,H,W 00000000 11111111		
$00106C_H$	SGDMAR1[W] B,H,W 00000000 00000000 00000000 00000000				
001070_H to $00107C_H$	-	-	-	-	Reserved
001080_H	-	SGDER2[R/W] B,H,W 00000000	SGCR2[R/W] B,H,W -0000-0- 000--000		Sound generator 2
001084_H	SGAR2[R/W] B,H,W 00000000 00000000		SGFR2[R/W] B,H,W 00000000	SGNR2[R/W] B,H,W 00000000	
001088_H	SGTCR2[R/W] B,H,W 00000000	SGIDR2[R/W] B,H,W 00000000	SGPCR2[R/W] B,H,W 00000000 11111111		
$00108C_H$	SGDMAR2[W] B,H,W 00000000 00000000 00000000 00000000				
001090_H to $00109C_H$	-	-	-	-	Reserved

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0010A0_H$	-	SGDER3[R/W] B,H,W 00000000	SGCR3[R/W] B,H,W -0000-0- 000--000		Sound generator 3
$0010A4_H$	SGAR3[R/W] B,H,W 00000000 00000000		SGFR3[R/W] B,H,W 00000000	SGNR3[R/W] B,H,W 00000000	
$0010A8_H$	SGTCR3[R/W] B,H,W 00000000	SGIDR3[R/W] B,H,W 00000000	SGPCR3[R/W] B,H,W 00000000 11111111		
$0010AC_H$	SGDMAR3[W] B,H,W 00000000 00000000 00000000 00000000				
$0010B0_H$ to $0010BC_H$	-	-	-	-	Reserved
$0010C0_H$	-	SGDER4[R/W] B,H,W 00000000	SGCR4[R/W] B,H,W -0000-0- 000--000		Sound generator 4
$0010C4_H$	SGAR4[R/W] B,H,W 00000000 00000000		SGFR4[R/W] B,H,W 00000000	SGNR4[R/W] B,H,W 00000000	
$0010C8_H$	SGTCR4[R/W] B,H,W 00000000	SGIDR4[R/W] B,H,W 00000000	SGPCR4[R/W] B,H,W 00000000 11111111		
$0010CC_H$	SGDMAR4[W] B,H,W 00000000 00000000 00000000 00000000				
$0010D0_H$ to $00112C_H$	-	-	-	-	Reserved
001130_H	-	-	-	CRCCR[R/W] B,H,W -0000000	CRC operation
001134_H	CRCINIT[R/W] B,H,W 11111111 11111111 11111111 11111111				
001138_H	CRCIN[R/W] B,H,W 00000000 00000000 00000000 00000000				
$00113C_H$	CRCR[R] B,H,W 11111111 11111111 11111111 11111111				
001140_H to $001FFC_H$	-	-	-	-	Reserved
002000_H	CTRLR0 [R/W] B,H,W --------000-0001		STATR0[R/W] B,H,W -------- 00000000		CAN0 (64msb)
002004_H	ERRCNT0 [R] B,H,W 00000000 00000000		BTR0[R/W] B,H,W -0100011 00000001		
002008_H	INTR0 [R] B,H,W 00000000 00000000		TESTR0[R/W] B,H,W -------- X00000--		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$00200C_H$	BRPER0 [R/W] B,H,W -------- ----0000		-		CAN0 (64msb)
002010_H	IF1CREQ0 [R/W] B,H,W 0------- 00000001		IF1CMSK0 [R/W] B,H,W -------- 00000000		
002014_H	IF1MSK20 [R/W] B,H,W 11-11111 11111111		IF1MSK10 [R/W] B,H,W 11111111 11111111		
002018_H	IF1ARB20 [R/W] B,H,W 00000000 00000000		IF1ARB10 [R/W] B,H,W 00000000 00000000		
$00201C_H$	IF1MCTR0 [R/W] B,H,W 00000000 0---0000		-		
002020_H	IF1DTA10 [R/W] B,H,W 00000000 00000000		IF1DTA20[R/W] B,H,W 00000000 00000000		
002024_H	IF1DTB10 [R/W] B,H,W 00000000 00000000		IF1DTB20 [R/W] B,H,W 00000000 00000000		
002028_H, $00202C_H$	Reserved				
002030_H, 002034_H	Reserved (IF1 data mirror)				
002038_H, $00203C_H$	Reserved				
002040_H	IF2CREQ0 [R/W] B,H,W 0------- 00000001		IF2CMSK0 [R/W] B,H,W -------- 00000000		
002044_H	IF2MSK20 [R/W] B,H,W 11-11111 11111111		IF2MSK10 [R/W] B,H,W 11111111 11111111		
002048_H	IF2ARB20 [R/W] B,H,W 00000000 00000000		IF2ARB10 [R/W] B,H,W 00000000 00000000		
$00204C_H$	IF2MCTR0 [R/W] B,H,W 00000000 0---0000		-		
002050_H	IF2DTA10 [R/W] B,H,W 00000000 00000000		IF2DTA20 [R/W] B,H,W 00000000 00000000		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
002054_H	IF2DTB10 [R/W] B,H,W 00000000 00000000		IF2DTB20 [R/W] B,H,W 00000000 00000000		CAN0 (64msb)
002058_H, $00205C_H$	Reserved				
002060_H, 002064_H	Reserved (IF2 data mirror)				
002068_H to $00207C_H$	Reserved				
002080_H	TREQR20 [R] B,H,W 00000000 00000000		TREQR10 [R] B,H,W 00000000 00000000		
002084_H	TREQR40 [R] B,H,W 00000000 00000000		TREQR30 [R] B,H,W 00000000 00000000		
002088_H	-		-		
$00208C_H$	-		-		
002090_H	NEWDT20 [R] B,H,W 00000000 00000000		NEWDT10 [R] B,H,W 00000000 00000000		
002094_H	NEWDT40 [R] B,H,W 00000000 00000000		NEWDT30 [R]B,H,W 00000000 00000000		
002098_H	-		-		
$00209C_H$	-		-		
$0020A0_H$	INTPND20 [R] B,H,W 00000000 00000000		INTPND10 [R] B,H,W 00000000 00000000		
$0020A4_H$	INTPND40 [R] B,H,W 00000000 00000000		INTPND30 [R] B,H,W 00000000 00000000		
$0020A8_H$	-		-		
$0020AC_H$	-		-		
$0020B0_H$	MSGVAL20 [R] B,H,W 00000000 00000000		MSGVAL10 [R] B,H,W 00000000 00000000		
$0020B4_H$	MSGVAL40 [R] B,H,W 00000000 00000000		MSGVAL30 [R] B,H,W 00000000 00000000		
$0020B8_H$	-		-		
$0020BC_H$	-		-		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0020C0_H$ to $0020FC_H$	Reserved				
002100_H	CTRLR1 [R/W] B,H,W --------000-0001		STATR1[R/W] B,H,W -------- 00000000		CAN1 (32msb)
002104_H	ERRCNT1 [R] B,H,W 00000000 00000000		BTR1[R/W] B,H,W -0100011 00000001		
002108_H	INTR1 [R] B,H,W 00000000 00000000		TESTR1[R/W] B,H,W -------- X00000--		
$00210C_H$	BRPER1 [R/W] B,H,W -------- ----0000		-		
002110_H	IF1CREQ1 [R/W] B,H,W 0------- 00000001		IF1CMSK1 [R/W] B,H,W -------- 00000000		
002114_H	IF1MSK21 [R/W] B,H,W 11-11111 11111111		IF1MSK11 [R/W] B,H,W 11111111 11111111		
002118_H	IF1ARB21 [R/W] B,H,W 00000000 00000000		IF1ARB11 [R/W] B,H,W 00000000 00000000		
$00211C_H$	IF1MCTR1 [R/W] B,H,W 00000000 0---0000		-		
002120_H	IF1DTA11 [R/W] B,H,W 00000000 00000000		IF1DTA21 [R/W] B,H,W 00000000 00000000		
002124_H	IF1DTB11 [R/W] B,H,W 00000000 00000000		IF1DTB21 [R/W] B,H,W 00000000 00000000		
002128_H, $00212C_H$	Reserved				
002130_H, 002134_H	Reserved (IF1 data mirror)				
002138_H, $00213C_H$	Reserved				
002140_H	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	
002144_H	IF2MSK21 [R/W] B,H,W 11-11111 11111111		IF2MSK11 [R/W] B,H,W 11111111 11111111		CAN1 (32msb)
002148_H	IF2ARB21 [R/W] B,H,W 00000000 00000000		IF2ARB11 [R/W] B,H,W 00000000 00000000		
$00214C_H$	IF2MCTR1 [R/W] B,H,W 00000000 0---0000		-		
002150_H	IF2DTA11 [R/W] B,H,W 00000000 00000000		IF2DTA21 [R/W] B,H,W 00000000 00000000		
002154_H	IF2DTB11 [R/W] B,H,W 00000000 00000000		IF2DTB21 [R/W] B,H,W 00000000 00000000		
002158_H, $00215C_H$	Reserved				
002160_H, 002164_H	Reserved (IF2 data mirror)				
002168_H to $00217C_H$	Reserved				
002180_H	TREQR21 [R] B,H,W 00000000 00000000		TREQR11 [R] B,H,W 00000000 00000000		
002184_H	-		-		
002188_H	-		-		
$00218C_H$	-		-		
002190_H	NEWDT21 [R] B,H,W 00000000 00000000		NEWDT11 [R] B,H,W 00000000 00000000		
002194_H	-		-		
002198_H	-		-		
$00219C_H$	-		-		
$0021A0_H$	INTPND21 [R] B,H,W 00000000 00000000		INTPND11 [R] B,H,W 00000000 00000000		
$0021A4_H$	-		-		
$0021A8_H$	-		-		
$0021AC_H$	-		-		
$0021B0_H$	MSGVAL21 [R] B,H,W 00000000 00000000		MSGVAL11 [R] B,H,W 00000000 00000000		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0021B4_H$	-		-		CAN1 (32msb)
$0021B8_H$	-		-		
$0021BC_H$	-		-		
$0021C0_H$ to $0021FC_H$	Reserved				
002200_H	CTRLR2 [R/W] B,H,W -------- 000-0001		STATR2[R/W] B,H,W -------- 00000000		CAN2 (32msb)
002204_H	ERRCNT2[R] B,H,W 00000000 00000000		BTR2[R/W] B,H,W -0100011 00000001		
002208_H	INTR2[R] B,H,W 00000000 00000000		TESTR2[R/W] B,H,W -------- X00000--		
$00220C_H$	BRPER2 [R/W] B,H,W -------- ----0000		-		
002210_H	IF1CREQ2[R/W] B,H,W 0------- 00000001		IF1CMSK2[R/W] B,H,W -------- 00000000		
002214_H	IF1MSK22 [R/W] B,H,W 11-11111 11111111		IF1MSK12[R/W] B,H,W 11111111 11111111		
002218_H	IF1ARB22 [R/W] B,H,W 00000000 00000000		IF1ARB12[R/W] B,H,W 00000000 00000000		
$00221C_H$	IF1MCTR2[R/W] B,H,W 00000000 0---0000		-		
002220_H	IF1DTA12 [R/W] B,H,W 00000000 00000000		IF1DTA22[R/W] B,H,W 00000000 00000000		
002224_H	IF1DTB12 [R/W] B,H,W 00000000 00000000		IF1DTB22[R/W] B,H,W 00000000 00000000		
002228_H, $00222C_H$	Reserved				
002230_H, 002234_H	Reserved (IF1 data mirror)				
002238_H, $00223C_H$	Reserved				
002240_H	IF2CREQ2[R/W] B,H,W 0------- 00000001		IF2CMSK2[R/W] B,H,W -------- 00000000		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
002244_H	IF2MSK22 [R/W] B,H,W 11-11111 11111111		IF2MSK12[R/W] B,H,W 11111111 11111111		CAN2 (32msb)
002248_H	IF2ARB22[R/W] B,H,W 00000000 00000000		IF2ARB12[R/W] B,H,W 00000000 00000000		
$00224C_H$	IF2MCTR2[R/W] B,H,W 00000000 0---0000		-		
002250_H	IF2DTA12[R/W] B,H,W 00000000 00000000		IF2DTA22[R/W] B,H,W 00000000 00000000		
002254_H	IF2DTB12[R/W] B,H,W 00000000 00000000		IF2DTB22[R/W] B,H,W 00000000 00000000		
002258_H, $00225C_H$	Reserved				
002260_H, 002264_H	Reserved (IF2 data mirror)				
002268_H to $00227C_H$	Reserved				
002280_H	TREQR22[R] B,H,W 00000000 00000000		TREQR12[R] B,H,W 00000000 00000000		
002284_H	-		-		
002288_H	-		-		
$00228C_H$	-		-		
002290_H	NEWDT22[R] B,H,W 00000000 00000000		NEWDT12[R] B,H,W 00000000 00000000		
002294_H	-		-		
002298_H	-		-		
$00229C_H$	-		-		
$0022A0_H$	INTPND22[R] B,H,W 00000000 00000000		INTPND12[R] B,H,W 00000000 00000000		
$0022A4_H$	-		-		
$0022A8_H$	-		-		
$0022AC_H$	-		-		
$0022B0_H$	MSGVAL22[R] B,H,W 00000000 00000000		MSGVAL12[R] B,H,W 00000000 00000000		
$0022B4_H$	-		-		
$0022B8_H$	-		-		
$0022BC_H$	-		-		

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$0022C0_H$ to $0022FC_H$	-	-	-	-	Reserved
002300_H	DFCTLR[R/W] B,H,W -0------ --------		-	DFSTR[R/W] B,H,W -----001	WorkFlash
002304_H	-	-	-	-	
002308_H	FLIFCTLR [R/W] B,H,W ---0--00	-	FLIFFER1 [R/W] B,H,W --------	FLIFFER2 [R/W] B,H,W --------	Flash/ WorkFlash
$00230C_H$ to $0023FC_H$	-	-	-	-	Reserved
002400_H	SEEARX[R] B,H,W --000000 00000000		DEEARX[R] B,H,W --000000 00000000		XBS RAM ECC control register
002404_H	EECSRX[R/W] B,H,W ----0000	-	EFEARX [R/W] B,H,W --000000 00000000		
002408_H	-	EFECRX [R/W] B,H,W -------0 00000000 00000000			
$00240C_H$ to $002FFC_H$	-	-	-	-	Reserved
003000_H	SEEARA[R] B,H,W -----000 00000000		DEEARA[R] B,H,W -----000 00000000		Backup RAM ECC control register
003004_H	EECSRA[R/W] B,H,W ----0000	-	EFEARA[R/W] B,H,W -----000 00000000		
003008_H	-	EFECRA [R/W] B,H,W -------0 00000000 00000000			
$00300C_H$ to $003FFC_H$	-	-	-	-	Reserved
004000_H to $005FFC_H$	Backup-RAM				Backup RAM area
006000_H to $00FEFC_H$	-	-	-	-	Reserved ($00F000_H$ to[S])
$00FF00_H$	DSUCR [R/W] B,H,W -------- -------0		-	-	OCDU [S]
$00FF04_H$ to $00FF0C_H$	-	-	-	-	Reserved [S]

Address	Address Offset Value / Register Name				Block
	+0	+1	+2	+3	
$00FF10_H$	PCSR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				OCDU [S]
$00FF14_H$	PSSR [R/W] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$00FF18_H$ to $00FFF4_H$	-	-	-	-	Reserved [S]
$00FFF8_H$	EDIR1 [R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				OCDU [S]
$00FFFC_H$	EDIR0 [R] B,H,W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

[S]: It is a system register. The illegal instruction exception (data access error) is generated when read/write is performed on these registers in the user mode.

[4]: The initial value is different by part number. For details, refer to the CSVCR register in chapter “Clock Supervisor”.

A.3 Lists of Interrupt Vector

List of Interrupt Vector is shown.

This list shows the assignments of interrupt factors and interrupt vectors/interrupt control registers.

Figure A-4. Interrupt Vector

Interrupt Factor	Interrupt Number		Interrupt Level	Offset	Default Address for TBR	RN[1]
	Decimal	Hexa- decimal				
Reset	0	00	-	$3FC_H$	$000FFFFC_H$	-
System reserved	1	01	-	$3F8_H$	$000FFFF8_H$	-
System reserved	2	02	-	$3F4_H$	$000FFFF4_H$	-
System reserved	3	03	-	$3F0_H$	$000FFFF0_H$	-
System reserved	4	04	-	$3EC_H$	$000FFFEC_H$	-
FPU exception	5	05	-	$3E8_H$	$000FFFE8_H$	-
Exception of instruction access protection violation	6	06	-	$3E4_H$	$000FFFE4_H$	-
Exception of data access protection violation	7	07	-	$3E0_H$	$000FFFE0_H$	-
Data access error interrupt	8	08	-	$3DC_H$	$000FFFDC_H$	-
INTE instruction	9	09	-	$3D8_H$	$000FFFD8_H$	-
Instruction break	10	0A	-	$3D4_H$	$000FFFD4_H$	-
System Reserved	11	0B	-	$3D0_H$	$000FFFD0_H$	-
System Reserved	12	0C	-	$3CC_H$	$000FFFCC_H$	-
System Reserved	13	0D	-	$3C8_H$	$000FFFC8_H$	-
Exception of illegal instruction	14	0E	-	$3C4_H$	$000FFFC4_H$	-
NMI request/ XBS RAM double-bit error detection/ Backup RAM double-bit error detection	15	0F	15 (F_H) Fixed	$3C0_H$	$000FFFC0_H$	-
External interrupt 0-7	16	10	ICR00	$3BC_H$	$000FFFBC_H$	0
External interrupt 8-15	17	11	ICR01	$3B8_H$	$000FFFB8_H$	1
Reload timer 0/1/4/5	18	12	ICR02	$3B4_H$	$000FFFB4_H$	2[2]
Reload timer 2/3/6	19	13	ICR03	$3B0_H$	$000FFFB0_H$	3[2]
Multi-function serial interface ch.0 (reception completed)/ Multi-function serial interface ch.0 (status)	20	14	ICR04	$3AC_H$	$000FFFAC_H$	4[3]

Interrupt Factor	Interrupt Number		Interrupt Level	Offset	Default Address for TBR	RN[1]
	Decimal	Hexa- decimal				
Multi-function serial interface ch.0 (transmission completed)	21	15	ICR05	$3A8_H$	$000FFFA8_H$	5
Multi-function serial interface ch.1 (reception completed)/ Multi-function serial interface ch.1 (status)	22	16	ICR06	$3A4_H$	$000FFFA4_H$	6[3]
Multi-function serial interface ch.1 (transmission completed)	23	17	ICR07	$3A0_H$	$000FFFA0_H$	7
LIN-UART2 (reception completed)	24	18	ICR08	$39C_H$	$000FFF9C_H$	8
LIN-UART2 (transmission completed)	25	19	ICR09	398_H	$000FFF98_H$	9
LIN-UART3 (reception completed)	26	1A	ICR10	394_H	$000FFF94_H$	10
LIN-UART3 (transmission completed)	27	1B	ICR11	390_H	$000FFF90_H$	11
LIN-UART4 (reception completed)	28	1C	ICR12	$38C_H$	$000FFF8C_H$	12
LIN-UART4 (transmission completed)	29	1D	ICR13	388_H	$000FFF88_H$	13
LIN-UART5 (reception completed)	30	1E	ICR14	384_H	$000FFF84_H$	14
LIN-UART5 (transmission completed)	31	1F	ICR15	380_H	$000FFF80_H$	15
LIN-UART6 (reception completed)	32	20	ICR16	$37C_H$	$000FFF7C_H$	16
LIN-UART6 (transmission completed)	33	21	ICR17	378_H	$000FFF78_H$	17
CAN0	34	22	ICR18	374_H	$000FFF74_H$	-
CAN1	35	23	ICR19	370_H	$000FFF70_H$	-
CAN2/ Up/down counter 0/ Up/down counter 1	36	24	ICR20	$36C_H$	$000FFF6C_H$	-
Real time clock	37	25	ICR21	368_H	$000FFF68_H$	-
Sound generator 0 / LIN-UART7 (reception completed)	38	26	ICR22	364_H	$000FFF64_H$	22
Sound generator 1 / LIN-UART7 (transmission completed)	39	27	ICR23	360_H	$000FFF60_H$	23
PPG0/1/10/11/20/21	40	28	ICR24	$35C_H$	$000FFF5C_H$	24
PPG2/3/12/13/22/23	41	29	ICR25	358_H	$000FFF58_H$	25
PPG4/5/14/15	42	2A	ICR26	354_H	$000FFF54_H$	26
PPG6/7/16/17	43	2B	ICR27	350_H	$000FFF50_H$	27
PPG8/9/18/19	44	2C	ICR28	$34C_H$	$000FFF4C_H$	28
Multi-function serial interface ch.8 (reception completed)/ Multi-function serial interface ch.8 (status) / HS_SPI reception interrupt request	45	2D	ICR29	348_H	$000FFF48_H$	29[4]

Interrupt Factor	Interrupt Number		Interrupt Level	Offset	Default Address for TBR	RN[1]
	Decimal	Hexa- decimal				
Main timer/Sub timer/PLL timer / Multi-function serial interface ch.8(transmission completed)/ HS_SPI transmission interrupt request	46	2E	ICR30	344_H	$000FFF44_H$	30[4]
Clock calibration unit (Sub oscillation) / Sound generator 4/ Multi-function serial interface ch.9 (reception completed) / Multi-function serial interface ch.9 (status)	47	2F	ICR31	340_H	$000FFF40_H$	31[5]
A/D converter	48	30	ICR32	$33C_H$	$000FFF3C_H$	32
Clock calibration Unit (CR oscillation) / Multi-function serial interface ch.9 (transmission completed)	49	31	ICR33	338_H	$000FFF38_H$	33[5]
Free-run timer 0/2/4	50	32	ICR34	334_H	$000FFF34_H$	-
Free-run timer 1/3/5	51	33	ICR35	330_H	$000FFF30_H$	-
ICU0/6 (fetching)	52	34	ICR36	$32C_H$	$000FFF2C_H$	36
ICU1/7 (fetching)	53	35	ICR37	328_H	$000FFF28_H$	37
ICU2/8 (fetching)	54	36	ICR38	324_H	$000FFF24_H$	38
ICU3/9 (fetching)	55	37	ICR39	320_H	$000FFF20_H$	39
ICU4/10 (fetching)	56	38	ICR40	$31C_H$	$000FFF1C_H$	40
ICU5/11 (fetching)	57	39	ICR41	318_H	$000FFF18_H$	41
OCU0/1/6/7/10/11 (match)	58	3A	ICR42	314_H	$000FFF14_H$	42
OCU2/3/4/5/8/9 (match)	59	3B	ICR43	310_H	$000FFF10_H$	43
Base timer 0 IRQ0 / Base timer 0 IRQ1 / Sound generator 2	60	3C	ICR44	$30C_H$	$000FFF0C_H$	44
Base timer 1 IRQ0 / Base timer 1 IRQ1 / Sound generator 3 / XBS RAM single bit error generation / Backup RAM single bit error generation	61	3D	ICR45	308_H	$000FFF08_H$	45[6]
DMAC0/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15	62	3E	ICR46	304_H	$000FFF04_H$	-
Delayed interrupt	63	3F	ICR47	300_H	$000FFF00_H$	-
System reserved (Used for REALOS™[7].)	64	40	-	$2FC_H$	$000FFEFC_H$	-
System reserved (Used for REALOS.)	65	41	-	$2F8_H$	$000FFEF8_H$	-
Used with the INT instruction.	66 | 255	42 | FF	-	$2F4_H$ | 000_H	$000FFEF4_H$ | $000FFC00_H$	-

[1]: It does not support the DMA transfer request by the interrupt generated from a peripheral to which no RN (Resource Number) is assigned.

[2]: Reload timer ch.4 to ch.6 does not support the DMA transfer by the interrupt.

[3]: The status of the multi function serial interface does not support the DMA transfer by I^2C reception.

[4]: HS_SPI does not support the DMA transfer by the interrupt.

[5]: The clock calibration unit does not support the DMA transfer by the interrupt.

[6]: It does not support the DMA transfer by the interrupt because of the RAM ECC bit error.

[7]: REALOS is the trademark of Cypress.

A.4 Pins Statuses in State of CPU

Pin statuses in state of CPU are shown.

Figure A-5. Pin Statuses

Pin Number	Pin Function Port Number/Bidirection Function /Output Function/Input Function	Pad circuit type	Input level	Active Level	Pull-Up/Down	External Reset Factor 1(*1)			External Reset Factor 2(*2)					Internal Reset Factor(*4)		Internal Reset Factor(*5)		Sleep Mode		Stop Mode		Watch Mode		Affected by GPORTEN ?
						While the reset factor is active	After the reset factor has gone		While the reset factor is active		After the reset factor has gone													
						Issuing internal reset	Issuing internal reset (including oscillation stabilization waiting)	After internal reset releasing	Before internal reset issuance	Issuing internal reset	Before internal reset issuance	Issuing internal reset (including oscillation stabilization waiting)	After internal reset releasing	Issuing internal reset (including oscillation stabilization waiting)	After internal reset releasing	Issuing internal reset (including oscillation stabilization waiting)	After internal reset releasing	SLVL[1:0] =0x	SLVL[1:0] =1x	SLVL[1:0] =0x	SLVL[1:0] =1x	SLVL[1:0] =0x	SLVL[1:0] =1x	
2	P015/D29_0/SEG5/D21_1/INT13_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked	Status before entering shutdown is retained.	Status before entering shutdown is retained.	Last status retained(*6)	Last status retained	Last status retained	Hi-Z/Input Blocked(*10)	Last status retained	Hi-Z/Input Blocked(*10)	Yes
3	P016/D30_0/SEG6/D22_1/INT14_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
4	P017/D31_0/SEG7/D23_1/INT15_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
5	P020/AIX/SEG8/ICU8_0/ICU9_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
6	P021/CS0X/SEG9/ICU7_0/ICU1_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
7	P022/CS1X/SEG10/ICU8_0/ICU2_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
8	P023/RDX/SEG11/ICU9_0/OCU3_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
9	P024/WRDX/SEG12/ICU10_0/OCU11_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
10	P025/WR1X/SEG13/ICU11_0/OCU10_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
11	P026/A00/SEG14/SPI_CS3/SIN6_1/OCU8_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
12	P027/A01/SEG15/SPI_CS2/SOT6_1/OCU6_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
13	P030/A02/SEG16/SPI_CS1/SCK6_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
14	P031/A03/SEG17/SPI_CS0/SIN9_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
15	P032/A04/SEG18/SPI_SIO3/SOT9_0/OCU7_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
16	P033/A05/SEG19/SPI_SIO2/SCK9_0/OCU5_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
17	P034/A06/SEG20/SPI_SIO1/SIN9_0/OCU5_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
18	P035/A07/SEG21/SPI_SIO0/SOT9_0/OCU4_1	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
19	P036/A08/SEG22/PPG11_0/SPI_CLK/SCK9_0	H	CMOS/AMTTL		D	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
22	P037/A09/SEG23/ST0/PPG12_0/SIN7_0	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
23	P040/A10/SEG24/ST1/PPG13_0/SOT7_0	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
24	P041/A11/SEG25/ST2/PPG14_0/SCK7_0	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
25	P042/A12/SEG26/ST3/PPG15_0/AIN0_0	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
26	P043/A13/SEG27/ST4/BIN0_0/SOA4_0/OCU6_1	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
27	P044/A14/SEG28/ST5/ZIN0_0/SCA4_0/OCU7_1	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
28	P045/A15/SEG29/ST6/AIN1_0/SIN8_2	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
29	P046/A16/SEG30/ST7/BIN1_0/SOT8_2	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
30	P047/A17/SEG31/ST8/ZIN1_0/SCK8_2	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
31	P050/A18/COM6/OCU8_1	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
32	P051/A19/COM1/OCU9_1	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
33	P052/A20/COM2/OCU10_1	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
34	P053/A21/COM3/OCU11_1	I	CMOS/AMTTL		UD	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked	Output "L"/Input Blocked									
35	P054/SYSCLK/V0/FRCK0_1	J2	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
38	P055/CS2X/V1/FRCK1_1	J2	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
39	P056/CS3X/V2/FRCK2_1	J2	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
40	P057/RDY/V3/FRCK3_1	J3	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
43	P060/PWM1P0/AN8	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
44	P061/PWM1M0/AN9/SIN1_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
45	P062/PWM2P0/AN10/ZIN1_1/SOT1_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
46	P063/PWM2M0/AN11/BIN1_1/SCK1_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
47	P064/PWM1P1/AN12/AIN1_1/SIN0_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
48	P065/PWM1M1/AN13/ZIN0_1/SOT0_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
49	P066/PWM2P1/AN14/BIN0_1/SCK0_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
50	P067/PWM2M1/AN15/AIN0_1/SIN8_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
53	P070/PWM1P2/AN16/SOT9_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
54	P071/PWM1M2/AN17/SCK9_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
55	P072/PWM2P1/AN18/ICU11_1/SIN8_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
56	P073/PWM2M2/AN19/ICU10_1/SOT8_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
57	P074/PWM1P3/AN20/PPG12_1/ICU9_1/SCK8_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
58	P075/PWM1M3/AN21/PPG13_1/ICU8_1/SIN7_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
59	P076/PWM2P1/AN22/PPG14_1/ICU7_1/SOT7_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
60	P077/PWM2M3/AN23/PPG15_1/ICU6_1/SCK7_1	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
63	P080/PWM1P4/AN24/SIN6_0/PPG16_0/AIN0_2	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
64	P081/PWM1M4/AN25/SOT6_0/PPG17_0/BIN0_2	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
65	P082/PWM2P4/AN26/SCK6_0/PPG18_0/ZIN0_2	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
66	P083/PWM2M4/AN27/ICU0_2/PPG19_0	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
67	P084/PWM1P5/AN28/ICU1_2/PPG20_0	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
68	P085/PWM1M5/AN29/ICU2_2/PPG21_0	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
69	P086/PWM2P5/AN30/ICU3_2/PPG22_0	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
70	P087/PWM2M5/AN31/ICU4_2/PPG23_0	K	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
73	P090/ADTG/PPG0_2	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
74	P100/SIN4_1/AN0/TIN0_1/PPG8_0	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
75	P101/SOT4_1/AN1/TIN1_1/PPG9_0	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
76	P102/SCK4_1/AN2/TIN2_1/PPG10_0/ICU6_2	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
77	P103/SIN5_1/AN3/TIN3_1/PPG11_1/ICU7_2	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
78	P104/SOT5_1/AN4/TOT0_1/PPG2_1/ICU8_2	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
79	P105/SCK5_1/AN5/TOT1_1/PPG3_1/ICU9_2	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
80	P106/AN6/PPG4_1/ICU10_2/SOA4_1	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
81	P107/AN7/PPG5_1/DAO1/ICU11_2/SCA4_1	L	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
85	P123/OCU1_0/PPG8_2/DAO0/AN29	L	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
88	P122/OCU0_0/SCK5_0/TOT3_0/PPG7_2/AN28	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
87	P121/FRCK0_0/SOT5_0/INT7_0/TOT2_0/PPG6_2/AN27	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
88	P120/FRCK1_0/SIN5_0/INT6_0/TOT1_0/PPG5_2/AN26	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
89	P117/SCK4_0/TOT0_0/SCS3/TRG4/FRCK2_0/AN25	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
90	P116/SOT4_0/TIN3_0/SGA3/FRCK3_0/AN24	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
91	P115/SIN4_0/TIN2_0/SGO2/FRCK4_0/AN23	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
92	P114/SCK3_0/TIN1_0/ICU5_1/SGA2/TRG3/AN22	J	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
95	P137 (*13)	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
96	P136 (*13)	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
98	P097/WOT/SOT3_0/INT8_0/TIN0_0/ICU4_1/PPG8_1	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
99	P094/SGO1/SIN3_0/INT15_0/ICU1_1/PPG9_1	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
100	P093/SGA1/SOT3_0/INT14_0/ICU3_1/PPG8_1	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
101	P092/SGO0/SCK2_0/INT13_0/TOT3_1/ICU0_1/PPG7_1	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
102	P091/SGA0/SIN2_0/INT12_0/TOT2_1/ICU2_1/PPG6_1	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
103	P110/TX1/PPG1_2/FRCK5_0	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
104	P111/RX1/INT10_0/PPG2_2	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
105	P112/TX2/PPG3_2	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
106	P113/RX2/INT11_0/PPG4_2	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
111	P095/TX0/PPG10_1	M	CMOS/AMTTL		UD	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled									No
112	P096/RX0/INT9_0	M	CMOS/AMTTL		UD	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled									
113	P124/OCU2_0/ICU5_2/PPG8_2	M	CMOS/AMTTL		UD	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled									
120	P125/OCU3_0/ICU0_0/PPG10_2	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									Yes
121	P126/TRG0/SIN0_0/INT1_0/OCU4_0	M	CMOS/AMTTL		UD	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled									
122	P127/SOT0_0/ICU3_0	M	CMOS/AMTTL		UD	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled									No
123	P130/SCK0_0/INT0_0/ICU1_0/TIOA0	N	CMOS/AMTTL		UD	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled	←Input Enabled									
124	P131/TRG1/SIN1_0/INT4_0/ICU2_0/TIOA1	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
125	P132/SOT1_0/INT2_0/ICU3_0/TIOB0	N	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
126	P133/SCK1_0/INT3_0/ICU4_0/TIOB1/PPG11_1/TRG5	N	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
127	P134/TRG2/INT5_0/ICU5_0/PPG1_3	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
131	P000/D16_0/SIN2_1/TIN0_2/PPG0_0/D24_1/INT0_1	M	CMOS/AMTTL		UD	←Input Blocked	←Input Blocked	←Input Blocked	←Input Enabled	←Input Blocked	←Input Enabled	←Input Blocked	←Input Blocked	←Input Blocked	←Input Blocked									
132	P001/D17_0/

Revision History

Document Revision History

Document Title: CY91570 Series FR Family FR81S Hardware Manual Document Number: 002-05574				
Revision	**Issue Date**	**ECN No.**	**Origin of Change**	**Description of Change**
**	10/13/2011	—	TORS	New release
*A	06/22/2017	5569290	TORS	Migrated Spansion Technical Reference Manual “MN705-00011-3v1-E” to Cypress format.
*B	03/26/2019	6521860	TORS	Change series name and part number: MB91570 -> CY91570 Change package dimensions.